Nancy-Université

**UFR mathématiques et informatique** **École doctorale IAEM Lorraine**

**Département de formation doctorale en informatique**

# Modèles graphiques probabilistes pour la reconnaissance de formes

## THÈSE

présentée et soutenue publiquement le 4 décembre 2009

pour l'obtention du

**Doctorat de l'université Nancy 2**

**(spécialité informatique)**

par

Sabine BARRAT

**Composition du jury**

| | |
|---|---|
| *Président :* | Pierre LOONIS |
| *Rapporteurs :* | Laurence LIKFORMAN-SULEM<br>Philippe MULHEM |
| *Examinateurs :* | Adam CICHON<br>Pierre LOONIS<br>Salvatore TABBONE (directeur de thèse)<br>Muriel VISANI |
| *Invité :* | Patrick NOURRISSIER |

**Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503**

Loria

Mis en page avec la classe thloria.

# Remerciements

Voici venue la section des remerciements, qui, contrairement à ce que l'on pourrait croire, est loin d'être la plus facile. En effet, il s'agit de n'oublier personne. Dans cette section, je vais essayer de remercier toutes les personnes qui m'ont aidée et permis de mener à bien la dure et longue épreuve qu'a été cette thèse, grâce à leurs conseils, leur présence, leur disponibilité et parfois leurs coups de pieds au derrière ! Avant de commencer, j'aimerais m'excuser pour tout éventuel oubli, et je tiens à souligner que l'ordre n'a aucune espèce d'importance dans les remerciements qui vont suivre. J'aimerais remercier Antoine Tabbone, mon directeur de thèse, qui m'a encadrée et conseillée, tout en me laissant une grande liberté dans mes choix, depuis mon DEA jusqu'à la fin de la thèse. Il a accepté mon mauvais caractère et compris comment le gérer pour me pousser à donner le meilleur de moi-même. De plus, sans lui, cette thèse n'aurait simplement pas été possible, puisque c'est lui qui a trouvé le financement initial CIFRE de ma thèse. De même, je tiens à remercier Patrick Nourrissier et la société Netlor Concept, qui a financé ma thèse pendant les trois premières années. J'aimerais aussi remercier Kamel Smaïli, qui m'a incitée à continuer mes études après ma maîtrise, et sans qui je ne serai pas en train d'écrire cette section !

Je remercie tous les membres de mon jury, c'est-à-dire Laurence Likforman-Sulem et Philippe Mulhem, pour avoir accepté d'être rapporteurs de ma thèse, et Adam Cichon, Pierre Loonis et Muriel Visani pour en avoir été examinateurs.

Je remercie mes collègues de l'équipe QGAR pour leur accueil, et en particulier mes amis, avec qui j'ai échangé rires, mais aussi parfois larmes et coups de gueule.

Je remercie Josep Lladós, pour m'avoir accueillie pendant 1 mois, au CVC, à Barcelone, durant ma première année de thèse.

Je remercie les personnes qui ont relu, corrigé et commenté tout ou partie de mon manuscrit et m'ont ainsi aidée à l'améliorer. Je remercie particulièrement Thomas Bonnotte qui a relu entièrement ma thèse, sans rien y comprendre, afin de corriger les fautes d'orthographes !

Merci également à David Bellot, Alain Delaplace et Cherif Smaïli pour leur aide et conseils sur les réseaux Bayésiens.

Je remercie toutes les personnes qui m'ont fait confiance et permis d'enseigner en tant que vacataire et ATER. Enfin, je remercie ma famille et mes amis, et particulièrement Caroline Lavecchia, Armelle Brun, Jean-Pierre Salmon, Emilie Balland, Paul Brauner, Slim Ouni, Emmanuel Didiot, Joseph Razik, Oanh Nguyen, Hervé Locteau et j'en oublie malheureusement, qui m'ont aidée, supportée mes sautes d'humeur, remonté le moral, changé les idées, bref, qui m'ont soutenue quand j'en avais besoin.

Merci aussi à Karinne Gallaire pour ses bons cafés et sa bonne humeur, qui m'ont aidée à tenir !

*Je dédie cette thèse à mes parents*

# Résumé

La croissance rapide d'Internet et de l'information multimédia a suscité un besoin en développement de techniques de recherche d'information multimédia, et en particulier de recherche d'images. On peut distinguer deux tendances. La première, appelée recherche d'images à base de texte, consiste à appliquer des techniques de recherche d'information textuelle à partir d'images annotées. Le texte constitue une caractéristique de haut-niveau, mais cette technique présente plusieurs inconvénients : elle nécessite un travail d'annotation fastidieux. De plus, les annotations peuvent être ambiguës car deux utilisateurs peuvent utiliser deux mots-clés différents pour décrire la même image. Par conséquent, plusieurs approches ont proposé d'utiliser l'ontologie Wordnet, afin de réduire ces ambiguïtés potentielles. La seconde approche, appelée recherche d'images par le contenu, est plus récente. Ces techniques de recherche d'images par le contenu sont basées sur des caractéristiques visuelles (couleur, texture ou forme), calculées automatiquement, et utilisent une mesure de similarité afin de retrouver des images. Cependant, les performances obtenues ne sont pas vraiment acceptables, excepté dans le cas de corpus spécialisés. De façon à améliorer la reconnaissance, une solution consiste à combiner différentes sources d'information : par exemple, différentes caractéristiques visuelles et/ou de l'information sémantique. Or, dans de nombreux problèmes de vision, on dispose rarement d'échantillons d'apprentissage entièrement annotés. Par contre, il est plus facile d'obtenir seulement un sous-ensemble de données annotées, car l'annotation d'un sous-ensemble est moins contraignante pour l'utilisateur. Dans cette direction, cette thèse traite des problèmes de modélisation, classification et annotation d'images. Nous présentons une méthode pour l'optimisation de la classification d'images naturelles, en utilisant une approche de classification d'images basée à la fois sur le contenu des images et le texte associé aux images, et en annotant automatiquement les images non annotées. De plus, nous proposons une méthode de reconnaissance de symboles, en combinant différentes caractéristiques visuelles.

L'approche proposée est dérivée de la théorie des modèles graphiques probabilistes et dédiée aux deux tâches de classification d'images naturelles partiellement annotées, et d'annotation. Nous considérons une image comme partiellement annotée si son nombre de mots-clés est inférieur au maximum de mots-clés observés dans la vérité-terrain. Grâce à leur capacité à gérer les données manquantes et à représenter d'éventuelles relations entre mots-clés, les modèles graphiques probabilistes ont été proposés pour représenter des images partiellement annotées. Par conséquent, le modèle que nous proposons ne requiert pas que toutes les images soient annotées : quand une image est partiellement annotée, les mots-clés manquants sont considérés comme des données manquantes. De plus, notre modèle peut étendre automatiquement des annotations existantes à d'autres images partiellement annotées, sans intervention de l'utilisateur. L'incertitude autour de l'association entre un ensemble de mots-clés et une image est représentée par une distribution de probabilité jointe sur le vocabulaire des mots-clés et les caractéristiques visuelles extraites de nos bases d'images. Notre modèle est aussi utilisé pour reconnaître des symboles en combinant différents types de caractéristiques visuelles (caractéristiques discrètes et continues). De plus, de façon à résoudre le problème de dimensionnalité dû à la grande dimension des caractéristiques visuelles, nous avons adapté une méthode de sélection de variables. Enfin, nous avons proposé un modèle de recherche d'images permettant à l'utilisateur de formuler des requêtes sous forme de mots-clés et/ou d'images. Ce modèle intègre un processus de retour de pertinence. Les résultats expérimentaux, obtenus sur de grandes bases d'images complexes, généralistes ou spé-

cialisées, montrent l'intérêt de notre approche. Enfin, notre méthode s'est montrée compétitive avec des modèles de l'état de l'art.

**Mots-clés:** Modèles graphiques probabilistes, réseaux Bayésiens, classification d'images, recherche d'images, annotation automatique d'images, combinaison de descripteurs

## Abstract

The rapid growth of Internet and multimedia information has shown a need in the development of multimedia information retrieval techniques, especially in image retrieval. We can distinguish two main trends. The first one, called « text-based image retrieval », consists in applying text-retrieval techniques from fully annotated images. The text describes high-level concepts but this technique presents some drawbacks : it requires a tedious work of annotation. Moreover, annotations could be ambiguous because two users can use different keywords to describe a same image. Consequently, some approaches have proposed to use Wordnet in order to reduce these potential ambiguities. The second approach, called « content-based image retrieval », is a younger field. These methods rely on visual features (color, texture or shape) computed automatically, and retrieve images using a similarity measure. However, the obtained performances are not really acceptable, except in the case of well-focused corpus. In order to improve the recognition, a solution consists in combining differents sources of information : for example different visual features and/or visual and semantic information. In many vision problems, instead of having fully annotated training data, it is easier to obtain just a subset of data with annotations, because it is less restrictive for the user. In this direction, this thesis deals with modeling, classifying, and annotating images. We present a scheme for natural image classification optimization, using a joint visual-text clustering approach and automatically extending image annotations. Moreover, we propose a symbol recognition method, by combining visual features.

The proposed approach is derived from the probabilistic graphical model theory and dedicated for both tasks of weakly-annotated image classification and annotation. We consider an image as weakly annotated if the number of keywords defined for it is less than the maximum defined in the ground truth. Thanks to their ability to manage missing values and to represent possible relations between keywords, probabilistic graphical models have been proposed to represent weakly annotated images. Therefore, the proposed model does not require that all images be annotated : when an image is weakly annotated, the missing keywords are considered as missing values. Besides, our model can automatically extend existing annotations to weakly-annotated images, without user intervention. The uncertainty around the association between a set of keywords and an image is tackled by a joint probability distribution over the dictionary of keywords and the visual features extracted from our collections of images. Our model is also used to recognize symbols by combining different kinds of visual features (continuous and discrete features). Moreover, in order to solve the dimensionality problem due to the large dimensions of visual features, we have adapted a variable selection method. Finally, a system of image retrieval has been proposed. This system enables keyword and/or image requests and relevance feedback process. The experimental results, obtained on large image databases (general or specialized databases), show the interest of our approach. Finally, the proposed method is competitive with a state-of-art model.

**Keywords:** Probabilistic graphical models, Bayesian networks, image classification, image retrieval, automatic annotation, descriptor combination

# Table des matières

# 1

# Introduction générale

## 1.1 Contexte

Quelle est l'utilité de la connaissance si on ne peut pas y accéder ? La numérisation en masse et l'extension des réseaux, apparus récemment, est une solution qui a permis de répondre à ce problème. En effet, Internet est désormais accessible à tout un chacun (à la maison, gratuitement dans des lieux publics, *etc.*), et le matériel multimédia (appareils photo numériques, téléphones portables, caméra-vidéo, ...) est maintenant à des prix accessibles et donc largement utilisé par le grand public, impliquant par là-même un accroissement des données multimédia. Chacun a donc accès à une très grande quantité d'information de toute nature (texte, image, vidéo, musique, *etc.*) à tout moment. L'information est donc désormais accessible à tous.

Cependant, face à cette augmentation, se pose le problème de la surabondance d'informations. La quantité de ressources disponibles est telle qu'il est désormais devenu difficile voire impossible d'accéder à l'information recherchée. Le besoin d'**indexer** les données se fait donc obligatoire, permettant ainsi de stocker et d'organiser ces données de telle façon que l'on puisse y accéder le plus rapidement possible.

Parmi ces média, l'image occupe une place prépondérante. En effet, l'image constitue le cœur de nombreuses thématiques scientifiques telles que l'analyse d'images, la vision par ordinateur, la reconnaissance de formes (visages, empreintes digitales, logos, *etc.*), l'imagerie spatiale et médicale, la recherche d'informations sur internet ou dans des bases de données, les télécommunications ... De plus, l'image est également présente dans d'autres disciplines telles que l'audiovisuel, l'art ou encore le design.

Compte tenu de la place prépondérante des images dans les documents multimédia et de la diversité de ses applications, nous nous intéressons au média « image »en particulier. Le contexte industriel de cette thèse a également influencé ce choix. En effet cette thèse a été effectuée dans le cadre d'un financement CIFRE avec l'entreprise Netlor Concept, dont l'objectif était de pouvoir reconnaître automatiquement des objets dans des photos numériques : par exemple reconnaître des monuments dans des photos de lieux populaires et/ou historiques, ou le reconnaissance d'articles dans des catalogues.

Une fois l'indexation des images effectuées, *i. e.* lorsqu'elles sont stockées et organisées, nous devons nous intéresser aux différents moyens de pouvoir accéder non seulement aux images pertinentes, mais également en un temps restreint.

En fonction des besoins de l'utilisateur qui souhaite accéder à ces images, nous pouvons distinguer trois façons d'accéder aux données :

– **parcours séquentiel** : lorsque l'utilisateur n'a pas une idée précise de ce qu'il recherche,

une des solutions qui s'offre à lui consiste à parcourir séquentiellement les images disponibles jusqu'à ce qu'il trouve celle qui l'intéresse. Cette approche a l'inconvénient d'être longue pour l'utilisateur, sans aucune garantie de trouver l'image recherchée.
- **recherche d'images** : dans le cas où l'utilisateur a une idée précise de ce qu'il recherche, il peut exprimer son besoin de plusieurs façons :
  - soit il dispose d'une image (ou d'une esquisse) pour laquelle il souhaiterait obtenir des images visuellement similaires : c'est ce que l'on appelle de la recherche d'image par le contenu.
  - soit il définit l'image qu'il recherche en fonction de mots-clés la décrivant : appelée recherche textuelle d'images.
- **classification d'images** : quand l'utilisateur n'a qu'une vague idée des images recherchées, sa tâche peut être facilitée si la base d'images est organisée en classes thématiques. La classification désigne le processus qui permet de regrouper entre elles des images ayant des thématiques proches. Appliquée à une base d'images, la classification va permettre de fournir une représentation simplifiée et ordonnée de cette base. Elle permettra ainsi une manipulation et un accès à l'information plus faciles et rapides dans de grandes bases d'images. En fait, la recherche se réduira au parcours d'une classe particulière. Dans ce cas, la classification évite le parcours séquentiel de la base.

Afin de répondre à ces différents besoins, cette thèse va s'articuler autour des thématiques d'indexation (au sens de l'extraction de caractéristiques), de recherche, et de classification d'images.

## 1.2 Problématique

La première étape qui intervient dans la réalisation des objectifs que nous venons de définir est de trouver une méthode d'indexation d'images adaptée à la recherche et à la classification d'images dans de grandes bases. Parmi les méthodes actuelles d'indexation, on distingue deux grandes tendances :

- la première consiste à indexer textuellement les images. On parlera d'**indexation textuelle**. Dans ce cas, chaque image est représentée par un ensemble de mots-clés. Cette indexation peut se faire de deux manières :
  - L'indexation peut tout d'abord être manuelle. Dans ce cas le choix des mots-clés est laissé à un ou plusieurs indexeurs humains. Cette méthode s'avère performante, mais elle est très coûteuse pour l'utilisateur et se révèle difficilement applicable aux grandes bases d'images. De plus, elle pose un problème de subjectivité de certains termes : un même mot peut avoir deux sens différents. Par exemple, le terme « chat », peut, suivant son contexte, désigner l'animal ou le chat d'une aiguille. De même, une même image pourra être interprétée et décrite différemment par deux utilisateurs. On parle de polysémie de l'image : le sens d'une image dépend de celui qui la regarde. Afin de bien visualiser ce problème, prenons l'exemple de la figure 1.1. Certains utilisateurs utiliserons le terme « ciel »pour décrire cette image, d'autres utiliseront plutôt le terme « nuages », « champ », ou « blé ».

FIGURE 1.1 – Image polysémique

- L'indexation peut également se faire automatiquement, à partir du nom, de métadonnées ou commentaires associés à l'image, par exemple. Dans le cas d'images issues d'Internet, les mots-clés peuvent être extraits à partir du texte environnant l'image, de l'URL, du titre ou d'autres attributs de la page. Le problème de l'indexation automatique est que le texte utilisé pour extraire les mots-clés n'a pas toujours un rapport avec l'image. De ce fait, les mots-clés extraits représentent souvent mal les images. L'indexation automatique a également l'inconvénient d'être moins performante que l'indexation manuelle.
- la deuxième approche, que nous appellerons **indexation visuelle**, consiste à représenter les images sur la base de leurs caractéristiques de couleur, forme ou texture. Cette méthode est efficace sur certaines bases d'images, *i. e.* qu'elle va permettre de retrouver des images recherchées, sur la base de leurs caractéristiques visuelles. Par contre ses performances décroissent sur des bases d'images plus généralistes. Dans ce cas, il devient très difficile de trouver une ou plusieurs caractéristiques visuelles permettant de discriminer correctement l'ensemble des formes représentées dans la base.

Un problème commun à ces deux types de méthodes d'indexation est qu'elles contraignent l'utilisateur dans sa façon d'exprimer ses besoins. En effet, dans le cas de l'indexation textuelle, l'utilisateur devra décrire avec des mots-clés l'image qu'il recherche. Le résultat de la recherche dépendra donc des termes choisis par l'utilisateur, ce qui soulève de nouveau le problème de subjectivité des termes et de polysémie de l'image.

Au contraire, dans le cas de l'indexation visuelle, l'utilisateur doit disposer d'une image pour exprimer ses besoins. Il ne peut les exprimer sous forme de mots-clés. En effet, il n'y a pas, *a priori*, de relations entre caractéristiques visuelles et textuelles, dans une image. Il s'agit du problème de fossé sémantique, qui, à notre sens, est bien illustré par cette citation de W.J.T. Mitchell, extraite du livre [Mitchell 96] :

« The domains of word and image are like two countries that speak different languages but that have a long history of mutual migration, cultural exchange, and other forms of intercourse ».

Ces problèmes dévoilent la problématique de cette thèse : trouver une méthode de classification et/ou recherche d'images, permettant de réduire le fossé sémantique, tout en laissant l'utilisateur le plus libre possible dans la façon d'exprimer ses besoins.

## 1.3 Contributions

Afin de résoudre les problèmes que nous venons de soulever, une solution consiste à combiner : non seulement plusieurs caractéristiques visuelles, pour permettre de discriminer plus de formes et proposer des méthodes de recherche d'images efficaces dans des bases plus larges, mais aussi

des caractéristiques textuelles, pour donner plus de liberté aux utilisateurs dans l'expression de leurs besoins. En effet, ceux-ci pourront, grâce à la combinaison des deux types d'indexation, exprimer leurs besoins sous forme d'image(s) exemple(s) et/ou de mots-clés.

C'est la solution que nous avons choisie et nous parlerons, dans la suite, d'indexation **visuo-textuelle**.

Notre défi est donc de proposer des techniques de recherche et de classification d'images adaptées à ce type d'indexation.

Compte tenu de la problématique établie, nous proposons, dans cette thèse, d'utiliser une approche à base de modèles graphiques probabilistes. En effet, ces modèles permettent de manipuler des caractéristiques hétérogènes (continues et discrètes), ce qui permet de combiner facilement des caractéristiques visuelles et textuelles. Nous avons d'ailleurs proposé un nouveau descripteur de forme fournissant une matrice de valeurs continues. La représentation de ces deux types de caractéristiques permettra à l'utilisateur d'exprimer ses besoins de façon plus souple : à l'aide d'image(s) exemple(s) et/ou de mots-clés. De plus, les modèles graphiques probabilistes sont particulièrement adaptés au traitement des données manquantes, ce qui s'avère être un atout particulièrement intéressant pour notre problème. En effet, ceci donne plus de liberté à l'utilisateur, qui peut annoter seulement un sous-ensemble des images par des mots-clés. Les mots-clés manquants sont considérés comme des données manquantes. Une fois les images indexées visuellement et/ou textuellement, les algorithmes d'inférence associés aux modèles graphiques probabilistes permettent de classer, rechercher et même d'étendre automatiquement des annotations existantes à d'autres images. L'annotation manuelle partielle par l'utilisateur et l'extension automatique d'annotation contribuent à la réducticon du fossé sémantique.

Cependant, les modèles graphiques probabilistes sont réputés pour être moins efficaces en présence de données de grande dimension. Or, les caractéristiques visuelles sont souvent représentées dans des espaces de grande dimension. Afin de pouvoir concilier judicieusement les modèles graphiques probabilistes et les caractéristiques de grande dimension, nous avons adapté une méthode de sélection de variables, le LASSO, ce qui nous a permis de réduire à la fois le nombre de nos variables, la taille de nos modèles et leur complexité en temps. Nos modèles sont ainsi compétitifs avec des méthodes comme les SVM réputés pour leur performance en grande dimension.

Ces contributions ont donné lieu à plusieurs publications dont la liste est fournie dans la section suivante.

## 1.4 Publications dans le cadre de la thèse

### Revues internationales avec comité de lecture

- S. Barrat and S. Tabbone. A progressive learning method for symbol recognition. *Journal of Universal Computer Science*, 14(2) :224–236, jan 2008.
- S. Barrat and S. Tabbone. A bayesian network for combining descriptors : application to symbol recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, publié en ligne le 26/11/2009.

### Revues nationales avec comité de lecture

- S. Barrat and S. Tabbone. Classification et extension automatique d'annotations d'images en utilisant un réseau bayésien. *Traitement du Signal (TS)*, accepté le 27/11/2009.

### Conférences internationales avec comité de lecture et actes

- S. Barrat and S. Tabbone. A progressive learning method for symbols recognition. In *ACM symposium on Applied computing (SAC 2007)*, pages 627–631, Seoul, Korea, March 11-15, 2007.
- S. Barrat and S. Tabbone. Visual features with semantic combination using bayesian network for a more effective image retrieval. In *19th IEEE International Conference on Pattern Recognition (ICPR 2008)*, Tampa, Florida, USA, December 8-11, 2008.
- S. Tabbone, O. Ramos Terrades, and S. Barrat. Histogram of radon transform. a useful descriptor for shape retrieval. In *19th IEEE International Conference on Pattern Recognition (ICPR 2008)*, Tampa, Florida, USA, December 8-11, 2008.
- S. Barrat and S. Tabbone. Modeling, classifying and annotating weakly annotated images using bayesian network. In *10th International Conference on Document Analysis and Recognition (ICDAR 2009)*, Barcelona, Catalonia, Spain, July 26-29, 2009.

### Workshops internationaux avec comité de lecture et actes

- S. Barrat, S. Tabbone, and P. Nourrissier. A bayesian classifier for symbol recognition. In *Seventh IAPR INternational Workshop on Graphics Recognition (GREC 2007)*, Curitiba, Brazil, September 20-21, 2007.
- S. Barrat and S. Tabbone. Classification and automatic annotation extension of images using bayesian network. In *Joint IAPR International Workshops on Structural and Syntactic Pattern Recognition and Statistical Techniques in Pattern Recognition (S+SSPR 2008, LNCS 5342)*, Orlando, Florida, USA, December 4-6, 2008.

### Conférences nationales avec comité de lecture et actes

- S. Barrat and S. Tabbone. Apprentissage progressif pour la reconnaissance de symboles dans les documents graphiques. In *Actes du 15ème Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA 2006), Tours*, 2006.
- S. Barrat and S. Tabbone. Modèles graphiques pour la combinaison de descripteurs : application à la reconnaissance de symboles. In *Actes du 16ème Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA 2008), Amiens*, pages 647–654, 2008.
- S. Barrat and S. Tabbone. Classification et extension automatique d'annotations d'images en utilisant un réseau bayésien. In *Actes du Colloque International Francophone sur l'Ecrit et le Document (CIFED 2008), Rouen*, 2008.
- S. Barrat and S. Tabbone. Modélisation, classification et annotation d'images partiellement annotées avec un réseau Bayésien. In *17ème Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA 2010), Caen*, 2010.

### Articles de revues internationales soumis et en cours de révision

- S. Barrat and S. Tabbone. Modeling, classifying and annotating weakly annotated images using bayesian network. *Journal of Visual Communication and Image Representation (JVCI)*, 2ème relecture.

## 1.5 Organisation du mémoire

Ce mémoire s'articule en deux parties :

- La **partie I** est consacrée à l'étude des méthodes existantes d'indexation, recherche, classification et annotation d'images. En particulier :
  - les différents types d'indexation, que nous venons d'introduire, et les méthodes de recherche d'images associées, sont présentés dans le **chapitre 2**.
  - Le **chapitre 3** présente un état de l'art des techniques de classification automatique.
  - Le problème d'annotation d'images fait l'objet du **chapitre 4**. L'étude de ces méthodes nous permet de justifier notre choix pour les modèles graphiques probabilistes.
- La **partie II** est dédiée à la présentation et l'évaluation de nos contributions en reconnaissance de formes.
  - Tout d'abord, un tutoriel sur les modèles graphiques probabilistes, leur fonctionnement et leur utilisation en classification, est proposé dans le **chapitre 5**.
  - Ensuite, dans le **chapitre 6**, nous proposons un modèle de reconnaissance de symboles par combinaison de descripteurs de forme. Un nouveau descripteur de forme y est d'ailleurs présenté.
  - Les chapitres 7 et 8 sont consacrés à la reconnaissance d'images naturelles dans des bases généralistes. En particulier, dans le **chapitre 7**, nous proposons différents modèles de représentation, classification et annotation d'images naturelles.
  - Le **chapitre 8**, aborde, quant à lui, une problématique un peu différente : la recherche d'images. Nous y proposons un modèle de représentation et recherche d'images. Afin d'améliorer les résultats de la recherche, le modèle autorise l'intervention de l'utilisateur dans le processus de recherche : on parle de retour de pertinence.

Enfin, dans le **chapitre 9**, nous faisons un bilan sur ces contributions avant d'introduire nos perspectives de recherche.

# Première partie

# État de l'art

# Chapitre 2

# Indexation et recherche d'images

**Sommaire**



Grâce à l'essor d'Internet et des nouvelles technologies, ces dernières années, tout un chacun peut maintenant, et à prix modique, s'équiper d'un ordinateur avec un scanner ou une petite caméra, puis diffuser ses images sur Internet. De plus, les bibliothèques numériques en ligne se sont développées ces dernières années. Par exemple, on peut citer Gallica [1], la bibliothèque numérique de la Bibliothèque nationale de France, qui donne accès à la consultation d'une partie des collections numérisées de la Bibliothèque nationale de France, ou Google Books [2], qui permet d'effectuer, en ligne, des recherches sur l'intégralité du texte de sept millions de livres. Ces bibliothèques constituent un nouveau mode de diffusion qui a l'avantage de faciliter l'accès à la culture, mais aussi de préserver le patrimoine en conservant les documents pour une plus grande durée que le support papier original. Ces bibliothèques numériques ont nécessité la numérisation de livres en masse : en 2009, 10 millions de livres ont déjà été scannés dans la bibliothèque numérique de Google Books. Cette numérisation massive de documents soulève une problématique liée à l'**indexation** des grosses quantités d'images et à la **recherche d'images** dans de grandes bases.

1. `http://gallica.bnf.fr/`)
2. `http://books.google.com/`

De même, la recherche d'images est un domaine de recherche en plein essor, compte tenu de la diversité de ses applications. En effet, les images occupent une place prépondérante dans les documents multimédias, de nos jours, omniprésents et mis à la disposition des professionnels et du grand public. Le réseau Internet en est un bon exemple. Les secteurs de la presse et de l'audiovisuel, ceux de l'industrie (imagerie scientifique), de la médecine ou encore ceux de la propriété industrielle collectent des quantités impressionnantes d'images qu'il faut pouvoir gérer. Or, souvent, le processus d'acquisition de ces images est plus rapide et simple que celui de l'indexation, ce qui fait naître un besoin urgent d'indexation automatique.

L'indexation d'images traite de la représentation, du stockage, de l'organisation et de l'accès aux images. Une fois les images indexées, on souhaite pouvoir les retrouver facilement. Les systèmes de recherche d'images (SRIm) tentent de résoudre ce problème. En effet, le but d'un système de recherche d'images est de retrouver, parmi une collection d'images préalablement stockées, celles qui répondent au besoin utilisateur exprimé sous forme de requête. Suivant le type d'indexation, une requête pourra être, par exemple, un ensemble de mots-clés ou une expression logique composée d'opérateurs logiques et de mots-clés décrivant les images recherchées par l'utilisateur ou les thèmes qu'elles véhiculent. Une requête peut aussi être une image fournie par l'utilisateur car il la considère similaire à celles qu'il recherche. Enfin, une requête peut être constituée à la fois d'une image et de mots-clés. Pour répondre à une requête, un SRIm met en œuvre un ensemble de processus de sélection des images pertinentes pour la requête. Une image est considérée comme pertinente si elle correspond au besoin de l'utilisateur exprimé par la requête. Le degré de correspondance est établi grâce à une mesure de similarité entre requête et réponses.

Dans ce chapitre, nous présentons un état de l'art succinct des méthodes d'indexation et de recherche d'images. En nous basant sur les avantages et inconvénients de ces méthodes, nous justifions et présentons notre choix pour une méthode d'indexation permettant des requêtes plus flexibles pour l'utilisateur, constituées par une image et/ou des mots-clés. Ce chapitre est organisé comme suit : en section 2.1, nous présentons les concepts de la recherche d'image (RIm) de façon générale. Nous y décrivons notamment les processus d'indexation et de recherche. Les deux types d'indexation possibles et les méthodes de recherche associées sont décrits sections 2.2 et 2.3. Enfin, grâce à l'étude des différentes méthodes d'indexation et de recherche d'images existantes, nous justifions, section 2.4 notre choix pour la combinaison des deux types d'indexation, que nous appellerons indexation visuo-textuelle, car les images sont indexées à la fois par des mots-clés et des caractéristiques visuelles.

## 2.1 Concepts

L'indexation d'images consiste à extraire des caractéristiques significatives de ces images puis à construire des index performants, *i. e.* à associer à chaque image un ensemble de caractéristiques pertinentes et à les organiser. L'index d'une base d'images est donc l'ensemble ordonné des caractéristiques extraites sur les images de cette base. L'organisation des caractéristiques doit permettre un accès plus rapide aux images qui seront recherchées. Ces caractéristiques peuvent être de type textuel (par exemple, un ensemble de mots-clés, aussi appelés termes) associés à chaque image), ou visuel (par exemple le nombre de pixels noirs d'une image). Ces caractéristiques doivent être pertinentes pour permettre une recherche rapide et efficace. En effet le choix des caractéristiques va dépendre de la base d'image considérée. Par exemple, le choix de la caractéristique « nombre de pixels noirs d'une image »pour indexer une base d'images couleurs, semble peu pertinent.

La qualité de l'indexation est difficile à évaluer. Elle dépend de la pertinence des caractéristiques, de la capacité de mise à jour, insertion ou suppression de données et du temps de réponse (temps pour accéder à l'image recherchée). Elle dépend aussi du degré de satisfaction de l'utilisateur.

Ainsi, la notion d'indexation fait référence à la fois au processus d'extraction des caractéristiques et à leur stockage. L'indexation constitue donc le processus permettant de construire la représentation interne des images d'une base et d'ordonner ces représentations. Le terme « représentation interne »d'une image désigne l'ensemble des caractéristiques calculées sur cette image.

Une fois ces données stockées, le problème est de pouvoir les retrouver rapidement. On parle de recherche d'images. Le processus général de recherche d'images fait donc ressortir deux mécanismes de base : le processus d'indexation et le processus de recherche. On distingue deux types de recherche d'images correspondant chacune à un type d'utilisateur :

- le premier est celui qui ne connaît pas exactement ce qu'il cherche et tente d'explorer la masse d'images à sa disposition. Dans ce cas, la navigation (dans une collection d'images personnelle ou sur Internet) constitue l'outil le plus approprié.
- Le second type d'utilisateur, le plus fréquent, est celui qui définit une requête correspondant à son désir d'information et qui attend une liste, précise et pertinemment ordonnée, des images. Le processus de recherche consiste alors à mettre en correspondance et à calculer le degré d'appariement des représentations internes des images de la base et de la requête. Les images qui correspondent au mieux à la requête (images dites pertinentes), sont alors retournées à l'utilisateur, dans une liste ordonnée par ordre décroissant de degré de pertinence lorsque le système le permet.

L'efficacité de la recherche est évaluée en fonction du nombre d'images pertinentes et non pertinentes, pour la requête, retrouvées dans une base : une recherche permettant de retrouver, dans une base d'images, toutes les images pertinentes pour la requête, et aucune image non pertinente, est parfaitement efficace.

Dans ce contexte, c'est donc l'indexation des images, *i. e.* leur représentation interne et l'organisation de ces représentations, qui permettra de les retrouver rapidement et facilement, en utilisant l'index des termes ou caractéristiques reliés à ces images.

On compte deux types d'indexation :

- une première technique (voir section 2.2) consiste à extraire automatiquement des mots-clés à partir des images elles-mêmes (leur nom ou métadonnées) ou des pages Web dans lesquelles elles sont contenues, par exemple. Il est aussi possible d'annoter manuellement ces images par des mots-clés. La recherche d'images pourra ensuite être effectuée à partir de ces mots-clés (indexation et recherche textuelle).
- Une autre voie (voir section 2.3) consiste à rechercher directement les images à partir de leur contenu même, *i. e.* à partir de leurs caractéristiques visuelles. Dans ce cas la recherche d'images se fait par mesure de similarité (de couleur, forme, ou texture) entre une image requête et les images du corpus utilisé (indexation et recherche par le contenu).

Enfin, à partir de ces deux types d'indexation, une solution consiste à combiner les informations sémantiques (les mots-clés) et visuelles. On parlera d'indexation et de recherche visuo-textuelle (solution section 2.4))

## 2.2 Indexation textuelle et recherche d'images à partir du texte associé aux images

Ces méthodes d'indexation et de recherche s'apparentent à celles utilisées classiquement dans une base de données textuelles. Simplement, dans le cas de la recherche d'images, les données indexées sont les informations textuelles se rapportant aux images. Ces informations textuelles proviennent d'un traitement préalable des images : soit elles ont été extraites de façon automatique à partir des images elles-mêmes (de leur nom ou métadonnées) ou du texte présent dans les pages Web les contenant, soit elles ont été fournies manuellement, par des indexeurs. Ces différents types d'indexation textuelle sont expliqués dans la section 2.2.1. Les méthodes de recherche d'images à partir de texte associé aux images sont introduites section 2.2.2. La reformulation de requêtes et les processus de retour de pertinence, seront abordés dans la section 2.2.3. Enfin, dans la section 2.2.4, nous ferons le point sur les méthodes d'indexation et de recherche textuelle d'images.

### 2.2.1 Indexation textuelle

En phase d'indexation, chaque image est analysée et les mots-clés (on pourra aussi parler de concepts) caractérisant son contenu informationnel sont extraits. Une fois extraits, les mots-clés sont stockés de manière organisée, à l'aide de bases de données, de fichiers inversés, ou de fichiers signatures, ... [Moffat 96, Kent 90].

Les mots-clés peuvent être extraits de façon automatique à partir de l'image elle-même (de son nom ou ses métadonnées) ou du texte présent dans la page Web la contenant, par exemple. Les mots-clés peuvent aussi être fournis manuellement, par des indexeurs.

Les mots-clés descriptifs du contenu sémantique d'une image sont appelés termes d'indexation (on peut aussi parler de descripteur). L'ensemble de tous les termes d'indexation des images d'une base constitue le langage (ou vocabulaire) d'indexation de cette base. Ce langage peut être libre ou contrôlé. Un langage d'indexation libre est constitué des termes extraits des images déjà analysées. Un langage libre est donc évolutif car il est susceptible d'être enrichi par un nouveau terme après chaque analyse d'image. Au contraire, un langage d'indexation contrôlé est construit avant de commencer à indexer les images. Dans ce cas, lorsqu'une image est analysée, on ne garde que les mots-clés qui appartiennent au vocabulaire préalablement défini. C'est-à-dire que les mots-clés associés à l'image qui sont absents du vocabulaire ne sont pas pris en compte.

Les termes d'un langage contrôlé peuvent être organisés dans un thésaurus. Un thésaurus est une liste structurée de termes constituant un vocabulaire normalisé, construit pour un besoin précis et un champ particulier. En effet, le choix des termes et leur structuration dépend du besoin d'information. Par exemple, dans une base d'images de véhicules (autos, trains, avions, *etc.*), les termes « coupé »et « cabriolé »pourront être indexés par le mot-clé général « auto », alors qu'ils seront différenciés dans un thésaurus dédié uniquement aux automobiles. De ce fait, la plupart des bases de données ont leur thésaurus. En plus des termes, un thésaurus fournit des explications sur ces termes et des relations entre eux [Cai 05]. En effet, les termes peuvent être reliés entre eux par des relations synonymiques, hiérarchiques et associatives. On parle souvent simplement de vocabulaire.

La figure 2.1 montre un extrait du thésaurus de l'UNESCO [3] correspondant à une recherche sur le mot-clé « thèse ». Le thésaurus fournit un ensemble de descripteurs pour le mot « thèse » : examen écrit, examen, évaluation de l'étudiant ...

3. `http://databases.unesco.org/thesfr/`

Ces termes constituent la note explicative du mot « thèse », et expliquent comment l'employer. Le symbole MT signifie que « Évaluation de l'éducation »est un micro thésaurus (un sous-ensemble particulier du thésaurus de l'UNESCO) auquel le terme « thèse »appartient. Le symbole EP signifie que le terme « dissertation »est proche du terme « thèse ». Le symbole TG indique que les termes examen écrit, examen, ... sont apparentés au mot « thèse »mais ce sont des termes plus généraux ils se situent au dessus du terme « thèse »dans la hiérarchie du thésaurus. Le symbole TA indique que « document primaire »est un terme associé au mot « thèse ». Enfin, le chiffre entre crochets situé à droite de chaque descripteur indique le nombre de documents de la base contenant ce descripteur. Wordnet [Fellbaum 98] est un autre exemple de thésaurus, dédiés aux termes de langue anglaise.

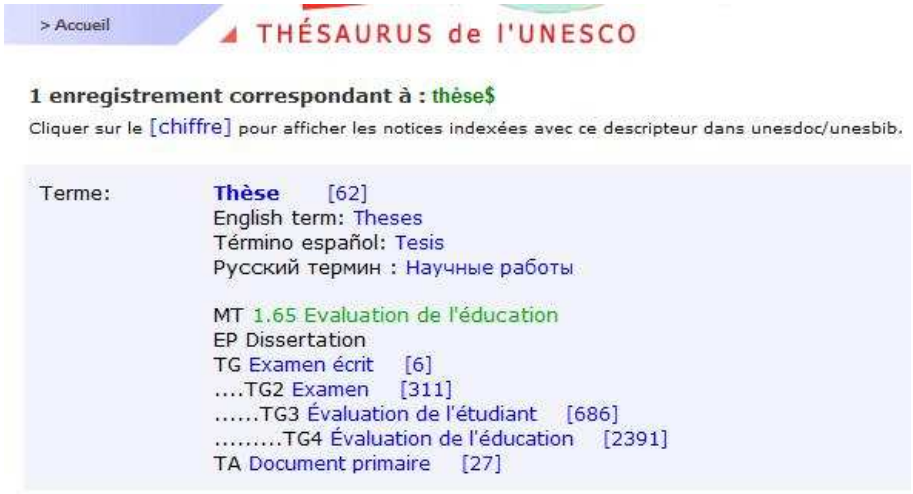



FIGURE 2.1 – Sous-ensemble du thésaurus de l'UNESCO

Techniquement, l'indexation peut-être automatique, manuelle ou semi-automatique [Salton 88]. Ces différentes techniques d'indexation textuelle sont abordées dans les sections 2.2.1.1, 2.2.1.2 et 2.2.1.3. Enfin, dans la section 2.2.1.4, nous présentons les solutions les plus populaires de stockage et d'organisation des données textuelles obtenues.

#### 2.2.1.1 Indexation automatique

En indexation automatique [Salton 68a, Maron 60], c'est un processus complètement automatisé qui se charge d'extraire les termes caractéristiques d'une image.

Dans un premier temps, les informations textuelles sont collectées. Pour les images du Web, des « robots d'indexation »sont utilisés. Ces logiciels visitent et analysent les pages Web (et les images, vidéo ou documents qu'elles contiennent), à une certaine fréquence, et enregistrent les informations collectées suivant des critères qui leurs sont propres. Les moteurs de recherche utilisent ces logiciels pour alimenter leurs bases. Une fois les informations collectées par les robots, les moteurs de recherche extraient automatiquement des mots, à l'aide d'expressions régulières (analyse lexicale) et à partir du code HTML des balises et attributs spécifiques aux images ou dans le texte, le titre des pages, les métadonnées ou la légende des images. Certains moteurs de recherches possèdent leur propre robot d'indexation. Par exemple, Lycos [4] et son

4. `http://www.lycos.com/`

robot « Lycos Spider »ou Google [5] et son robot « Googlebot ».

Concernant les images hors internet (provenant de collections personnelles), les mots sont extraits automatiquement à partir du nom des images, de leurs métadonnées ou annotations (le Chapitre 4 est consacré aux différentes techniques d'annotation).

A ce niveau, on obtient un ensemble de termes, dits termes d'indexation, pour chaque image. L'ensemble des termes d'indexation de la base constitue le langage d'indexation.

Puis ce langage est réduit en éliminant les mots trop rares ou trop fréquents du langage, ou non porteurs de sens. Les mots non porteurs de sens sont en général des pronoms, des déterminants, des conjonctions de coordination, des prépositions, *etc.*. Citons, par exemple : the, of, to, in et and en anglais et le, la, les, et, car, ... en français.

Les termes très rares sont souvent supprimés du langage. Cette suppression n'est pas toujours justifiée car certains mots peuvent être très rares mais très informatifs. Cependant, les mots très rares ne peuvent pas être utilisés par des méthodes à base d'apprentissage statistique (voir 2.2.2) du fait de leur très faible occurrence.

Concernant les mots trop fréquents, ils sont assimilés à des mots d'usage courant (par exemple : être, avoir, système, objet, chose , ...). Ces mots ne sont pas, en général, représentatifs du contenu d'une image spécifique,et peuvent être éliminés du langage d'indexation. Cette étape de suppression suppose que l'on dispose d'un thésaurus des mots courants.

Ensuite vient la normalisation du langage : chaque terme est remplacé par sa forme normalisée. Par exemple, la forme canonique d'un verbe conjugué correspond à son infinitif. Pour les autres mots, la forme canonique correspond au masculin singulier de ces mots, ...

L'intérêt de l'indexation automatique réside dans sa capacité à traiter les images nettement plus rapidement que l'approche manuelle, et de ce fait, elle est particulièrement adaptée aux corpus volumineux. Cependant elle présente plusieurs inconvénients majeurs :

- certain sites Web voient leur contenu actualisé plusieurs fois par jour. Dans ce cas il devient difficile pour les robots d'indexation de fournir des collections à jour.
- l'indexation automatique n'est applicable que si les images sont associées à du texte. C'est le cas des images provenant d'Internet, de catalogues, d'encyclopédies, de bibliothèques numériques, *etc.*. Par contre ce n'est pas souvent le cas des images issues de collections de photographies personnelles,
- les images présentes sur une même page Internet risquent d'être associées aux même termes alors qu'elles n'ont aucune raison de l'être,
- le texte accompagnant les images ne décrit pas toujours le contenu de celles-ci. Parfois le contenu de l'image et du texte sont complètement indépendants.

Les deux derniers points sont à l'origine de beaucoup d'erreurs d'indexation.

#### 2.2.1.2 Indexation manuelle

En indexation manuelle, ce sont des opérateurs humains, généralement experts d'un ou plusieurs domaines, qui se chargent de caractériser, selon leurs connaissances propres, le contenu des images. Cette méthode est nécessaire dans le cas d'images provenant de collections de photographies personnelles, par exemple, car les images ne sont pas accompagnées de texte. La collection d'images est alors annotée manuellement, c'est-à-dire que chaque image est associée à un petit texte (commentaire), ou un ensemble de mots-clés.

Cette approche présente deux inconvénients :

- elle est subjective, puisque le choix des termes d'indexation dépend des indexeurs et de leurs connaissances des domaines abordés par les images. De ce fait, un manque de cohérence

5. `http://www.google.com/`

peut apparaître entre les mots proposés par des indexeurs différents. Ceci est d'autant plus probables pour les bases d'images généralistes, ne touchant pas à un domaine particulier.
– elle risque de devenir obsolète sur le long terme, car on doit faire face à des corpus d'images de plus en plus volumineux et dont le contenu évolue régulièrement.

Néanmoins, tel que rapporté dans [Savoy 05], l'indexation manuelle est plus performante que l'indexation automatique, car les indexeurs organisent les données et choisissent leurs termes d'indexation de façon à retrouver facilement les images.

#### 2.2.1.3 Indexation semi-automatique

L'indexation semi-automatique [Jacquemin 02] est une combinaison des deux approches d'indexation précédentes. Par exemple, on peut définir un vocabulaire contrôlé, sous forme de thésaurus. L'indexation automatique va être utilisée normalement pour extraire des termes « candidats »d'une base d'images. Un processus d'appariement sera utilisé pour comparer les termes candidats aux termes du thésaurus. Le langage d'indexation final sera constitué des termes candidats retrouvés dans le thésaurus.

Une autre forme d'indexation semi-automatique consiste à utiliser l'indexation automatique pour fournir un ensemble de termes pour chaque image. Ensuite, un indexeur humain peut intervenir pour sélectionner parmi les termes ceux qu'il juge les plus pertinents. On pourra parler d'indexation semi-supervisée, pour faire un parallèle avec les techniques d'apprentissage semi-supervisées, dans lesquels l'utilisateur intervient à certains moments de l'apprentissage (voir Chapitre 3).

#### 2.2.1.4 Organisation des caractéristiques textuelles

Une fois qu'un ensemble de termes normalisés a été constitué pour chaque image de la base, il est parfois nécessaire de les organiser de façon à pouvoir retrouver, ensuite, le plus rapidement possible (en limitant la quantité de données textuelles examinées pendant la recherche), des images recherchées. En effet, l'organisation des données n'est pas obligatoire, mais, dès lors que la base de recherche devient conséquente et que les méthodes de recherche utilisées nécessitent le stockage des données textuelles, elle devient vivement conseillée. Cette organisation consiste à la construction de structures de données, appelées index.

Les deux formes d'index les plus populaires sont les « fichiers inversés »[Harman 92] et les « signatures numériques »[Faloutsos 92]. Un index sous forme de fichier inversé est une liste associant, à chaque terme d'indexation, les identifiants des images indexées par ce terme. Dans le cas des signatures numériques, l'indexation se fait par table de hachage. Une fonction de hachage associe une signature à chaque image à partir des termes l'indexant. Certaines études ont montré que les fichiers inversés sont en général plus performants, en terme de rapidité de recherche, pour l'indexation de données textuelles, que les signatures numériques [Zobel 98]. Cependant, les signatures permettent de construire l'index plus rapidement et semblent donc plus appropriées pour la recherche dans de grandes bases de données où le vocabulaire est large [Carterette 05].

Outre ces deux méthodes d'indexations, les bases de données, qui gèrent leur propre système d'index (en général des B-arbres, plus connus sous le nom de B-trees, en anglais [Bayer 72]), peuvent être utilisées pour organiser les termes d'indexation décrivant les images.

Enfin, les termes d'indexation peuvent être pondérés. Cette pondération consiste à associer un poids d'importance (ou degré de représentativité) à chaque terme d'une image. Cette pondération

peut être manuelle et fonction de la catégorie grammaticale des termes : par exemple les verbes, noms communs et noms propres ont des poids supérieurs aux adjectifs et adverbes.

Elle peut aussi être automatique et basée sur le modèle $tf * idf$ (de l'anglais « term frequency-inverse document frequency »), proposé par Salton [Salton 86] pour représenter l'importance d'un mot par rapport à un document, comparé aux autres documents. Dans ce cas, le poids augmente proportionnellement en fonction du nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus. Pour un terme $t_j$ et un document $d_i$, cette mesure est obtenue en prenant le produit de la fréquence du terme dans le document (term frequency) $tf_{i,j}$ et de la fréquence inverse de document (inverse document frequency) $idf_i$. La fréquence du terme du document est simplement le nombre d'occurrences de ce terme dans le document considéré. Cette somme est en général normalisée pour éviter les biais liés à la longueur du document :

$$tf_{i,j} = \frac{n_{j,i}}{\sum_{k=1}^{n} n_{k,i}}$$

où

$n_{i,j}$ est le nombre d'occurrences du terme $t_j$ dans $d_i$ et,

$n$ est le nombre de termes d'indexation.

La fréquence inverse de document est une mesure de l'importance du terme dans l'ensemble du corpus :

$$idf_j = log \frac{|D|}{|\{d_i : t_j \in d_i\}|}$$

où

$|D|$ est le nombre total de documents dans le corpus et le dénominateur désigne le nombre de documents où le terme $t_j$ apparaît.

Cette étape d'organisation, et éventuellement de pondération, des données, est très importante, car elle va permettre de retrouver les images rapidement, même dans de grandes bases d'images, en limitant le nombre de comparaisons. Elle constitue l'indexation à proprement parler, car elle représente la création d'index à partir des termes préalablement extraits. Cependant, le processus d'indexation relève à la fois de l'extraction des termes, puis de leur indexation (*i. e.* de la création d'index à partir de ces termes).

### 2.2.2 Recherche textuelle

Une fois les images indexées textuellement, le problème est de pouvoir les retrouver simplement. La recherche d'images se traduit alors par la mise en correspondance des représentations sémantiques des images et d'une représentation sémantique de la requête. Le terme « représentation sémantique »d'une image désigne l'ensemble des mots-clés la caractérisant. Il s'agit de la représentation interne de type textuel, d'une image. La mise en correspondance peut se faire grâce à des modèles de recherche d'information (RI) dans les documents textuels [Salton 68a]. On peut aussi parler de modèle de recherche de documents textuels, car ces modèles permettent de déterminer si un document répond ou ne répond pas à une question (requête). Pour adapter ces modèles à la recherche d'images, il suffit de considérer chaque image comme un document composé des termes (mots-clés) la décrivant et préalablement extraits. L'indexation permet de déterminer ces termes représentatifs des images et des requêtes (leur représentation sémantique), mais c'est le modèle de recherche qui va permettre d'interpréter et reformuler les requêtes à partir des termes les représentant, en vue de calculer le degré de similarité entre les requêtes et

chaque image de la base, à partir de leur représentation sémantique. Un système de recherche textuelle d'images peut finalement être décrit par le schéma de la figure 2.2.

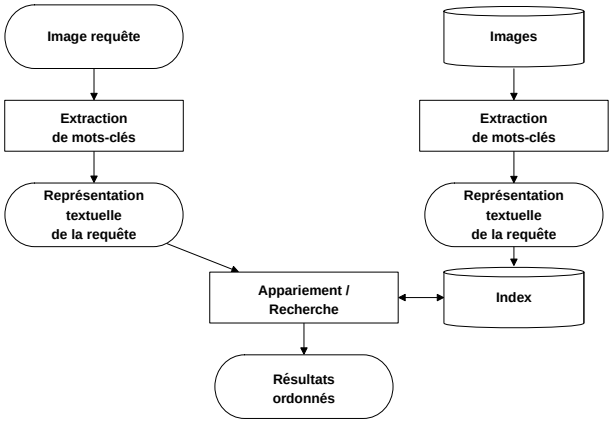



FIGURE 2.2 – Système de recherche textuelle d'images

Il existe trois grandes catégories de modèles de recherche de documents textuels : les modèles booléens [Cooper 70, Lashkari 09], les modèles vectoriels [Salton 71, Demartini 09] et les modèles probabilistes [Robertson 76, Losada 08]. Dans les sections 2.2.2.1, 2.2.2.2 et 2.2.2.3, nous décrivons, pour chacune de ces catégories, le modèle de base et quelques extensions de ces modèles. Pour plus d'informations, on renverra le lecteur au Chapitre 1 de la thèse [Boubekeur-Amirouche 08], dédiée à la recherche d'information, sur laquelle nous nous sommes appuyés pour décrire ces modèles.

#### 2.2.2.1 Modèles booléens

**Modèle booléen standard**

Dans le modèle [Cooper 70], chaque document est représenté par un ensemble de mots-clés (ou termes). Comme son nom l'indique, il a recours aux opérateurs booléens ($AND$, $OR$ et $NOT$) : une requête est une expression logique composée de termes connectés par des opérateurs booléens. Un document est sélectionné si et seulement si il satisfait l'expression booléenne.

Ce modèle est simple à mettre en œuvre et offre de bonnes performances, en termes de temps de calcul et de satisfaction de requêtes, et ce même sur de grandes collections de documents [Frakes 92]. De plus, il permet à l'utilisateur d'exprimer ses besoins de façon structurée. La possibilité d'utiliser des synonymes à l'intérieur d'une requête (grâce à l'opérateur $OR$) et de faire des phrases (grâce à l'opérateur $AND$) est utile pour formuler les requêtes [Marcus 91].

Par contre, il présente plusieurs inconvénients :

- certains trouvent que les requêtes booléennes sont difficiles à formuler [Belkin 92, Cooper 88]. En effet, les expressions booléennes ne sont pas accessibles à un large public et des confusions existent du fait de la différence de « sens »des opérateurs logiques $AND$ et $OR$ lorsqu'ils sont utilisés dans une requête et de leurs connotations respectives en langage naturel. Ils ont aussi des difficultés à utiliser les parenthèses. En partie à cause de cela, les expressions booléennes données par les utilisateurs correspondent souvent mal à leurs besoins. La qualité de la recherche s'en ressent.
- le modèle Booléen considère chaque terme comme étant absent ou présent d'un document. De ce fait, la correspondance entre un document et une requête est soit 1, soit 0. Ce

modèle ne permet donc de classer les documents que dans deux catégories, l'ensemble des documents pertinents et l'ensemble des documents non pertinents. Ce modèle ne permet donc pas de retrouver les documents ne correspondant que partiellement à la requête (appariement partiel).
– Enfin, tous les termes d'un document ou d'une requête sont d'égale importance. En effet ils sont pondérés à 1 si le terme se trouve dans le document et 0 sinon. Il est donc difficile d'exprimer qu'un terme est plus important qu'un autre dans leur représentation.

En conséquence, le système de recherche d'information textuelle (SRI) détermine un ensemble de documents non-ordonnés comme réponse à une requête. Il n'est pas possible de dire quel document est mieux qu'un autre. Cela crée beaucoup de problèmes aux utilisateurs, car ils doivent encore fouiller dans cet ensemble de documents non-ordonnés pour trouver des documents qui les intéressent. C'est difficile dans le cas où beaucoup de documents répondent aux critère de la requête

**Modèle booléen étendu** Le modèle booléen standard est simple et relativement efficace, mais il ne permet pes de classer les documents retrouvés. Le modèle standard a donc été étendu par Salton en 1983 [Salton 83], afin de prendre en compte les notions de pondération des termes (à la fois dans les documents et la requête) et d'appariement partiel.

En général, le poids d'un terme dans un document est fonction de la fréquence de ce terme dans le document et de la fréquence de ce terme dans le corpus (ensemble des documents disponibles) (voir modèle $tf * idf$ section 2.2.1.4). La requête demeure une expression booléenne classique, mais où les termes sont pondérés.

L'avantage de ce modèle par rapport au modèle standard se situe donc au niveau de la représentation des documents : grâce à la pondération des termes, on a une représentation plus raffinée. On peut exprimer dans quelle mesure un terme est important dans un document. De plus, cette extension du modèle standard permet un classement des résultats, mais selon des préférences exprimées par l'utilisateur dans sa requête.

Finalement le modèle booléen étendu est clairement plus performant que le modèle standard. Par contre il est plus complexe d'un point de vue calculatoire. Enfin, la distributivité de l'opérateur $AND$ n'est pas prise en compte dans l'ordonnancement des documents retrouvés. Soient $q_1$ et $q2$ deux requêtes, $t_1, t_2$ et $t_3$ des termes de la requête :

$$q_1 = (t_1 OR t_2) AND t_3 \, q_2 = (t_1 OR t_3) AND (t_2 OR t_3)$$

Soit $d_i$ un document retrouvé et $sim$ la mesure de similarité entre une requête et un document. Alors $sim(q_1, d_i) \neq sim(q_2, d_i)$ .

**Modèle booléen basé sur les ensembles flous** Le modèle booléen basé sur les ensembles flous est une autre extension du modèle booléen standard. Les requêtes et les documents sont représentés par des ensembles de termes d'indexation. Par contre, à la différence du modèle standard et du modèle étendu, l'appariement n'est pas strict. La similarité entre requête et document n'est pas évaluée par une fonction binaire mais par une fonction à valeurs dans $[0, 1]$. Pour plus d'informations quant à l'évaluation du degré de correspondance entre une requête et un document donné, on conseillera les lectures suivantes [Zadeh 65, Lukasiewicz 63].

Si on compare cette extension des deux modèles précédents, il est assez facile de voir son avantage : on peut mesurer le degré de correspondance entre un document et une requête dans $[0, 1]$. Ainsi, on peut ordonner les documents dans l'ordre décroissant de leur correspondance avec la requête. L'usager peut parcourir cette liste ordonnée et décider où s'arrêter.

#### 2.2.2.2 Modèles vectoriels

**Modèle vectoriel standard**

Une autre technique de RI consiste à utiliser le modèle vectoriel [Salton 71]. Dans ce modèle, un document est représenté par un vecteur de dimension $n$, dont les dimensions sont les termes d'indexation. Les coordonnées d'un vecteur document représentent les poids des termes correspondants. Formellement, un document $d_i$ est représenté par un vecteur de dimension $n$ :

$$d_i = (w_{i,1}, w_{i,2} \dots w_{i,n}), \forall i \in \{1, 2 \dots m\}$$

où

$w_{ij}$ est le poids du terme $t_j$ dans le document $d_i$, $m$, le nombre de documents dans la collection et $n$, le nombre de termes d'indexation du document $d_i$.

Une requête $q$ est aussi représentée par un vecteur défini dans le même espace vectoriel que le document :

$$q = (w_{q,1}, w_{q,2} \dots w_{q,n})$$

où

$w_{qj}$ est le poids de terme $t_j$ dans la requête $q$.

Ce poids peut être attribué manuellement par l'utilisateur ou être une forme de $tf * idf$ (voir 2.2.1.4).

La pertinence du document $d_i$ pour la requête $q$ est mesurée comme le degré de corrélation des vecteurs correspondants. Cette corrélation peut être exprimée par une des mesures de similarité sémantique classiques suivantes : le produit scalaire, la mesure du cosinus, la distance euclidienne, la mesure de Dice, la mesure de Jaccard, ... [Duda 01]. De telles mesures déterminent la ressemblance entre un document et une requête sur la base de la comparaison locale des termes qu'ils ont en commun.

Le modèle vectoriel standard présente l'inconvénient théorique de considérer que tous les termes sont indépendants [Berry 99]. De plus, le langage de requête est moins expressif que les expressions booléennes utilisées dans les modèles booléens. Par exemple, l'opérateur $NOT$ existant dans le modèle booléen ne peut pas être représenté dans le modèle vectoriel, de part l'utilisation de poids uniquement positifs. Enfin, de part l'appariement requête-document non strict, le fait qu'un document soit retrouvé plutôt qu'un autre est moins clair pour l'utilisateur.

Par contre, le modèle vectoriel standard a l'avantage de posséder un langage de requête plus simple que les expressions booléennes utilisées dans les modèles booléens. En effet, dans le cas du modèle vectoriel, les requêtes sont représentées par une simple liste de termes pondérés. De plus, grâce à la pondération des termes, les performances de ce modèle sont meilleures que celles des modèles booléens. Enfin, le modèle vectoriel standard permet des réponses plus précises aux requêtes : les documents répondant partiellement aux requêtes peuvent être retrouvés, et, surtout, les documents retrouvés peuvent être triés par ordre de pertinence grâce à la fonction d'appariement requête-document. Ces avantages font du modèle vectoriel standard un des modèles les plus populaires en RI [Salton 68b].

**Modèle connexionniste** Les SRI basés sur l'approche connexionniste utilisent le fondement des réseaux de neurones, tant pour la modélisation des termes que pour la mise en œuvre du processus de RI.

Deux modèles théoriques ont été utilisés : les modèles à auto-organisation [Lin 91] et les modèles à couches [Mothe 94]. Comme dans le modèle vectoriel standard, chaque document et

requête est représenté par un vecteur de poids. Les poids sont initialisés manuellement et mis à jour par la suite grâce à un processus d'apprentissage. En effet, la majorité des réseaux de neurones possède un algorithme « d'entraînement »qui consiste à modifier les poids synaptiques en fonction d'un jeu de données présentées en entrée du réseau. Le but de cet entraînement est de permettre au réseau de neurones « d'apprendre »à partir des exemples. On parle « d'apprentissage par expérience », et, de façon plus générale « d'apprentissage ».

Les modèles connexionnistes offrent des atouts intéressants pour la représentation des relations entre termes (par exemple la synonymie), entre documents (par exemple la similitude) et entre termes et documents (par exemple le poids d'un terme dans un document). De plus, leurs capacités d'apprentissage (qui sont à l'origine de la mise à jour des poids dans le réseau), permettent d'obtenir des SRI adaptatifs, c'est-à-dire des SRI permettant les processus de retour de pertinence (relevance feedback) et de reformulation de requête [Choi 99, Chen 06].

Les modèles développés dans la littérature ont fait leurs preuves [Syu 94]. Cependant, ils restent moins populaires que le modèle vectoriel standard, de part les problèmes liés à la définition de la structure du réseau et du coût de convergence de l'algorithme d'apprentissage [Turtle 92].

**Modèle d'indexation sémantique latente (LSI Latent Semantic Indexing)** L'objectif du modèle LSI est de transformer une représentation par des mots-clés en une autre qui soit telle que les documents et requêtes sémantiquement similaires sont plus proches avec la nouvelle représentation qu'avec la représentation par mots-clés initiale.

Pour ce faire, en partant de l'espace vectoriel de tous les termes d'indexation, le modèle LSI construit un espace d'indexation de taille réduite $k$, par application de la décomposition en valeurs singulières de la matrice de co-occurrence termes-documents, qui décrit les occurrences de termes dans les documents [Deerwester 90].

Dans ce nouvel espace de dimension $k$, les documents sémantiquement proches se sont « rapprochés ». Lorsqu'une requête est soumise, il est aussi possible de la représenter dans cet espace vectoriel de dimension réduite $k$. Un calcul de similarité (distance normalisée) permettra de mesurer la similarité entre la requête et chaque document de la base, et de retourner les documents les plus proches.

Ce modèle a montré des performances très intéressantes. Un de ses avantages est que des documents peuvent être retrouvés même s'ils n'ont aucun mot en commun avec la requête. Pour un corpus de petite taille, sa performance est supérieure au modèle vectoriel standard [Foltz 92]. Cependant, lorsque la taille du corpus de documents augmente, la différence avec le modèle standard semble négligeable. C'est la raison pour laquelle le modèle standard reste le modèle le plus populaire [Letsche 97].

#### 2.2.2.3 Modèles probabilistes

**Modèle probabiliste standard**

Le premier modèle probabiliste a été proposé par Maron et Kuhns [Maron 60] au début des années 60. Ce modèle est basé sur le modèle PRP [Robertson 77] (Probability Ranking Principle), qui a établi qu'un système de recherche d'information est supposé ordonner les documents retrouvés en fonction de leur probabilité de pertinence vis à vis d'une requête.

Le modèle probabiliste standard suppose que les documents peuvent être classés en deux classes : celle des documents pertinents, notée $R$ (de l'anglais « Relevant », signifiant « pertinent »), composée des documents que l'utilisateur souhaite retrouver parmi l'ensemble des documents disponibles, et celle des documents non pertinents, notée $\bar{R}$, composée du reste des

documents. Les ensembles $R$ et $\bar{R}$ sont donc disjoints et considérés comme deux variables aléatoires indépendantes.

Étant donnée une requête utilisateur, notée $q$, et un document $d_i$, le modèle probabiliste tente d'estimer la probabilité que le document $d_i$ appartienne à la classe des documents pertinents pour $q$. Comme dans le modèle vectoriel, $d_i$ et $q$ sont représentés par un vecteur de poids. Par contre ces vecteurs sont booléens *i. e.* qu'un poids vaut 1 si le terme correspondant se trouve dans le document et 0 sinon. Un document est alors sélectionné si la probabilité qu'il soit pertinent pour la requête $q$, notée $P(R|d_i)$, est supérieure à la probabilité qu'il soit non pertinent pour la requête $q$, notée $P(\bar{R}|d_i)$. Cette comparaison équivaut à calculer le degré de similarité entre le document $d_i$ et la requête $q$, noté $RSV(d_i, q)$, donné par :

$$RSV(d_i, q) = \frac{P(R|d_i)}{P(\bar{R}|d_i)}$$

En utilisant la formule de Bayes, on peut écrire :
$P(R|d_i) = \frac{P(R,d_i)}{P(d_i)}$ et $P(\bar{R}|d_i) = \frac{P(\bar{R},d_i)}{P(d_i)}$
et
$P(R, d_i) = P(d_i|R) \times P(R)$ et $P(\bar{R}, d_i) = P(d_i|\bar{R}) \times P(\bar{R})$
et
$P(d_i) = P(d_i|R) \times P(R) + P(d_i|\bar{R}) \times P(\bar{R})$
où
$P(\bar{R}) = 1 - P(R)$
Ainsi on peut simplifier $RSV$ :

$$RSV(d_i, q) = \frac{P(R, d_i)}{P(\bar{R}, d_i)} = \frac{P(d_i|R) \times P(R)}{P(d_i|\bar{R}) \times P(\bar{R})}$$

Si $RSV(d_i, q) > 1$ ou si $log(RSV(d_i, q)) > 0$ alors le document $d_i$ est pertinent pour la requête $q$.

L'inconvénient majeur de ce modèle réside dans le calcul des probabilités $P(R|d_i)$. En effet, ce calcul est difficile car on ne sait pas mesurer la pertinence d'un document pour un humain. Cependant, des modèles existent pour estimer ces probabilités. Le plus connu consiste à utiliser la règle de Bayes pour calculer $P(R|d_i)$ à partir des probabilités connues $P(R)$ (probabilité d'obtenir un document pertinent en piochant au hasard) et $P(d_i)$ (probabilité de piocher le document $d_i$ au hasard), et de la probabilité $P(d_i|R)$ que l'on peut obtenir par apprentissage à partir d'un ensemble de requêtes déjà résolues (cet ensemble est appelé échantillon d'apprentissage ou d'entraînement) et de la pondération des termes dans les documents. Pour des détails concernant les modèles d'estimation de ces probabilités, on conseillera au lecteur le troisième chapitre du livre [Madjid 04].

Malgré l'existence de ces modèles, le problème d'estimation des probabilités initiales (qui constituent les paramètres du modèle probabiliste standard) persiste donc. En effet, en l'absence d'échantillon d'apprentissage, cette estimation est difficile. De même, si on dispose d'un échantillon d'apprentissage mais qu'il est de petite taille, les probabilités seront mal estimées. Enfin, comme le modèle vectoriel standard, le modèle probabiliste standard présente l'inconvénient de supposer que la présence d'un terme dans un texte (document) est indépendant de la présence des autres termes, et que la pertinence d'un document ne dépend que des termes qu'il contient.

Par contre, cette approche probabiliste bénéficie de nombres d'avantages. Elle fournit aux utilisateurs un classement des documents retrouvés. En outre, les utilisateurs peuvent contrôler

le nombre de documents retrouvés en utilisant un seuil de pertinence au niveau des probabilités. Les requêtes sont aussi plus faciles à formuler qu'avec le modèle Booléen, car elles ne nécessitent pas l'apprentissage dans un langage de requêtes : les utilisateurs peuvent utiliser le langage naturel.

Les résultats du modèle probabiliste standard sont comparables à ceux du modèle vectoriel standard [Grossman 04, Jones 00]. Cependant, le modèle vectoriel reste le plus populaire car il ne nécessite pas de données d'apprentissage.

**Modèles de langage** Dans les modèles présentés ci-dessus, le texte est traité comme un « sac de mots », c'est-à-dire que l'ordre des termes dans le texte n'est pas pris en compte. Les termes d'un document sont supposés indépendants les uns des autres. En utilisant un modèle basé sur l'analyse des « $n$-grammes », cette hypothèse n'est plus nécessaire. La notion de « $n$-grammes »a été introduite par Claude Shannon dans ses travaux en théorie de l'information [Shannon 51], où il s'intéressait à la prédiction d'apparition de certains caractères en fonction des $n$ caractères précédents. Un « $n$-grammes »désignait alors une séquence de $n$ caractères. Les modèles de $n$-grammes généralisent cette notion à celle de sous-séquence de $n$ éléments construite à partir d'une séquence donnée. Ces modèles sont beaucoup utilisés en traitement automatique du langage naturel, c'est pourquoi on parle de modèles de langage. Dans ces modèles, un $n-grammes$ désigne une séquence ordonnée de $n$ termes. De plus, ces modèles reposent sur l'hypothèse simplificatrice que la prédiction d'un mot ne dépend que de la séquence des $n-1$ mots qui le précèdent.

Ces modèles sont différents des approches classiques en RI. En effet, plutôt que d'évaluer le degré de similarité entre documents et requêtes, les modèles de langage considèrent que la pertinence d'un document pour une requête est en rapport avec la probabilité que la requête puisse être générée par le document [Ponte 98]. C'est-à-dire, que, si la requête de l'utilisateur est une suite de mots, l'objectif est de calculer la probabilité de chaque document disponible, à partir du modèle de la requête. Ce calcul sera simplifié grâce à l'hypothèse simplificatrice évoquée ci-dessus : il suffira de calculer la probabilité du dernier mot de la requête étant donnée la suite de ses précédents.

L'inconvénient majeur de ce type de méthode est que, lorsque le $n$-grammes correspondant à la requête n'apparaît pas dans le corpus d'apprentissage, sa probabilité est systématiquement nulle. En effet, il est difficile de construire un corpus suffisamment représentatif pour contenir, de façon justement distribuée (c'est-à-dire correspondant à la distribution réelle) l'ensemble des $n$-grammes d'un langage. Ceci est d'autant plus probable si $n$ (la taille de l'expression) est grand. De ce fait, les modèles bi et trigrammes (avec un historique de un ou deux mots respectivement), sont le plus souvent utilisés. De plus, afin de pallier ce problème, des techniques de lissage peuvent être utilisées. Le lissage consiste à assigner des probabilités non nulles aux termes qui n'apparaissent pas dans les documents.

Les performances des modèles de langage étant dépendantes de la disponibilité de données pour l'apprentissage des paramètres nécessaires, elles sont comparables à celles obtenues par le modèle probabiliste standard. Cependant, ce modèle est plus précis car il permet de prendre en compte l'ordre des mots dans les documents. De plus, l'appariement entre documents et requête, qui se fait différemment dans les autres modèles de RI présentés, améliore les poids utilisés et en fait un modèle plus performant [Bennett 07].

Il existe d'autres modèles probabilistes utilisés en RI, comme les réseaux Bayésiens. Ces derniers ont fait leur apparition dans le domaine de la RI dans les années 90 [Callan 92, Turtle 91]. Comme dans le modèle probabiliste standard, ils permettent de calculer la probabilité qu'un

document représente bien une requête. Pour plus d'informations sur l'utilisation des réseaux Bayésiens en RI, on renverra le lecteur aux références suivantes [Myaeng 98, Ribeiro 96]. De plus, le chapitre 5 de cette thèse fait l'objet d'une étude des réseaux Bayésiens.

### 2.2.3 Reformulation des requêtes

Quel que soit le modèle de recherche utilisé, son efficacité est basée sur la capacité de la requête posée par un utilisateur à traduire son besoin d'information. Or, nous avons vu, dans la section 2.2.2, que l'utilisateur a parfois du mal à choisir les bons termes qui expriment le mieux ses besoins d'information, en particulier à cause de la subjectivité des termes d'indexation. La requête permet donc rarement d'aboutir à un résultat qui satisfait l'utilisateur.

Afin de pallier ce problème, la reformulation de requête a été introduite en RI. Le processus de RI est alors envisagé comme une suite de formulations et de reformulations de requêtes jusqu'à la satisfaction du besoin d'information de l'utilisateur. On distingue deux types de méthodes de reformulation de requête : les méthodes « globales »et les méthodes « locales ».

De façon générale, les méthodes globales [Navigli 03, Qiu 93] se basent sur l'expansion de requêtes en s'appuyant sur des ressources linguistiques (thésaurus ou ms), ou sur des techniques d'associations de termes telles que les règles d'association. C'est-à-dire que, pour chaque terme $t$, la requête peut être automatiquement étendue avec des termes extraits d'un thésaurus, synonymes de $t$, ou en relation avec $t$. Le système peut ainsi apparier la requête à des documents pertinents qui ne contiennent aucun des mots de la requête initiale.

Les méthodes locales ajustent une requête relativement aux documents qui sont retournés comme documents pertinents pour la requête initiale. Elles se basent sur la technique dite de retour (ou réinjection) de pertinence (relevance feedback) [Rocchio 71]. L'idée du retour de pertinence est de faire participer l'utilisateur dans le processus de recherche de sorte à améliorer l'ensemble final de résultats. Le procédé de base est le suivant :

- l'utilisateur formule sa requête initiale,
- le système renvoie un premier ensemble de résultats de recherche,
- l'utilisateur estime la pertinence de quelques documents retournés (par exemple en attribuant un poids de 1 aux documents qu'il considère pertinents, 0 sinon),
- le système calcule une meilleure représentation des besoins de l'utilisateur, à partir des documents jugés pertinents et non pertinents. Ceci aboutit à une nouvelle requête,
- le système renvoie un nouvel ensemble de résultats pour la nouvelle requête.

Le processus de retour de pertinence peut comporter plusieurs itérations de ce procédé de base.

Ce procédé de base, en recherche d'images, peut être représenté par le schéma de la figure 2.3.

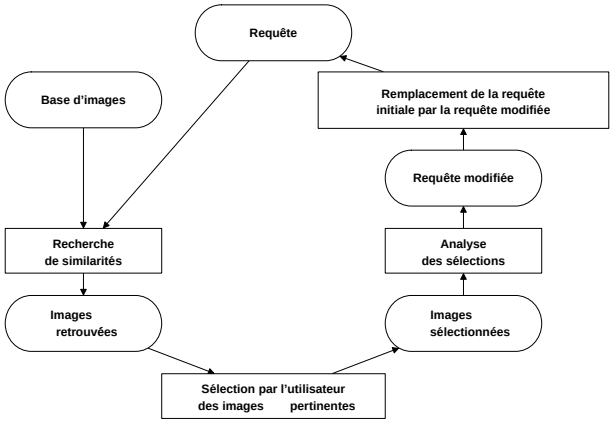



FIGURE 2.3 – Processus de base du retour de pertinence en recherche d'images

Les documents considérés et marqués comme pertinents par l'utilisateur sont appelés exemples positifs. Au contraire, les documents considérés et marqués comme non pertinents par l'utilisateur sont appelés exemples négatifs.

Chaque nouvelle requête, notée $q_1$, peut être construite grâce à la formule, souvent utilisée, de Rocchio [Rocchio 71], donnée ci-dessous, dont l'idée est d'ajouter à la requête initiale, notée $q_0$ les termes des documents pertinents (l'ensemble des documents pertinents est noté $R$) et de lui retrancher les termes des documents non pertinents (l'ensemble des documents non pertinents est noté $\bar{R}$). $d_i$ désigne un vecteur document.

$$q_1 = \alpha q_0 + \frac{\beta}{|R|} \sum_{d_i \in R} d_i - \frac{\gamma}{|\bar{R}|} \sum_{d \in \bar{R}} d_i$$

Les méthodes locales sont plus populaires que les méthodes globales, car elles ne nécessitent pas de thésaurus. Cependant, les méthodes locales, basées sur le retour de pertinence, présentent l'inconvénient d'être coûteuses pour l'utilisateur, qui, en plus de la formulation de sa requête initiale, doit marquer certains documents comme pertinents ou non à chaque renvoi du système des résultats d'une recherche. Pour pallier ce problème, il est possible de simuler l'interaction d'un utilisateur en supposant que les dix premiers documents trouvés par une première recherche sont pertinents et les cent derniers sont non pertinents. Cette adaptation n'empêche pas le processus de réinjection de pertinence d'être coûteux pour le système, son coût étant lié au nombre d'itérations du procédé de base décrit ci-dessus. Néanmoins, la reformulation automatique d'une requête grâce à l'utilisation de documents pertinents et non pertinents à été un succès de la recherche d'informations [Buckley 94].

### 2.2.4 Comparaison des modèles et conclusion sur l'indexation et la recherche textuelles d'images

Nous avons présenté dans cette section les concepts fondamentaux de la RI dans les documents textuels. Nous y avons en particulier exposé les techniques d'indexation automatique et les principaux modèles de recherche.

Les premiers modèles mis en place sont des modèles booléens, basés sur la théorie des ensembles. Ils sont simples à concevoir. Par contre, l'appariement utilisé est strict et ne permet de classer les documents que dans deux catégories : l'ensemble des documents pertinents et

l'ensemble des documents non pertinents. Ces modèles ne permettent donc pas de retrouver les documents répondant partiellement à la requête. Enfin, certains utilisateurs éprouvent des difficultés à construire des requêtes booléennes.

Les modèles vectoriels, algébriques, offrent la possibilité d'ordonner les documents retrouvés selon leur degré de similarité avec la requête. La facilité d'utilisation du modèle et sa robustesse, même si son calcul est coûteux, ont fait de lui un des modèles les plus populaires de RI. Cependant, ces modèles considèrent que tous les termes sont indépendants, ce qui constitue un inconvénient théorique important, car l'indépendance des termes se vérifie rarement, en réalité. De plus, leur pouvoir expressif est limité, contrairement aux modèles booléens. En particulier, l'opérateur *NOT* du modèle booléen n'a pas d'équivalent dans le modèle vectoriel.

Enfin, les modèles probabilistes, basés sur la théorie des probabilités, ont des performances comparables à celles des modèles vectoriels. Les documents y sont ordonnés selon leur probabilités de pertinence pour la requête. De plus, les requêtes sont plus faciles à exprimer qu'avec les modèles booléen. Leur inconvénient, cependant, réside dans l'estimation des probabilités utilisées pour l'évaluation de la pertinence.

Compte tenu des avantages et inconvénients que nous venons de citer pour les trois catégories de modèles de recherche, il en ressort que la plupart de ces modèles sont relativement efficaces. Tous ces modèles peuvent être appliqués à la recherche textuelle d'images [Rasiwasia 08, Rasiwasia 07, Fan 06]. Il suffit d'indexer les images textuellement puis de considérer chaque image (de la base de recherche ou image requête) comme un sac de mots, les mots étant les termes d'indexation. Finalement, les modèles de recherche d'information textuelle sont utilisés en considérant chaque image comme un document et la base d'images constitue l'ensemble des documents. Afin de tester l'efficacité des méthodes de recherche d'images par le texte, il existe des bases d'images standards, comme celles utilisées dans les Workshops ImageCLEF, dédiées à différents domaines d'application : des collections photographiques [Clough 07], des bases d'images médicale [Müller 08], etc.

L'approche [Rasiwasia 08] est originale dans le sens où la requête est uniquement textuelle, il n'y a pas d'image exemple. Par contre elle présente l'inconvénient de nécessiter que toutes les images de la base soient indexées textuellement, ce qui se révèle coûteux pour l'utilisateur dans le cas d'une indexation manuelle.

Appliqués à des images préalablement indexées textuellement, les modèles de RI permettent de retrouver rapidement et efficacement des images correspondant à une requête. Par contre, qu'ils soient stricts, pour les uns (booléens), ou flexibles, pour les autres (vectoriels et probabilistes), les modèles standards ne prennent pas en compte les préférences de l'utilisateur sur les critères de recherche et il est difficile de formuler des requêtes complexes. Par exemple, dans le modèle booléen standard, l'agrégation utilisée est exclusivement de type conjonctif (AND) et/ou disjonctif (OR). De ce fait, une requête telle « trouver un ballon rouge près d'un vélo, sur fond vert », va être très difficile à formuler. En effet, les utilisateurs manipulent mal les expressions logiques et construisent donc difficilement des expressions correspondant à leurs besoins, d'autant plus que les requêtes sont complexes. Afin de pallier ce problème de formulation et ainsi d'améliorer les résultats de recherche, des méthodes de reformulation de requêtes et de retour de pertinence existent [Lioma 08, Urban 06].

De plus, un problème majeur réside dans la phase d'indexation précédant la recherche. En effet, les méthodes d'indexation textuelle automatiques sont peu performantes et fournissent des ensembles d'images mal annotées, car elles utilisent l'URL, le titre de la page ou le texte proche de l'image dans le cas d'images provenant d'Internet, ou alors tout simplement le nom de l'image dans le cas d'images issues de collections personnelles. Pourtant, toutes les images présentes sur une même page ne devraient pas être indexées par les mêmes termes. De même, la plupart

des images ne sont pas nommées de façon pertinente, mais bien souvent par des noms générés automatiquement par les appareils numériques, comme « IMGP001.JPG », et qui ne portent pas de sens. De plus, ces images, issues de collections personnelles, sont rarement renommées par les utilisateurs, à cause du travail fastidieux que cela représente, et surtout parce qu'ils connaissent leurs images et ne voient pas l'utilité de l'annotation ou du renommage.

Quant à l'indexation textuelle manuelle, bien qu'elle soit plus performante que l'indexation textuelle automatique, elle est très coûteuse pour l'utilisateur et se révèle pratiquement inapplicable aux grandes bases d'images. De plus, elle est très subjective. En effet, une même image peut avoir plusieurs sens et donc contenir plusieurs termes. C'est le cas dans le tableau 2.1 présentant une image et différentes termes susceptibles de l'indexer. On parle de « polysémie »de l'image. De ce fait, la ou les personne(s) qui indexe(nt) une image et la personne qui la recherche ne choisiront pas forcément les mêmes termes pour la décrire.

| | |
|---|---|
|  | cheval<br>animal<br>voiture<br>automobile<br>véhicule<br>route<br>herbe |

TABLE 2.1 – Exemple d'une image et ses éventuels termes d'indexation

De plus, un même terme peut avoir des sens différents. Par exemple, le terme « glace »peut désigner un miroir, un aliment, ou de l'eau glacée.

Afin de pallier cette subjectivité des termes, des ressources sémantiques (thésaurus) ont été construites pour plusieurs langues, tel Wordnet [Fellbaum 98] pour l'anglais. WordNet est une base de données lexicales, définissant, pour chaque mot de la langue anglaise, ses classes d'équivalences sémantiques appelées « synsets ». Chaque « synset »regroupe tous les mots ayant le même sens que le mot considéré. Ainsi, les mots dans un même synset peuvent être considérés comme des synonymes. WordNet établit également d'autres types de relations sémantiques entre mots : des relations hiérarchiques (hyperonymie/hyponymie ou générique/spécifique), et des relations d'association. Des travaux ont été proposés pour le calcul de similarité sémantique entre deux mots dans WordNet [Pedersen 04]. Wordnet fournit donc un thésaurus qui semble être un outil précieux pour réduire la subjectivité inhérente à l'indexation textuelle manuelle. Un problème subsiste, cependant : les termes et relations qu'il contient ne sont pas dédiés à la description d'images. Cet état de fait explique bien des déconvenues en matière de requêtes effectuées à partir des onglets « images »des différents moteurs de recherche traditionnels, qui s'appuient encore majoritairement sur des techniques d'indexation textuelle.

A la différence de l'indexation automatique des textes, les images n'apportent pas directement d'information conceptuelle de haut niveau sémantique. De plus, le résultat d'une recherche textuelle dépend de la langue utilisée pour formuler la requête : celle-ci doit être formulée dans la même langue que la langue utilisée pour l'indexation des images disponibles. Enfin, nous avons vu que le choix des termes d'indexation est très subjectif. L'indexation textuelle ne semble donc pas la plus appropriée en vue de rechercher des images, même si elle reste d'actualité pour satisfaire les utilisateurs qui ne disposent pas d'image exemple pour exprimer leurs besoins, par exemple. A cet effet, de nouvelles approches ont été proposées récemment [Liu 09c, Grangier 08].

Par contre, quand l'utilisateur dispose d'image exemple, il est préférable de développer d'autres index qui soient pertinents pour permettre une recherche rapide et efficace, et des modèles de recherche qui permettent à l'utilisateur d'exprimer facilement ses besoins. A cet effet se développent, depuis plusieurs années, des outils de recherche par le contenu visuel, également désignés sous le sigle générique CBIR (Content-based image retrieval) [Smeulders 00].

## 2.3 Indexation et recherche d'images par le contenu

Les systèmes d'indexation et recherche d'images par le contenu permettent de rechercher les images d'une base en fonction de leurs caractéristiques visuelles. Ces caractéristiques, encore appelées caractéristiques de bas-niveau sont des représentations de la couleur, la texture, la forme, ...

Le principe de cette technologie respecte deux grandes phases. La première étape consiste à extraire les caractéristiques visuelles significatives de l'image toute entière ou de certaines parties (régions) de celle-ci. Cela permet de créer une « signature numérique », ou plus communément « signature », censée représenter l'image dans un index. La deuxième phase est la mise en adéquation entre les attributs choisis pour décrire les images des bases et les requêtes visuelles des usagers afin d'obtenir un appariement satisfaisant. L'appariement se fait grâce à des mesures de distances entre les caractéristiques ou des mesures de similarité globales entre deux images.

Une fois que la similarité entre une image requête et chaque image de la base d'images est calculée, on peut ordonner les images de la base de la plus pertinente à la moins pertinente, et présenter le résultat à l'utilisateur. L'image la plus pertinente est celle qui a le plus grand degré de similarité avec l'image requête.

Dans cette section, nous présentons brièvement les différents types de systèmes de recherche d'images par le contenu existants. Nous évoquons aussi la diversité des bases d'images (section 2.3.1). Ensuite, les différents types de requêtes visuelles possibles sont présentés (section 2.3.2). Puis nous exposons trois méthodes de caractérisation visuelle, fondées respectivement sur l'analyse de la forme (section 2.3.3.1), de la texture (section 2.3.3.2) et de la couleur d'une image (section 2.3.3.3). Dans la section 2.3.3.4, nous présentons les techniques d'indexation multidimensionnelle, permettant d'organiser ces caractéristiques visuelles. Quelques mesures de similarité et de distances, permettant de retrouver les images sur la base de caractéristiques visuelles préalablement extraites, sont présentées section 2.3.4. Enfin, dans la section 2.4, nous présentons notre choix de méthode d'indexation et de caractéristiques.

### 2.3.1 Motivations, applications et bases d'images

Comme nous l'avons vu dans la section 2.2, l'indexation textuelle des images n'est pas des plus efficaces. En effet, l'indexation automatique tient compte du texte que l'on peut trouver dans les pages Web contenant les images, ou dans les métadonnées ou le nom des images, ...

Quant à l'indexation textuelle, elle se révèle en général meilleure que l'indexation automatique, mais pose quand même problème. En effet, à moins que les images à indexer n'appartiennent à un domaine précis et que l'on dispose d'indexeurs humains spécialistes de ce domaine, il est difficile de rendre compte d'une image par des mots. Pour déterminer précisément le sujet d'une image, il est nécessaire de savoir à quoi va servir l'image, d'où elle provient et pourquoi on la recherche. De plus, les images sont polysémiques, *i. e.* que leur sens est défini par ceux qui la regardent. De ce fait, il est probable que des indexeurs différents interprètent une même image de façon différente et qu'ils utilisent des termes différents pour l'indexer.

Ces problèmes sont tous dus au fait que les images n'apportent pas directement d'information sémantique : on parle de fossé sensoriel entre le monde observé et l'image acquise [Smeulders 00, Boucher 05].

Devant ce problème de fossé sensoriel, le problème de recherche d'image a été redéfini : on ne cherche plus à décrire par des mots le contenu des images pour ensuite retrouver des images en utilisant des requêtes textuelles. On recherche maintenant à mesurer la similarité entre les images, sur la base de caractéristiques visuelles (caractéristiques de bas niveau). Cette nouvelle vision de la recherche d'images est plus connue sous le terme de recherche d'images par le contenu (en anglais CBIR, content based image retrieval). Nombre de systèmes d'indexation et de recherche d'images par le contenu ont vu le jour ces dernières années [Oussalah 08, Lew 06].

Les caractéristiques visuelles utilisées et les systèmes de recherche associés sont souvent dédiés à un type d'application, car la robustesse des caractéristiques est très dépendante de la base d'images utilisée, cette dernière étant souvent déterminante dans l'application associée. Parmi les nombreuses applications d'indexation d'images par le contenu, on peut citer : l'authentification (de visages [Visani 05, Toews 09], empreintes digitales [Uz 09, Sheng 09], symboles [Coustaty 08, LaViola 07], logos [Ballan 08, Wei 09]), la médecine (recherche ou détection de « scènes »anormales dans des images radio par exemple, en vue d'un diagnostic [Arzhaeva 09]), l'audiovisuel (par exemple identification de personnages dans les journaux télévisés [Everingham 09a]), l'art ou encore le design (par exemple recherche d'une texture particulière de tissus [Sánchez 08]), l'Internet (par exemple le but des moteurs de recherche et les portails de données sont d'identifier les données, images ou vidéo, correspondant à la recherche d'un utilisateur ou au domaine concerné par le portail (par exemple Google Similar Images [6], ...

Ces exemples nous mènent à distinguer deux types d'applications, correspondant à deux types de bases d'images : les bases généralistes et les bases spécialisées. Le type de base est souvent déterminant sur le type de besoin d'accès aux données : recherche d'images ou classification ? (les différents besoins ayant été présentés dans le Chapitre 1). En effet, les bases généralistes ont un contenu hétérogène : elles sont constituées d'images de couleurs et de formes variées, représentant divers objets. On parlera de bases d'images naturelles. Elles sont plutôt destinées à une utilisation grand public dans un contexte de recherche d'images. Les images que l'on peut trouver sur Internet constituent une large base d'images généralisées. C'est parmi ces images en ligne que les moteurs de recherche, comme Google Similar Images [7], Google Images [8] et Picsearch [9], vont rechercher des images correspondant à une idée, un besoin précis de l'utilisateur.

Au contraire, les bases d'images spécialisées ont un contenu homogène : les images présentent différentes vues d'un objet particulier, ou des objets de même type, ou des objets différents issus d'un domaine particulier. Par exemple, dans le domaine médical, on pourra utiliser des bases d'images de grains de beautés, de forme, de couleur ou d'aspect différents, dans le but d'identifier un cancer de la peau. C'est le cas du site Internet [Snyder 02], qui fournit 3 bases d'images spécialisées et de la base [Martinez 98] qui fournit une base de 4000 images de visages.

Ces bases sont souvent destinées à être utilisées par des experts (c'est le cas de l'exemple de la base d'images de grains de beautés : les experts sont des médecins qui utilisent ces images dans un but décisionnel, lié au diagnostic). Ces bases d'images spécialisées correspondent plus à un besoin de reconnaissance/identification d'objets que de recherche. En effet, reprenons l'exemple de la base d'images de grains de beauté : le but, ici, sera de classer chaque image en deux classes (sujet malade vs. sujet sain). Prenons un autre exemple, celui d'un dessin industriel : dans ce

6. `http://similar-images.googlelabs.com/`
7. `http://similar-images.googlelabs.com/`
8. `http://images.google.com/`
9. `http://www.picsearch.com/`

cas le but sera d'identifier chaque élément / symbole du schéma (résistance, ampoule, *etc.*). Ces tâches de reconnaissance et d'identification peuvent être résolues grâce à des techniques de classification et d'apprentissage, qui font l'objet du Chapitre 3. Dans la suite de cette section, on s'intéressera plus particulièrement aux techniques d'indexation et de recherche d'images.

### 2.3.2 Types de requêtes

Il existe deux façons de faire une requête dans un système d'indexation et de recherche d'images par le contenu : soit à l'aide d'exemples, soit par une esquisse / ébauche graphique (en anglais « sketch »).

#### 2.3.2.1 Requêtes à base d'exemples

Pour les systèmes de recherche d'images à base d'exemples, l'utilisateur, pour représenter ses besoins, utilise une image (ou une partie d'image) qu'il considère similaire aux images qu'il recherche. Cette image est appelée image « exemple »ou « requête ». L'image exemple peut soit être fournie par l'utilisateur, soit être choisie par l'utilisateur dans la base d'images utilisée. En anglais, cette technique est connue sous le nom de « query by image content »(QBIC).

Par exemple, la figure 2.4 montre une image requête (en haut de la page au centre) et les images les plus proches (en dessous de la requête) renvoyées par le système de recherche d'images en ligne Google Similar Images [10].

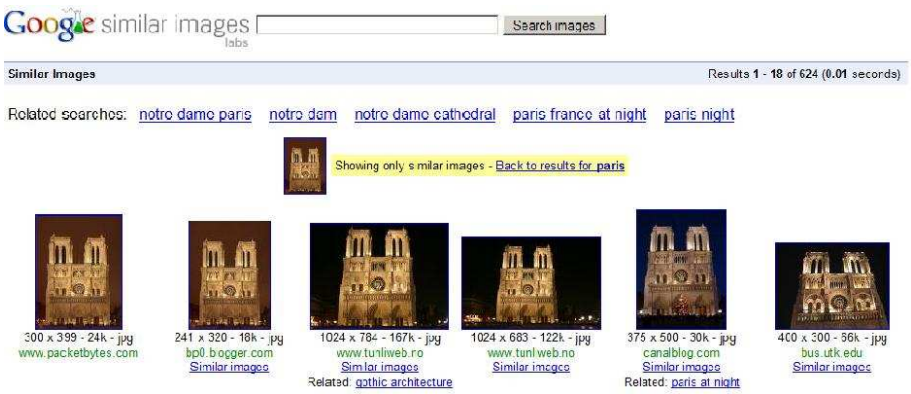



FIGURE 2.4 – Image requête et le résultat renvoyé par Google Similar Images

Lorsque l'utilisateur fournit une image entière en tant qu'exemple, on va comparer cette image aux images d'une base de recherche [Lin 09a, Torres 09].

Lorsque l'utilisateur fournit une partie d'image, on parle de « requête partielle ». En effet, un utilisateur ne désire pas toujours retrouver des images d'après l'apparence globale d'une image. Parfois, celui-ci peut rechercher des images ayant certaines zones (ou régions) semblables à celles

10. `http://similar-images.googlelabs.com/`

de l'image requête [Todorovic 08]. Il peut aussi rechercher des objets particuliers dans les images [Rahmani 08]. Dans ce cas, l'utilisateur fournit comme exemple une région particulière d'une image [Chiang 09], ou une image représentant l'objet qu'il désire retrouver dans les images de la base. L'indexation et la recherche d'images par requête partielle sont basées sur une segmentation préalable des images, ou sur leur structure spatiale.

On distingue deux types de segmentation d'images : la segmentation manuelle ou la segmentation automatique. Dans le cas de la segmentation manuelle, l'utilisateur trace, dans les images, le contour des régions qui l'intéressent. Quant à la segmentation automatique, elle consiste à partitionner les images en régions connexes et homogènes au sens d'un critère d'homogénéité précis (homogénéité de couleur, de texture, *etc.* au sein d'une même région) [Estrada 09]. Nombre d'approches ont déjà été proposées : elles reposent sur des outils variés tels que la morphologie mathématique [Hamarneh 09, Hodneland 09], la décomposition en ondelettes [Kim 07, Roudet 07], les contours actifs [Mille 09, Ying 09], ...

Certaines sont fondées sur la détection de contours ou de points d'intérêt [Zhang 09c, Papari 08]. D'autres, au contraire, sont basées sur l'identification de régions [Ayed 08, Zhang 07]. Pour plus d'informations sur les différentes techniques de segmentation, on conseille les lectures suivantes [Albatal 09, Freixenet 02].

Plusieurs problèmes sont liés aux requêtes partielles basées sur la segmentation d'images. En effet, la segmentation manuelle s'avère très coûteuse pour l'utilisateur, et difficile, sur de grandes bases d'images. Concernant la segmentation automatique, il est difficile d'obtenir une segmentation précise. De plus, la segmentation peut poser des problèmes d'invariance lors de l'appariement entre une requête et des images : certaines méthodes basées sur la segmentation ne permettront pas de retrouver les objets ou régions similaires à la requête s'ils n'ont pas la même taille, la même orientation ou la même position.

Une autre solution consiste donc à utiliser des requêtes partielles basées sur la structure spatiale des images. Une structure souvent utilisée est l'arbre quaternaire [Malki 99]. Dans cette approche multirésolution, une image est vue comme la réunion de quatre sous-images carrées de même taille. Chaque sous-image peut être décomposée de la même façon et ainsi de suite pour chaque niveau de résolution. Les caractéristiques visuelles choisies sont calculées et stockées pour chaque sous-image. Ainsi les relations spatiales entre les sous-images sont conservées. Quand un utilisateur recherche une image, il spécifie les régions qui l'intéressent pour chaque niveau de résolution. Les images de la base de recherche sont filtrées en augmentant au fur et à mesure le niveau de résolution. Cette approche évite la segmentation des images et offre des alternatives efficaces aux problèmes d'invariance.

Les requêtes partielles permettent donc de retrouver des objets ou des sous-images dans des images, contrairement aux requêtes à base d'exemple classiques. Les méthodes à base de requêtes partielles sont donc plus précises que celles à base d'exemples classiques. Cependant, étant basées, en général, sur une segmentation préalable des images, les requêtes partielles ont l'inconvénient d'être plus coûteuses. Outre le coût de la segmentation des images, il faut prendre en compte le stockage des caractéristiques : l'ensemble des caractéristiques choisies est calculé et stocké pour chaque région de chaque image, au lieu d'avoir un seul ensemble de caractéristiques par image dans le cas des requêtes par exemple classiques. Enfin, la recherche d'images est d'autant plus coûteuse qu'il y a de régions. Enfin, le choix entre requête partielle ou globale est fonction du besoin de l'utilisateur.

#### 2.3.2.2 Requêtes par esquisses

Dans les requêtes par esquisse (connues sous le terme « sketch-based image retrieval », en anglais) [Gavilan 08, Anelli 07], le système fournit à l'utilisateur des outils lui permettant de constituer une esquisse correspondant à ses besoins. L'esquisse fournie sera utilisée comme exemple pour la recherche. L'esquisse peut être une ébauche de forme ou contour d'une image entière ou une ébauche des couleurs ou textures des régions d'une image. L'utilisateur choisira, en fonction de la base d'images utilisée, de ses besoins et préférences, l'une ou l'autre de ces représentations.

Les requêtes par esquisse présentent l'inconvénient majeur qu'il est parfois difficile pour l'utilisateur de fournir une esquisse, malgré les outils qui lui sont fournis. C'est le cas des images provenant des bases généralistes. En effet, ces images peuvent être décrites par un large spectre de caractéristiques (texture, forme, couleur, composition spatiale, ...) Dans ce cas l'utilisateur a du mal à fournir une esquisse. Par contre, pour certaines bases spécialisées, comme des bases de symboles graphiques, il est assez facile de fournir une esquisse : par exemple, si l'utilisateur recherche toutes les images contenant le symbole d'une résistance, dans une base, il pourra dessiner facilement le contour du symbole et le fournir comme esquisse. La requête par esquisse est donc particulièrement utile pour la recherche d'images dont le contour est facilement identifiable et reproductible à la main.

Finalement, les résultats de la recherche dépendant de la qualité et de la précision de l'esquisse, une requête à base d'image exemple, globale ou partielle, issue de la base d'images utilisée pour la recherche, ou d'une collection personnelle de l'utilisateur, est généralement préférée à une esquisse. La requête par esquisse sera préférée pour la recherche dans certaines bases d'images spécialisées à forte teneur en composantes graphiques (symboles graphiques, par exemple) [Mas 07, Romeu 06].

### 2.3.3 Indexation par le contenu : extraction de caractéristiques

Une fois une requête exprimée, qu'elle soit partielle ou non, le système va extraire une ou des caractéristiques de la requête, les organiser, et va les comparer à celles des images de la base d'images. L'extraction, le stockage, et l'organisation des caractéristiques relèvent de l'indexation des images. La comparaison des caractéristiques d'images fait référence à la recherche d'images.

Dans le cas de requêtes partielles, le système va calculer les caractéristiques de façon locale, pour chaque région de la requête et de chaque image de la base. Dans le cas d'une requête à base d'exemple composée d'une ou plusieurs images globales, les caractéristiques seront calculées sur les images globales constituant la requête et sur chaque image de la base. Remarquons que pour ces dernières (requêtes à base d'exemple classiques composées d'une ou plusieurs images globales), les caractéristiques peuvent aussi être calculées de façon locale, sur chaque région de chaque image. Cette deuxième méthode est beaucoup plus coûteuse que la première, car elle va nécessiter une segmentation préalable de(s) image(s) requête(s) et de chaque image de la base, et un calcul et un stockage des caractéristiques plus coûteux. De même la recherche sera plus coûteuse. Cependant, cette méthode permettra de retrouver des objets ou sous-images dans les images, contrairement au calcul des caractéristiques sur les images globales.

L'étape d'extraction de caractéristiques visuelles est essentielle dans le processus d'indexation d'images. En effet, elle permet de passer de l'image à une description qui soit plus facilement utilisable. Une fois un ensemble de caractéristiques choisi et calculé pour chaque image, seules ces représentations visuelles (ces ensembles de valeurs de caractéristiques) seront utilisées pour la recherche d'images. Mais comment bien choisir un ensemble de caractéristiques ? En général, une caractéristique est d'autant plus pertinente que la différence entre les caractéristiques de

formes significativement différentes est grande, et que la différence entre les caractéristiques de formes similaires est faible. En fait, la description idéale doit être stable (robuste au bruit, à la luminosité, ...), concise (de dimension aussi petite que possible pour permettre un processus de recherche, basé sur un calcul de similarité entre caractéristiques, plus rapide), unique (une image donnée n'a qu'une valeur possible pour chaque caractéristique), accessible et invariante aux transformations géométriques.

Ci-dessous nous présentons trois méthodes de caractérisation visuelle, fondées respectivement sur l'analyse de la forme, de la couleur et de la texture. Pour chacune de ces méthodes de caractérisation, il existe de nombreuses descriptions possibles et en faire une énumération exhaustive serait fastidieux. Nous proposons plutôt ici de décrire les méthodes qui permettront d'avoir un aperçu de ce vaste champ de recherche.

#### 2.3.3.1 Description de la forme

Il existe plusieurs méthodes tentant de quantifier la forme comme le ferait l'intuition humaine. Celles-ci peuvent être classées en deux grands types : les méthodes statistiques, basées sur l'étude statistique des mesures que l'on effectue sur les formes à reconnaître, et les méthodes structurelles, qui tentent de représenter la structure physique des formes à reconnaître [Hastie 01, Lladós 02].

**Méthodes statistiques**

Avec les méthodes statistiques, chaque image est représentée par un vecteur caractéristique à $n$-dimensions. Ces vecteurs sont calculés grâce à des fonctions mathématiques, ou des algorithmes, appelés descripteurs. Un bon choix de caractéristiques pertinentes est indispensable pour atteindre un grand pouvoir de discrimination. Dans certaines applications, comme la reconnaissance de symboles, on cherchera aussi à atteindre l'invariance aux transformations affines (rotation, translation, effet d'échelle). De plus, l'extraction des caractéristiques doit être suffisamment robuste pour limiter la variabilité due au bruit et à la déformation.

Voici des exemples de caractéristiques :

- Basées sur les pixels de l'image : le vecteur caractéristique est composé d'une caractéristique pour chaque valeur de pixel, après que l'image ait été normalisée à une taille fixe. Dans ce cas, les caractéristiques sont obtenues grâce à des descripteurs dits « locaux »[Zhou 08, Tola 08]. Au lieu d'utiliser tous les pixels de l'image, il est possible de se concentrer sur les pixels les plus importants, appelés points d'intérêt. Les points d'intérêts sont obtenus grâce à des détecteurs de points d'intérêts [Herbert 06, Rosten 06].
  Les caractéristiques suivantes, sont issues de descripteurs dits « globaux ». A l'inverse des descripteurs locaux, ils calculent des caractéristiques à partir de contours ou de régions et permettent de décrire une image dans son ensemble.
- Caractéristiques géométriques : par exemple le centre de gravité, l'aire, le périmètre, les axes d'inertie, la circularité, les intersections de lignes, les « trous », la compacité, l'élongation, la rectangularité [Rosin 99], l'orientation ...
- Moments géométriques : moments invariants [Tzimiropoulos 09, Zhang 09a], moments de Zernike [Singhal 09, Revaud 09], ...
- Transformations : transformée de Fourier [Pan 09, Chen 09], transformée de Fourier-Mellin [Bin 08, Wang 07], et d'autres transformées spéciales pour obtenir des vecteurs caractéristiques (aussi appelés signatures) de l'image, comme la transformée de Radon [Hjouj 08, Tabbone 08].

Le vecteur caractéristique est un moyen de représentation simple et avec un coût de calcul faible. Cependant, le pouvoir discriminant et la robustesse face aux déformations dépend beaucoup du choix d'un ensemble optimal de caractéristiques pour chaque type d'application. De plus, le nombre de caractéristiques doit être relativement faible, pour permettre une recherche rapide. Enfin, ces méthodes nécessitent souvent une étape de segmentation préalable. Or, la segmentation n'est pas toujours facile, dans certains types d'applications, comme la reconnaissance de symboles par exemple, où les symboles sont parfois « encastrés »dans les dessins.

**Méthodes syntaxiques et structurelles**

Avec les méthodes syntaxiques et structurelles, les images sont représentées par une description de leur forme utilisant un ensemble de primitives géométriques appropriées et de relations entre elles [Lillholm 09, Dudek 97].

Les primitives usuelles de description de formes sont les lignes et les arcs, mais quelques fois, d'autres primitives géométriques, comme les cercles et les rectangles, sont utilisées [Berretti 00].

Ci-dessous on donne quelques exemples d'approches structurelles.

**Représentation des images par des graphes** Dans les approches basées sur les graphes [Pham 09, Backes 09, Luqman 09], chaque image est représentée par un graphe. Les sommets du graphe correspondent à des pixels ou des régions de l'image, et les arêtes correspondent à des relations entre ces points ou régions. C'est une représentation très intuitive et naturelle des images. Avec cette approche, les différents objets éventuellement contenus dans les images peuvent être vus comme sous-graphes de l'image entière, ce qui permet de faire la segmentation et la reconnaissance en même temps. L'inconvénient majeur de cette méthode est la complexité de l'algorithme d'appariement.

**Grammaires formelles - grammaires de graphes** Avec les méthodes à base de grammaires [Lin 09b, Siskind 07], la reconnaissance d'une image proposée consiste à analyser sa représentation pour tester si elle peut être générée par la grammaire. Cette méthode est utile pour les applications où les formes des images peuvent être définies précisément par un ensemble de règles.

Avec les méthodes structurelles, le choix des caractéristiques n'est pas si difficile parce qu'elles se fient aux représentations vectorielles, même si la vectorisation (transformation d'une image pixels en un ensemble de vecteurs) introduit des erreurs de représentation.

Il n'y a pas de structure générale ni optimale de représentation de la forme et il ne semble pas facile de définir une représentation à la fois assez générale et puissante pour être optimale dans les différents domaines et applications. Des signatures de forme ont émergé en matière de reconnaissance de formes et apparaissent comme des structures simples, générales et souples pour représenter les propriétés importantes des formes. Une signature est en général un ensemble de nombres (sous forme d'un vecteur ou d'une matrice) décrivant une forme donnée. Il n'est pas possible de reconstruire entièrement une forme à partir de sa signature, mais les signatures de différentes formes doivent être assez différentes pour permettre de les discriminer correctement. Ces signatures sont calculées grâce à des descripteurs.

Pour plus de détails sur les différentes descriptions de la forme, on conseillera au lecteur les études suivantes [Valveny 08b, Terrades 07]. Enfin, la norme de description de documents multimédia MPEG-7 [Manjunath 02] fournit un ensemble de descripteurs de forme. Pour plus de détails sur les descripteurs de forme de cette norme, on conseillera au lecteur l'étude comparative [Zhang 03].

#### 2.3.3.2 Description de la couleur

La couleur est un attribut important en reconnaissance d'images. Plusieurs systèmes cohérents ont été imaginés pour représenter fidèlement l'espace des couleurs. L'espace RGB (de l'anglais Red Green Blue) a été largement utilisé grâce à la grande disponibilité d'images au format RGB à partir d'images scannées. Il s'agit d'un espace vectoriel engendré par les trois composantes primaires (Rouge (Red), Vert (Green), Bleu (Blue)). En effet, la gamme infinie des couleurs naturelles peut être reproduite à partir de ces trois couleurs seulement. Ce principe de synthèse de la couleur se retrouve dans la plupart des dispositifs lumineux de restitution de la couleur : CRT, LCD, Plasma. Le modèle RGB propose donc, pour chaque pixel, 256 intensités de rouge, 256 intensités de vert et 256 intensités de bleu. Ainsi, chaque couleur naturelle peut être représentée par un système de coordonnées orthogonal à trois dimensions. Ce modèle présente quelques inconvénients : il ne tient pas compte des particularités de la perception visuelle des couleurs (les trois composantes RGB ne permettent pas de reconstituer réellement toutes les couleurs perceptibles par l'œil humain).

Afin de pallier ces inconvénients, la CIE (Commission Internationale de l'Éclairage) a défini un espace de représentation de la couleur basé sur trois couleurs primaires non visibles (dites virtuelles) X, Y et Z. Le passage de l'espace RGB à l'espace XYZ s'effectue simplement grâce à une transformation linéaire pouvant être interprétée comme un changement de base, tel que toutes les couleurs du spectre visible soient contenues dans le triangle XYZ. Ce système présente lui aussi des inconvénients : on perçoit plus de nuances en X et Z qu'en Y. D'autre part, certaines dimensions descriptives de couleur (clair/foncé, pur/délavé) ne sont pas accessibles directement.

Afin de caractériser les couleurs de façon plus intuitive, conformément à la perception naturelle des couleurs, l'espace HSV a été proposé. Celui-ci fait intervenir des critères psychophysiologiques : la teinte H (de l'anglais Hue), caractérise la couleur elle même (en général d'après sa position dans le disque chromatique) ; la saturation S (de l'anglais Saturation) est le niveau de pureté de la couleur représentée (il vaut 0 pour du noir ou du blanc et est maximum pour une couleur pure) ; la valeur V (de l'anglais Value) est la contenance relative de noir et de blanc. Cependant, ce modèle présente aussi des inconvénients : la teinte n'est pas significative pour les régions peu saturées, très claires, ou très sombres. De plus, le modèle HSV n'offre pas de corrélation simple avec les autres modèles.

Le modèle RGB reste donc le plus utilisé. Cependant il existe d'autres espaces de représentation des couleurs. Pour plus de détails sur les différents modèles on renverra le lecteur au chapitre 15 du livre [Watt 99] et la comparaison de certains modèles [Smith 90].

Nous avons présenté ici les espaces de couleur les plus répandus. Il en existe d'autres, moins réputés ou plus récents [Bratkova 09, Noda 07]. Ces nouveaux espaces sont en général basés sur les espaces plus classiques présentés ci-dessus. Par exemple, l'espace oRGB proposé dans [Bratkova 09] est, comme l'espace HSV, une transformation de l'espace RGB. Les couleurs primaires de ce modèle sont basées sur trois axes qui s'opposent (blanc-noir, rouge-vert et jaune-bleu). Ces axes relativement intuitifs font de lui un modèle simple. Ce modèle est adapté à nombre d'applications. En outre, comme oRGB est basé sur une transformation de RGB, les axes de oRGB vont des couleurs chaudes aux couleurs froides (du rouge au vert et du jaune au bleu). Ainsi oRGB a un concept quantitatif de la chaleur des couleurs, ce qui est nécessaire dans certaines applications artistiques.

A partir de ces modèles colorimétriques, plusieurs méthodes permettent de décrire la couleur d'une image. On peut simplement calculer la couleur moyenne ou dominante de l'image. Une méthode plus courante et efficace consiste à caractériser une image par la répartition des couleurs des différents pixels. En effet, quel que soit l'espace de représentation utilisé, l'information couleur

d'une image peut être représentée par un seul histogramme $3D$ ou 3 histogrammes $1D$ [Swain 91]. Ces modes de représentation de la couleur ont l'avantage d'être invariants à la translation et à la rotation. De plus, une simple normalisation de l'histogramme fournit aussi l'invariance à l'échelle.

Même si la représentation par histogramme est la plus utilisée, de par sa simplicité de mise en œuvre et son efficacité, il existe d'autres descripteurs de couleur [Hurtut 08, Yang 08]. Par exemple, l'approche proposée dans [Mignotte 08] est originale dans le sens où elle utilise une représentation par histogramme, mais qui combine plusieurs espaces colorimétriques.

Le descripteur à couleurs dominantes (DCD), utilisé dans [Min 09], fournit une représentation compacte et représentative d'une image. Ce descripteur consiste en un nombre de couleurs dominantes, et, pour chaque couleur dominante, sa valeur est exprimée sous forme d'un vecteur de composantes couleur et d'un pourcentage de pixels dans l'image qui correspondent à la composante. Ce descripteur est couramment utilisé. Cependant il présente l'inconvénient que les couleurs dominantes sont calculées pour chaque image au lieu d'être fixées dans l'espace colorimétrique.

Pour plus de détails sur les descripteurs de couleur, on conseillera aux lecteurs les études suivantes [López 08, Schettini 01, Manjunath 01]. Enfin, la norme MPEG-7 fournit aussi un ensemble de descripteurs de couleur [Yang 08].

#### 2.3.3.3 Description de la texture

Il n'existe pas de définition universelle de la texture. Cependant, comme pour les notions de forme et de couleur, la notion de texture est liée à la perception humaine. En effet, la texture désigne une surface, représentée par une image ou région d'image, offrant la possibilité de simuler l'apparence tactile (et donc 3D) de celle-ci. La notion de texture est donc liée aux sens du toucher et de la vision.

Une image dite « texturée »permet de donner la sensation de toucher l'image, de ressentir son grain, ses aspérités, comme si l'on passait son doigt dessus.

A partir de cette constatation, on peut définir deux types de textures :

- on peut considérer la texture comme la répétition spatiale d'un motif de base dans différentes directions de l'espace. C'est le cas, par exemple, de la texture d'un tissu, d'un mur de briques, d'écailles de poisson, d'un champ labouré, *etc.*. Les partisans d'une telle définition s'orientent généralement vers une approche fractale [Xu 09], spectrale [Liu 09a] ou structurelle [Liao 09, Peyré 09] de la texture.
- On peut aussi penser qu'une texture ne possède pas de contours francs, mais plutôt un certain désordre, c'est-à-dire une disposition aléatoire que l'on pourrait considérer comme visuellement homogène. On aurait alors une surface fermée, sans motifs isolables ou répétitifs. C'est le cas, par exemple, de photographies à distance d'herbe, de sable, de graviers *etc.* ... Les partisans d'une telle définition s'orientent généralement vers une approche probabiliste [Kokkinos 09, Choy 08].

Les approches citées peuvent être réparties en deux catégories, comme pour les descripteurs de forme : les méthodes statistiques et les méthodes structurelles [Umarani 07, Haralick 79].

**Méthodes statistiques**

Ces méthodes tentent de modéliser les notions qualitatives usuelles de texture : granularité, contraste, homogénéité, répétitivité, fragmentation, orientation, *etc.*.

Certaines utilisent des transformations orthogonales locales sur l'image (Fourier, Haar, Hadamard, Slant, Karhunen-Loeve) [Liu 09a, Chehel Amirani 09].

D'autres utilisent les statistiques locales de l'intensité lumineuse : des statistiques et moments d'ordre 1 (histogrammes $1D$, moyenne) [Ahonen 09, Andra 05], d'ordre 2 (matrices de co-occurrences, fonctions d'auto-corrélation) [Li 09b, Kiranyaz 08], des statistiques d'ordre supérieur (probabilités conditionnelles) [Ojala 01], et des outils de la morphologie Mathématique [Lin 08, Fernandez-Maloigne 08].

La plupart des techniques utilisant les paramètres statistiques du premier ou du second ordre sont souvent limitées quant à leur utilisation. En effet, certains paramètres sont difficiles à contrôler : par exemple les modèles auto-régressifs ne peuvent contrôler que les moments d'ordre 2. De plus certains modèles ne suffisent pas pour décrire entièrement la texture : même si les statistiques d'ordre 1 ont l'avantage d'avoir un temps de calcul réduit, ils ne permettent pas de prendre en compte les interactions entre plusieurs pixels. Enfin, certains modèles représentent l'interaction entre pixels (distance donnée entre deux pixels pour une direction donnée), mais posent des problèmes de temps de calcul et d'espace mémoire : c'est le cas des matrices de co-occurrences. Les statistiques d'ordre supérieur ont un temps de calcul réduit par rapport aux matrices de co-occurrences, néanmoins elles nécessitent une mémoire importante.

**Méthodes syntaxiques et structurelles**

Comme pour les descripteurs de forme, les méthodes syntaxiques et structurelles de représentation de la texture [Vartiainen 08, Le Borgne 07] modélisent des relations entre des primitives constituant l'image, et plus précisément des relations spatiales. Les règles de placement ou d'agencement spatial déterminent l'existence et la nature de la texture.

Ces méthodes peuvent toujours être vues comme la succession de deux phases : dans un premier temps, on cherche à définir les primitives constituant les images. Ensuite, on va caractériser les relations spatiales entre ces primitives. Pour identifier les primitives, une croissance de régions va d'abord être effectuée sur un ou plusieurs attributs (par exemple l'intensité lumineuse, surface, colorimétrie, ...) [Deng 01]. Les approches de croissance de régions consistent à faire croître chaque région autour d'un pixel de départ. L'agglomération des pixels n'exploite aucune connaissance *a priori* de l'image ou du bruit qui la dégrade. En fait, la décision de faire croître une région en y intégrant un pixel voisin repose seulement sur un critère d'homogénéité imposé à la zone en croissance : seuls les pixels les plus similaires suivant le critère sont ajoutés à la région et la font grandir. Les primitives seront alors caractérisées par leur forme et la valeur moyenne de l'attribut utilisé sur la région concernée. Dans un deuxième temps, on calcule les histogrammes du premier ordre ou du second ordre de ces paramètres afin de caractériser leur répartition dans l'image.

Les méthodes syntaxiques et structurelles sont particulièrement bien adaptées aux textures macroscopiques (comme les hachures, la peau d'un lézard ou un mur de brique). Par contre, sur des textures microscopiques (par exemple le sable, la laine tissée ou l'herbe), on préférera la caractérisation numérique grâce aux méthodes statistiques.

#### 2.3.3.4 Organisation des caractéristiques visuelles

Une fois qu'une ou plusieurs caractéristiques visuelles ont été calculées pour chaque image de la base, il est souvent nécessaire de les organiser de façon à pouvoir retrouver, ensuite, le plus rapidement possible (en limitant la quantité de données visuelles examinées pendant la recherche), des images recherchées. En effet, l'organisation des données n'est pas obligatoire, mais, dès lors que la base de recherche devient conséquente et que les méthodes de recherche utilisées nécessitent le stockage des caractéristiques visuelles, elle devient vivement conseillée. Cette organisation consiste à la construction de structures de données, appelées index, permet-

tant de gérer des données multidimensionnelles. On parle d'index multidimensionnels [Samet 05]. En effet, comme nous l'avons vu dans la section 2.3.3, les descripteurs fournissent souvent des caractéristiques multidimensionnelles (vecteurs, matrices, ...)

Dans la section 2.3.3, consacrée à l'organisation des caractéristiques textuelles, nous nous sommes intéressés aux structures d'index à 1 dimension. Ces structures permettent, à partir d'une clé de recherche simple (un terme ou plusieurs termes), de retrouver les images correspondant à cette clé. Or, la recherche d'images par le contenu manipule souvent des données dans des espaces à deux dimensions ou plus. Nous avons donc besoin de structures d'index multidimensionnelles. De telles structures permettront d'effectuer des requêtes en ne précisant des valeurs que pour certaines dimensions des caractéristiques de l'image requête. Il sera aussi possible d'ordonner les résultats différemment en fonction des dimensions.

Dans cette section, nous abordons, de façon succincte, les index multidimensionnels les plus réputés. Pour plus d'informations sur l'indexation multidimensionnelle, on conseillera au lecteur le chapitre 1 de la thèse [Berrani 04].

Les techniques d'indexation multidimensionnelles peuvent être classées en deux catégories :

- les techniques de première catégorie visent à regrouper les vecteurs caractéristiques en clusters. Chaque cluster est associé à un région de l'espace. Le X-tree [Berchtold 96] et le SS-tree [White 96] font partie de cette catégorie. Lors de la recherche, l'accès aux images recherchées sera plus rapide car on cherchera d'abord les clusters les plus proches du vecteur caractéristique de l'image requête, pour ensuite comparer le vecteur requête aux vecteurs des clusters les plus proches. Le nombre d'images est ainsi diminué, ainsi que le nombre de distances à calculer.
- les techniques de seconde catégorie ont pour principe de découper l'espace en région et d'indexer ces régions à l'aide de structures arborescentes, comme le K-D-B-Tree [Robinson 81] et le LSD-Tree [Henrich 89]. La structure d'index donne un accès direct à un sous-ensemble d'images de la base, regroupées dans une même région de l'espace.

Ces structures arborescentes sont les plus utilisées mais d'autres approches se distinguent. Par exemple, la méthode proposée dans [Mejdoub 09] utilise une structure de treillis pour indexer les images. De plus, les approches multirésolution, à base d'ondelettes, sont souvent utilisées [Luo 08, Hong 08]. Ces méthodes ont pour principe d'analyser et d'indexer les images à différents niveaux de résolution. Pour rechercher des images, on compare les images à différents niveaux de résolution.

Cette étape d'organisation des caractéristiques visuelles multidimensionnelles, est très importante, car elle va permettre de retrouver les images rapidement, même dans de grandes bases d'images, en limitant le nombre de comparaisons entre vecteurs caractéristiques. Elle constitue l'indexation à proprement parler, car elle représente la création de structures d'index multidimensionnelles à partir des caractéristiques visuelles préalablement extraites. Cependant, le processus d'indexation relève à la fois de l'extraction de ces caractéristiques, puis de leur indexation (*i. e.* de la création de structures d'index à partir de ces caractéristiques).

### 2.3.4 Recherche par le contenu

Comme pour la recherche textuelle, une fois les images indexées visuellement (*i. e.* que les caractéristiques visuelles ont été extraites et organisées), le problème est de pouvoir les retrouver simplement.

La recherche d'images se traduit alors par la mise en correspondance des représentations visuelles des images de la base et d'une représentation visuelle de la requête. Le terme « représentation visuelle »ou représentation du contenu d'une image, désigne l'ensemble des caractéris-

tiques calculées sur cette image pour la caractériser. Il s'agit de la représentation interne de type visuel, d'une image.

Dans le cas où une structure d'index a été utilisée pour organiser les représentations visuelles, la mise en correspondance ne se fait pas entre l'image requête et toutes les images de la base, mais uniquement entre la requête et les images d'une même région de l'espace à laquelle on a pu accéder directement grâce à l'index.

Un système de recherche d'images par le contenu peut finalement être décrit par le schéma de la figure 2.5.

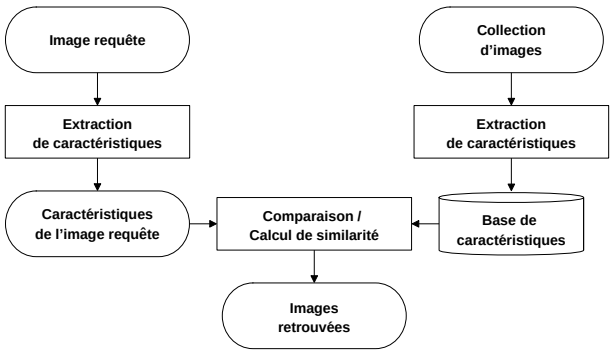



FIGURE 2.5 – Système de recherche d'images par le contenu

La qualité de la mise en correspondance de la représentation visuelle d'une image requête et des représentations visuelles des images disponibles pour la recherche va dépendre du choix d'une mesure de similarité ou distance adéquate. Ce choix va dépendre des caractéristiques visuelles utilisées. En effet, nous avons répertorié plusieurs modes de représentation du contenu des images : représentation basée sur des caractéristiques de forme, de texture ou de couleur des images, voire sur un ensemble de ces caractéristiques.

Suivant la représentation visuelle choisie, les caractéristiques correspondant à une image pourront être de format différent : matrice, vecteur, graphe, *etc.*. Pour aboutir à un système de recherche d'images performant, l'objectif est alors de choisir, à partir du format des représentations choisies, une mesure adéquate, parmi les mesures de distance ou de similarité, des calculs de probabilités, *etc.*.

Cette section a pour but de décrire brièvement les différentes mesures utilisées en recherche d'images basée sur le contenu, mais surtout d'expliquer dans quels cas et pourquoi il est plus utile d'utiliser telle mesure plutôt qu'une autre.

#### 2.3.4.1 Mesures de distance ou de similarité ?

Les notions de distance et de similarité sont proches, dans le sens où elles permettent toutes deux d'identifier si deux individus sont différents ou non. Cependant, à la différence des mesures de similarité, les mesures de distance accordent une valeur maximale à deux objets complètement différents et minimale (0) à deux objets identiques. De plus, une mesure $D$ peut être considérée comme une distance sur un ensemble $\mathbb{E}$ (où $\mathbb{E}$ est un $\mathbb{R}$-espace vectoriel), si et seulement $D$ est une application de $\mathbb{E} \times \mathbb{E}$ dans $\mathbb{R}^+$ telle que :

- $D(a, b) = 0 \Leftrightarrow a = b$ (réflexivité),
- $\forall a, b \in \mathbb{E}, D(a, b) = D(b, a)$ (symétrie),
- $\forall a, b, c \in \mathbb{E}, D(a, b) + D(b, c) \geq D(a, c)$ (inégalité triangulaire)

Il existe plusieurs mesures de similarité et de distances dans la littérature. Dans le paragraphe suivant, nous aborderons les distances les plus utilisées en recherche d'images par le contenu (les mesures de similarité étant, quant à elles, plus utilisées en recherche textuelle). Pour chaque distance abordée, nous essayerons d'expliquer quel contexte (avec quel types de données, ...) favorise l'utilisation de cette distance plutôt qu'une autre.

#### 2.3.4.2 Mesures de distance

**Distances de Minkowski d'ordre $i$**

Soit deux points $x$ et $y$ dans un espace à $n$ dimensions de coordonnées $x = \{x_1, x_2, \ldots, x_n\}$ et $y = \{y_1, y_2, \ldots, y_n\}$ de même dimension $n$. Alors les distances de Minkowski d'ordre $i$ entre $x$ et $y$, notées $L_i$ sont définies par :

$$L_i(x, y) = (\sum_{k=1}^{n} |x_k - y_k|^i)^{\frac{1}{i}}$$

Les seuls cas utilisés en pratique correspondent aux valeurs 1 et 2 de $i$. $L_2$ correspond à la distance euclidienne, et $L_1$ à la distance de Manhattan (aussi appelée « city-block »car elle correspond à la distance la plus courte entre deux points que doit parcourir un véhicule dans une ville composée de blocs carrés homogènes.). Même si la distance euclidienne est la plus intuitive, la distance de Manhattan est parfois meilleure. En effet, à la différence de la distance euclidienne, la distance de Manhattan ne maximise pas la pondération des valeurs extrêmes. Ainsi, des points ayant des valeurs proches sur la plupart des axes seront donc plus proches. Par rapport à la distance euclidienne, on observe alors une augmentation du contraste des distances. Par contre les temps de calcul de cette distance sont plus longs.

**Distance euclidienne**

Le plus populaire des indices de distance est la distance euclidienne (D1). Cette distance représente la distance géographique la plus courte entre deux points dans un espace multidimensionnel (distance à « vol d'oiseau »). Cet espace contient autant de dimensions qu'il y a de variables internes. Soit deux points $x$ et $y$ dans un espace à $n$ dimensions de coordonnées $x = \{x_1, x_2, \ldots, x_n\}$ et $y = \{y_1, y_2, \ldots, y_n\}$ de même dimension $n$. Alors la distance euclidienne entre $x$ et $y$ est donnée par :

$$D1(x, y) = \sqrt{\sum_{k=1}^{n} |x_k - y_k|^2}$$

Cette mesure n'a pas de borne supérieure. De plus, ses valeurs s'accroissent avec la dimension des vecteurs. En effet, pour des données réelles, la probabilité qu'une au moins des coordonnées d'un individu ait une valeur extrême peut être élevée, dès lors que le nombre de coordonnées est grand. Le point correspondant est alors rejeté aux frontières de l'espace occupé par les données, et sa distance aux autres données est grande. Comme cette situation est partagée par un grand nombre de données, le centre de l'espace occupé par ces dernières est largement dépeuplé. Ainsi, les distances euclidiennes entre les données de grande dimension (*i. e.* les données pour lesquelles le nombre de variables est très grand vis à vis du nombre d'observations) montrent en général peu de contraste, c'est-à-dire que les distances sont toutes du même ordre. Autrement dit, la distance euclidienne considère de la même manière deux points distants sur toutes les dimensions de l'espace et deux points assez proches partout sauf sur quelques dimensions. Enfin, et surtout,

la distance varie avec le domaine de chaque composante des vecteurs. C'est pour cette dernière raison que l'on calcule le plus fréquemment la distance euclidienne après centrage et réduction des variables.

**Distances de Mahalanobis**

La distance de Mahalanobis remonte aux années trente. Cette mesure permet de calculer la distance d'un point $x$ à la moyenne d'une distribution $\overline{X_j}$. Soit $W$ la matrice de covariance de la distribution. Alors la distance entre $x$ et la moyenne $\overline{X_j}$ est donnée par :

$$||x - \overline{X_j}||^2_{W^{-1}} = {}^t(x - \overline{X_j})\, W^{-1}(x - \overline{X_j}) \tag{2.1}$$

Cette distance a l'avantage de se révéler plus efficace que les autres lorsque les composantes des vecteurs caractéristiques ne sont pas homogènes ou très corrélées, car elle tient compte de la liaison entre les diverses composantes. Par contre, elle présente l'inconvénient de nécessiter suffisamment d'observations pour estimer la matrice de covariance $W$. De plus, Enfin, cette distance est couramment utilisée en l'analyse discriminante décisionnelle, technique de classification supervisée que nous abordons dans le chapitre 3.

Ces distances sont les plus utilisées en recherche d'images par le contenu. En particulier, la distance euclidienne est la plus populaire, de par ses bon résultats et sa faible complexité. Il existe cependant d'autres mesures de distances utilisées en recherche d'images par le contenu [Ni 09, Li 09a, Vivaracho-Pascual 09, Xiao 08]. Par exemple, l'approche [Li 09a] propose une mesure de distance entre deux images, prenant en compte à la fois les relations spatiales entre pixels et les niveaux de gris des images. Dans l'article [Vivaracho-Pascual 09], une distance fractionnaire est utilisée dans une application de reconnaissance de signatures. Les distances fractionnaires sont les distance de Minkowski d'ordre $i$ où $i$ est inférieur à 1. Cette distance a été utilisée, à la place de la distance euclidienne, car elle est plus adaptée aux problèmes en grande dimension. En effet, nous avons vu que plus la dimension est grande, plus la probabilité d'avoir une composante aberrante, parmi ces dimensions, est grande. Ceci rend la distance euclidienne moins performante dans de tels espaces, car elle est très sensible aux valeurs aberrantes. Au contraire, les distances fractionnaires minimisent le poids des valeurs aberrantes.

### 2.3.5 Conclusion sur l'indexation visuelle et la recherche par le contenu

Nous avons présenté dans cette section les principes fondamentaux de la recherche d'images par le contenu : à savoir, l'extraction et l'indexation de caractéristiques bas niveau de l'image, suivis de la définition de distances entre ces caractéristiques, afin de mesurer la similarité entre une image requête et l'ensemble des images de la base d'images disponibles.

Le choix des caractéristiques constitue la première étape de la recherche d'images par le contenu et est déterminant pour la qualité des résultats. En effet, quelle que soit la mesure de distance utilisée ensuite pour rechercher des images, cette distance est fonction des caractéristiques choisies et les résultats vont dépendre de celles-ci.

Afin de construire un système de recherche le plus efficace possible, il convient de choisir l'ensemble des caractéristiques les plus discriminantes pour les bases d'images à traiter et l'application voulue. En effet, dans le cadre de la recherche de symboles, par exemple, il conviendra plus d'extraire des caractéristiques de forme que des caractéristiques de couleur par exemple. Par contre, pour reconnaître des images de paysages, les caractéristiques de couleur et de texture paraissent plus appropriées. Ainsi, les techniques de recherche d'images par le contenu obtiennent des résultats satisfaisants pour certains types de requêtes et certains types de bases d'images.

On parle de méthodes dédiées à un type d'application ou d'images. Par contre, les résultats obtenus par les méthodes actuelles appliquées à différents types de bases d'images, ou à des bases généralistes, ne sont pas satisfaisants [Smeulders 00].

Afin d'améliorer ces résultats, il est possible d'utiliser des techniques de retour de pertinence, évoquées dans la section 2.2.3. Le procédé de base reste le même qu'en indexation textuelle : lorsque le système renvoie des images résultats pour une requête donnée, l'utilisateur a la possibilité d'estimer leur pertinence. Le système, à partir des images jugées pertinentes et non pertinentes, va procéder à une nouvelle recherche et renvoyer un nouvel ensemble de résultats. Ce procédé de base peut être réitéré plusieurs fois, jusqu'à satisfaction de l'utilisateur, par exemple. Le nom de « bouclage de pertinence »est donc parfois utilisé en lieu et place de « retour de pertinence », en référence à cette répétition d'un même procédé de base. Ce processus de bouclage de pertinence, en recherche d'images par le contenu, peut être représenté par le schéma de la figure 2.6. A gauche, l'image requête est présentée. Elle est analysée par le système de CBIR qui retourne 6 images résultats (à droite), considérées comme les plus proches de la requête. Parmi ces images, l'utilisateur sélectionne celles qu'il juge pertinentes pour la requête, et le système de CBIR effectue une nouvelle recherche à partir de ces nouvelles informations.

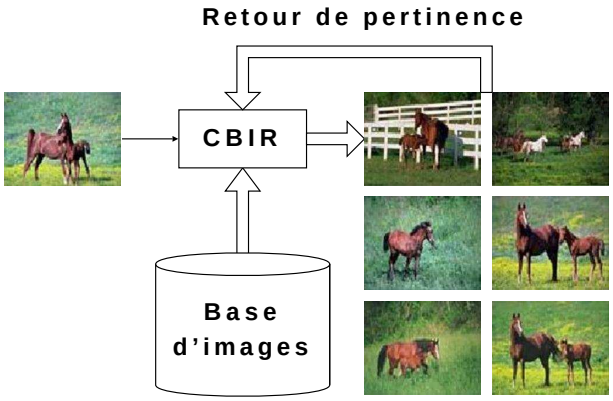



FIGURE 2.6 – Bouclage de pertinence dans un système de CBIR

Ces techniques de bouclage de pertinence améliorent les résultats mais restent coûteuses pour l'utilisateur et empêchent les systèmes d'être complètement automatiques. Pour plus de détails concernant l'utilisation des techniques de retour de pertinence en CBIR, on conseillera les lectures suivantes [Azimi Sadjadi 09, Liu 08b, Leon 07].

Enfin, pour améliorer encore les résultats des méthodes de recherche d'images par le contenu, une solution consiste à combiner différentes caractéristiques visuelles : par exemple plusieurs caractéristiques de forme dans une application de reconnaissance de symboles, ou des caractéristiques de forme et de couleur sur des bases plus généralistes. Les caractéristiques combinées peuvent être prises en compte avec la même importance, dans ce cas la dimension de l'espace vectoriel augmente avec le nombre de caractéristiques utilisées mais le calcul de similarité entre une image requête et les autres images reste le même. Cependant, dans le cas de la combinaison de caractéristiques, il peut être plus intéressant d'établir de nouvelles mesures de distances afin de pondérer chaque caractéristique. La combinaison de caractéristiques [Lin 09a, Kotsia 08, Terrades 07] améliore les résultats. Par contre, elle peut parfois poser des problèmes de stockage et de temps de calcul de part l'augmentation des dimensions.

Il est aussi possible de combiner des classificateurs [11] [Gunes 03, Ramos-Terrades 09]. A la différence de la combinaison de caractéristiques, où un seul classificateur est utilisé pour combiner plusieurs caractéristiques, les approches de combinaison de classificateurs prennent une décision globale à partir des décisions individuelles prises par chaque classificateur.

Malgré ces améliorations, les résultats obtenus en CBIR restent insatisfaisants sur des bases d'images généralistes ou « naturelles ». Trouver des méthodes de recherche d'images non dédiées reste un problème ouvert.

## 2.4 Synthèse et choix d'une méthode d'indexation : indexation visuo-textuelle

L'indexation est une phase très importante pour un système de recherche d'images car de sa qualité dépend la qualité des réponses du système et donc les performances de ce dernier. Une bonne indexation doit permettre de retrouver toutes les images pertinentes au besoin de l'utilisateur, et pas (ou peu) d'images non pertinentes pour celui-ci. Nous avons vu que l'indexation textuelle est intéressante dans le sens où elle va permettre de décrire le contenu sémantique des images, *i. e.* de mettre un mot sur une image. Par contre, dans le cas de l'indexation manuelle (la plus efficace), elle est coûteuse pour l'utilisateur mais surtout subjective. Pour pallier ce problème de subjectivité, la meilleure solution consiste à fournir aux utilisateurs un ensemble de mots ou termes d'indexation (vocabulaire), pour éviter que deux utilisateurs utilisent deux termes différents pour évoquer le même concept. Mais, dans ce cas, l'utilisateur a plus de contraintes dans sa façon de formuler sa requête. Enfin, la formulation des requêtes peut paraître difficile et peu intuitive pour un utilisateur non expert. Or, une requête mal formulée peut conduire à des erreurs de recherches. Une indexation des images basée uniquement sur l'information textuelle semble donc limitée.

Concernant l'indexation basée sur des caractéristiques visuelles de l'image, elle donne des résultats satisfaisants pour certains types de requêtes et certains types de base d'images. Cependant, l'utilisateur ne dispose pas toujours d'une image pour représenter ses besoins, et, dans ce cas, une requête textuelle semble plus appropriée. Enfin, la recherche d'images par le contenu ne fournit pas d'information sémantique. En effet, les résultats d'une recherche d'images par le contenu permettront de dire que deux images sont similaires, mais pas de mettre un terme sur l'information qu'elles contiennent. On parle de « fossé sémantique »(plus connu sous le nom de semantic gap, en anglais) [Smeulders 00]. Le fossé sémantique fait référence au fait qu'il est difficile de faire le lien entre les caractéristiques visuelles extraites des images (caractéristiques de bas niveau) et des informations sémantiques (caractéristiques de haut niveau).

Afin d'améliorer la reconnaissance, une solution consiste à combiner les informations visuelles et sémantiques : on parle d'approches visuo-textuelles. Plusieurs chercheurs ont déjà exploré cette possibilité [Ziou 09, Shotton 09, Magalhaes 07, Mulhem 06, Barnard 03, Grosky 01]. Par exemple, Barnard et al. [Barnard 03] segmentent les images en régions. Chaque région est représentée par un ensemble de caractéristiques visuelles et un ensemble de mots-clés. Les images sont alors classifiées en modélisant de façon hiérarchique les distributions de leurs mots-clés et caractéristiques visuelles. Grosky et al. [Grosky 01] associent des coefficients aux mots afin de réduire la dimensionnalité. Les vecteurs de caractéristiques visuelles et les vecteurs d'indices correspondant aux mots-clés sont concaténés pour procéder à la recherche d'images. Magalhaes et al. [Magalhaes 07] utilisent la théorie de l'information pour développer un modèle efficace

11. on peut également utiliser le terme « classifieur », par abus de langage par rapport au mot anglais « classifier »

sur plusieurs types de documents (documents textuels, images ou document contenant à la fois du texte et des images). Cependant, ces approches restent coûteuses pour l'utilisateur car elles requièrent de l'information textuelle pour chaque image. Or, cette information textuelle est de meilleure qualité lorsqu'elle est obtenue manuellement *i. e.* fournie par l'utilisateur.

Finalement, afin de réaliser le meilleur compromis possible entre efficacité de recherche et coût pour l'utilisateur, la solution semble être de trouver des systèmes de recherche d'images basés sur des caractéristiques visuelles et une indexation partielle par mots-clés appartenant à un vocabulaire pré-établi. Dans ce cas, le coût pour l'utilisateur est réduit dans le sens où il ne fournit des mots-clés que pour un sous-ensemble des images disponibles.

En dernier lieu, l'annotation automatique d'images peut éventuellement être utilisée pour obtenir les mots-clés manquants et améliorer la classification visuo-textuelle, sans aucun coût pour l'utilisateur. Dans ce cas, des modèles permettant à la fois les applications de recherche d'images et d'annotation automatique d'images semblent particulièrement indiqués. C'est le cas de certains modèles graphiques probabilistes sophistiqués, tels le « Gaussian-multinomial mixture model »(GM-Mixture), le « Latent Dirichlet Allocator »(LDA) et le « correspondence LDA »(CLDA), [Blei 03]. Ces modèles seront expliqués dans le chapitre 4 consacré aux techniques d'annotation. Enfin, le modèle [Metzler 04] est également adapté aux deux tâche de recherche et d'annotation : il utilise des méthodes non paramétriques pour estimer les probabilités d'un réseau d'inférence et peut être utilisé pour des tâches de recherche et d'annotation d'images.

# Chapitre 3

# Méthodes de classification

**Sommaire**



## 3.1 Introduction

Une fois les images indexées, le problème est de pouvoir y accéder simplement. Pour ce faire, il existe plusieurs solutions correspondant à des besoins bien particuliers. Comme on l'a vu dans l'introduction générale (chapitre 1), on va distinguer la recherche d'images (lorsque l'utilisateur a une idée précise de ce qu'il recherche, et qu'il est capable d'exprimer son besoin via des mots-clés et/ou une image exemple), de la classification (quand l'utilisateur n'a pas vraiment d'idée de ce qu'il recherche, la classification d'une base pourra l'aider dans sa recherche, car il parcourra la base classe par classe et évitera ainsi un parcours séquentiel).

En effet, de la recherche d'images proprement dite, le domaine évolue vers des tâches plus spécifiques, comme la classification qui permet de regrouper entre elles des images ayant des thématiques proches. Appliquée à une base d'images, la classification va permettre de fournir une représentation simplifiée et ordonnée de cette base. Elle permettra ainsi une manipulation et un accès à l'information faciles et rapides dans de grandes bases d'images.

La recherche d'images a fait l'objet du chapitre 2. Dans ce chapitre, nous nous intéressons aux différents types de méthodes de classification utilisées en reconnaissance d'images. Nous y décrivons les méthodes existantes de classification d'images. L'analyse des avantages et inconvénients de ces méthodes nous conduira à introduire l'approche que nous avons choisie.

Ce chapitre est organisé comme suit : en section 3.2, nous présentons brièvement deux types de méthodes de classification. Les méthodes de classification peuvent être réparties en trois catégories : la classification supervisée, dont différentes approches sont expliquées section 3.2.1, la classification non supervisée (plus connue sous le nom de « clustering », en anglais), qui fait l'objet de la section 3.2.2 et les approches semi-supervisées. Cependant, nous ne décrirons pas les approches semi-supervisées, car elles consistent, en général à faire intervenir l'utilisateur, en lui faisant valider ou non les résultats d'une classification non supervisé. Enfin, grâce à l'étude

des différentes méthodes de classification existantes, nous justifions, section 3.3 notre choix pour les modèles graphiques probabilistes. Une étude de ces modèles fera l'objet du chapitre 5.

## 3.2 Les différents types d'approches

Une fois l'ensemble des caractéristiques choisi, il faut donc choisir une méthode de classification. La classification consiste alors à identifier les classes auxquelles appartiennent les images à partir des caractéristiques préalablement choisies et calculées. Pour se faire, on va induire (apprendre) un classificateur (classifier en anglais) à partir des données. Un classificateur peut être défini comme étant une fonction qui associe une classe à un objet, représenté par un ensemble de caractéristiques. On distingue différents types d'approches de classification.

Lorsque les classes sont connues et que l'on dispose d'exemples de chaque classe, on parle de classification supervisée. Les images pour lesquelles la classe est connue vont servir d'échantillon d'apprentissage. Le problème consiste alors à trouver la classe d'une nouvelle image. Typiquement, l'ensemble des images étiquetées (dont on connaît la classe), est utilisé pour apprendre des descriptions des classes et on utilise ces descriptions afin d'affecter les nouvelles images à une classe. On parle d'apprentissage automatique (machine learning).

Dans le cas du clustering (classification non supervisée), le problème consiste à regrouper un ensemble d'images inconnues (non étiquetées) en clusters (groupes) significatifs. Dans un sens, chaque cluster correspond à une étiquette inconnue. Le nombre de clusters est également inconnu. Les clusters sont obtenus exclusivement à partir des données. Les méthodes non supervisées ne nécessitent aucune intervention humaine.

Enfin, il existe aussi des approches « interactives »de type apprentissage semi-supervisé [Wenyin 07] où l'utilisateur intervient dans le sens où il a la possibilité de corriger des mauvaise affectations possibles d'images déjà reconnues. L'apprentissage sera ainsi progressif, et on pourra atteindre un degré de précision de reconnaissance satisfaisant. Grâce à cet apprentissage progressif, le système acquiert plus de précision concernant l'information retenue dans les images et on obtient un meilleur taux de reconnaissance.

Nous présentons ici un bref aperçu des méthodes de classification supervisée 3.2.1 et non supervisée 3.2.2 les plus courantes, accompagné de leurs avantages et inconvénients respectifs. Pour les méthodes non supervisées, nous nous contentons de donner le principe général des méthodes couramment utilisées en reconnaissance de formes, et de citer leurs extensions et améliorations. De même, pour des informations détaillées sur les techniques de clustering, on conseillera au lecteur les études [Filippone 08, Berkhin 06]. Par contre, nous avons apporté plus d'attention à l'état de l'art des méthodes supervisées. Ce choix est lié au contexte industriel de cette thèse (financement CIFRE), évoqué dans l'introduction générale 1 , qui nous a orientés vers des applications supervisées.

### 3.2.1 Méthodes supervisées

Dans le cadre de la reconnaissance de formes, les méthodes de classification supervisée tentent d'apprendre, à partir d'images étiquetées (pour lesquelles la classe est connue), constituant un échantillon d'apprentissage, une fonction de classification. Cette fonction permettra d'associer une valeur de classe à chaque image non étiquetée.

#### 3.2.1.1 Méthodes basées sur le concept de similarité

Le moyen le plus simple de faire de la classification est de définir une fonction de distance entre les vecteurs caractéristiques, et d'affecter chaque image d'entrée inconnue à la classe dont le barycentre est le plus proche de l'image requête, selon la fonction de distance définie. Pour plus d'informations sur les différentes distances, on renverra le lecteur à la section 2.3.4.

#### 3.2.1.2 $k$ plus proches voisins (*kppv*)

Une légère variation des méthodes précédentes est l'algorithme des $k$ plus proches voisins (*kppv* ou *knn* en anglais pour $k$ nearest neighbor) : plusieurs représentants sont pris pour chaque classe et, pour chaque image d'entrée, l'ensemble des $k$ plus proches voisins est construit. L'image est affectée à la classe qui a le plus de représentants dans cet ensemble.

Afin d'illustrer cet algorithme, la figure 3.1 (respectivement 3.2) représente un problème de classification à 2 classes avec un algorithme *kppv* où $k = 1$ (respectivement $k = 3$). Sur les deux figures, un ensemble de points sont répartis dans le plan. Les données étiquetées de classe 1 (respectivement de classe 2) sont représentées par les points verts (respectivement par les points rouges). Deux observations non étiquetées (les données à classer), sont représentées par les points A et B.

Dans la figure 3.1, afin de classer les points A et B dans un voisinage de $k = 1$ point, on recherche le plus proche voisin de A et le plus proche voisin de B. Les deux cercles noirs entourent chaque point à classer et son plus proche voisin. Le plus proche voisin de A est un point rouge, A sera donc affecté à la classe des points rouges, *i. e.* la classe 2. Il en est de même pour les point B.

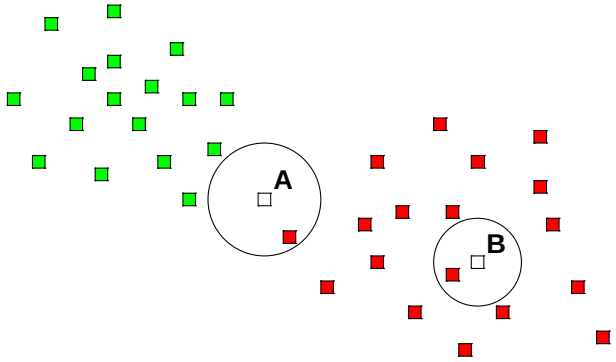



FIGURE 3.1 – Exemple de classification biclasses avec un *kppv* où $k = 1$

Considérons maintenant le même problème, mais avec un voisinage de $k = 3$ points (figure 3.2). Afin de classer les points A et B, on recherche les 3 points les plus proches de A et les 3 points les plus proches de B. Les deux cercles noirs entourent chaque point à classer et ses trois plus proches voisins. Parmi les 3 points les plus proches de A, il y en a 2 verts et 1 rouge. Un majoritaire est effectué et le point A sera donc affecté à la classe des points verts, *i. e.* la classe 1. Par contre, les 3 points les plus proches de B sont tous rouges. Le point B sera donc affecté à la classe des points rouges, *i. e.* la classe 2.

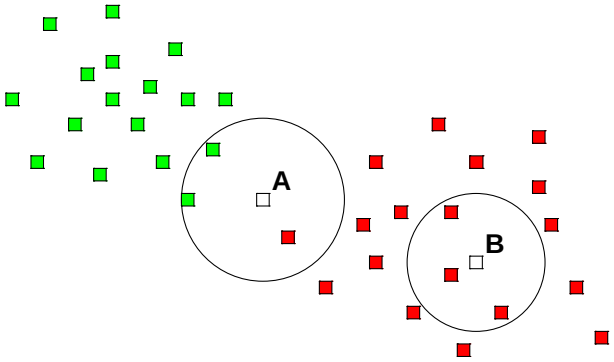



FIGURE 3.2 – Exemple de classification biclasses avec un *kppv* où $k = 3$

Contrairement aux autres méthodes de classification qui seront présentées dans cette section (arbres de décision, réseaux de neurones, ...), il n'y a pas d'étape d'apprentissage consistant en la construction d'un modèle à partir d'un échantillon d'apprentissage. C'est l'échantillon d'apprentissage, associé à une fonction de distance et d'une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle. Les *knn* rentrent alors dans la catégorie des modèles « non paramétriques ».

L'avantage principal de cette méthode est sa simplicité et le fait qu'elle ne nécessite pas d'apprentissage. L'introduction de nouvelles données permet d'améliorer la qualité de la méthode sans nécessiter la reconstruction d'un modèle. C'est une différence majeure avec des méthodes telles que les arbres de décision et les réseaux de neurones. Concernant l'interprétation des résultats, la classe attribuée à un exemple peut être expliquée à partir des plus proches voisins qui ont amené à ce choix. Cette méthode est également robuste au bruit. Enfin, elle peut être utilisée en présence de caractéristiques de grande dimension (*i. e.* les caractéristiques pour lesquelles le nombre de variables est très grand vis à vis du nombre d'observations).

Cependant, dans ce cas, il faut veiller à disposer d'un nombre assez grand d'enregistrements par rapport au nombre d'attributs et à ce que chacune des classes soit bien représentée dans l'échantillon choisi. Sans ça, la méthode donnera de mauvais résultats car la proximité sur les attributs pertinents sera noyée par les distances sur les attributs non pertinents. De plus, cette procédure de classification est lourde car chaque image requête est comparée (sur la base des ses caractéristiques) à toutes les images stockées (sauf si une structure d'index a été utilisée, auquel cas la comparaison se fera avec les images d'une région uniquement). D'autre part, la performance des *knn* est très dépendante de la mesure de distance utilisée et de la valeur de $k$ choisie. Enfin, cette méthode nécessite un espace mémoire important pour stocker les données.

Afin d'améliorer les performances de bonne classification en résolvant certains de ces problèmes, des extensions du *knn* standard ont été proposées [Samet 08, Angiulli 07, Ghosh 06, Keller 85]. Par exemple, l'approche proposée dans [Angiulli 07] utilise un *knn* standard mais en utilisant seulement un sous-ensemble des données étiquetées comme échantillon d'apprentissage. L'espace de recherche et le temps de calcul sont ainsi réduits. L'article [Ghosh 06], fournit, quant à lui, une méthode automatique de choix d'une valeur optimale de $k$. En 1985, Keller [Keller 85] a proposé une nouvelle approche, appelée « fuzzy KNN classifier algorithm »(*FKNN*), basée sur la combinaison de la théorie des ensemble flous et de l'algorithme *knn* standard. À la différence du *knn* standard le *FKNN* utilise la notion d'appartenance floue, donnée par la fonction suivante :

$$\mu_i(X) = \frac{\sum_{j=1}^{K} \mu_i(X_j) d(X, X_j)^{-2/p-1}}{\sum_{j=1}^{K} d(X, X_j)^{-2/p-1}}$$

où
– $\mu_i(X)$ est la valeur de l'appartenance floue de l'exemple $X$ à la classe $i$,
– $p$ est le coefficient flou,
– $K$ est le nombre de voisins considérés,
– $d(X, X_j)$ est la distance entre $X$ et son $j$-ième plus proche voisin,
– $\mu_i(X_j)$ est la valeur d'appartenance floue du $j$-ième plus proche voisin de $X$ à la classe $i$

Pour un exemple $X$ donné, son appartenance floue est calculée pour chaque classe $i$. L'exemple est affecté à la classe ayant la plus grande valeur d'appartenance floue. La performance du *FKNN* dépend du choix des valeurs des paramètres $p$ et $K$,

Les avantages de la méthode *knn* font d'elle une approche encore très répandue en reconnaissance de formes [Kondo 09, Blanzieri 08].

#### 3.2.1.3 Réseaux de neurones

Les réseaux de neurones sont des outils très utilisés pour la classification, l'estimation, la prédiction et la segmentation. Ils sont issus de modèles biologiques, sont constitués d'unités élémentaires (les neurones) organisées selon une architecture. Par exemple, la figure 3.3 présente un réseau de neurones couramment utilisé en classification : le perceptron multi-couches. Pour comprendre ce que sont les réseaux de neurones, et comment les mettre en œuvre dans un but de classification, on conseillera au lecteur les lectures suivantes [Bishop 95, Haykin 98].

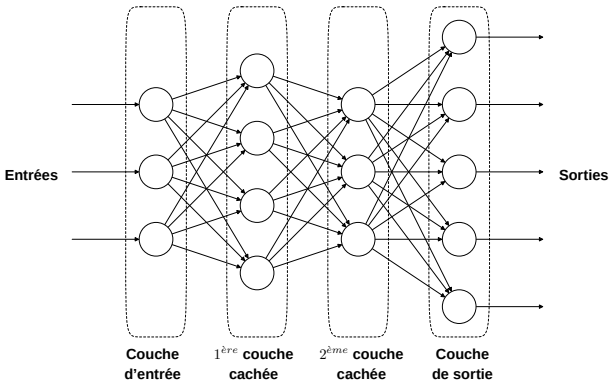



FIGURE 3.3 – Exemple d'un perceptron multicouches

Les réseaux de neurones sont réputés pour leurs bonnes performances, en terme de taux de reconnaissance et de vitesse de classification, dans plusieurs domaines et en particulier pour la reconnaissance de formes. Ils sont donc bien adaptés pour des problèmes comprenant des variables continues éventuellement bruitées. Ils ont aussi l'avantage de fonctionner en présence de données manquantes ou fausses et sont robustes au passage à l'échelle. Ces avantages font la popularité des approches neuronales en reconnaissance de formes [Humphreys 09, Fang 08].

Par contre ils sont plus sensibles aux valeurs aberrantes que les autres approches. De plus les réseaux de neurones ont un côté « boîte noire » : les résultats de la classification sont difficilement interprétables. En effet, le résultat de l'apprentissage est un réseau constitué de cellules

organisées selon une architecture, définies par une fonction d'activation et un très grand nombre de poids à valeurs réelles. Ces poids sont difficilement interprétables. Pour un vecteur d'entrée, il est donc difficile d'expliquer le pourquoi de la sortie (classe) calculée. De plus, il n'est pas facile, sans expérience approfondie, de choisir l'architecture et de régler les paramètres d'apprentissage. La construction d'un réseau de neurones est encore plus difficile pour des problèmes contenant un grand nombre de caractéristiques pour les entrées, ce qui est souvent le cas en reconnaissance de formes. Enfin, l'échantillon nécessaire à l'apprentissage doit être suffisamment grand et représentatif des sorties attendues, comme pour le *kppv*.

Comme pour les *kppv*, des améliorations ont été et continuent d'être proposées afin de résoudre ces problèmes [Plaza 09, Zhang 08, El-Bakry 07]. Par exemple, l'approche proposée dans [Plaza 09] vise à sélectionner l'échantillon d'apprentissage le plus informatif possible afin d'améliorer la performance de classificateur neuronal. Quant à l'approche [El-Bakry 07], elle a été proposée afin de réduire le temps de classification. Enfin, les approches combinant la théorie des réseaux de neurones et la logique floue, comme dans [Patil 07], permettent de réduire le manque d'interprétabilité des approches neuronales plus classiques.

#### 3.2.1.4 Arbres de décision

Un arbre de décision est une représentation graphique d'une procédure de classification. Chaque nœud interne correspond à un test, *i. e.* une condition spécifique sur la valeur d'une caractéristique particulière : ce sont des nœuds de décision. Les feuilles de l'arbre sont les classes. La classification est faite en descendant dans l'arbre le long des branches, selon le résultat du test au niveau de chaque nœud, jusqu'à ce que les feuilles de l'arbre soient atteintes. En effet, les arcs issus d'un nœud correspondent aux résultats du test correspondant à ce nœud. Par exemple, lorsque les tests sont binaires, le fils gauche correspond à une réponse positive au test et le fils droit à une réponse négative. Par exemple, la figure 3.4 montre un arbre de décision à 7 classes $D_1, \ldots D_7$.

Finalement, un arbre de décision fournit un ensemble règles de décision. Les arbres sont générés grâce à des algorithmes d'apprentissage. Les deux algorithmes les plus connus et les plus utilisés sont CART (Classification And Regression Trees [Breiman 84]) et C5 (version la plus récente après ID3 et C4.5 [Quinlan 93]). Ces algorithmes sont très utilisés car performants et car ils génèrent des procédures de classification exprimables sous forme de règles.

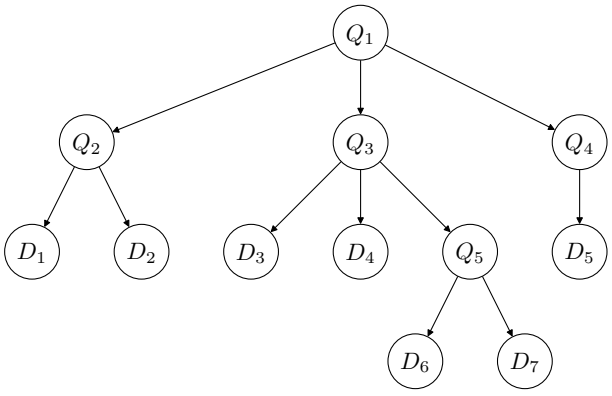



Figure 3.4 – Exemple d'arbre de décision

Cette approche a l'avantage de rendre le résultat de la classification interprétable pour l'uti-

lisateur, puisque la procédure de classification est représentée graphiquement. De plus, une fois l'arbre construit, le temps de calcul est réduit. En effet, l'attribution d'une classe à un exemple revient au parcours d'un chemin dans un arbre. Les arbres de décisions sont capables de traiter indifféremment les données discrètes et continues. Ils peuvent aussi gérer les valeurs manquantes et sont peu sensibles au bruit et aux points aberrants : ces points sont regroupés dans des feuilles de petite taille (*i. e.* des feuilles avec très peu d'individus). Enfin, cette méthode est non paramétrique, *i. e.* qu'elle ne postule aucune hypothèse *a priori* sur la distribution des données. Ces avantages rendent les arbres de décisions adaptés aux problèmes de reconnaissance de formes, et justifient l'engouement pour de telles méthodes dans ce domaine [Liu 08c, Zambon 06].

Cependant, ces méthodes exigent beaucoup de données d'apprentissage. D'autre part, les performances tendent à se dégrader lorsque le nombre de classes devient trop important. De plus, les variables situées en haut de l'arbre ont beaucoup de poids. En effet, elles influencent grandement le reste de la construction. Ceci peut conduire à changer complètement les prédictions en cas de modification d'une de ces variables, ce qui fait des arbres de décision une méthode d'apprentissage plutôt instable. Enfin, comme pour les réseaux de neurones, l'apprentissage n'est pas incrémental et, par conséquent, si les données évoluent avec le temps, il est nécessaire de relancer une phase d'apprentissage pour s'adapter à cette évolution. Afin de résoudre les problèmes d'instabilité, de nouvelles méthodes d'apprentissage d'arbres de décisions sont encore proposées [Ouyang 09, Kang 09].

**Forêts aléatoires (« Random forests »)** Les forêts aléatoires ont été introduites par Leo Breimain en 2001 [Breiman 01]. Une forêt aléatoire est un ensemble d'arbres de décision, utilisés pour calculer un vote pour la classe la plus populaire. Une forêt est dite « aléatoire »car elle peut être induite par exemple via un tirage aléatoire des caractéristiques qui définissent l'espace de description des données d'apprentissage, ou encore via un tirage aléatoire des données d'apprentissage utilisées pour chaque classificateur de base. Depuis qu'elles ont été introduites, les forêts aléatoires ont fait l'objet de plusieurs études comparatives [Rodriguez 06, Geurts 06]. Celles-ci ont montré que les forêts aléatoires sont compétitives avec les autres classificateurs.

En effet, en utilisant des ensembles d'arbres, on obtient une amélioration significative de la prévision, par rapports aux arbres de décision. De plus, globalement, la vitesse moyenne de d'apprentissage des forêts aléatoires est similaire à celle de l'apprentissage de la structure d'un arbre de décision unique. Par contre, les forêts aléatoires exploitent mieux les échantillons de petites et de grandes tailles [Breiman 01, Breiman 84].

Ainsi, grâce à ces qualités, les forêts aléatoires sont devenues un outil de plus en plus répandu en reconnaissance de formes [Ramirez 09, Moosmann 08].

#### 3.2.1.5 Analyse discriminante

L'analyse discriminante est une famille de techniques destinées à décrire et à classer des individus caractérisés par un nombre important de variables [Hastie 01]. L'origine de cette méthode remonte aux travaux de Fisher et de Mahalanobis dans les années trente du siècle passé. Son but est de trouver le sous-espace de projection qui sépare au mieux les observations.

Le problème est alors ramené à un test de Student, puisqu'il faut que les moyennes intra-groupes des groupes formés par la projection soient significativement différentes. L'idée est alors, qu'étant donné que les données d'une même classe doivent se projeter sur une classe la plus compacte possible et que les différentes classes doivent être les plus séparées possibles, le rapport des variances inter-classes et intra-classes dans la projection qui sépare le mieux les classes est

maximal. On va donc utiliser un test $\mathscr{F}$ de Fisher et maximiser $\mathscr{F}$ qui est le rapport des variances inter-classes et intra-classes.

L'analyse discriminante décisionnelle a pour but l'explication d'une variable qualitative $C$ à $m$ modalités par $p$ variables quantitatives $X_j, j = 1, 2 \dots p$. Les fonctions linéaires discriminantes sont des combinaisons linéaires de ces variables, séparant au mieux les classes. Disposant d'un nouvel individu sur lequel on a observé les $X_j$ mais pas C, il s'agit maintenant de décider de la modalité de C (ou de la classe correspondante) de ce nouvel individu. On parle donc de problème d'affectation. L'analyse discriminante fournit des règles de décision (ou d'affectation) à partir d'un échantillon d'apprentissage sur lequel les appartenances aux classes sont connues.

Soient :

- X, un vecteur représentatif d'une observation :

$$X = {}^t(X_1, X_2 \dots X_p)$$

  où p = nombre de composantes du vecteur caractéristique
- $X_{l,i}$, la i-ième composante de la l-ième classe :

$$X_{l,i} = {}^t(X_{1,l,i}, X_{2,l,i} \dots X_{p,l,i})$$

  $\forall i = 1, 2 \dots n_l$, où $n_l$ = nombre d'observations dans la classe l et
  $\forall l = 1, 2 \dots s$, où s = nombre de classes = nombre de modèles.

- $\overline{\overline{X}}$, le barycentre global
- $\overline{X_l}$, le barycentre de la classe l = vecteur (matrice 1 ligne, p colonnes) des moyennes de chaque variable dans la classe l
- W = matrice de covariances intra-classes supposée identique dans chaque classe, de dimension $p * p$, symétrique et régulière.
- Soit X un tableau de données, à $n = \sum_{l=1}^{s} n_l$ lignes (individus) et p colonnes (variables). Les n observations sont partitionnées en l groupes. On suppose que le j-ième groupe a une distribution $\mathscr{N}(\overline{X_j}, W)$ [12]
- Soit x une nouvelle observation (vecteur 1 ligne, p colonnes). On affecte cette nouvelle observation au groupe avec la probabilité *a posteriori* la plus grande. Supposons que la probabilité *a priori* d'appartenance à un groupe ne change pas ($p_1 = p_2 = \dots = p_s$). Ainsi, la probabilité *a posteriori* de la classe j s'écrit :

$$\frac{\exp\left(-\frac{1}{2}||x - \overline{X_j}||^2_{W^{-1}}\right)}{\sum_{i=1}^{s} \exp\left(-\frac{1}{2}||x - \overline{X_i}||^2_{W^{-1}}\right)}$$

- Pour pouvoir affecter cette nouvelle observation il faut maximiser cette probabilité *a posteriori* ou bien minimiser la quantité :

$$||x - \overline{X_j}||^2_{W^{-1}} = {}^t(x - \overline{X_j})\, W^{-1} (x - \overline{X_j}) \tag{3.1}$$

Cette méthode est fiable seulement si les conditions des mesures restent invariantes *i. e.* si les données sont observées sous les mêmes conditions pendant et après l'apprentissage. Cependant, ceci est peu vraisemblable dans la plupart des situations réelles car les conditions de mesure

12. Cette hypothèse est considérée pour affirmer le caractère optimal de la technique de discrimination linéaire. Pour des observations non normales, ce caractère optimal n'est pas assuré, mais la méthode peut encore être justifiée si l'on s'en tient aux propriétés du second ordre, c'est à dire portant seulement sur les moyennes et les variances. La loi normale va servir à quantifier les probabilités des risques pris lors des tests.

dépendent très souvent de facteurs de variabilité importante, mais contrôlée, que nous appellerons facteurs de dérive. Dans ce cas l'échantillon d'apprentissage ne réussit pas à bien déterminer l'appartenance à un groupe d'un individu supplémentaire. On parle alors d'apprentissage partiel.

Afin de résoudre cet effet de dérive, des améliorations ont été proposées [Pcekalska 09, Harrison 09, Baccini 01]. Par exemple, A. Baccini et al. ont proposé dans [Baccini 01], une méthode d'apprentissage progressif en analyse factorielle discriminante (DA) appelée analyse factorielle discriminante conditionnelle (CDA). Contrairement à la DA, la CDA voit le taux de mauvaises affectations diminuer de façon significative, au fur et à mesure que de nouvelles unités sont reconnues. Ces résultats prometteurs nous ont conduits à adapter la CDA au problème de la reconnaissance de symboles. Mes travaux de DEA ont été consacrés à cette tâche. Les bons résultats que nous avons obtenus ont donné lieu à trois publications (*cf.* chapitre 1). Ces résultats sont présentés dans la figure 3.5. Cette figure montre l'évolution des taux de reconnaissance de la CDA (avec ou sans retour de pertinence) comparée à la DA. On constate que le taux de reconnaissance de la CDA avec retour de pertinence (courbe bleue) ne cesse de croître, *i. e.* l'apprentissage s'améliore avec le temps. Par contre, les taux de reconnaissance de la DA (courbe rouge) et de la CDA sans retour de pertinence (courbe verte) décroissent régulièrement de 92% à 74% pour la DA et de 85% à 74% pour la CDA. Cependant, le taux de la CDA est légèrement meilleur que celui de la DA.

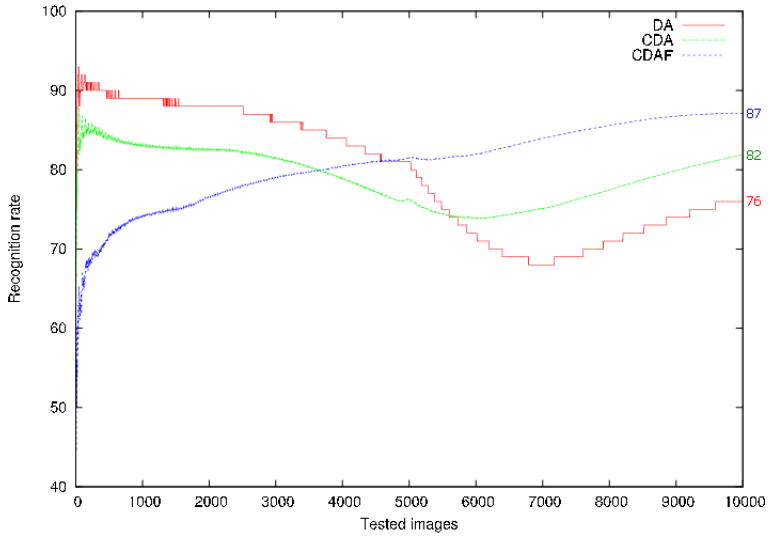



FIGURE 3.5 – Évolution du taux de reconnaissance de la CDA, avec ou sans retour de pertinence, comparée à la DA

Enfin, des approches non paramétriques ont aussi été proposées afin d'éviter l'hypothèse de normalité des données [Li 09c].

#### 3.2.1.6 « Séparateurs à vaste marge »ou « machines à vecteurs supports »(SVM)

Parmi les méthodes à noyaux, inspirées de la théorie statistique de l'apprentissage de Vladimir Vapnik [Vapnik 95], les SVM constituent la forme la plus connue. SVM est une méthode de classification binaire par apprentissage supervisé, introduite par Vapnik en 1995 [Cortes 95].

Cette méthode repose sur l'existence d'un classificateur linéaire dans un espace approprié. Puisque c'est un problème de classification à deux classes, cette méthode tente de séparer li-

néairement les exemples positifs des exemples négatifs dans l'ensemble des exemples. Elle est basée sur l'utilisation de fonction dites noyau (kernel) qui permettent une séparation optimale des données. En effet, grâce à la fonction de noyau, les caractéristiques initiales sont projetées dans un nouvel espace à grande dimension. Dans cet espace, les données sont linéairement séparables. La méthode SVM cherche alors l'hyperplan qui sépare les exemples positifs des exemples négatifs, en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale [Vapnik 06]. Intuitivement, cela garantit un bon niveau de généralisation car de nouveaux exemples pourront ne pas être trop similaires à ceux utilisés pour trouver l'hyperplan mais être tout de même situés franchement d'un côté ou l'autre de la frontière.

Les SVM sont réputés pour être très efficaces en reconnaissance de formes aussi ils ne cessent d'être adaptés à de telles applications [Adankon 09, Wang 09a]. Un autre intérêt est la sélection de Vecteurs Supports qui représentent les vecteurs discriminants grâce auxquels est déterminé l'hyperplan. Les exemples utilisés lors de la recherche de l'hyperplan ne sont alors plus utiles et seuls ces vecteurs supports sont utilisés pour classer un nouveau cas. Cela en fait une méthode très rapide. De plus, les SVM sont facilement extensibles à des frontières non-linéaires. Les SVM ont aussi l'avantage de pouvoir traiter à la fois des caractéristiques discrètes et continues. Enfin, ils sont surtout réputés pour être très performants en présence d'un grand nombre de variables (grâce à la régularisation), et ce en ayant besoin de peu d'exemples (données d'apprentissage), contrairement aux autres méthodes abordées dans cette section.

Au contraire, l'utilisation des SVM devient difficile lorsque la taille de la base d'apprentissage est importante. Mais le plus gros inconvénient des SVM réside dans le fait que les SVM standards ne possèdent pas d'extension naturelle à la discrimination à catégories multiples (on parle de « SVM multiclasses »). En effet, les SVM standards ne supportent que la classification biclasses, mais des extensions ont été proposées [Wu 08, Liu 06b] afin de prendre en compte les cas de classification multiclasses. Cependant, les performances des SVM multiclasses ne les distinguent pas significativement des méthodes de décomposition impliquant des SVM biclasses [Hsu 02, Allwein 00]. D'autre part, les SVM sont mal adaptés aux problèmes avec données manquantes. Enfin, ils nécessitent plusieurs expérimentations afin de déterminer les paramètres optimaux.

#### 3.2.1.7 La classification « Bayésienne »ou classification de Bayes

Les classificateurs Bayésiens utilisent des méthodes basées sur le Théorème de Bayes afin de déterminer les probabilités d'associer certaines classes à certaines instances selon les données d'entraînement (données d'apprentissage) [Mitchell 97]. En effet, un autre moyen de classer des objets représentés par un ensemble de caractéristiques $f_1, f_2, \ \dots \ , f_n$ est de considérer la classe et les caractéristiques comme des variables aléatoires et de calculer la distribution de probabilité conditionnelle $P(c_i|f), \forall i \in \{1, 2, \ \dots \ , C\}$ et d'affecter l'observation $f$ à la classe $i$ pour laquelle la probabilité *a posteriori* $P(c_i|f)$ est maximale. Ceci constitue un modèle probabiliste. Les probabilités $P(c_i|f)$ peuvent être calculées simplement grâce à la règles de Bayes, qui sera expliquée dans le chapitre 5, consacré aux modèles graphiques probabilistes.

Les méthodes Bayésiennes ont l'avantage d'être simples et efficaces, malgré les hypothèses d'indépendance (*cf.* chapitre 5) entre les variables, et les hypothèses sur les distributions de probabilités des variables, qui ne sont pas toujours vérifiées dans la réalité. Ceci peut être expliqué par le fait que la classification ne demande pas des estimations exactes des probabilités, mais seulement que la probabilité maximum soit donnée à la bonne classe. De même, des études comparatives des algorithmes de classification [Kotsiantis 07, Kim 04] ont prouvé que le plus simple classificateur Bayésien, appelé « Naïve Bayes »(*cf.* chapitre 5), avait des performances similaires à celles des arbres de décision et des réseaux de neurones. De plus, ils se sont montrés

précis et rapides appliqués à de grandes bases de données. En outre, le raisonnement Bayésien permet de supporter les données bruitées et les données manquantes. Il associe des probabilités aux prédictions, ce qui est utile dans les nombreux domaines où les connaissances sont incertaines. De cette façon, le résultat de la classification est facilement interprétable. D'autre part, le raisonnement Bayésien permet le support de connaissances *a priori*. Contrairement à d'autres approches (comme les réseaux de neurones et les arbres de décision), la classification Bayésienne permet le traitement incrémentale des données.

Les méthodes de classification probabilistes, ont, de ce fait, été largement utilisées en reconnaissance de formes, et continuent de l'être [Shahrokni 09, Likforman-Sulem 08, Mezghani 08, Shi 07]. Par exemple, l'étude présentée dans [Shi 07] compare deux approches Bayésienne de la combinaison de caractéristiques : l'une combine plusieurs caractéristiques au sein d'un classificateur Bayésien. L'autre combine les résultats de plusieurs classificateurs, où chacun est utilisé avec une caractéristique.

Par contre la classification Bayésienne est associée à certains désavantages : son application nécessite des probabilités dont la détermination requiert typiquement de grandes quantités de données ou plusieurs connaissances *a priori*. Les méthodes Bayésiennes nécessitent un coût de calcul relativement élevé pour déterminer l'hypothèse optimale dans un cas général. De plus un modèle probabiliste n'est parfois pas un concept intuitif pour un expert du domaine.

### 3.2.2 Méthodes non supervisées

Dans le cadre de la reconnaissance de formes, les méthodes de classification non supervisées (aussi appelées méthodes de clustering), tentent de partitionner de grands ensembles d'images, en plusieurs sous-ensembles de plus petite taille, regroupant des images similaires. Ces sous-ensembles sont appelés clusters.

#### 3.2.2.1 Méthodes basées sur le concept de similarité

Le moyen le plus simple de faire du clustering est de définir une fonction de distance entre les vecteurs caractéristiques. Les partitions sont constituées en minimisant les distances entre les vecteurs d'un même partition (cluster), et en maximisant les distances entre les vecteurs de clusters différents.

L'efficacité de ces méthodes dépend de la mesure de distance (ou similarité), utilisée. Il n'est pas évident de trouver une mesure de distance qui permette de donner plus d'importance à un attribut qu'à un autre. Pour plus d'informations sur les différentes distances, on renverra le lecteur à la section 2.3.4. Enfin, avec ces méthodes, le nombre de clusters possibles est difficile à contrôler.

#### 3.2.2.2 Méthode $K-means$

L'idée de base des $K-means$ [Lloyd 82, Macqueen 67] est d'utiliser le centre d'un cluster (sa moyenne, means en anglais, d'où le nom de $K-means$, car il y a $k$ centres de clusters), pour représenter ce cluster. Chaque vecteur caractéristique est affecté au cluster dont le centre est le plus proche. A chaque nouvelle affectation, le centre de chaque cluster est recalculé.

La méthode des moindres carrés est la plus couramment utilisée pour minimiser la distance entre un vecteur caractéristique et un centre de cluster. Cette méthode s'avère plutôt rapide et converge souvent vers un optimum local [Filippone 08].Par contre, comme pour les méthodes basées sur le concept de similarité, elle présente l'inconvénient d'être dépendante de la mesure de distance utilisée. De plus, elle requiert que l'on puisse calculer une moyenne des observations

pour chaque cluster. Or, il n'est pas toujours possible de définir une moyenne (dans le cas de caractéristiques qualitatives ou catégoriques, par exemple). En outre, les $K-means$ nécessitent de fixer à l'avance le nombre de clusters. Enfin, ils sont sensibles au bruit et aux valeurs aberrantes (plus connues sous le nom de « outliers », en anglais) [Duda 01, Duda 73].

Afin de résoudre ces problèmes, des variantes de l'algorithme de base ont été proposées [Fan 09, Zhang 02b, Pelleg 00]. Par exemple, la variante à base de noyaux permet de traiter des problèmes non linéaires [Zhang 02b]. La variante $X-means$ permet d'automatiser le choix du nombre de clusters [Pelleg 00].

#### 3.2.2.3 Classification hiérarchique

La classification hiérarchique [Kaufman 90, Johnson 67] crée, à partir d'un ensemble d'images (représentées par des vecteurs caractéristiques), un arbre dans lequel les feuilles représentent les images et les nœuds internes représentent la similarité entre ces points.

Afin de construire cet arbre, on commence par établir un cluster par image (feuilles de l'arbre). Ensuite, on regroupe les clusters les plus proches deux par deux. Cette opération de regroupement est répétée jusqu'à ce qu'il ne reste plus qu'un seul cluster, qui sera à la racine l'arbre. Ce type de classification hiérarchique est dite « agglomérative », car elle regroupe les clusters dans un parcours ascendant de l'arbre.

On distingue la technique agglomérative de la technique divisive, qui, à partir d'un cluster constitué de toutes les images, va diviser les clusters en deux dans un parcours descendant de l'arbre, jusqu'à obtenir un cluster par image au niveau des feuilles de l'arbre.

L'arbre obtenu par les deux techniques est appelé dendogramme. Le partitionnement en clusters des données initiales est obtenu en coupant horizontalement le dendogramme au niveau désiré. Les nœuds connectés correspondent aux données d'un même cluster. Afin d'illustrer cet algorithme, un exemple est fourni dans la figure 3.6. A gauche, on observe la répartition de 6 points dans le plan. Les clusters obtenus grâce à la classification hiérarchique sont représentés par les cercles. A droite, le dendogramme qui a permis d'obtenir ces clusters est présenté.

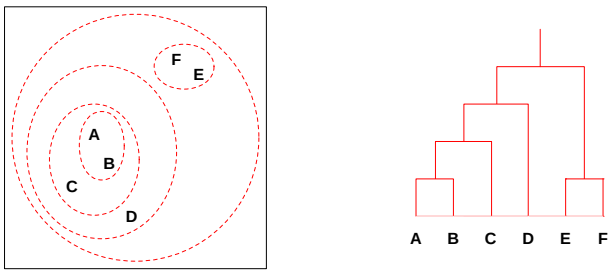



FIGURE 3.6 – Exemple de classification hiérarchique

Contrairement aux $K-means$, cette méthode a l'avantage de ne pas fixer à l'avance le nombre de clusters.Par contre, elle présente plusieurs inconvénients : elle supporte mal le passage à l'échelle. En effet, le temps de construction de l'arbre croît avec le nombre de données à classer [Berkhin 06]. Cependant, des algorithmes ont été proposés pour réduire la complexité des méthodes de classification hiérarchique [Dang 09, Goldberger 08]. De plus, comme les $K-means$, la classification hiérarchique est sensible au bruit et aux données aberrantes. Effectivement, si deux points de deux clusters différents sont proches, ces deux clusters risquent d'être regroupés

en un seul. Enfin, la qualité de la classification obtenue est dépendante de la mesure de distance entre clusters choisie, et du seuil de coupure.

#### 3.2.2.4 Cartes de Kohonen (Self-Organizing Maps)

Les cartes de Kohonen sont issues de la théorie des réseaux de neurones [Oja 02, Kohonen 98].

Dans les réseaux de neurones classiques (type perceptron multicouches) [Bishop 95], chaque neurone d'une couche est connecté à tous les neurones de la couche précédente et de la couche suivante (excepté pour les couches d'entrée et de sortie). Par contre il n'y a pas d'interactions entre les neurones d'une même couche.

Au contraire, dans une carte de Kohonen, les interactions entre neurones d'une même couche sont représentées : les neurones d'une carte de kohonen sont disposés sur une grille 2D. De plus, chaque neurone de la couche d'entrée est relié à chaque autre neurone de la carte. Le clustering est alors accompli en considérant une compétition entre les neurones. On calcule une distance entre le vecteur d'entrée (le vecteur caractéristique à reconnaître) et chaque autre neurone. Le principe du clustering est de considérer que les neurones qui sont proches les uns des autres interagissent différemment que les neurones qui sont loin les uns des autres : on encourage le neurone vainqueur (*i. e.* le plus proche du vecteur d'entrée) en faisant évoluer ses poids et ceux de son voisinage.

L'avantage principal de cette méthode est sa capacité à traiter les données de grande dimension.

Par contre, la convergence n'est pas assurée et le temps de convergence est long [Oja 02, Flanagan 01].

En termes d'applications au domaine de l'image, les techniques de clustering sont largement utilisées en segmentation [Yin 09, Scarpa 09], beaucoup plus que pour partitionner des bases d'images. Dans le cas de la segmentation, les techniques de clustering sont utilisées afin de partitionner une image en cluster. Chaque cluster correspond, à la fin de l'algorithme, à une région de l'image.

## 3.3 Synthèse et choix d'une méthode de classification et de recherche d'images

Dans le chapitre 2, nous avons vu qu'il semble judicieux de combiner information visuelle et sémantique afin de représenter et de rechercher des images. En effet, en terme de recherche d'images, on peut distinguer deux tendances. La première, appelée recherche d'images par le texte, consiste à appliquer des techniques de recherche de textes à partir d'ensembles d'images complètement annotés. L'efficacité de ces méthodes est étroitement liée à la qualité de l'indexation des images. Or, les méthodes d'indexation textuelle automatiques sont peu performantes et fournissent des ensembles d'images mal annotés, car elles utilisent l'URL, le titre de la page, ou d'autres attributs ou le texte proche de l'image dans le cas d'images provenant d'Internet, ou alors tout simplement le nom de l'image dans le cas d'images issues de collections personnelles. Quant à l'indexation textuelle manuelle, bien qu'elle soit plus performante que l'indexation textuelle automatique, elle est très coûteuse pour l'utilisateur et se révèle difficilement applicable ux grandes bases d'images.

Nous avons également vu, dans le chapitre 2, que la seconde approche, appelée recherche d'images par le contenu, est un domaine plus récent et utilise une mesure de similarité (similarité de couleur, forme ou texture) entre une image requête et une image du corpus utilisé. Ces

méthodes sont efficaces sur certaines bases d'images, mais leurs performances décroissent sur des bases d'images plus généralistes.

Afin d'améliorer la reconnaissance, une solution consiste à combiner les deux types d'informations. On parle d'indexation visuo-textuelle. C'est vers ce type d'indexation que nous voulons nous tourner. De plus, comme l'indexation textuelle manuelle est plus efficace que l'indexation automatique, c'est vers l'indexation manuelle que nous nous orientons pour extraire l'information textuelle des images. Par contre, afin d'être le moins contraignant possible pour l'utilisateur, nous permettrons à celui-ci d'indexer textuellement seulement un sous-ensemble d'images.

Il convient alors de trouver des méthodes de recherche et de classification d'images adaptées à ce type d'indexation. Or, l'information textuelle (sémantique), est en général fournie par un ensemble de mots-clés associés à une image. Concernant l'information visuelle, elle est souvent obtenue à l'aide de descripteurs qui fournissent des vecteurs caractéristiques (ou signatures) pour chaque image. De plus, ces vecteurs sont souvent de grande dimension. En terme de variables aléatoires, les mots-clés peuvent être vues comme des variables de type qualitatives (catégoriques) discrètes, et les vecteurs de caractéristiques visuelles fournissent en général des valeurs continues.

Ce constat va nous guider dans le choix d'une méthode de recherche et de classification adaptée. En effet, on souhaite s'orienter vers une méthode de classification ou de recherche pouvant combiner plusieurs types de variables : discrètes et continues, quantitatives et qualitatives. De cette façon, nous pourrons combiner à la fois plusieurs caractéristiques visuelles, ainsi que des caractéristiques textuelles. De plus, la méthode choisie se devra d'être efficace en grande dimension, afin de pouvoir manipuler les données visuelles. La robustesse en présence de données manquantes (dans le cas où l'information textuelle n'est pas disponible pour toutes les images), est également requise. Enfin, la méthode choisie devra supporter le passage à l'échelle pour pouvoir retrouver des images dans de grandes bases.

Parmi les techniques de recherche et de classification correspondant à ces spécificités, les SVM et les méthodes probabilistes semblent les plus adaptés. En effet, les SVM et les modèles probabilistes permettent de combiner différents types de variables. Ils sont aussi, et surtout, réputés pour être efficaces en grande dimension, contrairement aux modèles probabilistes. Par contre, les SVM sont moins adaptés aux données manquantes que les modèles probabilistes. Le problème d'efficacité en grande dimension des modèles probabilistes pouvant être évité en utilisant une méthode de réduction de dimension adaptée, c'est vers ces modèles que nous avons choisi de nous tourner. Parmi les modèles probabilistes, on va s'intéresser plus particulièrement aux modèles graphiques probabilistes, car ces derniers bénéficient non seulement des avantages des modèles probabilistes standards, mais ils possèdent en plus des avantages liés à leur représentation graphique. Ces avantages seront précisés dans le chapitre 5, qui est dédié à la définition et l'étude de ces modèles. Enfin, notre choix se justifie d'autant plus que les modèles probabilistes permettent à la fois de classer, rechercher et même annoter automatiquement des images. Cet aspect est abordé dans le chapitre suivant (chapitre 4), consacré aux techniques d'annotation.

# Chapitre 4

# Annotation d'images

**Sommaire**



## 4.1 Introduction

Dans le chapitre 2, nous avons vu que l'association d'information sémantique à une image est toujours un défi. En effet, l'indexation textuelle d'images se contente d'indexer les images sur la base de mots-clés représentatifs de l'image, mais sans utiliser le contenu (les caractéristiques visuelles) de l'image. De même, l'indexation d'images basée sur le contenu organise les images sur la base de caractéristiques visuelles de bas niveau (couleur, texture, forme, ...). Mais aucun lien entre information sémantique et visuelle n'est fait à ce niveau. On parle de « fossé sémantique »(plus connu sous le nom de semantic gap, en anglais) [Smeulders 00].

L'annotation d'images constitue une manière possible d'associer de la sémantique à une image, et ainsi de réduire le fossé sémantique. En effet, elle consiste à assigner à chaque image, un mot-clé ou un ensemble de mots-clés, destiné(s) à décrire le contenu sémantique de l'image. Cette opération peut être vue comme une fonction permettant d'associer de l'information visuelle, représentée par les caractéristiques de bas niveau (forme, couleur, texture, ...) de l'image, à de l'information sémantique, représentée par ses mots-clés.

L'annotation d'images va alors pouvoir être utilisée en amont de la recherche et de la classification d'images. En effet, les annotations pourront être utilisées pour indexer textuellement les images (voir chapitre 2). De ce fait, les annotations serviront à organiser et localiser les images recherchées et pour améliorer la classification et la recherche visuo-textuelles (voir chapitre 2 et 3).

Comme pour les méthodes de classification présentées dans le chapitre 3, les techniques d'annotation d'images peuvent être réparties en 3 catégories. On distingue l'annotation automatique

d'images, de l'annotation semi-automatique, qui nécessite l'intervention de l'utilisateur pour valider des annotations automatiques, dans un système de retour de pertinence, par exemple. Enfin on compte aussi l'annotation manuelle d'images, qui consiste à faire annoter des bases d'images par un ensemble d'utilisateurs, avec des mots-clés souvent issus d'un ensemble de mots-clés pré-défini appelé « vocabulaire ». L'annotation manuelle correspond donc à la première phase d'indexation textuelle des images, à savoir l'extraction de mots-clés (voir chapitre 2). L'annotation manuelle, bien que coûteuse (comme on l'a vu dans le chapitre 2), est souvent nécessaire pour créer des vérités-terrains, dans le cadre de la validation des approches automatiques.

Le but de ce chapitre est de présenter un bref état de l'art sur l'annotation d'images. A l'issue de ce chapitre, nous présenterons notre choix pour une méthode qui nous permette d'annoter nos bases d'images, ou d'étendre les annotations existantes à d'autres images, en réalisant le meilleur compromis possible entre la précision de l'annotation et le coût pour l'utilisateur.

On se restreint volontairement à l'étude des techniques d'annotation manuelle, qui sont souvent utilisées pour constituer les vérités-terrains (voir section 4.2), et des méthodes automatiques (voir section 4.3), puisque ce sont les moins coûteuses pour l'utilisateur. Nous n'étudions pas, ici, les méthodes semi-automatiques, car elles consistent, en quelque sorte, en un mélange des approches automatiques et manuelles. En effet, en général, les méthodes semi-automatique consistent à faire intervenir l'utilisateur pour valider les décisions du système. Cependant, pour plus d'informations sur ces techniques, on conseillera les lectures suivantes [Zhang 09b, Hanbury 08].

## 4.2 Annotation manuelle

Dans le chapitre 2, nous avons vu que l'indexation textuelle manuelle, dont la première phase, celle de l'extraction de mots-clés, est une étape d'annotation manuelle, était la plus efficace. Par contre, nous avons constaté que cette technique est coûteuse pour l'utilisateur et qu'elle devient très difficile à appliquer sur de grandes bases. De plus, l'image étant polysémique, on a vu qu'une même image pouvait être indexée différemment par différents indexeurs. Ce phénomène s'avère d'autant plus courant dans le cas de grandes bases d'images généralistes, où les indexeurs, même s'ils sont experts dans un ou plusieurs domaines, ne sont pas experts dans tous les domaines qui peuvent être représentés dans de grandes bases d'images généralistes.

Afin de pallier ces problèmes, et de pouvoir annoter et indexer manuellement de grandes bases d'images généralistes de façon correcte, des sites Web d'annotation d'images ont vu le jour.

Par exemple, l'outil d'annotation en ligne LabelMe [Russell 08], permet aux utilisateurs de segmenter les images en régions, puis d'annoter ces régions en choisissant des mots-clés dans une liste. La liste de mots-clés proposée est différente pour chaque image. Elle est établie en fonction des mots-clés déjà choisis par d'autres utilisateurs pour cette image. De cette façon, on limite les erreurs dues à l'ambiguïté des termes. De même, sur certaines images, on peut observer que des régions sont déjà délimitées. Cela veut dire que cette image a déjà été segmentée par un autre utilisateur, et que l'on peut observer les régions qu'il a délimitées. De plus, si une région a déjà été annotée, il suffit de passer la souris dessus pour afficher les mots-clés l'annotant. Par exemple, la figure 4.1 montre une image en cours d'annotation avec l'outil LabelMe. Détourées en bleu, rouge, et violet, on peut observer les régions déjà reconnues par d'autres utilisateurs. En particulier, le building, repéré en violet, est annoté par le mot-clé building. Les mots-clés qui ont déjà servi à annoter cette image figurent à droite de l'image. L'utilisateur peut alors segmenter l'image en entourant les régions/objets de son choix et/ou annoter des régions à

l'aide des mots-clés fournis.

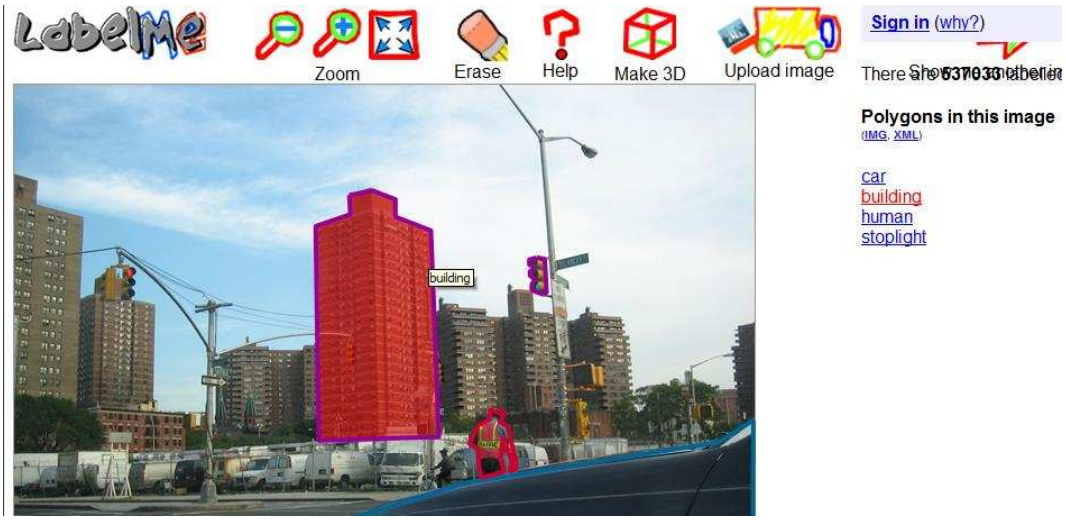



FIGURE 4.1 – Annotation d'une image avec LabelMe

Il existe d'autres outils d'annotation en ligne, présentés sous forme de jeu [Von Ahn 06], pour faire oublier à l'utilisateur le côté « coûteux »de l'annotation. L'exemple le plus connu est l'**ESP game** [von Ahn 04]. Dans ce jeu, la règle du jeu est la suivante : deux utilisateurs, connectés en même temps, se voient proposer par le système la même image. Chacun propose des mots-clés pour annoter cette image. Le système ne propose pas de vocabulaire. Par contre l'utilisateur n'est pas complètement libre d'utiliser les mots de son choix. En effet, une liste de mot tabous est fournie pour chaque image, et l'utilisateur ne peut pas choisir des mots de cette liste. Les mots tabous, pour une image donnée, correspondent aux mots déjà validés pour cette image. Si l'image n'a encore jamais été annotée, la liste de mots tabous est vide. Une fois que les deux utilisateurs sont d'accord sur un mot-clé, ce mot-clé est validé, et les deux utilisateurs gagnent des points. Une autre image leur est alors proposée, et ainsi de suite pendant 3 minutes. Quand un utilisateur atteint un certain nombre de points en une semaine, il se voit offrir un cadeau.

De ce fait, ces outils, en plus de motiver les internautes à annoter des images, grâce à l'appât du gain, résolvent aussi, en partie, le problème de polysémie de l'image. En effet, il faut que deux utilisateurs au moins aient choisi le même mot-clé pour la même image pour qu'il soit validé. Ceci réduit les erreurs d'annotation dues à la polysémie de l'image. Par contre, ce jeu conduit à des annotations redondantes. En effet, aucun vocabulaire n'est utilisé. La seule contrainte réside dans le fait que les mots déjà validés pour une image ne peuvent pas être choisis de nouveau. Ce fort degré de liberté peut conduire à annoter deux fois une même image avec le même mot mais en genre et/ou nombre différente. Par exemple, une image contenant des arbres pourra être annotée par les mots « tree »et « trees », qui seront considérés comme différents. Des approches ont donc été proposées afin d'améliorer la qualité des annotations en réduisant ce manque de normalisation [Robertson 09].

Un exemple de ce jeu sur une image est donné figure 4.2. Deux joueurs sont en train d'annoter cette image en même temps. On observe l'écran du joueur 1. Celui-ci peut soit proposer des mots pour annoter l'image (à l'aide du champ de saisie et du bouton « submit »en bas à droite de la page), soit passer à une autre image (à l'aide du bouton « pass ») s'il n'a pas d'idées de mots pour annoter l'image ou s'il n'arrive pas à se mettre d'accord sur un mot avec l'autre utilisateur.

On peut voir que le joueur 1 a déjà proposé 4 mots-clés (« water », « sky », « fire »et « building », affichés dans la liste à droite de l'image). A gauche de l'image, une liste de deux mots tabous (« lake »et « ship ») est donnée, dans laquelle les joueurs n'ont pas le droit de choisir leurs mots. Cela signifie que ces deux mots ont déjà été validés pour le système suite aux duels d'autres joueurs sur cette image. Au bout d'un certain temps, une fenêtre finit par afficher le message « Matched on : water ». Cela veut dire que le joueur 2 a enfin proposé le mot « water », en commun avec les mots déjà donnés par le joueur 1. Le système a donc validé ce mot comme annotation de l'image. Si l'image est de nouveau proposée à d'autres joueurs, la liste de mots tabous contiendra alors 3 mots : « lake », « ship »et « water ».


FIGURE 4.2 – Le jeu d'annotation d'images en ligne : EPS game

Enfin, dans notre équipe, nous avons mis en place notre propre site web d'annotation d'images, afin d'établir une vérité terrain sur nos bases d'images, pour faciliter l'évaluation de nos méthodes. Par exemple, la figure 4.3 montre la page de sélection d'une image à annoter. L'utilisateur choisit dans une liste déroulante la base d'images de son choix, et, dans cette base, une classe d'images de son choix. Les images de cette classe s'affichent alors en dessous. Ici, la classe « fleurs »a été choisie.


FIGURE 4.3 – Sélection d'une image à annoter - outil d'annotation de l'équipe

Une fois qu'une image a été sélectionnée dans une classe, cette image est affichée en bas de la page, ainsi qu'une liste de mots-clés, dans laquelle l'utilisateur peut choisir les mots-clés de son choix pour annoter l'image. Cette liste de mots correspond en fait à un vocabulaire contrôlé établi à la création de la base. Par exemple, la figure 4.4 montre l'image choisie dans la classe « fleurs »et les mots-clés du vocabulaire associé à cette base. Pour cette image, l'utilisateur a choisi, en les cochant, les mots-clés « flower »et « nature ». Une fois que les mots-clés choisis ont été validés, ils apparaissent surlignés en vert.

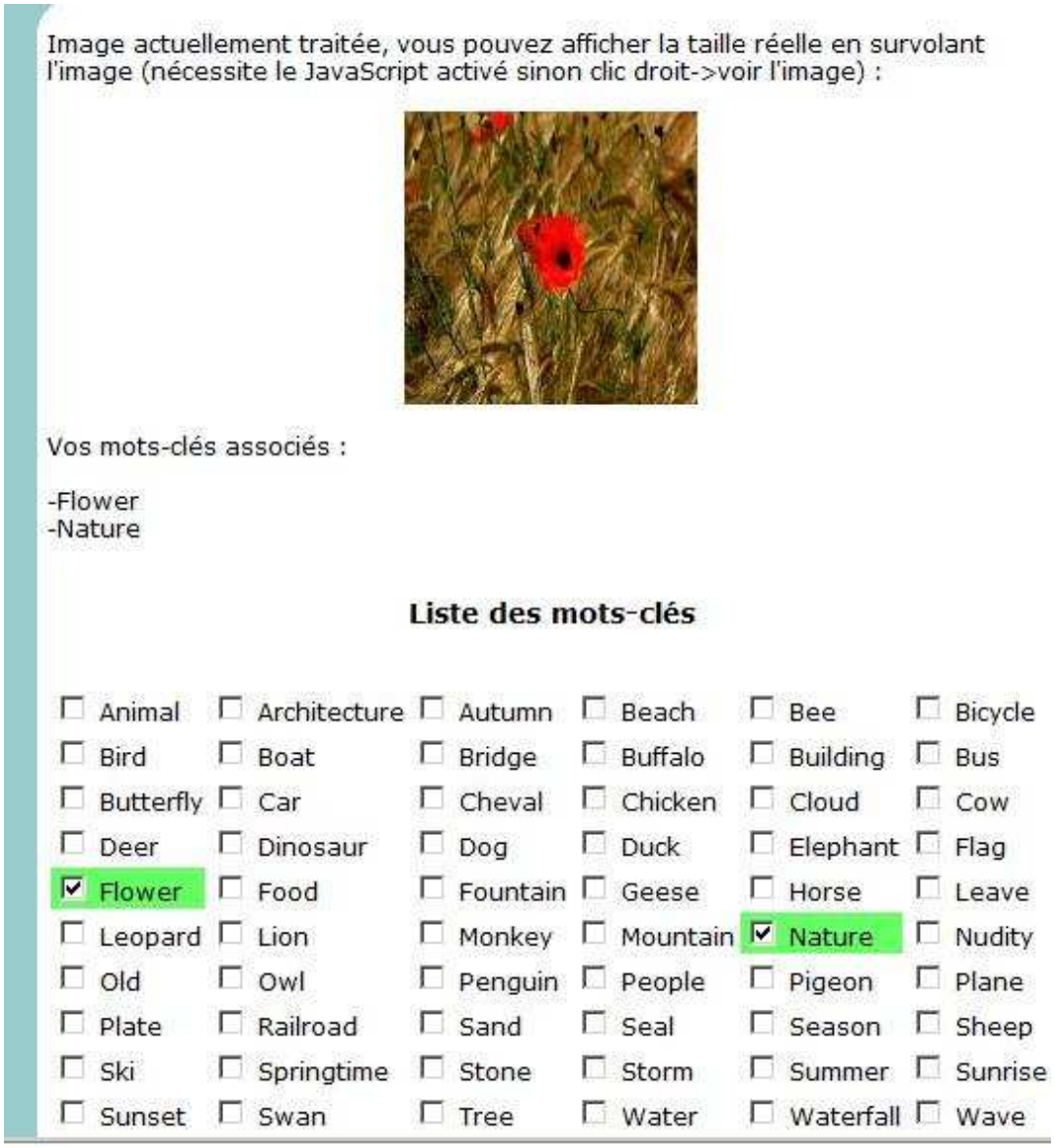



FIGURE 4.4 – Annotation d'une image - outil d'annotation de l'équipe

Ces sites web d'annotation d'images sont donc une idée astucieuse pour réduire les erreurs d'annotation liées à la polysémie de l'image. De plus, les sites sous forme de jeu, grâce à l'appât du gain et leur côté ludique et compétitif (chacun cherche à trouver le même mot que son adversaire, le plus rapidement possible), paraissent être la solution idéale pour motiver les utilisateurs à annoter des images, même dans de grandes bases. Par contre ils ne permettent pas aux utilisateurs de charger leurs propres images pour les proposer à l'annotation. De plus les images à annoter sont des images généralistes. L'annotation de grandes bases d'images liées à un domaine particulier ne peut donc pas être facilitée grâce à ces sites internet et reste la tâche, coûteuse, destinée à des indexeurs expert du domaine.

Pour pallier ce problème de coût de l'annotation qui subsiste malgré ces outils d'annotations, les méthodes d'annotation automatique ont fait leur apparition.

## 4.3 Annotation automatique

Nombre de travaux ont déjà été menés dans le domaine de l'annotation automatique [Wang 08, Wong 08, Fan 08]. Il en ressort plusieurs manières de traiter le problème d'annotation automatique. Cependant, la plupart de ces méthodes ont en commun le fait de nécessiter un échantillon d'apprentissage contenant des images annotées. Les annotations de nouvelles images seront apprises à partir de cet échantillon. Les sous-sections ci-dessous décrivent chaque type de méthode de façon générale. Puis certaines approches sont présentées plus précisément, de façon à cerner les problèmes qui restent ouverts encore aujourd'hui.

### 4.3.1 Méthodes basées sur les graphes

Récemment, les méthodes à bases de graphes se sont révélées efficaces pour résoudre de nombreux problèmes d'apprentissage, en particulier celui d'annotation d'images [Liu 09b, Rui 07, Liu 06a].

La plupart des méthodes d'annotation à base de graphes consiste à construire un graphe dit *de voisinage* ou *de similarité*. C'est à dire que chaque image est représentée par un ensemble de caractéristiques de bas niveau. Les vecteurs caractéristiques sont concaténés de façon à obtenir un vecteur de dimension $d$. Chaque image correspond donc à un point dans $R^d$. Le graphe de voisinage est construit de la façon suivante : chaque point constitue un sommet du graphe et les arêtes sont placées en respectant une propriété de voisinage particulière entre les points. Une fois le graphe construit, il s'agit d'attribuer des annotations à une nouvelle image inconnue du système. Le principe de base est « l'héritage ». En effet, il s'agit d'insérer la nouvelle image dans le graphe de voisinage, et de lui faire hériter des annotations de ses voisins en calculant des scores sur chaque annotation potentielle.

La plupart des méthodes existantes utilisent des méthodes de construction de graphes simples, telles que les graphes entièrement connectés, les méthodes $\epsilon$-ball, Minimum Spanning Tree (MST) ou les $k$ plus proches voisins. Ces méthodes de construction sont décrites brièvement ci-dessous.

#### 4.3.1.1 Graphes entièrement connectés

Le graphe de similarité est construit en reliant chaque sommet à tous les autres sommets. Un exemple de graphe entièrement connecté est donné dans la figure 4.5.

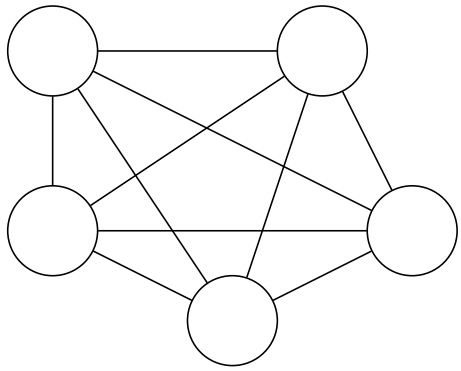


FIGURE 4.5 – Graphe entièrement connecté

#### 4.3.1.2 Graphes $\epsilon$-ball

Considérons $n$ points $P_1, P_2, \ldots, P_n$. Le graphe $\epsilon$-ball possède $n$ sommets $S_1, S_2, \ldots, S_n$, associés respectivement aux points $P_1, P_2, \ldots, P_n$. Deux sommets $S_i$ et $S_j$, $\forall (i,j) \in \{1, 2, \ldots, n\}$ et $i \neq j$ sont reliés par une arête si et seulement si $d(P_i, P_j) \leq \epsilon$ où $d$ désigne la distance euclienne.

#### 4.3.1.3 $k$ plus proches voisins

Le graphe des $k$ plus proches voisins est obtenu en reliant chaque point à ses $k$ plus proches voisins. Un exemple d'un tel graphe, avec $k = 2$ est donné dans la figure 4.6.

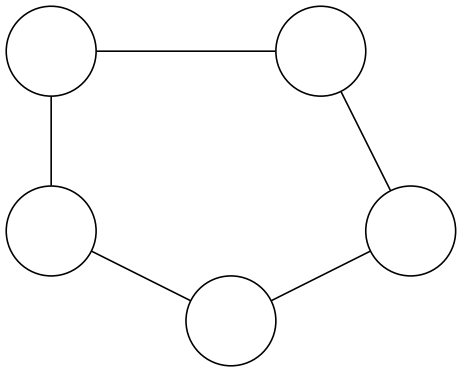


FIGURE 4.6 – Graphe des 2 plus proches voisins

#### 4.3.1.4 MST (Minimum Spanning Tree) ou *arbre couvrant minimal*

Pour construire un arbre couvrant minimal, on commence par construire un graphe pondéré (chaque arête est associée à un poids) entièrement connecté. Par exemple, on peut prendre en compte l'information mutuelle entre les individus afin d'attribuer des poids à chaque arête.

Ensuite, en sélectionnant tous les sous-graphes acycliques et connexes de ce graphe, on obtient l'ensemble des arbres couvrants du graphe initial. Un arbre couvrant minimal est un arbre couvrant dont le poids est plus petit ou égal à celui de tous les autres arbres couvrants du graphe. Le poids d'un arbre étant la somme des poids de ses arêtes.

#### 4.3.1.5 D'autres méthodes à base de graphes « inclassables »

Il existe d'autres types de méthodes à base de graphes. Par exemple,on peut citer l'approche de Pan et al. [Pan 04]. En effet, ils furent les premiers à proposer une méthode d'annotation automatique à base de graphes. Comme nous pouvons le voir dans la figure 4.7, dans leur travail, les images ($I_1, I_2 \ldots$), leurs annotations ($t_1, t_2 \ldots$) et leurs régions d'intérêt ($r_1, r_2 \ldots$, représentées par des attributs de bas niveaux) correspondent à des sommets de trois types différents. Chaque sommet de type « image »est relié par une arête pleine aux sommets correspondants aux attributs des régions et aux annotations de cette image. De plus, un autre type d'arêtes, en pontillés,est utiliser pour relier les sommets correspondants aux régions adjacentes d'une image. Enfin, les arêtes en pontillés sont également utilisées pour relier les régions les plus proches. Le graphe obtenu pour chaque image est appelé graphe « MMG »(un exemple est représenté dans la figure 4.7).

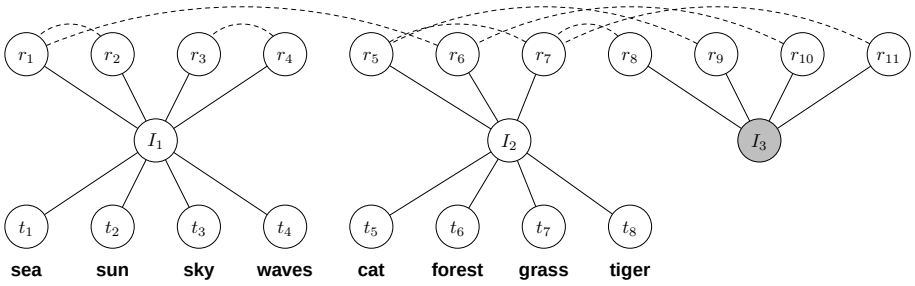



FIGURE 4.7 – Graphe « MMG »

Cette approche est plus complète que les méthodes à base de graphes de voisinages classiques puisqu'elle prend en compte d'autres types de relations. Par contre les caractéristiques de bas niveaux calculées sur chaque région sont la plupart du temps des données continues, tandis que les annotations constituent des données discrètes issues d'un vocabulaire fini. Il est difficile de traiter ces deux types de sommets, issus de différentes modalités, dans un seul graphe. De plus ce graphe ne représente pas le éventuelles relations entre les mots-clés.

#### 4.3.1.6 Conclusion sur les méthodes à base de graphes

Les méthodes à base de graphes ont l'avantage d'être indépendantes du domaine et simples à paramétrer. Mais, dans le problème d'annotation d'images, il est nécessaire de représenter une large quantité de données dans le graphe, d'autant plus que les caractéristiques de bas niveau utilisées sont de grande dimension. Ainsi les graphes $\epsilon$-ball ou les méthodes basées sur les $k$ plus proches voisins, sont plus adaptées que les méthodes à base de graphes entièrement connectés. Cependant, ces méthodes dépendent des paramètres $k$ et $\epsilon$ utilisés. Un mauvais réglage de ces paramètres peut entraîner des oublis d'arêtes ou au contraire de mauvaises connections entre les données. Les méthodes à base de MST semblent donc plus appropriées, mais les arbres couvrants minimaux présentent souvent beaucoup de branches et de connections entre un sommet et plusieurs autres sommets, ce qui engendre une plus grande complexité.

### 4.3.2 Méthodes basées sur la classification

Dans les méthodes de classification, le problème d'annotation automatique peut être vu comme un type de classification multiclasses avec un grand nombre de classes, aussi large que la taille du vocabulaire. Les travaux les plus représentatifs de ce type de méthodes sont l'indexation linguistique automatique d'images [Barnard 01, Li 03], les méthodes d'annotation basées sur le contenu avec des SVMs (Support Vector Machines) [Cusano 04, Gao 06] ou des BPMs (Bayes Point Machines) [Chang 03], qui estiment les distributions de probabilités des caractéristiques visuelles des images associées à chaque mot-clé.

Les méthodes d'indexation linguistique automatique utilisent souvent des modèles statistiques pour organiser les collections d'images en intégrant l'information sémantique fournie par le texte associé à l'image et l'information visuelle fournie par les caractéristiques de l'image. Les modèles apprennent ainsi les relations entre texte et image et peuvent être utilisés pour associer des mots à de nouvelles images.

La plupart des méthodes à base de SVM commencent par une étape de segmentation des images en région d'intérêts. Chaque région est alors classifiée indépendamment par un SVM multiclasses. L'image hérite alors des mots-clés associés aux classes trouvées.

La méthode de Chang et al. [Chang 03] est un peu différente car elle utilise un ensemble de classificateurs binaires. En effet, la procédure d'annotation commence par l'annotation manuelle d'un ensemble d'apprentissage, chaque image de l'échantillon étant annotée par un unique mot-clé. Un classificateur binaire est alors associé à chaque mot-clé. Chaque classificateur détermine si une image peut être associée ou non au mot-clé correspondant. L'ensemble des classificateurs est appliqué aux images tests. Pour chaque image, les mots-clés associés aux classificateurs renvoyant la réponse « vrai »sont alors retenus pour son annotation. On peut ainsi obtenir plusieurs mots-clés par image. Dans cette approche, les classificateurs de type BPM se sont révélés plus efficaces que ceux de type SVM.

Il existe d'autres méthodes basées sur la classification, qui, à mon sens, se distinguent des approches que nous venons de citer. En effet, elles ont la particularité que chaque classe correspond à plusieurs mots-clés. Par exemple, Wang et al. [Wang 06] proposent, dans un premier temps, de segmenter chaque image en régions. Plusieurs caractéristiques de bas niveau (position, couleur, forme, texture) sont calculés sur chaque région. Un algorithme de classification est alors appliqué sur chaque région, pour regrouper les images ayant des caractéristiques visuelles similaires. Ensuite, pour chaque classe, les caractéristiques les plus pertinentes sont sélectionnées, en se basant sur l'analyse des histogrammes des caractéristiques. Des poids sont alors associés à chaque caractéristique, les poids les plus important correspondant aux caractéristiques les plus pertinentes. Ainsi les caractéristiques auront des poids différents suivant les classes. Par exemple au sein d'une classe « tigre »les caractéristiques de couleur auront un poids élevé. Au contraire, au sein d'une classe « ballon », ce sont les caractéristiques de forme qui auront les poids les plus élevés. Des liens sont alors créés entre certains mots-clés et certaines classes, pour chaque région. Pour annoter automatiquement une nouvelle image, on calcule, pour chaque région, la distance entre les caractéristiques visuelles de cette image et les caractéristiques visuelles de chaque centre de classe. Pour chaque région, on associe le mot-clé du centre le plus proche.

Cette méthode, contrairement aux méthodes classiques basées sur la classification, peut être appliquée sur de plus grand corpus de données. Cependant, elle requiert, comme les méthodes classiques, un lourd travail d'annotation manuelle à cause de la segmentation en régions.

#### 4.3.2.1 Conclusion sur les méthodes basées sur la classification

Ce type de méthode a l'avantage de ne pas nécessiter d'hypothèse sur la distribution des caractéristiques graphes. Cependant, la plupart de ces méthodes présente l'inconvénient de ne pouvoir être utilisée dans le cadre de grandes bases d'images et de grands vocabulaires, puisqu'elles associent un mot-clé à un classificateur.

### 4.3.3 Méthodes probabilistes

Les méthodes probabilistes d'annotation consistent à apprendre des modèles probabilistes d'association entre des images et des mots-clés.

Le premier travail remarquable de ce type, proposé par Mori et al. [Mori 99] en 1999, est un modèle de co-occurrence. Ce modèle consiste à compter les co-occurrences de mots-clés et de caractéristiques graphiques à partir des images d'un échantillon d'apprentissage, et à les utiliser pour prédire les mots-clés annotant d'autres images. Ce modèle présente l'inconvénient de nécessiter des vecteurs de caractéristiques discrètes, ou une discrétisation préalable de ces vecteurs. Ce modèle a alors été amélioré, en 2002, par Duygulu et al. [Duygulu 02] par l'introduction d'un modèle de traduction statistique. Dans cette approche, les images sont d'abord segmentées en régions. Ces régions sont ensuite classifiées en fonction de leurs caractéristiques graphiques. Une

relation entre les classes de régions et les mots-clés est alors apprise, en utilisant une méthode basée sur l'algorithme EM. Ce processus est analogue à l'apprentissage d'un lexique à partir d'un corpus bilingue aligné (deux textes qui sont les traductions l'un de l'autres). Cette méthode supporte des vecteurs de caractéristiques continues mais nécessite une annotation manuelle des régions d'un sous-ensemble d'images.

D'autres travaux cherchent à calculer, pour les images non ou partiellement annotées, la distribution des mots-clés conditionnellement aux caractéristiques visuelles. En effet cette distribution représente une prédiction des mots-clés manquants pour ces images. Il existe plusieurs travaux dans ce sens. Par exemple, Blei et al. [Blei 03] ont proposé trois modèles probabilistes hiérarchiques pour représenter et classifier des données annotées : un modèle de mélange de distributions Gaussiennes et multinomiales (modèle GM-Mixture), le modèle Gaussian-Multinomial LDA (GM-LDA) , et, le plus efficace appelé *correspondence* LDA (CORR-LDA). Ces modèles introduisent une variable aléatoire latente (cachée) pour faire le lien entre les caractéristiques graphiques et les mots-clés. Par exemple, le modèle GM-Mixture (voir figure 4.8) suppose que les caractéristiques visuelles et les mots-clés d'une image ont été générés conditionnellement au même facteur caché (variable latente $z$), qui représente la classe cachée de chaque image. Un vecteur de caractéristiques visuelles est calculé sur les $N$ régions de l'image. Ces vecteurs caractéristiques sont supposés avoir une distribution Gaussienne de paramètres $(\mu, \sigma)$. En plus de ces caractéristiques visuelles, chaque image est annotée par $M$ mots-clés, chacun étant supposé suivre une distribution multinomiale. Comme on peut le voir dans la figure 4.8, une image et sa légende sont supposées avoir été générées en choisissant d'abord la valeur de $z$ puis en répétant l'échantillonnage des caractéristiques $r_n$ des $N$ régions et des $M$ mots-clés $w_m$ conditionnellement à la valeur de $z$. La distribution de probabilité jointe $P(z, r, w)$ est données par :

$$p(z, r, w) = p(z|\lambda) \prod_{n=1}^{N} p(r_n|z, \mu, \sigma) \prod_{m=1}^{M} p(w_m|z, \beta)$$

Une boîte englobante autour d'une variable aléatoire représente une répétition. Par exemple, la boîte autour de a variable $r$ représente $n$ répétitions de $r$, ce qui donne le premier produit dans l'équation ci-dessus.

Ce modèle présente l'inconvénient de nécessiter une segmentation préalable des images, sans pour autant annoter textuellement les régions d'images. En effet, les mots-clés sont associés à l'image entière. Avec ce modèle, on a finalement les inconvénients du découpage en régions des images (*i. e.* le coût de la segmentation préalable), sans les avantages (l'annotation n'est pas plus précise car chaque mot-clé reste associé à une image entière).

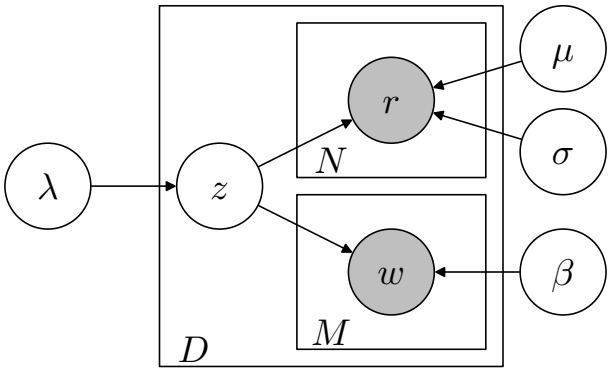



FIGURE 4.8 – Modèle GM-Mixture

Afin de résoudre ce problème, le modèle GM-LDA (présenté figure 4.9), suppose que les caractéristiques visuelles et les mots-clés peuvent provenir de différents facteurs cachés. La variable latente $z$ représente donc le facteur caché de génération des caractéristiques visuelles. De même, la variable latente $v$ représente le facteur caché de génération des mots-clés. Enfin, la variable latente $\theta$ représente finalement une classe cachée de chaque image associée à des mots-clés. Comme dans le modèle GM-Mixture, un vecteur caractéristique est calculé sur les $N$ régions de l'image. Ces vecteurs caractéristiques sont supposés avoir une distribution Gaussienne de paramètres $(\mu, \sigma)$. En plus de ces caractéristiques visuelles, chaque image est associée à $M$ mots-clés, chacun étant supposé suivre une distribution multinomiale. La distribution de probabilité jointe de ce modèle est donnée par :

$$p(r, w, \theta, z, v) = p(\theta|\alpha)(\prod_{n=1}^{N} p(z_n|\theta)p(r_n|z_n, \mu, \sigma))(\prod_{m=1}^{M} p(v_m|\theta)p(w_m|v_m, \beta))$$

L'utilisation de deux facteurs cachés, un pour les caractéristiques visuelles et un autre pour les mots-clés, permet d'établir une correspondance entre une région spécifique de l'image et un mot-clé précis. De ce fait, avec ces modèles, on a l'inconvénient du coût lié au découpage préalable des images, par contre on a l'avantage d'une annotation plus fine.

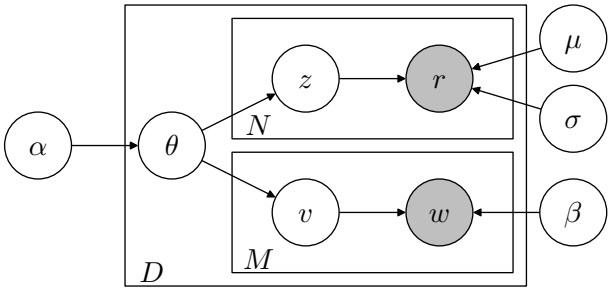



FIGURE 4.9 – Modèle GM-LDA

Enfin, le modèle correspondence-LDA (présenté figure 4.10) permet une représentation d'une image et ses mots-clés encore plus efficace que les deux précédents. En effet, les caractéristiques visuelles sont d'abord générées, et les mots-clés ensuite. De ce fait, l'annotation consiste à sélectionner, pour chaque mot-clé d'une image, une région. De cette façon, le système permet une

annotation plus souple que les deux précédents modèles : un mot-clé peut être associé plusieurs régions, et plusieurs mots-clés peuvent être associés à une même région.

La distribution de probabilité jointe de ce modèle est données par :

$$p(r, w, \theta, z, y) = p(\theta|\alpha)(\prod_{n=1}^{N} p(z_n|\theta)p(r_n|z_n, \mu, \sigma))(\prod_{m=1}^{M} p(y_m|N)p(w_m|y_m, z, \beta))$$

La flèche orientée de $N$ vers $y$ signifie que les caractéristiques $r_n$ des $N$ régions de l'images sont d'abord générées. Ensuite, pour chacun des $M$ mots-clés, une des régions est sélectionnées dans l'image et un mot-clé correspondant $w_m$ est déterminé, conditionnellement au facteur qui a généré la région sélectionnées.

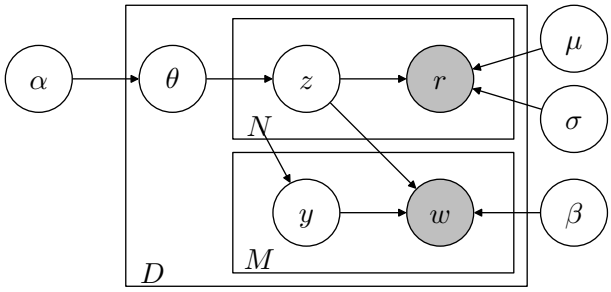



FIGURE 4.10 – Modèle Corr-LDA

Finalement, ces trois modèles possèdent l'avantage de pouvoir effectuer, en plus de l'annotation automatique, les tâches de clustering et de recherche d'images basée sur le texte, grâce aux facteurs cachés. Par contre, ces modèles présentent l'inconvénient de considérer que les mots-clés constituant l'annotation d'une image sont indépendants. De plus le nombre de mots-clés annotant une image est limité.

Jeon et al. [Jeon 03] ont introduit le modèle *Cross-Media Relevance Model* (CMRM) qui utilise les mots-clés communs à des images similaires pour annoter de nouvelles images. En effet, comme dans l'approche de [Duygulu 02], les images sont supposées être décrites par un petit vocabulaire associé aux classes de régions de l'image. En utilisant un échantillon d'apprentissage contenant des images annotées, la distribution de probabilité jointe des classes de régions et des mots-clés est apprises. Cette méthode a ensuite été améliorée par le modèle *Continuous-space Relevance Model* [Lavrenko 03] et le modèle *Multiple Bernoulli Relevance Model* (MBRM) [Feng 04]. L'approche [Lavrenko 03] suppose que chaque image est divisée en régions, chacune étant décrite par un vecteur caractéristique à valeurs continues. Étant donné un échantillon d'apprentissage constitué d'images annotées, un modèle probabiliste des caractéristiques et des mots-clés est appris, permettant de prédire la probabilité de générer un mot-clé étant données les caractéristiques des régions d'images. De même, la méthode [Feng 04] suppose que l'on dispose d'un échantillon d'apprentissage constitué d'images, ou de vidéos, annotées par des mots-clés issus d'un vocabulaire. Chaque image est alors partitionnée en un ensemble de régions rectangulaires et des vecteurs de caractéristiques continues sont calculés sur ces régions. Le modèle proposé est une distribution de probabilité jointe des annotations et des vecteurs caractéristiques, calculée à partir de l'échantillon d'apprentissage. Les probabilités des mots-clés sont estimées en utilisant un modèle de Bernoulli multiple et les probabilités des caractéristiques en utilisant une estimation de densité non-paramétrique. Ce modèle présente l'inconvénient d'émettre une hypothèse sur le type de distribution de probabilité des mots-clés.

Dans [Zhang 05], l'algorithme EM et la règle de Bayes sont utilisés pour connecter chaque caractéristique à des mot-clés. On obtient ainsi des concepts sémantiques. Un concept sémantique est un ensemble de mots-clés. Les caractéristiques visuelles sont supposées avoir été générées à partir de plusieurs distributions Gaussiennes, chacune correspondant à un concept sémantique. Les paramètres de ce mélange de Gaussiennes sont estimés grâce à l'algorithme EM. Une nouvelle image sera annotée par le mot clé du vocabulaire ayant la plus grande probabilité étant données les caractéristiques visuelles de l'image. Cette probabilité est obtenue à l'aide des concepts obtenus et de la règle de Bayes. Cette approche a l'avantage de ne faire de supposition quant à la distribution des mots-clés. Par contre le nombre de concepts est fixé et la recherche d'images nécessite qu'elles soient toutes annotées.

Jin et al. [Jin 04] proposent, quant à eux, un modèle de langage pour l'annotation d'images. Le modèle proposé a l'avantage de considérer les relations de dépendances entre les mots-clés alors que la plupart des modèles considèrent que les mots-clés sont indépendants. Pour introduire la corrélation entre les mots-clés, la probabilité d'un ensemble de mots-clés étant donnée une image est calculée. Cette méthode pose problème par le nombre exorbitant d'ensembles de mots-clés distincts. Pour pallier ce problème, les probabilités sont estimées grâce à un modèle de langage. Un autre avantage de ce modèle est qu'il permet une longueur d'annotation variable. En effet certaines images plus complexes que d'autres nécessitent plus de mots-clés pour les décrire. Toutes les images ne sont donc pas annotées par le même nombre de mots-clés. Par contre cette approche limite l'annotation des images par 5 mots-clés maximum. Enfin, l'annotation des images est soumise à un seuil, *i. e.* que les images ne sont annotées automatiquement par un mot-clé que si celui-ci a la plus grande probabilité et que cette probabilité est supérieure à 0.5. Certaines images ne sont donc pas annotées automatiquement. Celles-ci sont proposées à l'utilisateur pour qu'il les annote manuellement. Cette méthode d'annotation n'est donc pas entièrement automatique.

Le papier [Yavlinsky 05] propose également une méthode probabiliste d'annotation automatique d'images. Celle-ci consiste à calculer la probabilité de chaque mot du vocabulaire étant donnée une image. Chaque image est représentée par un ensemble de caractéristiques et une signature. Cette méthode est assez classique à l'exception qu'elle utilise des tests non paramétriques (méthode statistique) pour estimer les probabilités.

Enfin, dernièrement, l'approche proposée dans [Wang 09b] s'est distinguée car elle utilise à la fois des caractéristiques visuelles globales et locales des images.

#### 4.3.3.1 Conclusion sur les méthodes probabilistes

Les approches probabilistes présentent l'avantage de pouvoir être utilisées, en général, pour les deux tâches de classification et d'annotation.

Par contre, elles nécessitent souvent de grands échantillons d'apprentissage. Enfin, l'inconvénient majeur de la plupart des méthodes probabilistes que nous venons de décrire est que leur efficacité dépend fortement des techniques de segmentation utilisées.

### 4.3.4 Évaluation de l'annotation

Il existe de nombreuses façons d'évaluer les résultats de l'annotation automatique. Ces différentes manières sont très liées aux méthodes d'annotation. En effet, dans le cadre d'un modèle permettant d'effectuer à la fois les tâches d'annotation automatique et de recherche d'images, une façon d'évaluer la qualité de l'annotation peut être d'annoter l'ensemble de la base d'images disponibles et d'effectuer des requêtes à base de mots-clés. Une image sera retrouvée si l'anno-

tation automatique établie contient le mot-clé requête. Ensuite on peut, par exemple, évaluer la précision et le rappel de la méthode. La précision est le rapport entre le nombre d'images retournées contenant le mot-clé requête dans la « bonne »annotation et le nombre d'images retournées. Le rappel correspond au rapport entre le nombre d'images retournées contenant le mot-clé requête dans la « bonne »annotation et le nombre d'images de l'échantillon test étant annotées par le mot-clé requête.

Une autre manière d'évaluer la qualité de l'annotation peut être d'annoter automatiquement la base d'images disponibles, et de calculer le taux moyen d'avoir un mot-clé correct parmi les mots-clés retournés. On peut aussi calculer la longueur (nombre de mots-clés constituant l'annotation) minimale moyenne des annotations retournées qui contiennent tous les mots-clés corrects.

Toutes ces méthodes nécessitent de connaître les annotations « correctes »des images. C'est-à-dire qu'un lourd travail d'annotation manuelle de la base de test est nécessaire, pour pouvoir comparer les annotations estimées aux annotations réelles (voir section 4.2 pour les solutions possibles au problème d'annotation manuelle).

Enfin, la diversité des méthodes d'évaluation rend la comparaison des méthodes existantes très difficile. Il existe donc des campagnes d'évaluation et des compétitions dédiées au problème d'annotation [Smeaton 09, Everingham 09b], qui proposent des bases d'images standards permettant la comparaison des méthodes.

## 4.4 Synthèse et choix d'une méthode d'annotation

La plupart des techniques d'annotation automatique visent à apprendre, grâce à des méthodes d'apprentissage automatique et à partir d'exemples d'images annotées, des relations entre mots-clés et caractéristiques visuelles. Les relations apprises sont ensuite utilisées afin d'attribuer des mots-clés à des images non annotées. Outre leurs avantages et inconvénients liés aux méthodes d'apprentissages utilisées, on distingue, d'autres points forts et problèmes.

Le problème majeur de la plupart des méthodes que nous avons présentées réside dans le fait qu'un mot-clé est choisi pour annoter une image sans prendre en compte les autres mots-clés éventuels de cette image. De plus, la majorité des approches décrites effectue une annotation locale des images, *i. e.* qu'un ou plusieurs mots-clés sont attribués à une région d'image et non pas à l'image entière. L'annotation est alors plus fine, mais l'efficacité de ces approches dépend aussi des techniques de segmentation utilisées. Enfin, l'inconvénient commun à toutes ces méthodes est qu'elles nécessitent un échantillon d'apprentissage composé d'images annotées. Les méthodes d'annotation automatique semblent donc particulièrement utiles pour compléter les annotations existantes dans des bases d'images déjà partiellement annotées manuellement.

Malgré ces inconvénients, certains modèles probabilistes nous semblent sortir du lot, car, en plus d'annoter automatiquement des images, ils permettent de classer ces images, et/ou de rechercher des images à partir de requêtes très souples pour l'utilisateur, car elles peuvent être composées à la fois d'images exemples et de mots-clés.

Compte tenu de ces avantages et inconvénients, l'annotation automatique d'images semble être un outil intéressant afin d'atteindre nos objectifs. En effet, dans les chapitres 2 et 3, nous avons élaboré notre stratégie, consistant à combiner plusieurs sources d'informations (visuelles et/ou sémantiques) afin d'améliorer la reconnaissance et la classification d'images. Or, nous avons vu dans le chapitre 2, que l'indexation textuelle manuelle conduit à de meilleurs résultats que l'indexation automatique, mais elle est coûteuse pour l'utilisateur. Suite à l'étude des techniques d'annotations, un bon compromis permettant de réaliser ces objectifs apparaît : à partir de bases

d'images partiellement annotées manuellement, on souhaite, grâce l'annotation automatique, compléter les annotations de ces bases, pour pouvoir rendre plus efficaces nos méthodes de recherche et/ou classification visuo-textuelles. En effet, même si les méthodes que nous allons proposer permettent de traiter les données manquantes et donc de fonctionner sur des bases partiellement annotées, la réduction du nombre de données manquantes, grâce à l'annotation automatique, serait un avantage non négligeable.

De plus, la comparaison des techniques d'annotation automatique a soulevé l'existence de modèles graphiques probabilistes évolués permettant d'effectuer les deux tâches de classification et d'annotation [Blei 03]. Or, dans le chapitre 3, nous avons déjà annoncé notre choix pour les modèles graphiques probabilistes afin de réaliser les tâches de recherche et de classification visuo-textuelles. Nous souhaitons donc utiliser les mêmes modèles graphiques probabilistes à des fins d'annotation et de recherche et/ou classification d'images. Par contre, à la différence des modèles présentés dans [Blei 03], nous souhaitons que notre approche prenne en compte les éventuels mots-clés existants d'une image afin d'en prédire de nouveaux, car nous souhaitons non seulement annoter automatiquement des images sans annotations, mais aussi compléter les annotations d'images partiellement annotées. Enfin, afin de proposer des méthodes dont l'efficacité ne dépend pas de la segmentation préalable des images, les méthodes que nous proposerons procéderont à une annotation globale des images.

L'étude des méthodes existantes de recherche, classification et recherche d'images, présentée dans les chapitre 2, 3 et 4, nous a donc conduits à choisir les modèles graphiques probabilistes. Cependant, suite à cette étude, on ne voit pas clairement les avantages des modèles graphiques probabilistes par rapport aux modèles probabilistes standards.

Par conséquent, dans la partie II, avant de présenter nos contributions, nous proposons un tutoriel (chapitre 5) dédié à la définition et l'étude de ces modèles, en particulier dans un but de classification.

## Deuxième partie

# Contributions en reconnaissance de formes

# Chapitre 5

# Préambule : tutoriel sur les modèles graphiques probabilistes

**Sommaire**



## 5.1 Introduction

La partie I nous a amenés à choisir les modèles probabilistes pour traiter notre problème de reconnaissance de formes, incluant la recherche, la classification et l'annotation d'images. Parmi ces modèles, nous orientons notre choix vers les modèles graphiques. En effet, ceux -ci bénéficient non seulement des avantages des modèles probabilistes, mais, de plus, ils présentent des avantages supplémentaires liés à leur représentation graphique. En effet, les modèles graphiques probabilistes permettent de visualiser la structure et les propriétés (incluant les relations de dépendance conditionnelle) du modèle probabiliste correspondant. Enfin, des calculs complexes, parfois requis par les processus d'apprentissage de paramètres et d'inférence dans les modèles probabilistes complexes, peuvent être simplifiés à l'aide de manipulations graphiques.

Grâce à ce chapitre, on souhaite apporter au lecteur les acquis nécessaires à la compréhension des modèles graphiques utilisés et proposés dans les chapitres 6, 7 et 8. Nous proposons donc un petit tutoriel sur les modèles graphiques probabilistes, fournissant les définitions nécessaires, ainsi qu'un aperçu des méthodes couramment utilisées dans les processus d'apprentissage de

paramètres et d'inférence. Les modèles graphiques probabilistes utilisés classiquement comme classificateurs, sont également présentés. Concernant les méthodes d'inférence et d'apprentissage, nous insisterons uniquement sur les méthodes que nous avons été amenés à utiliser. Pour les autres, nous nous contenterons de les aborder brièvement. En effet, ce chapitre n'est pas dédié à fournir un état de l'art complet des modèles graphiques et des processus d'apprentissage et d'inférence associés, car nous n'apportons pas de contribution à ce niveau. Par contre, ce chapitre doit rappeler ou donner les acquis nécessaires à l'adaptation des modèles graphiques probabilistes au problème de la reconnaissance de formes.

Si le lecteur veut plus de détails sur les modèles graphiques et sur l'inférence dans ces modèles, on lui conseillera le chapitre 8 du livre [Bishop 06] et les livres [Lucas 07, Jordan 99].

Ce chapitre est organisé de la façon suivante. D'abord, une définition des modèles graphiques probabilistes est donnée dans la section 5.2. Les deux formes de modèles graphiques probabilistes les plus répandues sont aussi présentées dans cette section. Dans la section 5.3, nous donnons une définition formelle des réseaux Bayésiens, ainsi que leur représentation graphique. Les techniques d'apprentissage des paramètres des modèles sont abordées dans la section 5.4. La section 5.5 est dédiée à la notion d'inférence probabiliste et aux méthodes pour la réaliser. Enfin, les modèles graphiques probabilistes couramment utilisés comme classificateurs sont présentés dans la section 5.6.

## 5.2 Définition

Un modèle graphique est une famille de distributions de probabilités définie en terme de graphe orienté ou non. Les nœuds du graphe représentent des variables aléatoires, et les distributions de probabilités jointes sont définies en faisant le produit de fonctions définies sur les sous-ensembles de nœuds connectés. Issus de la théorie des graphes, des algorithmes généraux permettent de calculer les probabilités marginales et conditionnelles recherchées. De plus, le formalisme fournit un contrôle sur la complexité de calcul associée à ces opérations.

On distingue réellement deux formes de modèles graphiques probabilistes : les modèles orientés et les modèles graphiques non orientés, basés respectivement sur les graphes acycliques orientés et les graphes non orientés. Cependant, certains modèles mélangent les arcs orientés et non orientés [Jordan 99]. Les modèles orientés sont plus connus sous le nom de réseaux Bayésiens. Les modèles de Markov cachés (HMM) [Rabiner 90] sont un exemple de réseaux Bayésiens couramment utilisés en reconnaissance de la parole. A l'opposé, les champs de Markov constituent un exemple de modèle non orienté [Rue 05].

Dans ce chapitre, on s'intéressera uniquement aux modèles orientés, les réseaux Bayésiens, car ils sont plus adaptés à la prise de décision (et donc à la classification). En effet, ils représentent des relations déterministes, des liens de cause à effet entre les variables. Dans les modèles non orientés, le fait que deux variables interagissent entre elles est représenté, mais on ne sait pas de quelle façon ni quelle variable est à l'origine de l'interaction. Pour des détails sur les modèles non orientés, on conseillera les lectures suivantes [Bishop 06, Jordan 03].

## 5.3 Les réseaux Bayésiens

Soit $G(\mathscr{V}, \varepsilon)$, un graphe acyclique orienté, où $\mathscr{V}$ est l'ensemble de nœuds, et $\varepsilon$ celui des arcs. Ce graphe peut être statique, ou dynamique. Dans ce deuxième cas on parlera de réseau Bayésien dynamique [Murphy 02]. Ces graphes ont la particularité d'évoluer au cours du temps : des arcs peuvent être ajoutés entre chaque pas de temps. Le coût de l'inférence probabiliste sera donc

plus important dans ce genre de réseau.

Soit $X = \{X_v : v \in \mathscr{V}\}$ un ensemble de variables aléatoires. Chaque variable $X_v$ correspond à un nœud $v$ du graphe. Pour chaque nœud $v \in \mathscr{V}$, on définit $\pi_v$, l'ensemble de ses parents dans le graphe. $|x_{\pi_v}$ désigne l'ensemble des valeurs observées pour les parents de $v$.

Soit $\theta$, l'ensemble de probabilités conditionnelles $\{\theta_v\} = \{p(x_v|x_{\pi_v})\}, v \in \mathscr{V}$.

**Remarque** : La notation $p(x_v)$ signifie que l'on calcule $p(X_v = x_v)$ où $x_v$ est une valeur possible pour la variable $X_v$. Cette notation est un abus de langage mais allège les expressions et améliore la lisibilité. Dans la suite de ce manuscrit, lorsque nous lirons $p(a)$, il faudra comprendre que l'on calcule $p(A = a)$, où $a$ est une valeur possible pour la variable $A$. Les caractères minuscules représentent les valeurs et les majuscules les variables.

Le couple $(G, \theta)$ définit un réseau Bayésien et la distribution de probabilité jointe associée à ce réseau, sur l'ensemble $\mathscr{V}$ des variables du modèle, est définie comme suit :

$$p(x) = \prod_{v \in \mathscr{V}} p(x_v|x_{\pi_v}) \tag{5.1}$$

Cette probabilité jointe est en fait une expression simplifiée. La simplification a pu être obtenue grâce au raisonnement suivant :

posons $X = \{X_1, X_2, \ \dots \ , X_n\}$. La probabilité jointe de ces variables est notée $p(x) = p(x_1, x_2, \ \dots \ , x_n)$.

Grâce à la règle de Bayes [Bayes 63], qui stipule que $p(x_1, x_2) = p(x_2|x_1) \times p(x_1)$, la probabilité jointe peut être décomposée de la façon suivante :

$$p(x) = p(x_1, x_2, \ \dots \ , x_n) = p(x_n|x_{n-1}, \ \dots \ , x_2, x_1) \times \dots \times p(x_2|x_1)p(x_1) = p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}, \ \dots \ , x_1) \tag{5.2}$$

Supposons maintenant que les probabilités conditionnelles de certaines variables $X_i$ ne dépendent que d'un sous-ensemble des prédécesseurs de $X_i$, les prédécesseurs de $X_i$ étant $X_1, X_2 \dots X_{i-1}$. Notons $X_{\pi_i}$ l'ensemble de ces prédécesseurs. On peut alors écrire $p(x_i|x_{i-1}, \ \dots \ , x_1) = p(x_i|x_{\pi_i})$. Ceci nous permet de simplifier la décomposition obtenue dans l'équation 5.2 de la façon suivante :

$$p(x) = p(x_1, x_2, \ \dots \ , x_n) = \prod_{i=1}^{n} p(x_i|x_{\pi_i})$$

Cette simplification correspond bien à l'équation 5.1.

Derrière tout réseau Bayésien se cache donc une hypothèse essentielle : chaque variable est indépendante de ses non descendants étant donnés ses parents dans le graphe. Les propriétés de dépendance et d'indépendance conditionnelles d'un tel modèle sont visualisables dans son graphe. Afin d'observer graphiquement les notions de dépendance conditionnelle, considérons le réseau Bayésien représenté figure 5.1. Ce réseau modélise la probabilité (jointe) qu'une randonnée soit annulée et que les fleurs poussent, suite à une cause possible : la pluie. Cet exemple modélise donc les relations qui relient les variables « pluie », « randonnée »et « fleurs ». La pluie provoque

l'annulation d'une randonnée et la pousse des fleurs, *i. e.* que le maintien de la randonnée et la pousse des fleurs dépendent conditionnellement de la pluie.

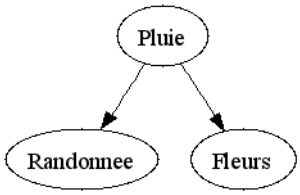



FIGURE 5.1 – Exemple d'un réseau Bayésien à 3 variables

Les réseaux Bayésiens peuvent traiter deux types de variables : discrètes ou continues. Dans le cas de variables discrètes, la somme des probabilités $p(x_v|x_{\pi_v})$ vaut 1. Dans le cas de variables continues, c'est l'intégrale qui vaut 1.

Chaque probabilité conditionnelle $\theta_i$ est considérée comme un paramètre du modèle. Il n'y a donc pas de distinction entre données et paramètres : les paramètres d'un modèle sont ses probabilités. Il est possible de représenter les paramètres au niveau des nœuds du graphe représentant le modèle. La représentation de ces paramètres permettent d'enrichir la structure graphique du modèle en quantifiant les relations entre les variables.

La figure 5.2 donne un exemple de réseau Bayésien où les paramètres sont représentés. Ce modèle, proposé dans l'article [Blei 03], a déjà été présenté dans le chapitre 4. Les lettres $\alpha$, $\theta$, $z$, $v$, $r$, $w$, $\sigma$, $\mu$ et $\beta$, associées aux variables, sont les paramètres du modèles. Des boîtes englobantes peuvent aussi être utilisées pour représenter des répétitions de sous-parties d'un modèle. Par exemple, sur la figure 5.2, il y a trois boîtes englobantes. La boîte où la lettre $N$ figure en bas à gauche de la boîte signifie que les variables englobées (de paramètres $z$ et $r$) sont répétées $N$ fois. Il est de même pour les lettre $M$ et $D$ qui caractérisent le nombre de répétitions des variables que leurs boîtes englobent.

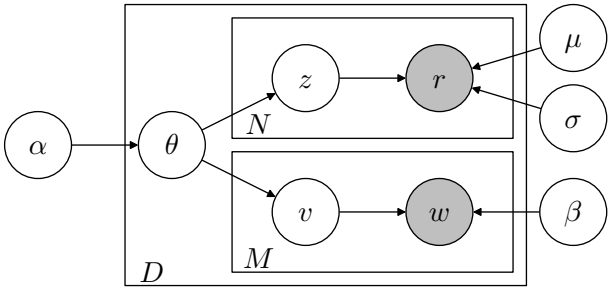



FIGURE 5.2 – Modèle GM-LDA

Finalement, les graphes fournissent une représentation visuelle compacte et attractive d'une distribution de probabilité, mais ils apportent beaucoup plus :

– d'abord, quelle que soit la forme des probabilités conditionnelles $p(x_v|x_{\pi_v})$, la probabilité jointe de l'équation 5.1 implique un ensemble d'hypothèses d'indépendance conditionnelle entre les variables $X_v$ : chaque variable est indépendante de ses non descendants étant

donnés ses parents. L'ensemble de ces hypothèses d'indépendance conditionnelle peut être obtenu automatiquement, en parcourant le graphe et en utilisant un critère appelé la « d-séparation »[Pearl 88].
– De plus, comme on va le voir dans la section 5.5, la structure graphique peut être exploitée par des algorithmes pour l'inférence probabiliste.

## 5.4 Apprentissage de paramètres d'un réseau

Une fois que la description d'un modèle est établie, à savoir sa structure graphique et les lois de probabilités des variables, on cherche à estimer les valeurs numériques de chaque paramètre.

Supposons que l'on dispose de variables continues ou discrètes (ou un mélange des deux), et d'un ensemble de données représentatif de plusieurs cas possibles pour chaque variable. L'ensemble des données peut être complet ou incomplet. Suivant le cas, une solution différente va être utilisée.

Dans le cas où l'ensemble des données ne présente pas de données manquantes, la méthode la plus simple et la plus utilisée est l'estimation statistique de la probabilité d'un évènement par la fréquence d'apparition de l'évènement dans la base de données. Cette méthode est appelée « maximum de vraisemblance ». Soit $\mathcal{D}$ un ensemble de données, alors $P(d|M)$ est la probabilité qu'une donnée $d \in \mathcal{D}$ soit générée par le modèle $M$, et est appelée la vraisemblance de $M$ étant donné $d$. Par conséquent, la vraisemblance de $M$, étant donné l'ensemble complet $\mathcal{D}$, est :

$$L(M|\mathcal{D}) = P(\mathcal{D}|M) = \prod_{d \in \mathbb{D}} P(d|M)$$

Pour des raisons de simplicité de calcul, le logarithme est souvent utilisé à la place de la vraisemblance :

$$L(M|\mathcal{D}) = \sum_{d \in \mathcal{D}} logP(d|M)$$

Par conséquent, le principe du maximum de vraisemblance préfère choisir les paramètres avec le plus grande vraisemblance :

$$\hat{\theta} = argmax_{\theta} L(M_{\theta}|\mathcal{D})$$

En général, le maximum de vraisemblance est obtenu en comptant la fréquence de l'évènement dans la base.

Dans le cas où l'on ne dispose pas d'assez de données pour représenter tous les cas possibles (présence de données manquantes), un des algorithmes les plus populaires est l'algorithme Espérance-Maximisation (EM) (« Expectation Maximization »en anglais). Cet algorithme commence par initialiser aléatoirement les paramètres (distributions de probabilités) du modèle. Ensuite, il consiste à calculer, de manière itérative, le maximum de vraisemblance, quand les observations peuvent être vues comme données incomplètes. Chaque pas d'itération de l'algorithme consiste en une étape de calcul d'espérance, suivie par une étape de maximisation. D'où son nom d'algorithme EM (Espérance Maximisation). Le principe général de cet algorithme est expliqué en détail dans [Dempster 77].

Dans un réseau Bayésien, la première étape de l'algorithme EM peut être faite facilement en utilisant un algorithme d'estimation du maximum *a posteriori* (ou « map »en anglais, de

« Maximum *a posteriori* estimation »). Il s'agit de calculer les valeurs les plus probables pour les données manquantes, étant données les variables observées. La seconde étape de l'algorithme EM (celle de maximisation) est alors exécutée. Cette seconde étape peut être effectuée avec un algorithme d'optimisation, si aucune forme du maximum de vraisemblance n'est connue, ou avec l'approche précédente (« map »). Les deux étapes (E et M), sont répétées jusqu'à convergence.

Dans les cas où des variables continues sont supposées suivre la loi Normale, l'algorithme EM est aussi utilisé pour apprendre les paramètres des distributions Gaussiennes associées à cette loi. En effet, ces paramètres peuvent être considérés comme des données manquantes.

## 5.5 Inférence probabiliste

Le problème général de l'inférence probabiliste est de calculer la probabilité de n'importe quelle variable d'un modèle probabiliste, à partir de l'observation d'une ou plusieurs autres variables.

Dans cette section, on s'intéresse aux algorithmes exécutant de tels calculs. Il existe nombre d'algorithmes d'inférence et nous ne pouvons en faire, ici une étude exhaustive. Nous nous concentrons sur les algorithmes les plus utilisés, et que nous avons utilisés dans cette thèse. Les autres seront abordés plus brièvement. Pour une présentation plus complète, on conseillera les lectures suivante [Jojic 05, Jordan 02].

Dans la section 5.5.1, nous expliquons le problème de l'inférence de façon générale, et montrons qu'il peut être effectué en termes de manipulations graphiques. Nous verrons que la structure du graphe joue un rôle important dans la complexité de ces calculs et dans le choix de la méthode d'inférence. En effet, les algorithmes d'inférence peuvent être classés en deux catégories : l'inférence exacte (section 5.5.2), et l'inférence approximative (section 5.5.3), et le choix d'une catégorie ou d'une autre sera fonction de la structure du graphe, des distributions de probabilité utilisées et du nombre de données d'apprentissage.

### 5.5.1 Approche générale de l'inférence

Dans cette section, nous allons montrer pourquoi l'inférence peut être réalisée à partir du graphe associé à un réseau Bayésien et pourquoi des algorithmes spécifiques sont nécessaires.

Soit $X = \{X_1, X_2, \ \dots \ , X_n\}$ un ensemble de variables. On suppose qu'une de ces variables $X_j \in X$ est observée, et que sa valeur observée est $x_j$. L'observation $X_j = x_j$ est appelée *évidence*. Dans ce contexte, l'inférence correspond au calcul de $p(X_i = x_i | Xj = x_j), \forall X_i \in X \ et \ i \neq j$.

Le calcul de $p(X_i = x_i | X_j = x_j)$, peut, dans le cas discret et grâce à la règle de Bayes, être décomposé de la façon suivante :

$$p(X_i = x_i | X_j = x_j) = \frac{p(X_i = x_i, X_j = x_j)}{p(X_j = x_j)} = \frac{\sum_{\{X_k\}k \ \in \ \{1, \ \dots \ ,n\}\backslash\{i,j\}} p(X_k = x_k, X_i = x_i, X_j = x_j)}{\sum_{\{X_k\}k \ \in \ \{1, \ \dots \ ,n\}\backslash\{j\}} p(X_k = x_k, X_j = x_j)} \quad (5.3)$$

Le calcul de $p(X_i = x_i | X_j = x_j)$ peut ainsi être exprimé en un quotient de lois jointes. Or, un réseau Bayésien encode une probabilité jointe sur un ensemble de variables aléatoires. De plus, le même réseau permet de décomposer la probabilité jointe en facteurs de probabilités conditionnelles (voir équation 5.1) Le graphe d'un réseau Bayésien va donc pouvoir être utilisé pour simplifier le calcul de $p(X_i = x_i | X_j = x_j)$.

Le problème à résoudre est finalement un problème de marginalisation : on souhaite marginaliser les probabilités jointes obtenues dans l'équation 5.3. Afin d'expliquer la notion de marginalisation, prenons cet exemple simple : soient $X$ et $Y$ deux variables aléatoires. La marginalisation

consiste, à partir de la probabilité jointe $p(X = x, Y = y)$, à retrouver les lois marginales $p(x)$ et $p(y)$, à l'aide de la règle de Bayes :

$$p(Y = y) = \sum_{X} p(X = x, Y = y)$$

On dit que l'on a « marginalisé »sur la variable $X$, *i. e.* que pour chaque valeur $x$ de $X$, on calcule $P(X = x, Y = y$ et on fait la somme de ces probabilités.

Revenons à notre exemple de réseau Bayésien à $n$ variables $X_1, X_2, \ldots, X_n$. La marginalisation de $p(x_1, x_2, \ldots, , x_n)$ sur une variable $X_k \in X$ ne pose pas de problème de calcul mais un problème de temps. En effet, si chacune de ces variables est binaire, le calcul $\sum_{x_k} p(x_1, \ldots, x_n)$ prendra un temps de $O(2^n)$.

Certains algorithmes d'inférence permettent de faire exactement ces calculs. On les appelle des algorithmes d'inférence exacte (voir section 5.5.2). Mais, dans le cas de réseaux avec un grande nombre de variables et/ou beaucoup d'arcs entre ces variables, leur utilisation devient déconseillée car ils sont trop coûteux en temps, comme on vient de l'évoquer. Dans ce cas, on leur préférera des algorithmes d'inférence approximative (voir section 5.5.3).

### 5.5.2 Algorithmes d'inférence exacte

Les algorithmes d'inférence exacte les plus réputés sont l'algorithme de passage de messages de Pearl [Pearl 88], l'algorithme d'élimination de Bucket [Dechter 98] et l'algorithme d'arbre de jonction [Jensen 90].

Par exemple, l'algorithme d'élimination de Bucket [Dechter 98] consiste à marginaliser la distribution de probabilité jointe d'un réseau, en procédant variable par variables. Chaque marginalisation sur une variable $X_i$ donne lieu à une somme des probabilités de cette variable. Parfois, cette somme vaudra 1, ce qui conduira à l'élimination de la variable $X_i$. On procédera alors à la marginalisation sur une des variables restantes et ainsi de suite jusqu'à ce que la distribution soit marginalisée.

Le problème de cet algorithme est que l'ordre dans lequel les variables sont éliminées détermine la quantité de calcul nécessaire pour marginaliser la distribution de probabilités jointe et donc la complexité de l'algorithme.

L'algorithme de passage de messages [Pearl 88] est le plus courant. Nous détaillons cet algorithme dans la section ci-dessous (section 5.5.2.1), car c'est un de ceux que nous avons utilisés dans nos approches (voir partie II).

#### 5.5.2.1 L'algorithme de passage de messages

Dans cette technique, à chaque nœud est associé un processeur qui peut envoyer des messages de façon asynchrone à ses voisins, jusqu'à ce qu'un équilibre soit atteint, en un nombre fini d'étapes. Cet algorithme ne s'applique qu'aux arbres. Donnons ici une définition d'un arbre :

soit un graphe $G$ possédant $n$ nœuds. $G$ est un arbre si et seulement si $G$ est sans cycle et qu'il possède $n - 1$ arêtes. Un graphe sans cycle est un graphe dans lequel il n'est pas possible de revenir à un point de départ sans faire le chemin en sens inverse. Ceci engendre le fait que chaque nœud d'un arbre n'a qu'un seul parent (sauf la racine qui n'a aucun parent).

Cette méthode a été étendue aux réseaux quelconques pour donner l'algorithme de l'arbre de jonction qui fera l'objet de la section 5.5.2.3.

L'algorithme de passage de messages et les différents types de messages sont expliqués ci-dessous :

- soit $G(\mathscr{V}, \varepsilon)$, un graphe acyclique orienté (un arbre), où $\mathscr{V}$ est l'ensemble de nœuds, et $\varepsilon$ celui des arcs,
- soit $X = \{X_v : v \in \mathscr{V}\}$ un ensemble de variables aléatoires. Chaque variable $X_v$ correspond à un nœud $v$ du graphe. Pour chaque nœud $v \in \mathscr{V}$, on définit $\pi_v$, l'ensemble de ses parents dans le graphe,
- soit $\theta$, l'ensemble de probabilités conditionnelles $\{\theta_v\} = \{p(x_v|x_{\pi_v})\}$, $v \in \mathscr{V}$, alors le couple $(G, \theta)$ définit un réseau Bayésien.
- Soit $E \in X$ le sous-ensemble des variables observées de $E$,
- soit $X_i \in X$ une variable quelconque, associée au nœud $i$ de $G(\mathscr{V}, \varepsilon)$,
- Soit $N_i$ l'ensemble des nœuds parents observés de $i$, et $D_i$ l'ensemble des nœuds enfants observés. La figure 5.3 montre un nœud $i$, l'ensemble de ses nœuds parents $N_i$ et l'ensemble de ses nœuds enfants $D_i$.
- soit $N_{X_i}$ l'ensemble des variables associées aux nœuds de $N_i$ et $D_{X_i}$ l'ensemble des variables associées aux nœuds de $D_i$.

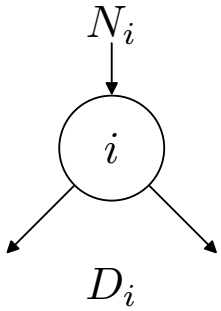



FIGURE 5.3 – Sous-ensemble d'un arbre : un nœud, son parents et ses enfants

On va alors distinguer deux types de messages, $\lambda$ et $\pi$ :
$\lambda(X_i) \propto p(D_{X_i}|X_i)$ et $\pi(X_i) \propto p(X_i, N_{X_i})$.
De plus, soit $E_i \in E$, une variable observée. Alors :

$$P(X_i|E_i = e_i) \propto \lambda(X_i)\pi(X_i)$$

Expliquons maintenant comment calculer chaque type de message :

**Calcul des messages $\lambda$ en chaque nœud $i$ :**
Pour chaque nœud $j$ enfant du nœud $i$, *i. e.* pour chaque variable $X_j \in D_{X_i}$, où $i, j \in \mathscr{V}$ et $i \neq j$, on a :

$$\lambda_{X_j}(X_i = x_i) = \sum_{x_j} p(X_j = x_j|X_i = x_i)\lambda(X_j = x_j)$$

$\lambda_{X_j}(X_i = x_i)$ signifie que l'on fait une sommation sur toutes les valeurs possibles $x_j$ de la variable $X_j$.
Posons $X = \{X_1, X_2, \ldots, X_n\}$. Les messages $\lambda$ se calculent de la façon suivante :
- Si la variable $X_i$ est observée, alors $\lambda(X_i)$ est un vecteur de taille égale au domaine de $X_i$, *i. e.* le nombre de valeurs possible que la variable $X_i$ peut prendre. Ce vecteur vaut 0 partout sauf à la place de la valeur observée où il vaut 1.
- Si le nœud $i$ est une feuille de l'arbre, alors le vecteur $\lambda(X_i)$ vaut 1 partout.

– Sinon, $\lambda(X_i = x_i) = \prod_{X_j \in D_{X_i}} \lambda_{X_j}(X_i = x_i)$

**Calcul des messages $\pi$ en chaque nœud $i$ :**
Soit $j$ l'unique nœud parent de $i$, *i. e.* soit $X_j \in N_{X_i}$, où $i, j \in \mathscr{V}$ et $i \neq j$. On a

$$\pi_{X_i}(X_j = x_j) = \pi(X_j = x_j) \prod_{X_k \in D_{X_i} \ X_i} \lambda_{X_k}(X_j = x_j)$$

Posons $X = \{X_1, X_2, \ \ldots \ , X_n\}$. Les messages $\pi$ se calculent de la façon suivante :
– Si la variable $X_i$ est observée, alors $\lambda(X_i) = \pi(X_i)$ est un vecteur de taille égale au domaine de $X_i$, *i. e.* le nombre de valeurs possible que la variable $X_i$ peut prendre. Ce vecteur vaut 0 partout sauf à la place de la valeur observée où il vaut 1.
– Si le nœud $i$ est la racine de l'arbre (*i. e.* si $i$ n'a pas de parent), alors $\pi(X_i) = p(X_i)$
– Sinon, $\pi(X_i = x_i) = \sum_{x_j} p(X_i = x_i | X_j = x_j) \pi_{X_i}(X_j = x_j)$

#### 5.5.2.2 Exemple de propagation d'un message par l'algorithme de Pearl

Dans cette section, nous allons dérouler l'algorithme de passage de messages de Pearl, que l'on vient d'expliquer, sur un exemple simple, pour bien en comprendre le principe :

**Énoncé de l'exemple :**
Reprenons le réseau Bayésien représenté de la figure 5.1 :

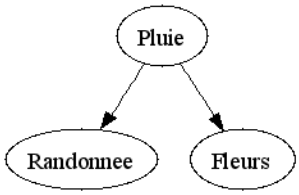



FIGURE 5.4 – Exemple d'un réseau Bayésien à 3 variables

La description des nœuds est données ci-dessous :
– Le nœud *Pluie* représente la variable aléatoire discrète $M$. Cette variable est qualitative et prend ses valeurs dans le domaine $\{nulle = m_0, moyenne = m_1, forte = m_2\}$
– La variable *Fleurs* représente la variable aléatoire discrète $F$. Cette variable est qualitative et prend ses valeurs dans le domaine $\{bonnepousse = Bo, mauvaisepousse = Ma\}$
– Le nœud *Randonnee* représente la variable aléatoire discrète $R$. Cette variable est qualitative et prend ses valeurs dans le domaine $\{oui = O, non = N\}$

Les tables de probabilités conditionnelles des variables $M$, $F$ et $R$ sont données dans les tableaux 5.1, 5.2 et 5.3 respectivement.

**Arrivée d'une nouvelle observation :**
Supposons que l'on a une nouvelle observation de la valeur $m_2$ sur la variable $M$. Par contre, on n'observe pas du tout les autres valeurs $m0$ et $m_1$.

Cette observation s'appelle *évidence* et on écrit $e_M = \{0, 0, 1\}$.

On souhaiterait savoir comment les autres variables vont réagir étant donnée cette observation.

| $M$ | $P(M)$ |
|---|---|
| $m_0$ | 0.30 |
| $m_1$ | 0.60 |
| $m_2$ | 0.10 |

TABLE 5.1 – Table de probabilités de la variable $M$

| $P(R\|M)$ | $O$ | $N$ |
|---|---|---|
| $m_0$ | 0.85 | 0.15 |
| $m_1$ | 0.50 | 0.50 |
| $m_2$ | 0.05 | 0.95 |

TABLE 5.2 – Table des probabilités conditionnelles de la variable $R$ étant donnée la variable $M$

Pour ce faire, on va propager le message que fournit l'évidence.

Commençons par calculer les messages au niveau de la variable observée, car le calcul des messages $\lambda$ et $\pi$, au niveau des variables observées, est un cas trivial :

**Calcul des messages $\lambda$ et $\pi$ au nœud observé $M$ :**

- **Calcul des messages $\lambda$ et $\pi$ :**

  $M$ est observé. On a donc $\lambda(M) = \pi(M) = \begin{bmatrix}0\\0\\1\end{bmatrix}$

- **Le message à transmettre à l'enfant $R$ est calculé comme suit :**

  $$\pi_R(M) = \pi(M) \cdot \lambda_F(M) = \begin{bmatrix}0\\0\\1\end{bmatrix} \cdot \lambda_F(M = m_2)$$

  Or,
  $\lambda_F(M = m_2) = \sum_f p(F = f|M = m_2)\lambda(F = f) =$
  $p(F = Bo|M = m_2)\lambda(F = Bo) + p(F = Ma|M = m_2)\lambda(F = Ma) =$
  $0.90 * 1 + 0.10 * 1 = 1$

  Donc $\pi_R(M) = \begin{bmatrix}0\\0\\1\end{bmatrix} \cdot 1 = \begin{bmatrix}0\\0\\1\end{bmatrix}$

- **De même, le message à transmettre à l'enfant $F$ est calculé comme suit :**

  $$\pi_F(M) = \pi(M) \cdot \lambda_R(M) = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

On peut ensuite calculer les messages au niveau des deux autres nœuds $R$ et $F$. Les deux autre nœuds sont des feuilles de l'arbre. Le calcul des messages $\lambda$ sera donc trivial au niveau de ces nœuds. On peut traiter $R$ et $F$ dans n'importe quel ordre car aucun de ces nœuds n'est racine de l'arbre, et aucun de ces nœuds n'est observé.

| $P(F\|M)$ | $Bo$ | $Ma$ |
|---|---|---|
| $m_0$ | 0.20 | 0.80 |
| $m_1$ | 0.75 | 0.25 |
| $m_2$ | 0.90 | 0.10 |

TABLE 5.3 – Table des probabilités conditionnelles de la variable $F$ étant donnée la variable $M$

**Calcul des messages $\lambda$ et $\pi$ au nœud $R$ :**

- **Calcul du message $\lambda$ :**

  $R$ est une feuille donc on a $\lambda(R) = \begin{bmatrix}1\\1\end{bmatrix}$
- **Calcul du message $\pi$ :**

  $R$ n'est ni une variable observée, ni la racine de l'arbre. On a donc :
  $\pi(R) = p(R|M).\pi_R(M) = \begin{bmatrix}0.85 & 0.5 & 0.05\\0.15 & 0.5 & 0.95\end{bmatrix} \cdot \begin{bmatrix}0\\0\\1\end{bmatrix}$, soit $\pi(R) = \begin{bmatrix}0.05\\0.95\end{bmatrix}$
- Enfin, on a $P(R|M = e_M) \propto \lambda(R) \cdot \pi(R) = \begin{bmatrix}0.05\\0.95\end{bmatrix}$

  On peut en conclure que l'observation d'une pluie forte peut conduire (avec un risque de 5%), à l'annulation d'une randonnée.

**Calcul des messages $\lambda$ et $\pi$ au nœud $F$ :**

- **Calcul du message $\lambda$ :**

  $F$ est une feuille donc on a $\lambda(F) = \begin{bmatrix}1\\1\end{bmatrix}$
- **Calcul du message $\pi$ :**

  $F$ n'est ni une variable observée, ni la racine de l'arbre. On a donc :
  $\pi(F) = p(F|M) \cdot \pi_F(M) = \begin{bmatrix}0.20 & 0.75 & 0.90\\0.80 & 0.25 & 0.10\end{bmatrix} \cdot \begin{bmatrix}0\\0\\1\end{bmatrix}$, soit $\pi(R) = \begin{bmatrix}0.90\\0.10\end{bmatrix}$
- Enfin, on a $P(F|M = e_M) \propto \lambda(F) \cdot \pi(F) = \begin{bmatrix}0.90\\0.10\end{bmatrix}$

  On peut en conclure que l'observation d'une pluie forte peut conduire (avec un risque de 10%), à la pousse des fleurs.

#### 5.5.2.3 Algorithme d'arbre de jonction

Le problème de l'algorithme de passage de messages de Pearl est qu'il ne s'applique qu'aux arbres. Une généralisation a donc été proposée : l'algorithme d'arbre de jonction [Jensen 90], qui permet de faire de l'inférence sur n'importe quel type de graphe.

Cet algorithme peut être vu comme une combinaison des idées de l'algorithme d'élimination et l'algorithme de passage de messages. L'idée de base est de transformer le graphe acyclique $G$ du réseau en un arbre non orienté $T$. Cette transformation opère en trois étapes :

- la première étape est la **moralisation** du graphe $G$. Elle consiste à « marier »deux à deux les parents de chaque nœud, en les reliant par un arc non orienté. A l'issue de cette étape, il reste encore des arcs orientés entre chaque nœud et chacun de ses parents. On finit de moraliser le graphe en enlevant des directions de chaque arc orienté. On aboutit alors au graphe moralisé $G^m$.
- La deuxième étape est la **triangulation** du graphe $G^m$. Cette étape consiste à extraire de $G^m$ un ensemble de cliques de nœuds. Une clique est un sous-graphe du graphe $G^m$ dont tous les nœuds sont connectés deux à deux. Le graphe $G^t$ obtenu est triangulé quand l'ensemble de ses nœuds peuvent être éliminés. Un nœud peut être éliminé s'il appartient à une clique dans le graphe.
- Cette dernière étape correspond à la construction de l'arbre de jonction. A partir du graphe $G^t$ obtenu à l'issue de la triangulation, le problème est de calculer l'arbre couvrant de poids minimum. Pour ce faire, on va procéder à l'élimination des nœuds qui font partie d'une clique. Ce processus d'élimination n'est pas sans rappeler l'algorithme d'élimination de Bucket. L'arbre $T$ obtenu est un arbre non orienté, dans lequel les nœuds sont des cliques.

L'algorithme de passage de messages est ensuite lancé sur cet arbre de jonction $T$ et permet de calculer les probabilités marginales de tous les nœuds pour chaque clique. La complexité de cet algorithme est déterminée par la plus grande clique.

Pour résumer, les algorithmes d'inférence exacte calculent les probabilités marginales en exploitant systématiquement la structure graphique. On cherche à exploiter l'information d'indépendance conditionnelle encodée par les arcs, dans les graphes.

De nombreux modèles graphiques probabilistes, comme les modèles de Markov cachés, ou les réseaux dont le graphe est déjà un arbre ont affaire à ce type d'algorithmes. Mais le problème de ces algorithmes est leur complexité, dépendante de la taille des graphes, du fait que les graphes sont fortement connectés ou non.

Pour pallier ce problème, on peut utiliser des méthodes d'inférence approximative, qui ont une complexité moindre. De la même façon que l'on manipule des probabilités initiales inexactes, car elles sont souvent issues d'une estimation, obtenue grâce à des méthodes d'apprentissage de paramètres (voir section 5.4), les méthodes d'inférence approximative vont fournir des probabilités *a posteriori* (ce sont les probabilités obtenues par inférence) approximatives.

### 5.5.3 Algorithmes d'inférence approximative

Parmi les algorithmes d'inférence approximative, on peut citer les algorithmes d'échantillonnage, tels l'échantillonnage de Gibbs [Gilks-Thomas 94] et les chaînes de Monte Carlo [Liu 08a, Robert 05], les algorithmes variationnels [Wainwright 08], les méthodes de recherche de masse [Henrion 90].

## 5.6 Les réseaux Bayésiens comme classificateurs

Rappelons qu'un moyen de classifier des objets représentés par un ensemble de caractéristiques $f = \{f_1, f_2, \ldots, f_n\}$ est de considérer la classe et les caractéristiques comme des variables aléatoires et de calculer la distribution de probabilité conditionnelle $P(c_i|f), i \in \{1, 2, \ldots, C\}$. L'observation $f$ sera alors affectée à la classe $i$ pour laquelle ladite probabilité est maximale. Ceci constitue un modèle probabiliste, qui peut être représenté graphiquement par des réseaux Bayésiens.

Dans cette section, nous présentons les réseaux Bayésiens classificateurs les plus classiques : le Naïve Bayes est présenté section 5.6.1. La section 5.6.2 présente une extension directe du Naïves Bayes, le TAN, établissant des liens entre variables caractéristiques. Enfin, la section 5.6.3 présente le Multinets, un ensemble de plusieurs classificateurs Bayésiens.

Il existe bien sûr d'autres réseaux Bayésiens classificateurs, que nous avons pu rencontrés dans les chapitre 3 et 4. Nous avons choisi d'étudier ces trois là en particulier pour plusieurs raisons :

- ces réseaux sont simples, rapide à implémenter mais efficaces,
- ils sont couramment utilisés,
- comme nous le verrons dans les chapitres suivants, ils nous ont servis de point de départ pour établir nos propres modèles et/ou nous les avons utilisés à but comparatif.

### 5.6.1 Classificateur Bayésien naïf (Naïve Bayes)

#### 5.6.1.1 Formulation et notations

On rappelle que notre objectif est d'affecter une instance particulière de $f = \{f_1, f_2, \ldots, f_n\}$ à une classe $c_i$ parmi $k$ classes. Des travaux en apprentissage supervisé [Domingos 96, Friedman 97b, Friedman 97a] ont montré que le classificateur Bayésien naïf a de bonnes performances. De même, des travaux plus récents [Kotsiantis 07, Kim 04, Huang 03, Friedman 97b] ont prouvé que le « Naïve Bayes »avait des performances similaires à celles des arbres de décision et des réseaux de neurones.

Notons $F_1, F_2, \ldots, F_n$ les $n$ variables aléatoires représentant les caractéristiques des observations, et $C$ la variable aléatoire représentant la classe. Ce classificateur apprend à partir d'un échantillon d'apprentissage (composé de données étiquetées) la probabilité conditionnelle de chaque caractéristique $F_j$, $\forall j \in \{1, 2, \ldots, n\}$ étant donnée la classe $C$. La classification est alors obtenue en appliquant la règle de Bayes pour calculer la probabilité que la variable aléatoire $C$ prenne la valeur $c_i$ étant donnée une observation des caractéristiques $F_1, F_2, \ldots, F_n$, $\forall i \in \{1, 2, \ldots, k\}$ et en prédisant la classe avec la probabilité *a posteriori* maximale. Ce calcul est possible en faisant une forte hypothèse d'indépendance : toutes les caractéristiques $F_j$ sont conditionnellement indépendantes étant donnée la valeur de la classe $C$. Ainsi on obtient cette formule :

$$P(c_i|f_1, f_2, \ldots, f_n) = \frac{P(f_1, f_2, \ldots, f_n, c_i)}{P(f_1, f_2, \ldots, f_n)} = \frac{P(f_1, f_2, \ldots, f_n|c_i) \times P(c_i)}{P(f_1, f_2, \ldots, f_n)}$$

où

$$P(f_1, f_2, \ldots, f_n) = \sum_{i=1}^{k} P(f_1, f_2, \ldots, f_n, c_i) = \sum_{i=1}^{k} P(f_1, f_2, \ldots, f_n|c_i) \times P(c_i)$$

Représenté par un réseau Bayésien, le classificateur Bayésien naïf a la simple structure décrite en Fig. 5.5.

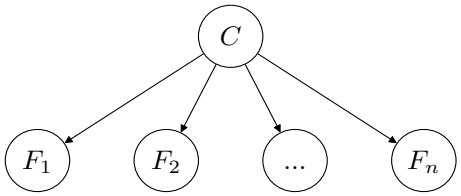



FIGURE 5.5 – Classificateur Bayésien naïf

Ce réseau encode l'hypothèse principale sur laquelle est basé le classificateur Bayésien naïf, à savoir que chaque caractéristique (chaque feuille dans le graphe) est indépendante du reste des caractéristiques, étant donné l'état de la variable classe (la racine du graphe).

### 5.6.2 Classificateur Bayésien naïf augmenté : TAN (tree-augmented naïve Bayesian)

#### 5.6.2.1 Formulation et notations

La performance du classificateur Bayésien naïf est quelque chose de surprenant, car l'hypothèse d'indépendance est clairement irréaliste avec des données réelles. Aussi ce fait soulève la question suivante : peut-on améliorer les performances du classificateur Bayésien en évitant les hypothèses d'indépendance infondées à cause des données ?

En fait il est possible d'apprendre automatiquement la structure de réseaux Bayésiens à partir des données. De nombreuses approches d'apprentissage de structure ont déjà été proposées [François 06, Friedman 98, Buntine 96]. Il s'agit d'une forme d'apprentissage non supervisé, dans le sens que l'on ne distingue pas la variable classe des variables caractéristiques dans les données. Le processus d'optimisation est mis en pratique en utilisant des techniques de recherche heuristique pour trouver le meilleur candidat dans l'espace des réseaux possibles : le processus de recherche repose sur une fonction de score qui évalue le mérite de chaque réseau candidat. Or, en utilisant une telle procédure d'apprentissage et en prenant en compte le statut particulier de la variable classe, Friedman a montré dans [Friedman 97b] que l'on améliore la performance des réseaux Bayésiens comme classificateurs. Un moyen de réaliser ça est d'imposer partiellement la structure du réseau, comme dans le réseau Bayésien naïf, de telle façon qu'il y ait un arc orienté de la variable classe vers chaque variable caractéristique. Ainsi, dans le réseau appris, la probabilité $P(c_i|f_1, f_2, \ \ldots \ , f_n)$ prendra en compte toutes les caractéristiques.

Pour améliorer la performance du classificateur basé sur cette hypothèse, Friedman propose d'augmenter la structure du réseau Bayésien naïf avec un arc entre les caractéristiques, quand c'est nécessaire. De cette façon on évite l'hypothèse d'indépendance entre toutes les caractéristiques. Cette structure est appelée Réseau Bayésien naïf augmenté et les arcs entre variables caractéristiques sont appelés arcs augmentés. Dans un tel réseau, un arc orienté de la variable caractéristique $F_i$ vers la variable caractéristique $F_j$, $i \neq j$ implique que l'influence de la variable $F_i$ sur la valeur de la variable classe $C$ dépend aussi de la valeur de la variable $F_j$. Un exemple de TAN est donné ci-dessous.

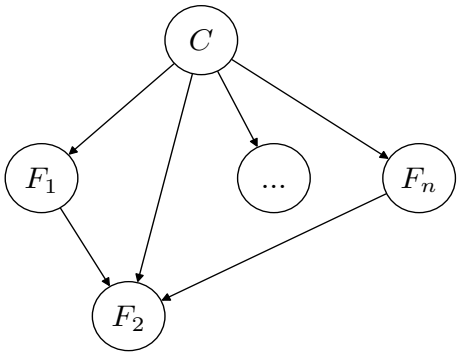



FIGURE 5.6 – Reseau Bayésien naïf augmenté

#### 5.6.2.2 Apprentissage de la structure des TAN

Comme on vient de le voir, lorsque la structure d'un réseau Bayésien n'est pas fournie, ou l'est seulement en partie, il est possible d'apprendre cette structure à l'aide des données. C'est le cas du TAN où l'on sait qu'il existe un arc orienté de la variable classe $C$ vers chaque variable caractéristique $F_i$. Par contre les autres arcs doivent être déterminés automatiquement.

Comme pour l'apprentissage des paramètres d'un réseau, on va distinguer deux cas. Supposons que l'on dispose de variables continues ou discrètes (ou un mélange des deux), et d'un ensemble de données représentatif de plusieurs cas possibles pour chaque variable. L'ensemble des données peut être complet ou incomplet. Suivant le cas, une solution différente va être utilisée.

Dans le cas où l'ensemble des données ne présente pas de données manquantes, deux solutions sont envisageables. La première consiste à analyser les relations d'indépendance conditionnelle entre les variables afin de construire un graphe non orienté. Ce graphe sera ensuite orienté afin d'obtenir un réseau Bayésien. Le problème de cette méthode est que sa complexité augmente avec le nombre de variables et le nombre d'indépendances conditionnelles entre ces variables.

La deuxième solution consiste à construire l'ensemble des graphes possibles. Un score est alors associé à chaque graphe. On choisira le graphe associé au plus grand score. Cette solution pose aussi un problème de complexité car le nombre de graphes possibles, déterminant la base de recherche, peut être très grand.

Ces deux solutions ont, devant ces problèmes, fait l'objet d'améliorations, et sont à la base des méthodes actuelles [Holmes 08, Pena 05].

Dans le cas où l'on ne dispose pas d'assez de données pour représenter tous les cas possibles (présence de données manquantes), l'agorithme EM va être utilisé afin d'estimer ces données manquantes (voir section 5.4).

### 5.6.3 Multinets

#### 5.6.3.1 Formulation et notations

Avec le classificateur Bayésien naïf, les arcs entre les variables caractéristiques sont identiques quelle que soit la valeur de la variable classe. Une généralisation simple est d'avoir des arcs augmentés différents pour chaque classe et ainsi une collection de réseaux comme classificateur.

Pour implanter ce modèle, on partitionne l'échantillon d'apprentissage en $k$ classes. Ensuite, pour chaque classe $c_i \forall i \in \{1, 2, \ \ldots \ , k\}$, on construit un réseau Bayésien $B_i$ avec les

variables caractéristiques $F_1, F_2, \ldots, F_n$. Le réseau $B_i$ encode la distribution de probabilité jointe $P_{B_i}(F_1, F_2, \ldots, F_n)$, étant donnée la classe $c_i$ associée au réseau $B_i$. On a donc $P_{B_i}(F_1, F_2, \ldots, F_n) = P(F_1, F_2, \ldots, F_n | C = c_i)$. Le réseau Bayésien $B_i$ associé à la classe $c_i$ est appelé *réseau local pour* $c_i$. L'ensemble des réseaux locaux combinés aux *a priori* sur la variable classe $C$ est appelé Multinet Bayésien [Heckermann 91, Geiger 96].

Formellement, un multinet est un tuple $M =< P_C, B_1, B_2, \ldots, B_k >$ où $P_C$ est une distribution de probabilité sur la variable classe $C$ et $B_i$ est un réseau Bayésien sur les variables caractéristiques $F_1, F_2, \ldots, F_n \forall i \in \{1, 2, \ldots, k\}$. Le multinet $M$ définit la distribution de probabilité jointe.

$$P_M(C, F_1, F_2, \ldots, F_n) = P_C(c_i) \times P_{B_i}(F_1, F_2, \ldots, F_n), i \in \{1, 2, \ldots, k\}$$

#### 5.6.3.2 Apprentissage des stuctures du multinets

Pour apprendre un multinet, on pose $P_C(c_i)$ égal à la fréquence de la classe $c_i$ dans l'échantillon d'apprentissage. Puis on apprend chaque réseau $B_i$ de la même manière que les TAN. Ensuite on affecte une observation $f$ caractérisée par les valeurs $f_1, f_2, \ldots, f_n$ pour les variables caractéristiques $F_1, F_2, \ldots, F_n$ à la classe $c_i$ pour laquelle la probabilité $P_M(c_i, f_1, f_2, \ldots, f_n)$ est maximale.

En partitionnant les données d'apprentissage suivant la variable classe, cette méthode assure que les interactions entre la variable classe et les variables caractéristiques sont bien prises en compte. Le multinet est simplement une généralisation du classificateur Bayésien naïf augmenté, dans le sens qu'un réseau Bayésien naïf augmenté peut être facilement simulé par un multinet dans lequel tous les réseaux locaux ont la même structure.

On remarque que la complexité en temps de l'apprentissage des arcs augmentés entre les variables caractéristiques est aggravée par le besoin d'apprendre un réseau différent par valeur possible pour la variable classe : la recherche de la structure avec le meilleur score est faite plusieurs fois, autant de fois qu'il y a de classes, et donc chaque fois avec un échantillon d'apprentissage différent.

## 5.7 Conclusion

Nous avons vu, dans ce chapitre, comment construire et utiliser un réseau Bayésien pour faire de l'inférence, en particulier dans un but de classification. De plus, on distingue maintenant mieux les avantages et inconvénients que peut avoir un modèle graphique probabiliste face à un modèle graphique standard.

En effet, les réseaux Bayésiens possèdent tous les avantages des modèles probabilistes standards, ainsi que des avantages supplémentaires liés à leur représentation graphique. Cette représentation graphique va permettre le raisonnement de façon bidirectionnelle (*i. e.* en suivant les relations de dépendances entre variables dans les deux directions). De même, leur représentation facilite la compréhension dans un domaine de connaissances (elle représente directement les connaissances du domaine et non des procédures de raisonnement). Enfin, cette représentation modélise explicitement tous les liens de dépendances entre variables.

Par contre, les réseaux Bayésiens souffrent des mêmes inconvénients que ceux des modèles probabilistes standards, mais présentent des inconvénients supplémentaires liés à leurs représentations : la compréhension des réseaux peut devenir difficile avec un grand nombre de variables et/ou liens de dépendances.

Dans les chapitres suivant, nous présentons nos contributions en relation avec les modèles graphiques probabilistes, pour représenter, rechercher, classer et annoter automatiquement des images.

# Chapitre 6

# Reconnaissance de symboles

## 6.1 Contexte

Dans cette section, nous nous intéressons à un problème particulier : la reconnaissance de symboles. Cette discipline est au cœur de la reconnaissance de formes. Son principe est le suivant : étant donnée une base d'images, où chaque image contient un symbole, notre objectif est de reconnaître le symbole « parfait »(aussi appelé modèle), représenté dans chaque image. Mais, les symboles contenus dans les images ne sont pas parfaits : ils peuvent être bruités, déformés, ou présenter des « occlusions »(« trous »). Ce problème de reconnaissance peut être vu comme un problème de classification : notre but est d'affecter chaque image à une classe, correspondant au symbole parfait (le modèle) de cette image. Cependant, nous ne disposons d'aucun symbole parfait. Par conséquent, notre problème ne peut pas se réduire à la minimisation des distances entre chaque image de la base et chaque symbole parfait. Par contre, cette tâche de classification peut être résolue en utilisant une méthode d'apprentissage supervisé, à partir d'un sous-ensemble de la base pour lequel la classe est connue pour chaque image.

## 6.2 Combinaison de descripteurs

De nombreuses méthodes ont déjà été proposées afin de résoudre de tels problèmes [Ramos-Terrades 09, Valveny 08a, Zhang 06, Yang 05, Lladós 01]. Dans le même sens, de nombreux descripteurs de forme ont déjà été proposés (voir les études [Terrades 07, Zhang 04a, Jain 00]). Ces études ont montré que les différents descripteurs n'ont pas les mêmes performances sur différentes bases de symboles. En effet, d'une base à l'autre, les symboles ne sont pas affectés par le même type de bruits ou le même type de transformations géométriques. Or, certains descripteurs robustes à un type de bruit ne le sont pas nécessairement à un autre. De même, un descripteur invariant à une transformation géométrique donnée ne l'est pas nécessairement à une autre. Il en ressort qu'un seul descripteur n'est en général pas suffisant pour décrire tous les types de formes et donc pour donner des taux de reconnaissance satisfaisants.

Une solution consiste à combiner différents descripteurs de façon à être robuste à plus de bruits et plus de transformations géométriques (comme la rotation, la translation, l'homothétie, ...) qu'en utilisant seulement un seul descripteur. La combinaison peut se faire en utilisant un seul classificateur [Wendling 07] (dans ce cas on pourra parler de fusion précoce), ou plusieurs (un par descripteur) et de combiner leurs sorties [Ramos-Terrades 09, Ramos-Terrades 08] (dans ce cas on parlera de fusion tardive). Les approches de fusion tardive utilisent en général des classificateurs simples. Par contre, si la fusion finale des sorties de chaque classificateur est

réalisée avec un classificateur, ces techniques peuvent poser un problème de sur-apprentissage. Quant aux approches de fusion précoce, elles peuvent poser des problèmes de normalisation du fait de la diversité des descripteurs. De même, la concaténation des vecteurs caractéristiques peut entraîner des problèmes de temps de calcul du fait de l'augmentation de la dimension. Par contre, elles prennent mieux en compte les corrélations entre les composantes des vecteurs caractéristiques, comparées aux approches de fusions tardive. Enfin, elles paraissent plus simples, plus intuitives. De ce fait, nous avons choisi une approche de fusion précoce.

Plus particulièrement, nous nous sommes orientés vers les modèles graphiques probabilistes. Aussi, dans les sous-sections ci-dessous, nous présentons les descripteurs que nous avons choisi de combiner (section 6.3. Dans la section 6.4, nous justifions le choix des modèles graphiques probabilistes, avant de présenter, dans les sections 6.5 et 6.6, les modèles que nous avons proposés. La section 6.7 est dédiée à l'évaluation de ces méthodes. Enfin, dans la section 6.8, nous proposons une critique de ces modèles.

## 6.3 Choix des caractéristiques à combiner

Avant de construire un classificateur à partir d'échantillons d'apprentissage, il convient d'étudier les données disponibles. Dans le cadre de la reconnaissance de symboles, nous manipulons des images noir et blanc, qui peuvent être décrites par des caractéristiques ou vecteurs caractéristiques, obtenus à partir de descripteurs de forme. Le choix de ces descripteurs n'est pas très important dans le sens on l'on souhaite montrer que combiner plusieurs descripteurs améliore le taux de reconnaissance par rapport à l'utilisation d'un seul descripteur, quelles que soient les caractéristiques utilisées. Cependant, certains descripteurs possèdent des propriétés intéressantes qui ont guidé notre choix : robustesse au bruit, invariance à différentes transformations géométriques. D'autres descripteurs sont aussi couramment utilisés dans l'état de l'art et/ou ils sont disponibles dans l'équipe et ils convenait de les utiliser. Quatre descripteurs de forme et trois mesures de forme ont ainsi été choisis : le Generic Fourier Descriptor (GFD), le descripteur Zernike et la $\mathcal{R}$-signature $1D$ et le descripteur *HRT* concernant les descripteurs et la compacité, la rectangularité et l'ellipticité concernant les mesures de formes.

La distinction entre descripteurs et mesures de forme est déterminée par la taille des caractéristiques. Nous considérons les caractéristiques composées d'une seule valeur (*i. e.* des vecteurs unitaires caractéristiques) comme des mesures de forme, et les vecteurs caractéristiques (de plus d'une composante) comme des descripteurs de forme. De plus, les mesures de forme sont discrétisées avec un seuil de discrétisation fixé à 0.5. Cette discrétisation a du sens, avec ce type de caractéristiques, car chacune d'elles est composée d'une seule valeur normalisée entre 0 et 1. Enfin, cette discrétisation permet de considérer les mesures de formes comme des variables discrètes, ce qui nous permettra de montrer l'intérêt de combiner des caractéristiques discrètes et continues pour améliorer la reconnaissance de formes.

Ces descripteurs et mesures de forme, et leurs propriétés, sont définis ci-dessous. Plus particulièrement, les descripteurs de formes sont présentés dans la section 6.3.1, tandis que la section 6.3.2 est dédiée aux mesures de forme.

### 6.3.1 Descripteurs de forme

#### 6.3.1.1 Generic Fourier Descriptor

Generic Fourier Descriptor (GFD) est basé sur la transformée de Fourier [Zhang 02a]. La transformée n'étant pas invariante à la rotation, Dengsheng Zhang et Guojun Lu proposent

dans [Zhang 02a] d'utiliser la transformée de Fourier modifiée polairement (MPFT). La MPFT est définie par :

$$PF(\rho,\phi) = \sum_r \sum_i f(r,\theta_i) exp[j2\pi(\frac{r}{R}\rho + \frac{2\pi i}{T}\phi)] \tag{6.1}$$

où $0 \leq r = [(x-x_c)^2+(y-y_c)^2]^{\frac{1}{2}} < R$ et $\theta_i = i(2\pi/T)(0 \leq i < T)$, $x_c$ et $y_c$ sont les coordonnées du centre de la forme ; $0 \leq \rho < R$, $0 \leq \phi < T$ où R et T sont les résolutions angulaire et radiale. Après normalisation, $PF(\rho,\phi)$ décrit une signature invariante à la rotation et à l'échelle.

#### 6.3.1.2 Zernike

Le descripteur Zernike [Kim 00] est basé sur les moments de Zernike. Ces moments sont construits à partir de polynômes complexes (formule 6.2) et forment un ensemble orthogonal complet défini sur le disque unité.

$$A_{mn} = \frac{m+1}{\pi} \int_x \int_y I(x,y)[V_{mn}(x,y)]\,dydx \tag{6.2}$$

$m$ et $n$ représentent l'ordre du moment et $I(x,y)$ le niveau de gris d'un pixel sur l'image. $V_{mn}$, le polynôme de Zernike, est exprimé en coordonées polaires :

$$V_{mn}(r,\theta) = R_{mn}(r)e^{-jn\theta} \tag{6.3}$$

et $R_{mn}$ étant le polynôme radial orthogonal :

$$R_{mn}(r) = \sum_{s=0}^{\frac{m-|n|}{2}} (-1)^s \frac{(m-s)!}{s!(\frac{m+|n|}{2}-s)!(\frac{m-|n|}{2}-s)!} r^{m-2s} \tag{6.4}$$

Une fois les moments calculés, on obtient un vecteur pour chaque image. Ce descripteur est invariant à la rotation et au changement d'échelle.

#### 6.3.1.3 $\mathcal{R}$-signature $1D$

La $\mathcal{R}$-signature[Tabbone 02] se base sur la transformée de Radon pour représenter une image. La transformée de Radon est la projection d'une image dans un plan particulier. Elle est définie par la formule 6.5 où $\delta(.)$ est la fonction telle que $\delta(x) = 1$ *si* $x = 0$ *et* $0$ *sinon*, $\theta \in [0,\pi[$ et $\rho \in ]-\infty,\infty[$.

$$T_{R^f}(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(xcos\theta + y\sin\theta - \rho)\,dx\,dy \tag{6.5}$$

Cette projection possède des propriétés géométriques intéressantes qui font d'elle un bon descripteur de forme. Suivant ces propriétés géométriques, une signature de la transformée, notée $\mathcal{R}_f$, est créée :

$$\mathcal{R}_f(\theta) = \int_{-\infty}^{\infty} T^2_{Rf}(\rho, \theta) d\rho$$

Cette signature vérifie les propriétés d'invariance à certaines transformations géométriques, telles que la translation et le changement d'échelle (après normalisation). Par contre l'invariance à la rotation est restaurée par permutation cyclique de la signature ou directement à partir de sa transformée de Fourier. La $\mathcal{R}$-signature $1D$ fournit un vecteur de 180 caractéristiques par image.

#### 6.3.1.4 Descripteur *HRT*

Le descripteur *HRT* (« Histogram of Radon Transform ») fait partie des contributions de cette thèse. Il a été présenté à la conférence ICPR'2008 (*cf.* liste des contributions dans le chapitre 1).

Ce descripteur est une matrice de fréquences calculées sur chaque colonne de la transformée de Radon (*i. e.* le paramètre « angle »de la transformée) d'une image. Ainsi le descripteur *HRT* représente un histogramme $2D$ des longueurs de la forme à chaque orientation, pour des images noir et blanc.

**Définition préalable : histogramme d'une fonction**

Soit $f$ une fonction réelle définie sur un domaine $X : f : X \to Y$. Notons aussi $\#$ la cardinalité d'un ensemble et $|X|$ la taille (longueur) du domaine $X$. Alors l'histogramme de $f$ est défini par :

$$H(f)(y) = \frac{\#\{x \in X | y = f(x)\}}{|X|}$$

**Définition du descripteur *HRT***

Une fois ces notions préalables rappelées, on peut définir le descripteur *HRT*, noté $HRT_f$. La valeur de $HRT_f$ pour une fonction $f$ et pour chaque orientation $\theta$ est :

$$HRT_f(y, \theta) = H(Rf(\cdot, \theta))(y), \quad \theta \in [0, \pi)$$

On peut observer que la somme des fréquence doit être de 1 pour chaque angle. Le descripteur *HRT* est invariant à la translation et au zoom. En effet, l'invariance à la translation est due au fait que la translation d'un objet implique seulement un décalage du paramètre radial. Comme l'histogramme est calculé angle par angle, la distribution reste inchangée. De plus, l'invariance au zoom est obtenue grâce aux propriétés de la transformée de Radon. En effet, le zoom d'une image (forme) provoque uniquement un effet sur le paramètre radial de la transformée de Radon. Ici encore, la distribution de chaque colonne reste donc inchangée. Par contre, le descripteur *HRT* n'est pas directement invariant (comme la transformée de Radon et la $\mathcal{R}$-signature $1D$ ) à la rotation. Cependant, grâce aux propriétés de la transformée de Radon, l'invariance à la rotation peut être obtenue par permutation circulaire de chaque colonne (histogramme) de la matrice du descripteur *HRT*. En effet, un changement d'orientation provoque un décalage du paramètre angulaire de la transformée et pour une rotation $\theta'' = \theta' + \theta$ :

$$HRT_f(y, \theta'') = H(Rf(\cdot, \theta' + \theta))(y)$$

Ainsi, une rotation d'angle $\theta'$, à partir d'une position initiale $\theta$ provoque un décalage de valeur $\theta'$ dans la matrice *HRT*. La figure 6.1 montre le descripteur *HRT* d'une même forme

après zoom, translation et rotation. On remarque que seule la rotation implique une décalage de l'histogramme.

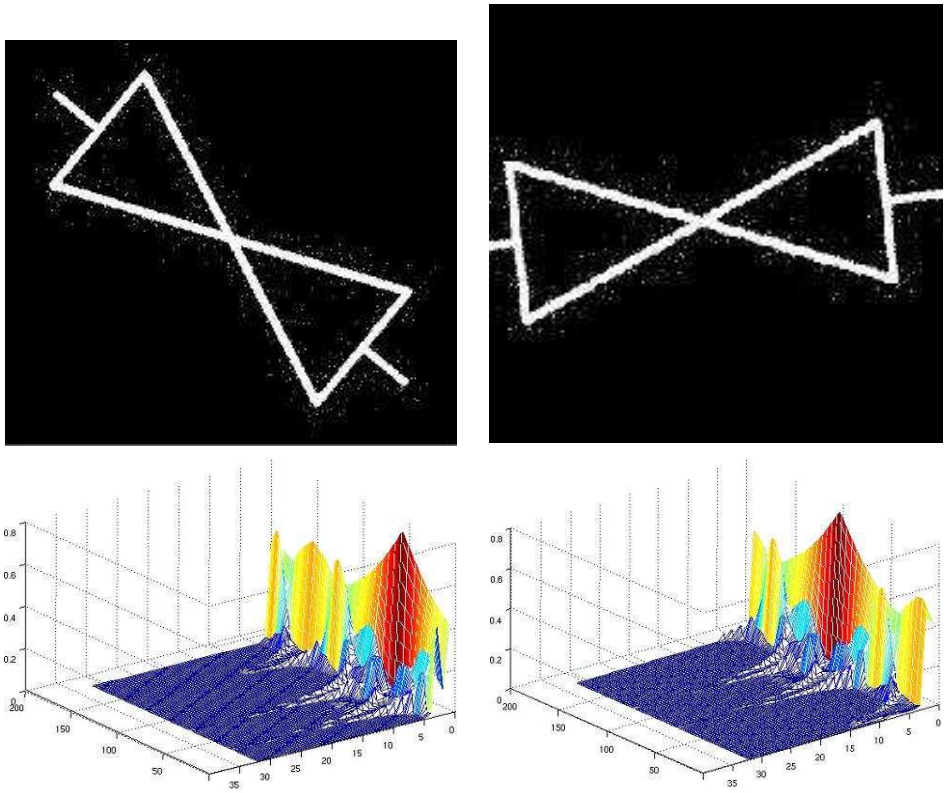


FIGURE 6.1 – Le descripteur *HRT* face à la rotation

**Performances du descripteur** *HRT*

La performance du descripteur *HRT* a été évaluée sur différentes bases de d'images et comparée à celle d'autres descripteurs couramment utilisés dans les problèmes de reconnaissance de formes. Nous avons utilisé trois bases d'images différentes : une base de formes, une base de logos, et une base de symboles. La base de formes est constituées de 18 classes de formes pleines et de 12 images par classe. Les bases de logos et de symboles ont été construites en utilisant le portail fourni par le projet Eperies [13]. Chacune de ces deux bases est composée de 20 classes d'images ayant subi des rotations des translations ou des zooms. Chaque base comporte 300 images mais ces images ne sont pas réparties de façon uniforme dans les classes. La performance d'un descripteur étant dépendante de l'application et des propriétés géométriques des formes, ces 3 bases ont été choisies de façon à couvrir une grande variété de formes. En effet, la base de formes contient des formes pleines connectées, tandis que la base de logos est composée de formes pleines non connectées. Enfin, la base de symboles est constituée de symboles électriques non connectés.

Le descripteur *HRT* a été comparé à la signature angulaire (ART, de Angular Radial Transform, en anglais) [Kim 99] et au descripteur shape context (SC) [Belongie 02], ainsi qu'à la $\mathcal{R}$-signature $1D$ et au Generic Fourier Descriptor que nous venons de présenter. Tous ces des-

13. `http://www.epeires.org/`

cripteurs sont des descripteurs calculés sur des régions de l'image, à l'exception du descripteur SC qui est basé sur les contours.

Afin d'évaluer la performance du descripteur *HRT*, comparé aux 3 autres, sur les 3 bases d'images, nous avons mesuré le taux d'images pertinentes en fonction du rang de ces images (ARR, pour Average Relevant Rank, en anglais). En considérant toutes les images d'une base en tant que requête, la mesure ARR calcule la moyenne des images pertinentes parmi les $k$ premières images retournées. Les résultats de la comparaison des descripteurs sur les 3 bases sont présentés dans la figure 6.2. On constate que le descripteur *HRT* est plus performant, dans la plupart des cas.

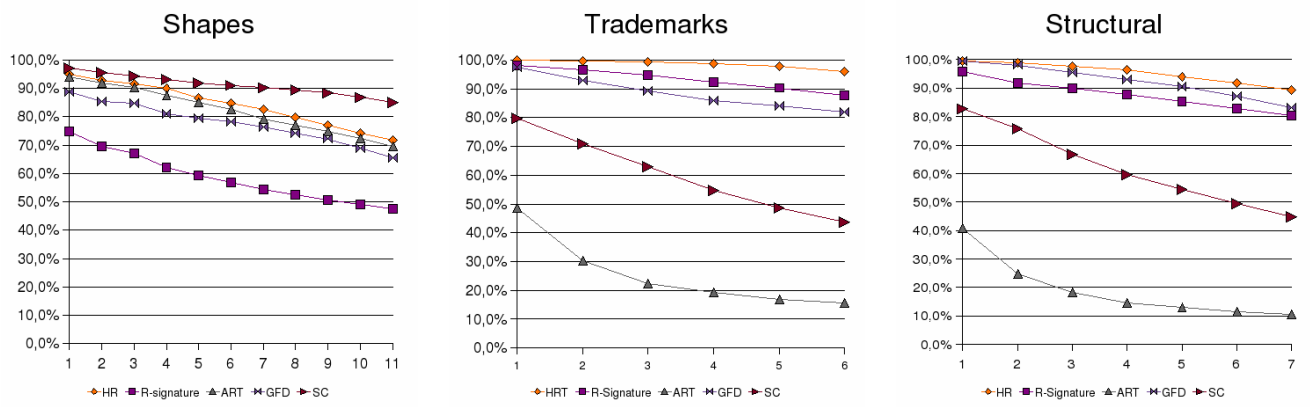



FIGURE 6.2 – Taux d'images pertinentes en fonction du rang

De plus, nous avons testé la robustesse au bruit du descripteur *HRT*. Les deux bases de logos et de symboles ont été contaminées par du bruit de type Kanungo [Kanungo 00]. Ces bruits sont similaires à ceux qui peuvent apparaître dans un document lorsqu'il est scanné, imprimé ou photocopié. La figure 6.3 présente les résultats obtenus par les 4 descripteurs sur ces deux bases bruitées. On remarque que les performances des quatre descripteurs diminuent avec le bruit. Par contre, notre descripteur *HRT* est le plus performant sur la base de logos. Sur la base de symboles, les performances de *HRT* sont similaires à celles obtenues par *GFD*.

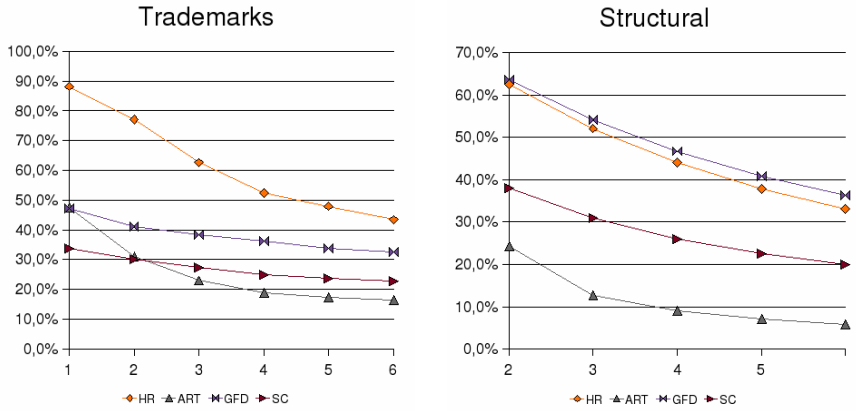



FIGURE 6.3 – Taux d'images pertinentes en fonction du rang - bases bruitées

### 6.3.2 Mesures de forme

#### 6.3.2.1 Compacité

La mesure de compacité $C$ représente le ratio entre l'aire de la forme et l'aire du cercle (la forme la plus compacte) ayant le même périmètre :

$$C = \frac{4\pi A}{P^2}$$

où $P$ est le périmètre et $A$, l'aire.

Cette mesure est invariante à la translation, à la rotation, et au zoom.

#### 6.3.2.2 Rectangularité

Le degré de rectangularité [Rosin 99] $R$ est égal au ratio entre l'aire de la forme et l'aire de sa boîte englobante minimale :

$$R = \frac{A}{L * l}$$

où $A$ est l'aire de la forme et $L$ (respectivement $l$) sont la longueur (respectivement la largeur) de la boîte englobante minimale.

#### 6.3.2.3 Ellipticité

Le degré d'ellipticité $\epsilon$ est obtenu à partir du ratio entre l'axe majeur et l'axe mineur [Teague 79] :

$$\epsilon = 1 - \frac{b}{a}$$

où $a$ est l'axe majeur et $b$ l'axe mineur.

Cette mesure est invariante à la rotation, la translation et à l'homothétie.

## 6.4 Combinaison de descripteurs avec des classificateurs Bayésiens

### 6.4.1 Pourquoi choisir les modèles probabilistes ?

Nous avons vu que notre problème de reconnaissance de symboles peut être vu comme un problème de classification d'images. Cette tâche nécessite un classificateur, *i. e.* une fonction qui associe une classe (correspondant à un symbole parfait) à des observations (des images de symboles bruités, déformés, *etc.*) décrites par un ensemble de caractéristiques. La construction de classificateurs à partir d'échantillons d'apprentissages (ensemble de données pour lesquelles la classe est connue) est un problème central en apprentissage : c'est un problème d'apprentissage supervisé. En effet, dans nombre d'applications, l'objectif est d'affecter un vecteur caractéristique $f = \{f_1, f_2, \ldots, f_n\}$ à une classe $c_i$ parmi $k$ classes, représentées par le vecteur $c = \{c_1, c_2, \ldots, c_k\}$. Comme nous l'avons vu dans l'état de l'art (Chapitre 3), plusieurs approches à ce problème sont basées sur des représentations fonctionnelles, telles que les arbres de décision, les réseaux de neurones, les graphes décisionnels, *etc.*. [Bishop 06, Safavian 91, Zhang 00, Nielsen 09], associés à des règles de décision.

Les approches probabilistes jouent aussi un rôle central en classification [Zhang 04b, Paek 00, Kitamoto 99]. Un moyen d'atteindre le précédent but, en utilisant les probabilités, est de calculer

la distribution de probabilité conditionnelle $P(c_i|f), \forall i \in \{1, 2, \ldots, k\}$ et d'affecter l'observation $f$ à la classe $c_i$ pour laquelle cette probabilité est maximale. On parle de modèle probabiliste.

### 6.4.2 Pourquoi est-il peut-être plus judicieux de choisir les modèles graphiques probabilistes ?

Afin de représenter les distributions de probabilités d'un large ensemble de variables, des hypothèses d'indépendance conditionnelle peuvent être introduites afin de réduire la complexité des modèles probabilistes et de fournir des modèles tractables. Or, parmi les modèles graphiques probabilistes évoqués dans l'état de l'art [Jordan 03], une classe de modèles appelés réseaux Bayésiens, permet de représenter efficacement les distributions de probabilités, en les factorisant grâce à des hypothèses d'indépendances. Cette factorisation aide à réduire la complexité du modèle. De plus, les réseaux Bayésiens sont associés à des algorithmes d'inférence (*i. e.* de calcul des probabilités *a posteriori*) et d'apprentissage (factorisation des paramètres, calcul des distributions de probabilités, ...).

Les modèles probabilistes, et plus particulièrement des modèles graphiques, semblent donc être adaptés à la résolution de notre problème. Aussi, dans les sous-sections suivantes, nous présentons les différents modèles que nous avons proposés.

## 6.5 Le Naïve Bayes et d'autres réseaux Bayésiens usuels

### 6.5.1 Motivations : pourquoi un classificateur Bayésien naïf ?

Parmi les classificateurs Bayésien, le Naïve Bayes (voir section 3.4), est le plus facile à mettre en œuvre. De plus, il est efficace, malgré les hypothèses d'indépendance non vérifiée, en général, avec des données réelles. Il semblait donc intéressant de l'utiliser dans le cadre de notre problème de reconnaissance de symboles.

Nous disposons d'un ensemble de descripteurs et mesures de forme, fournissant, soit des vecteurs de grande dimension de valeurs continues (pour les descripteurs de forme), soit des valeurs uniques continues (pour les mesures de forme). Or, le classificateur Bayésien naïf requiert des variables discrètes. De plus, il est sensible à la dimensionnalité des données.

D'un premier abord, les SVMs, réputés pour leur capacité à traiter à la fois des données discrètes et continues, et leur robustesse à la dimensionnalité des données, semblaient donc plus adaptés à notre problème de classification. Cependant, nous montrons, dans cette section, que la discrétisation des données, associée à une méthode de sélection de variables appropriée, améliore significativement la performance de classification des SVMs, aussi bien que celle du Naïve Bayes. De plus, le Naïve Bayes se montre alors compétitif, en termes de taux de reconnaissance et de complexité, aux SVMs.

Dans le reste de cette section, nous expliquons donc comment nous avons adapté le classificateur Naïve Bayes, et d'autres réseaux Bayésiens usuels, au problème de reconnaissance de symboles et montrons comment nous avons adapté une méthode de sélection de variables à notre problème. Enfin, nous présentons la méthode de discrétisation utilisée.

### 6.5.2 Adaptation du Naïve Bayes et autres réseaux Bayésiens usuels

#### 6.5.2.1 Réduction de dimensionnalité

Pour combiner l'ensemble des $n$ descripteurs et $m$ mesures de forme choisis, on commence par les calculer sur l'ensemble de la base d'images. On obtient alors $n$ signatures et $m$ valeurs

par image. Chaque image sera alors représentée par un vecteur de caractéristiques correspondant à la concaténation des $n$ signatures et $m$ valeurs obtenues. Ce vecteur de caractéristiques est de dimension égale à la somme des dimensions des signatures qui le composent, à laquelle on ajoute le nombre de mesures de formes. Or *GFD* fournit un vecteur de 225 composantes, Zernike un vecteur de 34 composantes, la $\mathcal{R}$-signature $1D$ un vecteur de 180 composantes, et le descripteur *HRT* une matrices de $32 * 180 = 5760$ valeurs. Les signatures initiales sont donc déjà de grande dimension. De plus, leur concaténation nous fournit un vecteur pouvant aller de 6199 composantes par image en utilisant tous les descripteurs de forme disponibles. Ceci nous place face à un problème de dimensionnalité. En effet, afin d'adapter le classificateur Bayésien naïf à notre problème, on considère chaque composante du vecteur caractéristique comme une variable aléatoire. Chacune de ces variables correspond à un nœud de type caractéristique du réseau Bayésien. On dispose donc d'un réseau composé de 6200 variables si on utilise tous les descripteurs : 6199 variables caractéristiques et une variable classe. Ce nombre de variables est trop important comparé au nombre de données d'apprentissage disponibles. On parle de *malédiction des grandes dimensions* ou du *Small Sample Size problem* (SSS).

Pour pallier ce problème, nous devons utiliser une méthode de réduction de dimension. Un nombre important d'algorithmes de réduction de dimension ont été proposés dans la littérature et des études comparatives existent [Denoeux 07, Kudo 00, Dash 97].

Nous présentons,ici, quelques méthodes couramment utilisées dans l'État de l'art : l'analyse en composantes principales (ACP), les méthodes séquentielles et le LASSO. Ces méthodes sont décrites ci-dessous.

**Analyse en composantes principales : ACP**

L'analyse en composantes principales (ACP) est une analyse de la matrice variance-covariance, *i. e.* une analyse de la variabilité/dispersion des données, représentées par plusieurs variables.

Excepté si l'une des variables peut s'exprimer en fonction d'une autre, on a besoin de toutes les variables pour prendre en compte toute la variabilité du système. L'objectif de l'ACP est de décrire à l'aide d'un sous-ensemble des variables initiales un maximum de cette variabilité.

L'ACP nous fournit des composantes principales, bien sûr en nombre inférieur à celui des variables initiales, qui sont des combinaisons linéaires des variables initiales, et que l'on considère comme de nouvelles variables.

Nous avons appliqué cette méthode à nos caractéristiques issues des descripteurs de forme. Elle s'est révélée inefficace car :

- On n'obtient pas un sous-ensemble des variables initiales, mais de nouvelles variables. Il faut donc recalculer pour chaque observation un nouveau vecteur de caractéristiques, qui est la projection dans l'espace des composantes principales du vecteur de caractéristiques initial.
- La réduction de dimension est faite indépendemment de la classe.

Parmi les méthodes de réduction de dimension, on préférera donc s'orienter vers les méthodes de sélection de variables en particulier. En effet, ces méthodes permettent non seulement de réduire la dimension, mais aussi de sélectionner un sous-ensemble des variables initiales. Ces méthodes de sélection de variables sont plus adaptées à notre problème, car notre objectif est de réduire le nombre de caractéristiques afin de réduire la taille de notre réseau Bayésien et la complexité de notre méthode.

**Méthodes séquentielles**

Afin de réduire la dimension tout en sélectionnant réellement des variables dans l'ensemble

des variables initiales, des heuristiques basées sur des parcours séquentiels sont souvent préférées aux méthodes type ACP. Elles consistent à rajouter ou éliminer itérativement des variables [Devijver 82]. Dans ces approches, il est possible de partir d'un ensemble de variables vide et d'ajouter des variables à celles déjà sélectionnées (il s'agit de la méthode Sequential Forward Selection (SFS)) ou de partir de l'ensemble de toutes les variables et d'éliminer des variables parmi celles déjà sélectionnées (dans ce cas on parle de Sequential Backward Selection (SBS)). Ces méthodes sont connues pour leur simplicité de mise en œuvre et leur rapidité. Cependant, comme elles n'explorent pas tous les sous-ensembles possibles de variables et ne permettent pas de retour arrière pendant la recherche, elles sont donc sous-optimales. Pour réduire cet effet, des méthodes alternent les procédures SFS et SBS, permettant ainsi d'ajouter des variables et puis d'en retirer d'autres. Ces méthodes sont apparues en 1976 sous le nom de $PTA(l, r)$ de « plus $l$-take away $r$ », en anglais [Stearns 76]. Ces méthodes sont réputées pour leur simplicité et leur rapidité. Cependant, elles sont aussi connues pour leur instabilité. De plus, comme les n'explorent pas tous les sous-ensembles de variables possibles et qu'elles ne permettent pas de retour arrière durant le processus de sélection, elles ne sont pas optimales.

Il existe d'autres méthodes de sélection de variables basées sur de la régression linéaire. C'est le cas de la méthode du LASSO que nous présentons ci-dessous.

**Least Absolute Shrinkage and Selection Operator : LASSO**

Le LASSO est une méthode de sélection de variables et de diminution de coefficients pour la régression linéaire (on peut parler de méthode de rétrécissement), introduite par R. Tibshirani [Tibshirani 96]. Son objectif est de minimiser la somme des erreurs quadratiques, avec un seuil associé à la somme des valeurs absolues des coefficients :

$$\beta^{lasso} = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

avec la contrainte

$$\sum_{j=1}^{p} |\beta_j| \leq s \tag{6.6}$$

Pour adapter cette méthode à notre problème, nous avons appliqué la forme linéaire du LASSO à nos données d'apprentissage, dans une étape de pré-traitement. Le LASSO a été utilisé uniquement sur les vecteurs caractéristiques issus des descripteurs de forme. En effet, une mesure de forme correspondant à une seule variable ne nécessite pas de sélection.

Pour chaque échantillon d'apprentissage disponible, $y_i$ représente la somme des composantes du vecteur caractéristique moyen de la classe $i$, et $x_j = \{x_{j1}, x_{j2}, \ldots, x_{jp}\}$ le vecteur des $p$ caractéristiques de l'observation $j$. Une fois les variables sélectionnées, seul le sous-ensemble des ces variables est utilisé dans notre classificateur Bayésien.

Le LASSO utilise une pénalité $L_1$ : $\sum_{j=1}^{p} |\beta_j|$. Cette contrainte implique que pour des petites valeurs de $s$, $s \geq 0$, certains coefficients $\beta_j$ seront égaux à 0. Ainsi choisir $s$ revient à choisir le nombre des variables prédictives du modèle de régression. Par conséquent, seules les variables correspondant aux coefficients non nuls sont sélectionnées. La valeur de $s$ est choisie de telle façon que ce soit la plus grande valeur permettant d'annuler au moins 1 coefficient. Le calcul des solutions du LASSO est un problème de programmation quadratique, qui peut être traité par

des algorithmes standards d'analyse numérique. Un des plus adaptés est l'algorithme de Least Angle Regression (LAR). En effet, il exploite la structure particulière du problème du LASSO et fournit un moyen efficace de calculer simultanément les solutions pour toutes les valeurs de $s$. Cet algorithme est présenté en détails dans [Efron 04], mais nous donnons ci-dessous son fonctionnement général :

- commencer par initialiser les coefficients $\beta_j$ à zéro
- trouver la variable prédictive $x_j \in \{x_1, x_2, \ldots, x_p\}$ la plus corrélée avec la variable $y_i$. On rappelle que $p$ désigne le nombre de caractéristiques
- modifier le coefficient $\beta_j$ en fonction du signe de la corrélation de $x_j$ avec $y_i$, *i. e.* augmenter $\beta_j$ si la corrélation est positive, le diminuer sinon. Récupérer le résidus $r = y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j$. Continuer à modifier $\beta_j$ jusqu'à ce qu'une autre variable prédictive $x_k$ ait une meilleure corrélation avec $r$ que $x_j$.
- modifier le couple $(\beta_j, \beta_k)$ en fonction de son écart-type, jusqu'à ce qu'une autre variable prédictive $x_m$ ait une meilleur corrélation que $x_k$ avec le résidus $r$.
- continuer jusqu'à ce que tous les prédicteurs soient dans le modèle.

Pour utiliser cet algorithme dans le cadre de la résolution du problème du LASSO, il suffit de supprimer une variable $x_j$, associée à un coefficient $\beta_j$ , lorsque celui-ci atteint zéro.

La méthode du LASSO a finalement été choisie pour sa stabilité et sa facilité d'implémentation. De plus, elle s'est avérée plus efficace expérimentalement, du point de vue du taux de reconnaissance, que la méthode SFS, qui reste la plus populaire (voir section 6.7. Enfin, le LASSO permet de sélectionner des variables, contrairement à l'ACP, qui réduit la dimension, en fournissant des combinaisons linéaires des variables initiales. En sélectionnant un sous-ensemble de variables, on parvient ainsi à réduire la taille de notre réseau, ainsi que sa complexité.

La sélection d'un sous-ensemble des variables initiales représente donc la première étape de l'adaptation du Naïve Bayes à notre problème. Il s'agit d'un pré-traitement des données. Cette étape est indépendante de la phase de discrétisation que nous présentons dans le paragraphe suivant, et aussi indépendante de la phase finale de classification.

#### 6.5.2.2 Discrétisation des données

Les descripteurs de forme décrits précédemment nous fournissent des signatures de valeurs continues. Ainsi, une fois que l'on a sélectionné un sous-ensemble de variables, on dispose d'un ensemble réduit de variables continues, ainsi qu'un ensemble de valeurs discrètes issues des mesures de formes. Or, le classificateur Bayésien naïf requiert des variables discrètes. Une deuxième étape de pré-traitement des données est donc nécessaire afin de discrétiser nos variables continues.

La discrétisation est un processus de transformation de variables à valeurs continues en variables à valeurs discrètes, en créant un ensemble d'intervalles contigus (ou, de manière équivalente un ensemble de points de rupture) afin de répartir les valeurs de variables. On peut classer les méthodes de discrétisation en deux catégories distinctes : les méthodes non supervisées, qui n'utilisent pas l'information contenue dans la variable cible (la variable classe), et les méthodes supervisées, qui, elles, l'utilisent. Des études ont démontré [Data 96, Liu 97] que la discrétisation supervisée est plus bénéfique pour la classification que la discrétisation non supervisée. Ainsi nous nous sommes concentrés sur les méthodes supervisées. Typiquement, ces méthodes discrétisent une variable en un simple intervalle si la variable a peu ou pas de corrélation avec la variable cible.

Nous avons utilisé la méthode présentée dans [Colot 04], parce qu'elle permet une réduction importante des données, en passant de données continues à discrètes, sans perte d'information. De plus elle est facile à implémenter. Concrètement, cette méthode consiste à approximer des lois

de probabilités par des histogrammes. Ces histogrammes doivent approximer de façon optimale, au sens du maximum de vraisemblance et de l'erreur quadratique moyenne, les lois de probabilités inconnues d'un processus aléatoire avec un seul échantillon. Afin d'obtenir le nombre de classes de l'histogramme et la distribution pour chaque classe, le critère d'information de Akaike (AIC), habituellement utilisé pour la sélection d'ordres de modèles, a été généralisé à ce problème. Plus précisément, on commence par construire un histogramme à $m$ classes à partir de l'échantillon. Ensuite, l'idée pour réduire la taille de l'histogramme est de fusionner les deux classes adjacentes qui maximisent la différence entre la valeur du critère avant et après la fusion. Ce processus est répété jusqu'à ce que cette différence soit négative.

Par exemple la figure 6.4 montre une $\mathcal{R}$-signature $1D$ (courbre de gauche) et la signature discrétisée (courbe de droite).

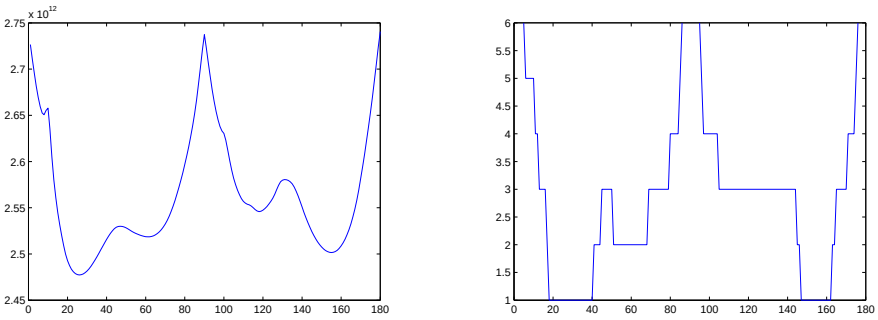



FIGURE 6.4 – Une signature (à gauche) et sa discrétisation (à droite)

Nous avons appliqué cette méthode pour toutes les variables de la base d'exemple. On dispose donc d'un échantillon par variable et la méthode nous fournit un histogramme par variable. Considérons alors un histogramme : chaque classe correspond à une valeur discrète possible pour la variable correspondant à cet histogramme, et on obtient la probabilité de chaque valeur possible grâce au nombre d'éléments dans chaque classe. Ainsi on estime la distribution de probabilité de chaque variable.

La discrétisation des données continues représente la deuxième étape de l'adaptation du Naïve Bayes à notre problème. Il s'agit, comme l'étape de sélection de variables, d'une étape de pré-traitement des données. Cette étape est indépendante de la phase finale de classification grâce au Naïve Bayes.

#### 6.5.2.3 Classification

Afin de classifier un ensemble de symboles, la démarche est la suivante : d'abord l'ensemble des descripteurs et mesures de formes est calculé sur toute la base d'images disponibles. La base d'images est alors divisée en deux sous-bases : les images pour lesquelles la classe est connue constituent l'échantillon d'apprentissage, le reste des images sont celles à reconnaître et constituent l'échantillon de test. La méthode du LASSO est alors appliquée sur l'échantillon d'apprentissage, en considérant uniquement les variables issues des descripteurs de forme (chaque variable correspond à une composante d'un vecteur caractéristique obtenu grâce à un descripteur de forme), afin de sélectionner un sous-ensemble de variables. Une fois cette première étape de pré-traitement effectuée, on a à notre disposition un nouvel ensemble de variables caractéristiques : un sous-ensemble des variables continues initiales, ainsi que les variables discrètes corres-

pondant chacune à une mesure de forme. Intervient alors une deuxième étape de pré-traitement des données : celle de discrétisation des variables continues sélectionnées. On aboutit alors à un ensemble de variables caractéristiques discrètes. Le Naïve Bayes peut alors être construit.

Il faut maintenant estimer les distributions de probabilité de ces variables caractéristiques et de la variable classe. Les variables étant toutes discrètes, leur domaine est obtenu à partir de la discrétisation. Ainsi toutes les variables caractéristiques et la variable classe suivent une loi multinomiale. Les probabilités associées à chaque valeur possible sont estimées à partir de l'échantillon d'apprentissage, en utilisant la méthode du maximum de vraisemblance expliquée dans le chapitre 5. Finalement le naïve Bayes est construit de cette façon : il possède une variable discrète classe et une variable discrète pour chaque composante du vecteur caractéristique initial.

Pour classifier une nouvelle image, la notion d'inférence est utilisée, en particulier l'algorithme de passage de message présenté dans le chapitre 5. En effet, cet algorithme est bien adapté à la structure d'arbre du Naïve Bayes. La probabilité *a posteriori* de la variable classe sera calculée pour chaque image requête. Cette image sera affectée à la classe qui a la plus grande probabilité. Dans ce cas où l'on a uniquement des variables discrètes, ce problème peut être résolu simplement à partir de manipulations algébriques. La représentation graphique n'est donc ici pas indispensable. Le naïve Bayes peut cependant être représenté par un graphe et être utile d'un point de vue visuel : en effet, dans ce cas, même si la représentation graphique n'est pas indispensable, elle permet de visualiser facilement les dépendances conditionnelles entre les variables. La représentation graphique du Naïve Bayes est redonnée ci-dessous. Comme le veut la structure d'un classificateur naïf (*cf.* section 5.6.1), on a un arc orienté de la variable classe vers chaque variable caractéristique.

– C désigne la variable aléatoire « Classe »
– $F_1, F_2, \ldots, F_n$ sont les variables caractéristiques

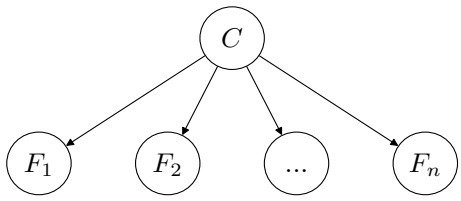



FIGURE 6.5 – Naïve Bayes

Enfin, le réseau Bayésien naïf augmenté (TAN) et le Multinets, tous les deux présentés dans le chapitre 5, ont été adaptés de la même façon. Les structures du TAN et des réseaux du Multinets ont été apprises avec l'algorithme MWST (arbre de recouvrement maximal) [François 04].

## 6.6 Modèle de mélange de lois de Bernoulli et de mélanges de Gaussiennes

### 6.6.1 Pourquoi les réseaux usuels ne suffisent pas ?

Le Naïve Bayes, le TAN et le Multinets sont des modèles simples et efficaces, mais, dans leur forme standard, ils requièrent des variables discrètes. En particulier, le Naïve Bayes, associé à un pré-traitement de discrétisation des données, il se montre compétitif avec les classificateurs couramment utilisés dans l'État de l'art (voir section 5.6.1 et dans notre étude comparative section 6.7), comme les SVMs, réputés pour leur robustesse en présence de données de grande dimension, et adaptés à la classification de données à valeurs continues.

Cependant, on souhaiterait classifier les images en limitant au maximum le nombre de pré-traitements. Nous avons donc besoin d'un classificateur permettant de gérer les données continues (valeurs provenant des descripteurs de forme) et discrètes (valeurs provenant des mesures de forme). Ce type de modèle est trop grand pour être représenté par une unique distribution de probabilité jointe. Par conséquent, il est nécessaire d'introduire de la connaissance structurelle *a priori* : le Naïve Bayes doit être étendu de façon à prendre en compte ces deux types de valeurs. Le pré-traitement de discrétisation sera ainsi évité, ce qui générera un gain de temps et de précision dans l'estimations des distributions de probabilités.

### 6.6.2 Comment étendre le Naïve Bayes ? Avec des modèles graphiques

Les modèles graphiques probabilistes, et en particulier les réseaux Bayésiens, sont un bon moyen de résoudre ce type de problème. En effet, dans les réseaux Bayésiens, la distribution de probabilité jointe est remplacée par une représentation structurelle, uniquement entre les variables s'influençant les unes les autres. Les interactions entre les variables indirectement reliées sont ensuite calculées par inférence, par propagation des croyances dans le graphe au travers des connections directes. Ainsi, les réseaux Bayésiens sont un moyen simple de représenter la distribution de probabilité jointe d'un ensemble de variables aléatoires, de visualiser les propriétés de dépendance conditionnelle et d'effectuer des opérations complexes comme l'estimation de probabilité, l'inférence, selon des calculs à base de graphes.

### 6.6.3 Modèle de mélange GM-B

Un réseau Bayésien, combinant des variables discrètes et continues, est proposé. Dans cette partie, nous présentons un modèle probabiliste hiérarchique : le modèle de mélange de lois de Bernoulli et de mélanges de Gaussiennes (GM-B). En effet, l'observation de la distribution sur les différents histogrammes des composantes des vecteurs caractéristiques fournis par les descripteurs de forme nous ont conduits à considérer que les caractéristiques visuelles continues peuvent être estimées par des densités à mélanges de Gaussiennes. Les variables discrètes issues des mesures de forme suivent une loi de Bernoulli. En effet, ces variables peuvent prendre deux valeurs : 0 si la mesure de forme correspondante fournit une valeur inférieure à $0,5$, 1 sinon.

- Soit $\{MF_1, MF_2, \ldots, MF_n\}$ l'ensemble des mesures de forme calculées. $n$ désigne le nombre maximal de mesures de forme utilisées.
- Chaque variable $MF_j, \forall j \in \{1, \ldots, n\}$ peut prendre deux valeurs : 0 ou 1

Chaque variable $MF_j, \forall j \in \{1, \ldots, n\}$ suit une loi Bernoulli de paramètre $p$, où $p$ est la probabilité associée à la valeur 1 : $p(MF_j = 0) = p$ et $p(MF_j = 1) = 1 - p$.

Le modèle proposé reste quand même inspiré du Naïve Bayes dans le sens où la variable classe est connectée à toutes les autres (*cf.* figure 6.7).

- Soit $F$ un échantillon d'apprentissage composé de $m$ individus $f_{1_i}, \ldots, f_{m_i}, \forall i \in \{1, \ldots, n\}$,
- $n$ est la dimension des signatures obtenues par concaténation des vecteurs caractéristiques issus du calcul des descripteurs sur chaque image de l'échantillon.
- Chaque individu $f_j, \forall j \in \{1, \ldots, m\}$ est caractérisé par $n$ variables continues.

Comme nous l'avons vu dans la section 6.1, nous sommes dans le cadre d'une classification supervisée. Les $m$ individus sont donc divisés en $k$ classes $c_1, \ldots, c_k$.

- Soit $G_1, \ldots, G_g$ les $g$ groupes dont chacun a une densité Gaussienne avec une moyenne $\mu_l, \forall l \in \{1, \ldots, g\}$ et une matrice de covariance $\sum_l$.
- Soit $\pi_1, \ldots, \pi_g$ les proportions des différents groupes,
- soit $\theta_l = (\mu_l, \sum_l)$ le paramètre de chaque Gaussienne,

– soit $\Phi = (\pi_1, \pi_2, \ldots, \pi_g, \theta_1, \ldots, \theta_g)$ le paramètre global du mélange.

Alors la densité de probabilité de $F$ conditionnellement à la classe $c_i, \forall i \in \{1, \ldots, k\}$ est définie par $P(f, \Phi) = \sum_{l=1}^{g} \pi_l p(f, \theta_l)$ où $p(f, \theta_l)$ est la Gaussienne multivariée définie par le paramètre $\theta_l$.

Ainsi, nous avons un modèle de mélange de Gaussiennes (GMM) par classe. Ce problème peut être représenté par le modèle probabiliste de la figure 6.6, où :

– Le nœud « Classe »est un nœud discret, pouvant prendre $k$ valeurs correspondant aux classes prédéfinies $c_1, \ldots, c_k$.
– Le nœud « Composante »est un nœud discret correspondant aux composantes (*i. e.* les groupes $G_1, \ldots, G_g$) des mélanges. Cette variable peut prendre $g$ valeurs, *i. e.* le nombre de Gaussiennes utilisé pour calculer les mélanges. Il s'agit d'une variable latente qui représente le poids de chaque groupe (*i. e.* les $\pi_l, \forall l \in \{1, \ldots, g\}$).
– Le nœud « Gaussienne »est une variable continue représentant chaque Gaussienne $G_l, \forall l \in \{1, \ldots, g\}$ avec son propre paramètre ($\theta_l = (\mu_l, \sum_l)$). Il correspond à l'ensemble des vecteurs caractéristiques dans chaque classe. Ces vecteurs caractéristiques ont une distribution gaussienne de paramètres $\theta_l$
– Enfin, les arêtes représentent l'effet de la classe sur le paramètre de chaque Gaussienne et son poids associé. Le cercle vert sert à montrer la relation entre le modèle graphique proposé et les GMMs : nous avons un GMM (entouré en vert), composé de Gaussiennes et de leur poids associé, par classe. Chaque GMM a son propre paramètre global.

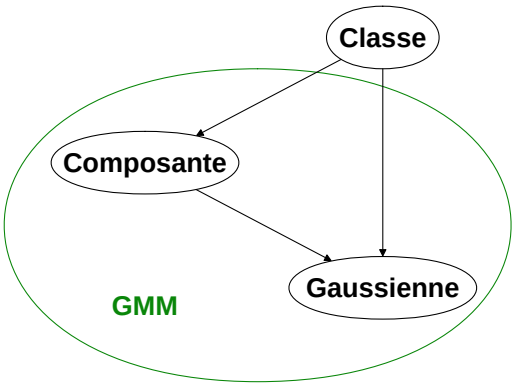



FIGURE 6.6 – GMMs représentés par un modèle graphique probabiliste

La probabilité jointe de ce modèle est obtenue en prenant le produit des distributions conditionnelles des nœud étant donnés leurs parents. Ainsi, la distribution de probabilité jointe du modèle est donnée par :

$$p(C, \pi_l, f_j) = p(\pi_l | C) p(f_j | C, \pi_l, \theta_l)$$

où $f_j$ représente le vecteur caractéristique ($n$ caractéristiques continues) d'une image et $C$ est la variable classe.

Maintenant le modèle peut être complété par les variables discrètes, issues des mesures de forme. Ces variables sont notées $MF\ 1, \ldots, MF\ n$, où $n$ est le nombre de mesures de formes utilisées et $MF_i$ représente la valeur de chaque mesure de forme. Des *a priori* de Dirichlet [Robert 97], ont été utilisés pour l'estimation de ces variables. Plus précisément, on introduit des pseudo comptes supplémentaires à chaque instance de façon à ce qu'elles soient toutes virtuellement représentées dans l'échantillon d'apprentissage. Ainsi, chaque instance, même si elle

n'est pas représentée dans l'échantillon d'apprentissage, aura une probabilité non nulle. Comme les variables continues correspondant aux descripteurs de forme, les variables discrètes correspondant aux mesures de forme sont incluses dans le réseau en les connectant à la variable classe.

Notre classificateur peut alors être décrit par la figure 6.7. La variable latente $\alpha$ montre qu'un *a priori* de Dirichlet a été utilisé. La boîte englobante autour de la variable *MF* indique $n$ répétitions de *MF*, pour chaque mesure de forme. G

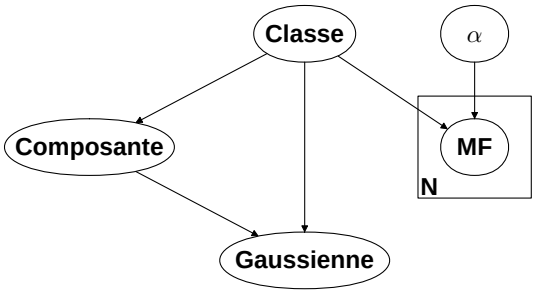



FIGURE 6.7 – Modèle de mélange de lois de Bernoulli et de mélanges de Gaussiennes, pour la combinaison de descripteurs

Ce classificateur signifie que les caractéristiques continues et discrètes représentant les images sont supposées avoir été générées conditionnellement à la même classe. Par conséquent, les paramètres des mélanges de Gaussiennes et des lois de Bernoulli résultants doivent correspondre : concrètement, si une image, représentée par des descripteurs visuels continus, a une grande probabilité pour une certaine classe, alors ses mesures de forme discrètes doivent avoir une grande probabilité pour cette même classe.

#### 6.6.3.1 Classification

Pour classifier une image requête $f_j$, le nœud classe *Classe* est inféré grâce à l'algorithme de passage de message. Cette image, caractérisée par ses caractéristiques de forme continues $v_{j_1}, \ldots, v_{j_m}$ et discrètes $MF\ 1_j, \ldots, MF\ k_j$ est considérée comme une « évidence »représentée par :

$$P(f_j) = P(v_{j_1}, \ldots, v_{j_m}, MF\ 1_j, \ldots, MF\ n_j) = 1$$

quand le réseau est évalué. Grâce à l'inférence, les probabilités de chaque nœud sont mises à jour en fonction de cette évidence. Après propagation, on connaît, $\forall i \in \{1, \ldots, k\}$, les probabilités *a posteriori*

$$P(c_i|f_j) = P(c_i|v_{j_1}, \ldots, v_{j_m}, MF\ 1_j, \ldots, MF\ n_j)$$

La requête $f_j$ est affectée à la classe $c_i$ qui maximise cette probabilité.

## 6.7 Évaluation et résultats

### 6.7.1 Données

Nous avons utilisé des symboles de la base GREC pour effectuer nos tests [Valveny 04]. Cette base (voir figure 6.8), a été créée spécialement dans le cadre du concours de reconnaissance de symboles à GREC' 2005.

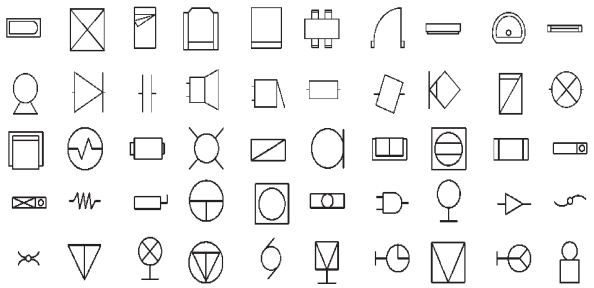


FIGURE 6.8 – Base de symboles GREC.

Les symboles de cette base sont issus principalement de deux domaines d'application, architecture et électronique, car ces symboles sont le plus largement utilisés par les équipes et représentent un grand nombre de formes différentes. Cette base constitue une base de référence dans le domaine (benchmark).

Cinquante modèles différents ont été utilisés, plusieurs d'entre eux ayant des formes similaires. On dispose donc d'une base de 50 symboles parfaits, sur lesquels nous avons effectué des dégradations basées sur le modèle Kanungo [Kanungo 00]. Ces dégradations sont similaires au bruit obtenu quand un document est scanné, imprimé ou photocopié (bruits de type global et local et fermeture morphologique). Nous avons aussi appliqué aux symboles des rotations de différents angles et différents zooms, de façon à obtenir, dans un premier temps, une base de 3600 symboles, constituée de 72 images différentes par modèle.

Ensuite, la base initiale de 3600 images a été étendue à une base de 5400 images en générant aléatoirement des « occlusions »sur la moitié des images de chaque classe de la base initiale. Ces « trous »sont de différentes tailles et leurs positions dans les images ont été choisies aléatoirement. Nous disposons maintenant d'une plus grande base d'images plus bruitées, composée de 108 images par classe. Par exemple, la figure 6.9 présente 4 symboles parfaits (première colonne) et 5 images bruitées et présentant des « trous », dérivées de ces modèles.

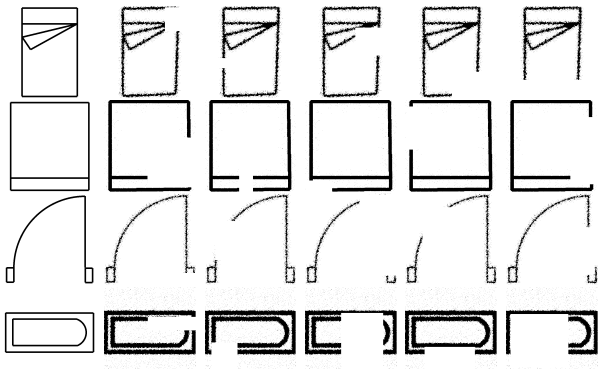


FIGURE 6.9 – Exemples de symboles bruités et présentant des « occlusions »pour différents modèles

### 6.7.2 Protocole expérimental

Sur la première base de 3600 images, nous avons défini plusieurs tests d'apprentissage : nous avons constitué des échantillons d'apprentissage et de test variés et de différentes tailles. Nous avons testé la méthode en effectuant diverses combinaisons des trois mesures de forme et des signatures issues de trois descripteurs de forme décrits section 5.3 : le Generic Fourier Descriptor (GFD), le descripteur Zernike et la $\mathcal{R}$-signature $1D$ pour les descripteurs et la compacité, la rectangularité et l'ellipticité pour les mesure de forme.

Nous avons évalué notre méthode en procédant à une validation croisée, en utilisant 75% de la base d'images pour l'apprentissage, et les 25% pour les tests. Les tests sont répétés 4 fois de façon à ce que chaque image de la base soit utilisée pour l'apprentissage et pour les tests. Afin d'évaluer nos résultats de classification, nous avons choisi de calculer le taux de reconnaissance, *i. e.* le taux de bonne classification, correspondant au ratio entre le nombre d'images bien classées et le nombre d'images classifiées. Le taux de reconnaissance pour la validation croisée est obtenu en faisant la moyenne des taux de reconnaissance des 4 tests.

Sur la deuxième base de 5400 images, notre méthode a été évaluée en procédant à 3 validations croisées, dont chaque échantillon d'apprentissage représente 25%, 50% et 75% de la base, les 75%, 50% et 25% restants (respectivement) étant utilisés pour les tests. En effet, cette base étant plus grande que la précédent, il nous a paru plus judicieux de procéder à plusieurs validations croisées, en faisant varier la taille des échantillons d'apprentissage, dans le but d'éviter le « sur »ou le « sous »apprentissage. Dans chaque cas les tests sont répétés 10 fois de façon à ce que chaque image de la base soit utilisée pour l'apprentissage et les tests. Pour chaque taille d'échantillon d'apprentissage, le taux de reconnaissance est obtenu en prenant la moyenne des taux de reconnaissance des 10 tests.

De plus, le descripteur *HRT* a été calculé sur l'ensemble des images de la nouvelle base. Pour l'intégrer dans nos modèles, nous avons concaténé les vecteurs colonnes de la matrice *HRT*. Or, chaque vecteur colonne de *HRT* correspond à un histogramme calculé sur la matrice de Radon pour une valeur du paramètre angulaire $\theta$ donnée. En faisant cette concaténation, on perd l'information de la structure matricielle, et, en particulier, l'information sur le paramètre angulaire. Cette perte d'information explique le fait qu'il y ait moins de différence entre les résultats du descripteur *HRT* et ceux de la $\mathcal{R}$-signature $1D$, avec nos modèles (voir section 6.7.3), que dans les résultats présentés dans la section 6.3.1.4. De ce fait, les résultats présentés dans ce chapitre, et les chapitres 7 et 8 ne présenteront pas tous les performances du descripteur *HRT*.

Enfin, sur cette base de 5400 images, le Naïve Bayes, le Naïve Bayes augmenté (TAN) et le Multinets, ont été utilisés après discrétisation des variables continues issues des descripteurs de forme, avec la méthode de discrétisation [Colot 04] présentée dans la section 6.5.1.

Dans cette section on souhaite montrer que la combinaison de descripteurs et la sélection de variables améliorent la classification. Dans nos expériences nous allons donc comparer :

- la classification après sélection de variables avec la méthode du LASSO, à la classification sans sélection de variables automatique, avec sélection aléatoire et avec une méthode courante de sélection de variables,
- la classification en combinant 2 ou 3 descripteurs de forme (GFD, Zernike et la $\mathcal{R}$-signature $1D$ ou *HRT*) à la classification avec un seul descripteur de forme,
- la classification en combinant des caractéristiques discrètes et continues à la classification avec seulement les caractéristiques continues,
- la classification en combinant 3 descripteurs continus avec deux méthodes de l'état de l'art.

### 6.7.3 Résultats

**Base de 3600 images**

Tout d'abord, on peut remarquer que la réduction de dimension des vecteurs caractéristiques améliore le taux de reconnaissance pour tous les classificateurs. De plus, la sélection de variables avec le LASSO nous a permis de diminuer significativement le nombre de variables. Le tableau 6.1 montre le nombre de variables moyen sélectionnées pour chaque descripteur avec la méthode du LASSO, comparé à la méthode Sequential Forward Selection (SFS method) [Pudil 94] (*cf.* section 6.5.2). On peut voir que le LASSO nous a permis de sélectionner moins de variables que la méthode *SFS* (voir tableau 6.1).

Le tableau 6.2 montre le taux de reconnaissance en fonction des différentes méthodes de sélection de variables, en combinant 3 descripteurs avec notre modèle de mélange de lois de Bernoulli et de mélanges de Gaussiennes (GM-B) et deux classificateurs de l'état de l'art : un classificateur SVM classique [Chang 01], et le $k$ plus proches voisins « flou »(FKNN) [Keller 85]. Les taux de reconnaissance pour ces trois classificateurs sans sélection de variable et après sélection d'un sous-ensemble de variables avec les méthodes *SFS* ou LASSO ou avec des sélections aléatoires du même nombre de variables que celui obtenu avec le LASSO, sont comparés. Le FKNN a été calculé avec $k = 1$ et $k = m$ où $m$ est le nombre moyen d'images par classe dans l'échantillon d'apprentissage.

Les résultats du tableau 6.2 montrent que la sélection de variables avec le LASSO améliore le taux de reconnaissance de 8.7% en moyenne comparée à la classification sans sélection de variables, de 5.3% en moyenne comparée à la sélection aléatoire et de 1.8% en moyenne comparée à la sélection avec SFS. Ainsi, la LASSO s'est montré plus robuste sur cette base et d'un point de vue expérimental, que la méthode SFS. En effet, les méthodes de rétrécissement comme la LASSO sont réputées pour être plus stables que les méthodes itératives comme SFS, pour sélectionner des variables dans un large ensemble de variables et avec peu d'exemples. Ainsi, dans la suite de cette section, les variables seront sélectionnées, avec la méthode du LASSO, avant d'opérer la classification.

| Nombre de variables de | GFD | Zernike | $\mathcal{R}$-signature $1D$ |
|---|---|---|---|
| sans sélection | 225 | 34 | 180 |
| SFS | 141 | 34 | 48 |
| LASSO | 13 | 15 | 13 |

TABLE 6.1 – Nombre moyen de variables en fonction de la méthode de sélection de variables

| Méthode de sélection de variables | SVM | FKNN $k = 1$ | FKNN $k = m$ | GM-B |
|---|---|---|---|---|
| Sans sélection | 87.6 | 89.9 | 88.6 | 89.8 |
| Sélection aléatoire | 90.8 | 93.7 | 91.5 | 93.3 |
| SFS | 94.1 | 97.2 | 95.3 | 96.7 |
| LASSO | 95.7 | 98.8 | 96.2 | 100 |

TABLE 6.2 – Taux de reconnaissance moyens (en %) pour les classificateurs SVM, FKNN et GM-B en fonction de la méthode de sélection de variables

Considérons maintenant le tableau 6.3. La notation G (respectivement Z et R) signifie que

le descripteur GFD (respectivement les descripteurs Zernike et la $\mathcal{R}$-signature $1D$) a été utilisé. L'opérateur « + »indique que les descripteurs représentés par les opérandes sont combinés.

Les taux de reconnaissance confirment que la combinaison de 2 ou 3 descripteurs implique une meilleure classification qu'avec un seul de ces descripteurs. En effet, on observe que la combinaison de 2 descripteurs augmente le taux de reconnaissance de 18% en moyenne comparé à l'utilisation d'un seul descripteur. De plus, on peut noter que la combinaison de 3 descripteurs est meilleure, de 18.3% en moyenne, à l'utilisation d'un seul d'entre eux. D'autre part, même si l'on obtient un taux de reconnaissance élevé avec le descripteur de Zernike, le taux de reconnaissance ne diminue pas si on combine ce descripteur avec un ou deux autres descripteurs, quels que soient ces descripteurs, et même s'ils ont un faible taux de reconnaissance (c'est le cas de la $\mathcal{R}$-signature $1D$), *i. e.* que le mauvais comportement d'un descripteur ne pénalise pas le comportement des autres descripteurs auxquels on le combine.

| G | Z | R | G+Z | G+R | Z+R | G+Z+R |
|---|---|---|---|---|---|---|
| 99 | 100 | 46.1 | 100 | 99.3 | 100 | 100 |

TABLE 6.3 – Taux de reconnaissance moyens (en %) du GM-B après sélection de variables avec le LASSO

Enfin, la dernière ligne du tableau 6.2 montre l'efficacité de notre approche comparée aux classificateurs SVM et FKNN. Les résultats ont été obtenus en combinant les trois descripteurs et après sélection de variables avec le LASSO. Il apparait que les résultats du modèle proposé GM-B sont toujours meilleurs que ceux du SVM et du FKNN.

**Base de** 5400 **images**

Sur cette base, le LASSO nous a permis de sélectionner environ le même nombre de variables que sur la base intiale : 12 variables en moyenne à partir des caractéristiques de GFD, 13 à partir des caractéristiques de Zernike et 13 à partir des caractéristiques de la $\mathcal{R}$-signature $1D$. Enfin, 83 variables ont été sélectionnées à partir des caractéristiques du descripteur *HRT* calculé sur cette nouvelle base. Considérons maintenant le tableau 6.4. Les notations utilisées sont les mêmes que celles utilisées dans le tableau 6.3. De plus, la notation MF signifie que les trois mesures de forme ont été utilisées. Les taux de reconnaissance montrent l'intérêt de la combinaison de descripteurs. En effet, même si la classification est moins efficace que sur la première base (car la base de 5400 images est plus complexe que la première, du fait des « occlusions »sur certaines images), les résultats montrent que la combinaison des descripteurs continus améliore le taux de reconnaissance. De plus, l'ajout des 3 mesures de formes (caractéristiques discrètes) améliore encore ces résultats. En effet, l'intégration de caractéristiques discrètes améliore le taux de reconnaissance de 3.8% en moyenne comparé au taux de reconnaissance obtenu en combinant les 3 descripteurs continus *GFD*, *Zernike* et $\mathcal{R}$-signature $1D$.

De plus, le tableau 6.4 montre que l'utilisation du descripteur *HRT* à la place de la $\mathcal{R}$-signature $1D$ permet d'améliorer encore le taux de reconnaissance de 0.4% en moyenne.

Enfin, le tableau 6.5 montre que le classificateur proposé GM-B offre de meilleures performances que les classificateurs SVM et FKNN.

De même, le modèle proposé GM-B montre de meilleurs taux de reconnaissance que les classificateurs Bayésiens usuels (Naïve Bayes (BN), Naïve Bayes augmenté (TAN) et Multinets) présentés section 5.6. On remarque que le Naïve Bayes et le TAN ont des performances similaires. Ceci est du au fait que lors de l'apprentissage de structure du TAN, peu d'arcs ont été créés entre

les variables caractéristiques. Aussi la structure du réseau est quasi identique à celle du naïve Bayes. On remarque que le Naïve Bayes ont un comportement similaire au FKNN. De même, les performances du Naïve Bayes et du TAN sont proches de celles du SVM. Cependant, le SVM présente des taux de reconnaissance légèrement supérieurs lorsque 25% et 50% de la base sont utilisés pour l'apprentissage, alors que les modèles graphiques probabilistes sont plus performants lorsque la taille de la base d'apprentissage passe à 75% de la base totale. Ceci confirme le fait que les SVM sont moins efficaces en présence de beaucoup de données d'apprentissage, à la différences des modèles graphiques probabilistes. Enfin, le Multinets se montre plus performant que le Naïve Bayes et le TAN.

| apprentissage | G | Z | R | G+Z | G+R | Z+R | G+Z+R | G+Z+R+MF | G+Z+*HRT*+MF |
|---|---|---|---|---|---|---|---|---|---|
| 25% | 70,4 | 79 | 39,3 | 85,5 | 75,5 | 82,2 | 93,3 | 96,8 | 97,5 |
| 50% | 71 | 80,7 | 40,2 | 87,6 | 76,3 | 83,4 | 93,7 | 98,6 | 98,8 |
| 75% | 75,7 | 85,1 | 41,2 | 89,4 | 79,1 | 87,6 | 96,2 | 99,2 | 99,5 |

TABLE 6.4 – Taux de reconnaissance moyens (en %) du modèle GM-B après sélection de variables avec le LASSO - base de 5400 images

| apprentissage | SVM | FKNN $k = 1$ | FKNN $k = m$ | BN | TAN | Multinets | GM-B |
|---|---|---|---|---|---|---|---|
| 25% | 89,2 | 91,9 | 91,7 | 88,8 | 89,1 | 95,3 | 96,8 |
| 50% | 91 | 95,2 | 93 | 90,1 | 90,7 | 97,2 | 98,6 |
| 75% | 92,5 | 97,1 | 94,7 | 93,6 | 94,6 | 98,6 | 99,2 |

TABLE 6.5 – Taux de reconnaissance moyens (en %), en combinant les caractéristiques continues et discrètes (G+Z+R+MF), avec le SVM, le FKNN, le Naïve Bayes, le Naïve Bayes augmenté (TAN), le Multinets et le modèle GM-B après sélection de variables avec le LASSO - base de 5400 images

Le tableau 6.6 montre les valeurs maximales et minimales, ainsi que la moyenne et l'écart-type des taux de reconnaissances obtenus par les 3 classificateurs comparés, durant les 10 tests et pour un apprentissage sur 50% de la base. L'écart-type est faible, quel que soit le classificateur utilisé, et montre une faible variabilité du taux de reconnaissance en fonction des différents échantillons d'apprentissage et de test.

| Mesure | SVM | FKNN $k = 1$ | FKNN $k = m$ | GM-B |
|---|---|---|---|---|
| Min | 90.4 | 94.8 | 92.9 | 98.5 |
| Max | 91.7 | 95.4 | 93.03 | 98.65 |
| Moyenne | 91 | 95.2 | 93 | 98.6 |
| Ecart-type | 0.4 | 0.17 | 0.04 | 0.07 |

TABLE 6.6 – Mesures statistiques (en %) sur les taux de reconnaissance des classificateurs SVM, FKNN et GM-B, après sélection de variables avec le LASSO, en combinant des caractéristiques continues et discrètes (G+Z+R+MF) - base de 5400 images (échantillon d'apprentissage = 50% de la base)

Enfin, Le tableau 6.7 montre les temps CPU du SVM, du FKNN et du modèle proposé, pour

les phases d'apprentissage et de test, dans les même conditions expérimentales que celles du tableau 6.5. Toutes les expérimentations ont été menées avec un processeur Intel Core 2 Duo 2,40 GHz 2,40 Ghz, 2 Go RAM, Windows. Les trois classificateurs ont été exécutés avec Matlab©. Si on considère uniquement les phases de test (l'apprentissage étant fait hors-ligne pour le SVM et le GM-B), le SVM est plus rapide que les deux autres. Le temps CPU est plus élevé pour le modèle GM-B car il dépend du nombre de Gaussiennes, et de la précision prédéfinie dans l'algorithme EM. Ici, nous avons utilisé deux Gaussiennes. Ce nombre a été déterminé expérimentalement, de façon à réaliser le meilleur compromis entre temps de calcul et taux de reconnaissance. Comme nous pouvons le voir dans la figure 6.10, le mélange à 2 Gaussiennes a offert le meilleur taux de reconnaissance. Les tests de comparaison ont été effectués en utilisant 50% de la base pour l'apprentissage et de 1 à 10 Gaussiennes. Les taux présentés sont les moyennes obtenues après validation croisée.

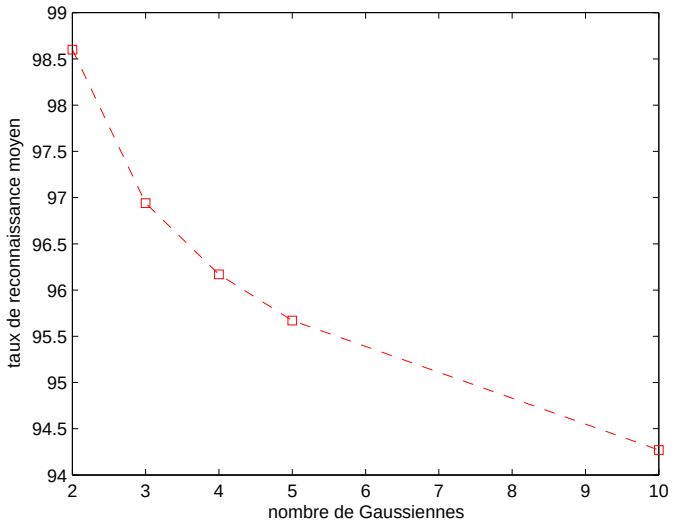



FIGURE 6.10 – Taux de reconnaissance moyen en fonction du nombre de Gaussiennes

On remarquera, que, malgré un temps CPU supérieur pour le modèle GM-B, ce temps reste inférieur à $0,03$ s par image.

Finalement, comme nous l'avons évoqué dans la section 6.5.2.1, on peut remarquer que la sélection de variables permet de diminuer considérablement les temps de calcul du classificateur GM-B.

| Apprentissage | SVM | | FKNN $k=1$ | FKNN $k=m$ | GM-B avec LASSO | | GM-B sans sélection | |
|---|---|---|---|---|---|---|---|---|
| | app | test | | | app | test | app | test |
| 25% | 4 | 5 | 40 | 41 | 58 | 78 | 2726 | 25608 |
| 50% | 10 | 6 | 56 | 58 | 117 | 52 | 5696 | 17291 |
| 75% | 19 | 4 | 42 | 45 | 168 | 24 | 8110 | 8453 |

TABLE 6.7 – Temps CPU (en secondes), du SVM, du FKNN et du GM-B. Les temps sont donnés pour la classification de toutes les images test

## 6.8 Conclusion

Dans ce chapitre, nous avons montré qu'un simple classificateur standard, le Naïve Bayes, associé à une méthode de sélection de variables originale, le LASSO, et une méthode de discrétisation des variables continues adaptées, s'est montré compétitif avec des classificateurs courants comme le SVM et le FKNN. Nous avons aussi remarqué que la sélection de variables avec le LASSO améliore également les résultats du SVM, pourtant réputé pour son efficacité en présence de caractéristiques de grande dimension. De plus, l'adaptation du LASSO nous a permis de résoudre notre problème de dimensionnalité et ainsi de diminuer la complexité des modèles graphiques, tout en améliorant le taux de reconnaissance.

Nous avons montré que la combinaison de caractéristiques, associée à la méthode de sélection de variables LASSO, améliore le taux de reconnaissance, quel que soit le classificateur utilisé. En effet, la combinaison fournit un classificateur plus robuste à la variabilité et au passage à l'échelle.

Enfin, nous avons proposé un réseau Bayésien, permettant de représenter des symbole sur la base de caractéristiques discrètes et continues. Nous avons montré que la combinaison de caractéristiques discrètes et continues améliore le taux de reconnaissance, par rapport à l'utilisation de données continues uniquement. De plus, ce classificateur s'est montré compétitif avec les classificateurs SVM et FKNN couramment utilisés dans l'État de l'art. De plus, le classificateur proposé a donné de meilleurs résultats que les classificateurs Bayésiens standards.

# Chapitre 7

# Classification et annotation d'images de scènes naturelles

## 7.1 Contexte

Dans ce chapitre, nous nous intéressons à un problème un peu plus général que le précédent : la reconnaissance d'images naturelles. Cette discipline est également au cœur de la reconnaissance de formes. Ici, on souhaite reconnaître le ou les objets contenus dans les images, un objet correspondant à une classe. Cette tâche peut être assimilée à de la classification supervisée et peut être résolue en utilisant une méthode d'apprentissage supervisée, à partir d'un sous-ensemble de la base pour lequel la classe est connue pour chaque image.

## 7.2 Combinaison d'information visuelle et sémantique

Comme nous l'avons vu dans le chapitre 2, l'indexation textuelle est efficace lorsqu'elle est manuelle, mais dans ce cas elle est très coûteuse pour l'utilisateur. Quant à l'indexation visuelle, elle est efficace sur certaines bases d'images, mais ses performances décroissent sur des bases d'images plus générales, comme les bases d'images naturelles auxquelles on s'intéresse dans cette section. Pour pallier ces problèmes, nous avons vu qu'une solution semble d'être de combiner ces deux types d'information : visuelle et sémantique. De plus, afin de réduire le coût que représente l'annotation manuelle d'une base d'images dans sa totalité, on recherche une méthode permettant de traiter les données manquantes, et qui serait ainsi efficace sur des bases partiellement annotées. Nous considérons qu'une base est partiellement annotée si un sous-ensemble des images ne possède pas d'annotations, ou si certaines images sont partiellement annotées. De même, nous considérons une image comme *partiellement annotée* si elle ne possède pas le nombre maximal de mots-clés disponibles par image dans la vérité-terrain.

Afin de résoudre ce problème, nous nous sommes de nouveau orientés vers les modèles graphiques probabilistes, car ils permettent justement de combiner différents types d'informations, et de traiter les données manquantes. Aussi, dans les sections ci-dessous, nous présentons les descripteurs que nous avons choisi de combiner (section7.3). Dans la section 7.4, nous justifions le choix des modèles graphiques probabilistes, avant de présenter, dans les sections 7.5 et 7.6, les modèles que nous avons proposés. La section 7.7 est dédiée à l'évaluation de ces méthodes. Enfin, dans la section 7.8, nous concluons sur ces modèles.

## 7.3 Choix des caractéristiques à utiliser

Avant de construire un classificateur à partir d'échantillons d'apprentissage, il convient d'étudier les données disponibles. Dans le cadre de la reconnaissance d'images de scènes naturelles, nous manipulons des images de toute sorte : images couleur, noir et blanc et accompagnées de mots-clés, ou non. Ces images peuvent être décrites par des vecteurs caractéristiques, obtenus à partir de descripteurs de forme, de descripteurs de couleur ou de texture, comme nous l'avons vu au chapitre 2.

Nous avons pour l'instant utilisé deux descripteurs de forme présentés précédemment, la $\mathcal{R}$-signature $1D$ et le descripteur $HRT$, ainsi qu'un descripteur de couleur. En effet, nous avons vu que la couleur est un attribut important en reconnaissance d'images [Swain 91]. L'humain ne semble pas affecté par de petites variations de couleur comme par les variations de valeurs de niveaux de gris. Aucun descripteur de texture n'a été utilisé, mais il serait facile d'en intégrer un (ou plusieurs). De même on pourrait facilement intégrer un ou plusieurs autres descripteurs de forme et de couleur : il suffirait de concaténer les vecteurs caractéristiques. Enfin, l'information textuelle portée par les mots-clés, quand ils sont disponibles, est aussi utilisée.

Le descripteur de couleur utilisé est un simple histogramme des composantes RGB, présenté ci-dessous.

### 7.3.1 Histogramme des composantes RGB

Comme on l'a vu dans le chapitre 1, il existe un grande nombre de modes de représentation de l'espace des couleurs (par exemple RGB et HSI). L'espace RGB a été largement utilisé grâce à la grande disponibilité d'images au format RGB à partir d'images scannées. Quel que soit l'espace de représentation, l'information couleur d'une image peut être représentée par un seul histogramme $3D$ ou 3 histogramme $1D$ [Swain 91]. Ces modes de représentation de la couleur ont l'avantage d'être invariants à la translation et à la rotation. De plus une simple normalisation de l'histogramme fournit aussi l'invariance à l'échelle : notons $H_i$ l'histogramme d'une image, où l'indice $i$ représente un intervalle de l'histogramme. Alors l'histogramme normalisé $I$ est défini par :

$$I(i) = \frac{H(i)}{\sum_i H(i)}$$

Finalement, pour chaque image, le vecteur de caractéristiques couleur correspond à la concaténation des 3 histogrammes $1D$ normalisés (un pour chaque composante R,G et B), de 16 valeurs chacun.

On obtient ainsi un vecteur caractéristique de 48 valeurs par image.

Dans la figure 7.1 nous présentons deux images provenant de la même classe (classe « fleurs ») et leurs histogrammes des composantes RGB. Sur l'histogramme de la première image (la rose blanche), on aperçoit distinctement 3 pics, un par intervalle de 16 valeurs, *i. e.* un par composante R, G et B. L'histogramme est très net car on distingue très clairement deux couleurs dans l'image, correspondant chacune au blanc de la rose et au noir du fond.

Par contre l'histogramme de la deuxième image est moins « interprétable ». En effet, l'image présente plus de couleurs que la première et surtout, ces couleurs ne permettent pas de distinguer facilement un objet ou une zone particulière dans l'image. Typiquement, on va pouvoir distinguer, grâce aux deux pics plus hauts que les autres dans l'histogramme, deux couleurs majoritaires : le bleu du ciel et le rouge des coquelicots.

Le caractère discriminant de ce descripteur est donc vraiment dépendant des images.

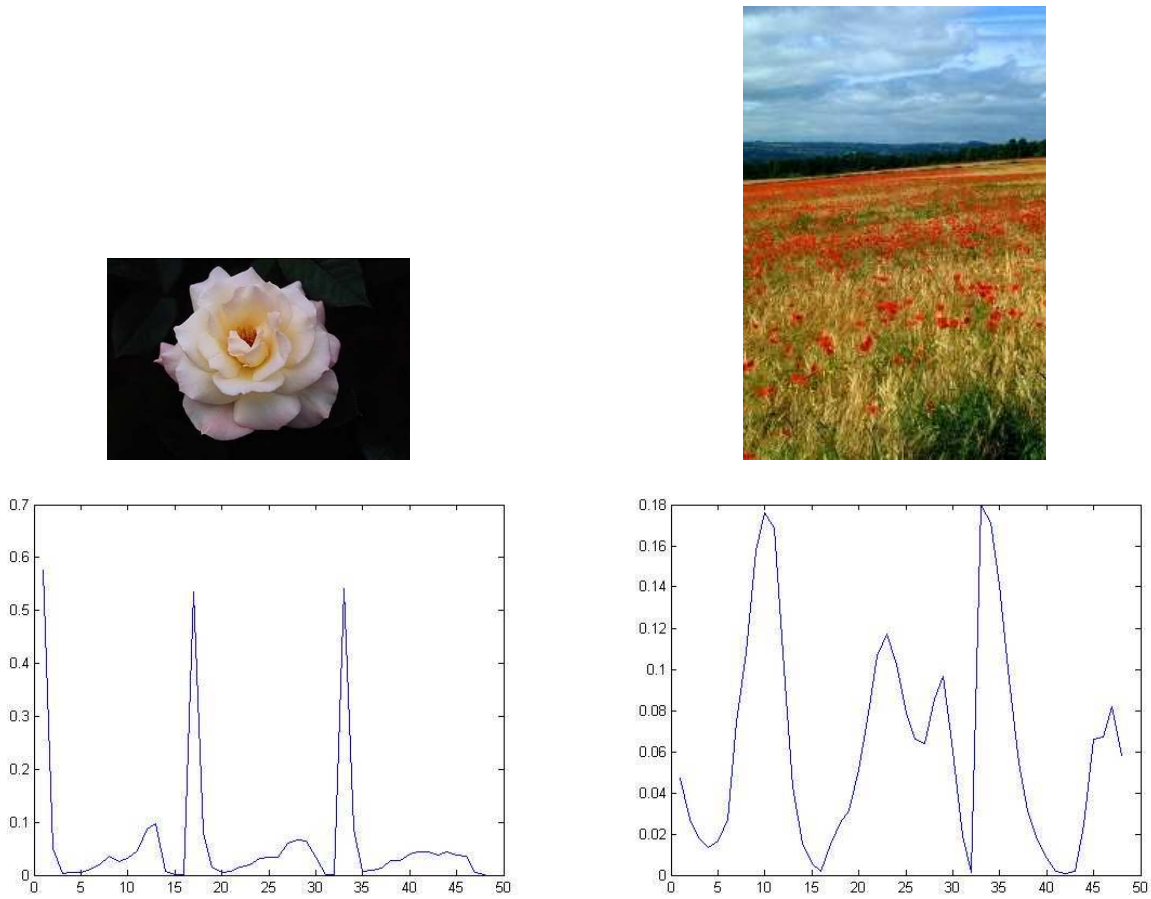



FIGURE 7.1 – L'histogramme des composantes RGB

## 7.4 Combinaison d'information visuelle et sémantique avec des modèles graphiques probabilistes

Nous avons vu que notre problème de reconnaissance d'images naturelles peut être vu comme un problème de classification d'images. Nous avons vu aussi (dans le chapitre 3) que les modèles probabilistes sont efficaces pour résoudre ce genre de problème, en particulier le simple Naïve Bayes.

Cependant, ici, on doit à la fois combiner des données continues (caractéristiques issues des descripteurs de forme et couleur) et discrètes (mots-clés éventuellement associés aux images). De plus, le modèle proposé doit pouvoir gérer les données manquantes (dans les cas où les images ne sont pas annotées). Le Naïve Bayes n'est alors plus suffisant. Mais, les modèles graphiques probabilistes permettent de résoudre ce genre de problèmes.

Aussi, dans les sous-sections suivantes, nous présentons les différents modèles que nous avons proposés.

## 7.5 Modèle de mélange GM-Mult

### 7.5.1 Définition du modèle

Nous présentons un modèle hiérarchique probabiliste multimodal (images et mots-clés associés) pour classifier de grandes bases de données d'images annotées. Nous rappelons que les caractéristiques visuelles sont considérées comme des variables continues, et les éventuels mots-clés associés comme des variables discrètes. De plus, on considère que notre échantillon de caractéristiques visuelles suit une loi dont la fonction de densité est une densité de mélange de Gaussiennes. Les variables discrètes sont supposées suivre une distribution multinomiale sur le vocabulaire des mots-clés. Le vocabulaire peut être constitué de deux manières : soit il correspond à l'ensemble des termes observés dans les données, soit c'est un vocabulaire contrôlé qui a été établi avant même la construction de la base (les notions de vocabulaire libre et contrôlé sont définies dans le chapitre 2).

- Notons $T_i, \forall i \in \{1, \ldots, N\}$ chaque terme d'un vocabulaire de taille $N$
- Soit $\{KW_1, KW_2, \ldots, KW_n\}$ l'ensemble des mots-clés d'une image. $n$ désigne le nombre maximal de mots-clés pour une image.
- Chaque variable $KW_j, \forall j \in \{1, \ldots, n\}$ peut être réprésentée un vecteur booléen dans l'espace des $N$ termes du vocabulaire : on a $KW_j = \{m_1, m_2, \ldots, m_N\}$, où $m_i = 0\ ou 1, \forall i \in \{1, \ldots, N\}$ et $\sum_{i=1}^{N} m_i = k$.

Chaque variable $KW_j, \forall j \in \{1, \ldots, n\}$ suit une loi multinomiale de paramètres $(k, p_1, p_2, \ldots, p_N)$, où $p_i, \forall i \in \{1, \ldots, N\}$ est la probabilité associée à chaque valeur $m_i$ si la distribution de $KW_j$ est la suivante :

$$p(m_1 = p_1, \ldots, m_N = p_N) = \frac{k!}{m_1! m_2! \ldots m_N!} p_1^{m_1} p_2^{m_2} \ldots p_N^{m_N}$$

Cette distribution est une distribution multinomiale.

Nous proposons d'étendre le Naïve Bayes afin de prendre en compte ces distributions de probabilités : le modèle proposé est un modèle de mélange de lois multinomiales et de densités à mélange de Gaussiennes (noté « modèle de mélange GM-Mult »). La structure du Naïve Bayes est conservée c'est-à-dire que l'on dispose d'une variable « Classe », connectée à chaque variable caractéristique (*cf.* figure 7.3).

Comme dans le modèle de mélange de lois de Bernoulli et de mélanges de Gaussiennes du chapitre proposé dans le chapitre 6, le cœur du modèle présenté ici est le modèle probabiliste représenté ci-dessous et expliqué section 6.6 :

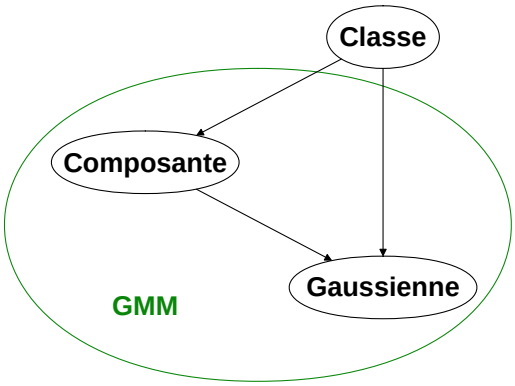



FIGURE 7.2 – GMMs représentés par un modèle graphique probabiliste

Maintenant le modèle peut être complété par les variables discrètes, notées $KW\ 1, \ldots, KW\ n$, correspondant aux éventuels mots-clés associés aux images. Des *a priori* de Dirichlet [Robert 97], ont été utilisés pour l'estimation de ces variables. Plus précisément, on introduit des pseudo comptes supplémentaires à chaque instance de façon à ce qu'elles soient toutes virtuellement représentées dans l'échantillon d'apprentissage. Ainsi, chaque observation, même si elle n'est pas représentée dans l'échantillon d'apprentissage, aura une probabilité non nulle. Comme les variables continues correspondant aux caractéristiques visuelles, les variables discrètes correspondant aux mots-clés sont incluses dans le réseau en les connectant à la variable classe.

Notre classificateur peut alors être décrit par la figure 7.3. La variable latente $\alpha$ montre qu'un *a priori* de Dirichlet a été utilisé. La boîte englobante autour de la variable $KW$ indique $n$ répétitions de $KW$, pour chaque mot-clé.

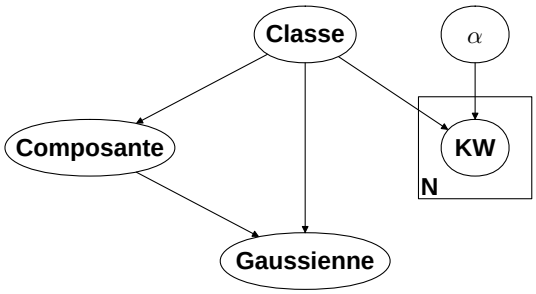



FIGURE 7.3 – Modèle de mélange GM-Mult

### 7.5.2 Classification

Pour classifier une nouvelle image $f_j$, le nœud classe *Classe* est inféré grâce à l'algorithme de passage de messages [Kim 83]. Ainsi, une image requête $f_j$, représentée par ses caractéristique visuelles $v_{j_1}, \ldots, v_{j_m}$ et ses éventuels mots-clés $KW\ 1_j, \ldots, KW\ n_j$, est considérée comme une observation (aussi appelée « évidence ») représentée par :

$$P(f_j) = P(v_{j_1}, \ldots, v_{j_m}, KW\ 1_j, \ldots, KW\ n_j) = 1$$

quand le réseau est évalué. En effet, une évidence correspond à une information nous donnant avec une certitude absolue la valeur d'une variable, d'où la probabilité égale à 1.

Grâce à l'algorithme d'inférence (*i. e.* l'algorithme de passage de messages [Kim 83]), les probabilités de chaque nœud sont mises à jour en fonction de cette évidence. On parle de « propagation de croyance ou de propagation de l'évidence ». Il s'agit de la phase de calcul probabiliste à proprement parler où les nouvelles informations concernant les variables observées sont propagées à l'ensemble du réseau, de manière à mettre à jour l'ensemble des distributions de probabilités du réseau. Ceci se fait en passant des messages contenant une information de mise à jour entre les nœuds du réseau. A la fin de cette phase, le réseau contiendra la distribution de probabilité sachant les nouvelles informations.

Après la propagation de croyances, on connaît donc, $\forall i \in \{1, \ldots, k\}$, la probabilité *a posteriori* :

$$P(c_i|f_j) = P(c_i|v_{j_1}, \ldots, v_{j_m}, KW\ 1_j, \ldots, KW\ n_j)$$

L'image requête $f_j$ est affectée à la classe $c_i$ maximisant cette probabilité.

### 7.5.3 Extension d'annotation

Étant donnée une image sans mot-clé, ou partiellement annotée, le modèle proposé peut être utilisé pour calculer une distribution des mots-clés conditionnellement à une image et ses éventuels mots-clés existants. En effet, pour une image $f_j$ annotée par $k, \forall k \in \{0, \ldots, n\}$ mots-clés, où $n$ est le nombre maximum de mots-clés par image, l'algorithme d'inférence permet de calculer la probabilité *a posteriori* $P(KW\ i_j | f_j, KW\ 1_j, \ldots, KW\ k_j), \forall i \in \{k+1, \ldots, n\}$. Cette distribution représente une prédiction des mots-clés manquants d'une image. Pour chaque annotation manquante, le mot-clé du vocabulaire ayant la plus grande probabilité est retenu, si cette probabilité atteint un certain seuil. Ce seuil a été défini à 0.5 dans nos expérimentations. Ainsi, toutes les images ne seront pas annotées par le même nombre de mots-clés à l'issue de l'extension automatique d'annotations.

Par exemple, considérons le tableau 7.1 présentant 3 images avec leurs éventuels mots-clés existants et les mots-clés obtenus après l'extension automatique d'annotations. La première image, sans mot-clé, a été automatiquement annotée par deux mots-clés appropriés. De même, la seconde image, annotée au départ par deux mots-clés, a vu son annotation s'étendre à trois mots-clés. Le nouveau mot-clé, « coucher de soleil »est approprié. Enfin, la troisième image, initialement annotée par un mot-clé, a été complétée par deux nouveaux mots-clés. Le premier nouveau mot-clé, « nuage », est correct. Par contre, le second, « coucher de soleil », ne convient pas. Cette erreur est due au grand nombre d'images de la base annotées par les trois mots-clés « pont », « nuage »et « coucher de soleil »(donc la probabilité jointe de ces trois mots-clés est grande), et à l'algorithme d'inférence.

Nous verrons de façon plus complète, dans la partie expérimentale (section 7.7), l'intérêt d'étendre automatiquement des annotations.

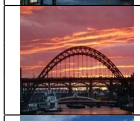
| image | mots-clés initiaux | mots-clés après extension automatique d'annotations |
|---|---|---|
|  | | pont<br>eau |
|  | pont<br>nuage | pont<br>nuage<br>coucher de soleil |
|  | pont | pont<br>nuage<br>coucher de soleil |

TABLE 7.1 – Exemple d'images et de leurs éventuels mots-clés, avant et après extension automatique d'annotations, en utilisant le modèle GM-Mult

## 7.6 Modèle de mélange GM-B

Dans cette section, nous proposons une amélioration du modèle GM-Mult que nous venons de présenter. Nous justifions nos choix sur les paramètres de ce nouveau modèle, et expliquons comment l'utiliser pour classer et annoter automatiquement des images.

### 7.6.1 Définition du modèle

Le modèle de mélange GM-Mult est efficace mais présente deux inconvénients : l'ordre des mots-clés annotant une image intervient, *i. e.* qu'une annotation composée des deux mots-clés (cheval, animal) n'est pas équivalente à l'annotation (animal,cheval) par exemple. De plus, un nombre maximal de mots-clés par annotation doit être fixé lors de la construction du modèle, puisque chaque nœud $KW$ du modèle correspond à un mot-clé composant les annotations.

Afin de pallier ces problèmes, nous proposons d'utiliser le modèle de mélange de lois de Bernoulli et de mélanges de Gaussiennes (GM-B) proposé pour la reconnaissance de symboles (dans le chapitre 6). Ce modèle a donc une structure commune à celle du modèle de mélange GM-Mult. On retrouve le sous-modèle correspondant au modèle de mélange de Gaussiennes par classe (présenté section 6.6), ainsi que des variables discrètes correspondant aux mots-clés.

Les différences par rapport au modèle de mélange GM-Mult résident dans le fait, qu'ici, chaque variable discrète $KW$ correspond à un terme du vocabulaire. Ces variables suivent une loi de Bernoulli. En effet, pour une image donnée, chaque variable mot-clé peut prendre deux états : 1, dans le cas où le terme correspondant fait partie de l'annotation de de cette image, et 0 sinon. Une loi de Bernoulli a été préférée a une loi multinomiale car la loi de Bernoulli nous permet de représenter des dépendances entre les variables mots-clés. Ceci n'est pas possible en utilisant la loi multinomiale, car, avec cette dernière, chaque variable discrète correspond à une composante (mot-clé) d'une annotation. Chacune de ces composantes prend ses valeurs dans l'ensemble des termes du vocabulaire et chaque terme se voit associer une probabilité. Ici, nous préférons considérer chaque terme du vocabulaire comme une variable discrète suivant une loi de Bernoulli. Cette loi permet d'associer une probabilité $p$ à la présence du terme correspondant dans l'annotation d'une image. La probabilité associée à l'absence du terme dans l'annotation est donc $1 - p$ (conformément à la définition de cette loi donnée dans la section 6.6). De cette façon, l'ordre des mots-clés dans chaque annotation n'intervient plus et il n'est pas nécessaire de choisir la longueur de chaque annotation (nombre de mots-clés par image) lors de la construction du modèle.

Enfin, le modèle a été amélioré par la prise en compte d'éventuelles relations sémantiques entre les mots-clés du vocabulaire. Ces relations sont représentées par des dépendances (arcs dans le graphe) entre certaines variables discrètes $KW$ (voir figure 7.5). Par exemple, les mots-clés « dog »et « animal »sont clairement dépendants. En effet, ils appartiennent au même groupe de concepts, comme défini dans Wordnet [Fellbaum 98]. Cette dépendance est représentée par un arc orienté du nœud « dog »vers le nœud « animal ». Les mots-clés sont en anglais car ils proviennent de Wordnet, qui est une base lexicale anglaise.

Comme nous venons de le voir, il n'est pas possible de représenter ce type de dépendances dans le modèle GM-Mult, ni dans le modèle GM-Mixture [Blei 03], présenté dans le chapitre 4. En effet, l'utilisation de la loi multinomiale implique d'utiliser une variable discrète multinomiale pour chaque mot-clé composant une annotation. Outre le fait qu'un nombre maximal de mots-clés par image doit être défini à l'avance, cette représentation ne permet pas de représenter des relations entre mots-clés du vocabulaire. Par contre, il serait possible de représenter d'éventuelles relations entre les différents mots composant les annotations. Cependant, de telles relations ne sont pas évidentes, et, pour les définir, il faudrait envisager l'utilisation d'un algorithme d'apprentissage de structure.

Le modèle de mélange de lois de Bernoulli et de mélanges de Gaussiennes est représenté par la figure 7.4. Les variables $KW\ 1, \ldots, KW\ n$ correspondent chacune à un terme du vocabulaire. $n$ désigne la taille du vocabulaire. Aussi la boîte englobante autour de la variable $KW$ représente $n$ répétitions de $KW$, pour chaque terme du vocabulaire. Dans un soucis de clarté, les arcs

représentant les relations sémantiques entre mots-clés n'ont pas été représentés dans la figure 7.4. La figure 7.5 représente plus précisément les variables mots-clés et leurs éventuelles dépendances. Dans cette dernière, tous les nœuds correspondant aux mot-clés n'ont pas été dessinés. De ce fait, seules quelques dépendances entre mots-clés sont représentées. Par exemple, les mots « bird »et « animal »ont une relation sémantique représentée par un arc orienté du nœud « bird »vers le nœud « animal ». De la même façon, un arc est observé entre les nœuds « duck »et « animal »et les nœuds « duck »et « bird ». Des exemples de dépendances entres termes, issues de Wordnet, sont donnés dans le tableau 7.2. La première colonne contient les mots-clés sources d'une dépendance avec un autre mot-clé donné dans le seconde colonne. La troisième colonne donne le type de la relation sémantique, définie dans Wordnet, entre les deux mots-clés de la même ligne. À partir des ces relations sémantiques, nous avons défini, dans notre graphe, une relation de dépendance entre 2 mots-clés du même « synset »(groupe sémantique) comme défini dans Wordnet.

| source de la dépendance | destination | type de la relation sémantique |
|---|---|---|
| autumn | season | hypernyme direct |
| bird | animal | hypernyme hérité |
| buffalo | animal | hypernyme hérité |
| beetle | animal | hypernyme hérité |
| butterfly | animal | hypernyme hérité |
| beach | sand | méronyme de substance |
| cow | animal | hypernyme hérité |
| chicken | animal | hypernyme hérité |
| chicken | bird | hypernyme hérité |
| deer | animal | hypernyme hérité |
| dog | animal | hypernyme hérité |
| duck | animal | hypernyme hérité |
| duck | bird | hypernyme hérité |
| elephant | animal | hypernyme hérité |
| flower | nature | hypernyme hérité |
| goose | animal | hypernyme hérité |
| goose | bird | hypernyme hérité |
| horse | animal | hypernyme hérité |
| leopard | animal | hypernyme hérité |
| lion | animal | hypernyme hérité |
| leaf | nature | hypernyme hérité |
| monkey | animal | hypernyme hérité |
| owl | animal | hypernyme hérité |
| owl | bird | hypernyme hérité |
| penguin | animal | hypernyme hérité |
| penguin | bird | hypernyme hérité |
| pigeon | animal | hypernyme hérité |
| pigeon | bird | hypernyme hérité |
| sheep | animal | hypernyme hérité |
| seal | animal | hypernyme hérité |
| swan | animal | hypernyme hérité |
| swan | bird | hypernyme hérité |
| springtime | season | hypernyme direct |
| summer | season | hypernym direct |
| waterfall | water | hypernyme hérité |
| dinosaur | animal | hypernyme hérité |

TABLE 7.2 – Exemples de relations de dépendances entre mots-clés issues de Wordnet

On rappelle que Wordnet est une grande base de données lexicales anglaises, où des mots (noms, verbes, adjectifs et adverbes) sont classés en ensembles de synonymes cognitifs (appelés « synsets »), chacun exprimant un concept différent. C'est-à-dire que deux mots ayant une relation sémantique appartiennent au même synset.

Ces relations sémantiques sont représentées par des dépendances dans notre modèle, c'est-à-dire par des arcs orientés dans le graphe. De manière générale, un arc est ajouté entre deux mots-clés du même synset. Chaque arc est orienté du mot-clé le plus spécifique (hypernyme), vers

le mot-clé le plus général (hyponyme). Concernant les arcs entre les termes dont un terme est une partie de l'autre, ces arcs sont orientés du terme qui est une partie de l'autre (méronyme), vers le mot-clé qui constitue le « tout »(holonyme). De cette façon, nous représentons les ontologies de Wordnet.

La structure de ce modèle a été établie à la main. Aucun algorithme d'apprentissage de structure n'a été utilisé. De la même façon, les relations sémantiques ont été établies à la main, en prenant chaque couple de mots-clés du vocabulaire et en recherchant dans Wordnet une éventuelle relation entre les termes de chaque couple.

Ce réseau Bayésien (7.4), signifie que chaque image et ses mots-clés sont supposés avoir été générés conditionnellement à la même classe. Par conséquent, les paramètres résultant du modèle de mélange de lois de Bernoulli et de mélanges de Gaussiennes doivent correspondre. Concrètement, si une image, représentée par des descripteurs visuels, a une grande probabilité dans une certaine classe, alors ses mots-clés doivent aussi avoir une grande probabilité dans cette classe. Nous gardons cependant à l'esprit que les notions de « mots-clés »et de « classes »sont différentes. En effet, une base est classée en plusieurs classes. Chaque image d'une base peut être annotée par plusieurs mots-clés. La valeur d'un mot-clé ne détermine pas la classe de l'image.

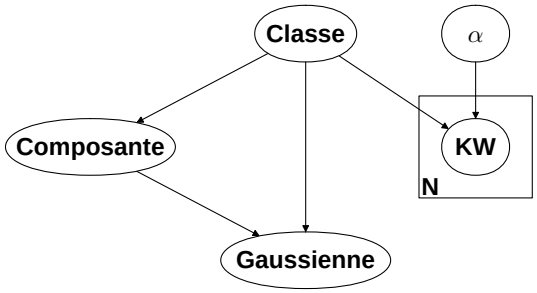



FIGURE 7.4 – Modèle de mélange de lois de Bernoulli et de mélanges de Gaussiennes, pour la combinaison de caractéristiques visuelles et sémantiques

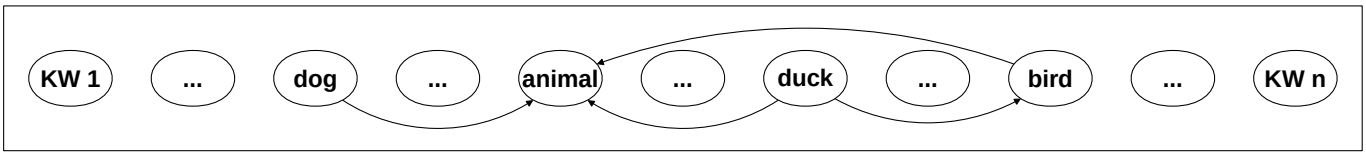



FIGURE 7.5 – Dépendances entre mots-clés

### 7.6.2 Classification

Pour classifier une image requête $f_j$, le nœud classe *Classe* est inféré grâce à l'algorithme de passage de message. Cette image, caractérisée par ses caractéristiques de forme continues $v_{j_1}, \ldots, v_{j_m}$ et discrètes $KW\ 1_j, \ldots, KW\ k_j$ est considérée comme une « évidence »représentée par :

$$P(f_j) = P(v_{j_1}, \ldots, v_{j_m}, KW\ 1_j, \ldots, KW\ n_j) = 1$$

quand le réseau est évalué. Grâce à l'inférence, les probabilités de chaque nœud sont mises à jour en fonction de cette évidence. Après propagation, on connaît, $\forall i \in \{1, \ldots, k\}$, les probabilités *a posteriori*

$$P(c_i|f_j) = P(c_i|v_{j_1}, \ \ldots \ , v_{j_m}, KW\ 1_j, \ \ldots \ , KW\ n_j)$$

La requête $f_j$ est affectée à la classe $c_i$ qui maximise cette probabilité.

### 7.6.3 Extension d'annotation

Étant donnée une image sans mot-clé ou partiellement annotée [14], le réseau Bayésien proposé ci-dessus peut être utilisé pour calculer une distribution de probabilité des mots-clés manquants conditionnellement aux caractéristiques d'une image et ses mots-clés existants. En effet, pour une image requête $f_j$ annotée par un ensemble de $k, \forall k \in \{0, \ \ldots \ , n\}$ mots-clés, notés EKW (pour « Existing KeyWords ») où $n$ est la taille du vocabulaire, l'algorithme d'inférence permet de calculer la probabilité *a posteriori* $P(KW_{i_j}|f_j, EKW)\ \forall KW_{i_j} \notin EKW$. Cette distribution représente la prédiction des mots-clés manquantes pour cette image.

Par exemple, considérons le tableau 7.3 qui présente 4 images et ses éventuels mots-clés existants, ainsi que les mots-clés obtenus après extension d'annotations avec (colonne 3) ou sans (colonne 2) prise en compte des éventuelles relations sémantiques entre mots-clés.

L'annotation de la première image, composée de 3 mots-clés au départ, a été étendue par un mauvais mot-clé. En effet, le bon mot-clé manquant est « shrubs ». Cette erreur est due au grand nombre d'images de la base annotées par les 4 mots « bear », « black », « water »et « grass », ce qui génère une grande probabilité de cet ensemble de mots-clés.

Considérons la seconde image. Son annotation n'a pas été étendue sans prendre en compte les relations sémantiques entre mots-clés. Ceci est à un seuil prédéfini pour sélectionner les mots-clés. En effet, un mot-clé est sélectionné comme annotation si la probabilité de ce mot-clé est strictement supérieure à un seuil, défini à 0.5 dans nos expérimentations. L'annotation de la seconde image n'a pas été étendue car aucune probabilité (pour aucun terme) n'a atteint ce seuil. Au contraire, en prenant en compte les relations sémantiques, la seconde image a été annotée par un mot-clé correct « water », grâce à la relation sémantique existant entre les termes « river »et « water », qui augmente la probabilité du mot-clé « water »étant donné le mot-clé « river ».

Enfin, la troisième et la dernière images appartiennent à la même classe « penguin ». La troisième image est complètement annotée. Au contraire, la quatrième a deux mots-clés manquants. Son annotation a été étendue par le mot-clé incorrect « iceberg »(avec ou sans prise en compte des relations sémantiques). En effet, le troisième mot-clé aurait du être « snow ». Cette erreur est due à la similarité de couleur élevée entre la troisième et la quatrième image. Finalement, grâce aux relations sémantiques, l'annotation de la quatrième image a été étendue par un mot-clé correct « bird ». En effet, la relation sémantique entre les mots-clés « penguin »et « bird »(voir tableau 7.2) augmente la probabilité du mot-clé « bird »étant donné le mot « penguin ».

14. on rappelle que l'on considère une image comme partiellement annotée si son nombre de mots-clés est inférieur au nombre maximal de mots-clés observés pour une image dans la vérité terrain

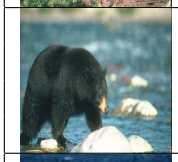
| image | mots-clés initiaux | après annotation sans RS | après annotation avec RS |
|---|---|---|---|
|  | bear<br>black<br>water | bear<br>black<br>water<br>grass | bear<br>black<br>water<br>grass |
|  | bear<br>black<br>river | bear<br>black<br>river | bear<br>black<br>river<br>water |
|  | penguin<br>iceberg<br>water<br>bird | penguin<br>iceberg<br>water<br>bird | penguin<br>iceberg<br>water<br>bird |
|  | penguin<br>water | penguin<br>water<br>iceberg | penguin<br>water<br>iceberg<br>bird |

TABLE 7.3 – Exemples d'images et de mots-clés associés, avant et après extension d'annotations avec ou sans relations sémantiques, en utilisant le modèle GM-B

## 7.7 Évaluation et résultats

### 7.7.1 Modèle de mélange GM-Mult

#### 7.7.1.1 Données

Dans cette section, nous présentons une évaluation de notre modèle sur plus de 3000 images provenant d'Internet, et fournies par Kherfi et al. [Kherfi 04]. Ces images ont été réparties manuellement en 16 classes. Le nombre de classes a été choisi arbitrairement, en fonction des données. Chaque classe contient 230 images. Par exemple, la figure 7.6 présente quatre images de la classe « cheval ».

FIGURE 7.6 – Exemples d'images de la classe « cheval »

Cette base nous a été fournie partiellement annotée : 65% de la base est annotée par 1 mot-clé, 28% par 2 mots-clés et 6% par 3 mot-clés, en utilisant un vocabulaire de 39 mots-clés. La notion de mot-clé est à différencier de la notion de classe. En effet, un même mot-clé peut être présent dans les annotations d'images de classes différentes. Par exemple, le mot-clé nature annote plusieurs images des classes « feuilles », « fleurs »et nature. De même, on pourra trouver le mot-clé « eau »dans les annotations d'images des classes « chutes d'eau », « ponts »et « mer ». Enfin, certains types d'images font partie de deux classes différentes : par exemple certaines

images de la classe « feuilles »peuvent faire partie de la classe « forêts »et *vice versa*. Les images annotées ont été choisies aléatoirement et les mots-clés sont distribués non uniformément, parmi les images et les classes : c'est-à-dire que toutes les images ne sont pas annotées par le même nombre de mots-clés. De même toutes les classes n'ont pas le même nombre d'images annotées. Par exemple, parmi les quatre images de la figure 7.6, la première est annotée par 2 mots-clés, « animal »et « cheval ». La seconde est annotée par 1 mot-clé seulement : « animal ». Les deux autres images n'ont aucune annotation.

Les caractéristiques visuelles utilisées sont issues d'un descripteur de couleur : un histogramme des composantes RGB et de deux descripteurs de forme : la $\mathcal{R}$-signature $1D$ , et le descripteur *HRT* (voir section 6.3.1.4). Ceci va nous permettre, en plus d'évaluer notre modèle, de comparer l'efficacité du descripteur *HRT*, par rapport à la $\mathcal{R}$-signature $1D$, sur une base d'images de scènes naturelles.

Comme dans le chapitre 6, la méthode du LASSO a été utilisée pour réduire le nombre de variables initiales.

#### 7.7.1.2 Protocole expérimental

**Classification**

Notre méthode a été évaluée en effectuant cinq validations croisées, dont chaque proportion de l'échantillon d'apprentissage est fixée à 25%, 35%, 50%, 65% et 75% de la base. Les 75%, 65%, 50%, 35% et 25% respectivement restants sont retenus pour l'échantillon de test. Dans chaque cas, les tests ont été répétés 10 fois, de façon à ce que chaque observation ait été utilisée au moins une fois pour l'apprentissage et les tests. Ici encore le mode d'évaluation choisi est le taux de reconnaissance. Pour chacune des 5 tailles de l'échantillon d'apprentissage, on calcule le taux de reconnaissance moyen en effectuant la moyenne des taux de reconnaissance obtenus pour les 10 tests. Dans tous les tests, notre modèle de mélange de mélanges de Gaussiennes et de lois multinomiales (noté mélange GM-Mult), a été appris et exécuté avec des mélanges de 2 Gaussiennes et des matrices de covariance diagonales.

Le nombre de Gaussiennes a été déterminé expérimentalement, de façon à réaliser le meilleur compromis entre temps de calcul et taux de reconnaissance. Pour ce faire, nous avons effectué les mêmes tests que dans la figure 6.10 de la section 6.7.

**Extension d'annotations**

Considérons maintenant le problème d'extension d'annotations. Il est nécessaire que chaque annotation comprenne au moins un mot-clé pour comparer les annotations après l'extension automatique d'annotations à la vérité terrain. 99% de la base d'images, annotée par au moins 1 mot-clé, a donc été sélectionnée comme vérité terrain. Afin d'évaluer la qualité de l'extension d'annotations, une validation croisée a été effectuée. La proportion de chaque échantillon d'apprentissage est fixée à 50% du sous-ensemble pré-sélectionné de la base comme vérité-terrain. Les 50% restants sont retenus pour l'échantillon de test. Les tests ont été répétés 10 fois, de façon à ce que chaque observation ait été utilisée au moins une fois pour l'apprentissage et les tests. Ici le mode d'évaluation utilisé est le taux de bonnes annotations. Le taux moyen de bonnes annotations est obtenu en effectuant la moyenne des taux de bonnes annotations obtenus pour les 10 tests. Pour chaque test, le taux de bonnes annotations correspond à la proportion de mots-clés corrects parmi les mots-clés obtenus par extension.

Il n'était pas possible de calculer les mesures de rappel et précision voir chapitre (4) en fonction du nombre de mots-clés, couramment utilisées dans les problèmes d'annotation automatique. Ceci est du au fait que l'extension d'annotation n'aboutit pas à des annotations de

même taille pour chaque image, de par l'utilisation d'un seuil sur la probabilité des mots-clés. Ainsi, après extension d'annotations, certaines images seront annotées par 1 mot-clé, d'autres par 3, d'autres par aucun. De même, les annotations initiales (avant extension) ne sont pas de même taille.

Le seuil utilisé pour l'annotation a été fixé à 0.5. C'est-à-dire que, pour une image donnée, un mot-clé sera sélectionné pour étendre son annotation si sa probabilité d'annoter cette image, étant donnés les caractéristiques visuelles et les éventuels mots-clés existants de cette image, est strictement supérieure à 0.5. Cette valeur a été déterminée expérimentalement, de façon à réaliser le meilleur compromis entre le nombre de mots-clés et la qualité de l'annotation. La figure 7.7 présente les résultats des tests que nous avons effectués afin de choisir cette valeur. Plus précisément, cette figure montre le taux moyen de bonnes annotations (courbe verte) et le nombre moyen de mots-clés (courbe bleue) obtenus en fonction de la valeur du seuil. Les tests de comparaison ont été effectués en utilisant 50% de la base pour l'apprentissage et en faisant varier le seuil de 0.1 à 0.9. Les taux présentés sont les moyennes obtenues après validation croisée. On remarque que plus la valeur du seuil augmente, plus le taux moyen de bonnes annotations est élevé. Cependant, le nombre de termes sélectionnés pour l'annotation diminue quand la valeur du seuil augmente. Nous avons donc choisi de fixer le seuil à 0.5, car cette valeur présente un bon compromis entre qualité de l'annotation et nombre de mots-clés. En effet, il s'agit (approximativement) de la valeur du seuil pour laquelle les deux courbes s'interceptent.

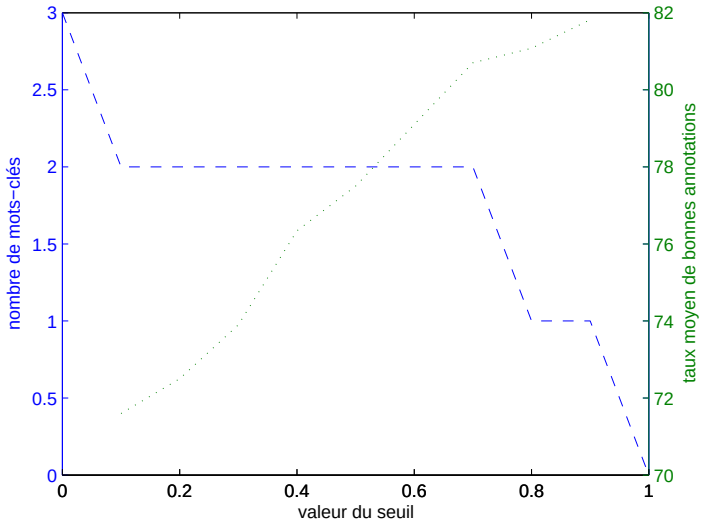



FIGURE 7.7 – Taux moyen de bonnes annotations et nombre moyen de mots-clés obtenus en fonction de la valeur du seuil

#### 7.7.1.3 Résultats

**Classification**

Tout d'abord, considérons le tableau 7.4, présentant le nombre de variables sélectionnées pour chaque descripteur avec la méthode du LASSO, comparé à celui obtenu avec une autre méthode de sélection de variables : « Sequential Forward Selection »(notée SFS) [Pudil 94]. Ce tableau montre que la sélection de variables avec la méthode du LASSO nous a permis de réduire

significativement le nombre de variables issues de la $\mathcal{R}$-signature $1D$. Par contre, seulement 3 variables ont été supprimées de l'ensemble de caractéristiques issues de l'histogramme de couleur. Ceci peut s'expliquer par la faible dimension initiale de l'histogramme de couleurs (16 valeurs pour chaque composante R, G et B) : globalement, il y a plus de variables pertinentes et moins de redondance entre les variables, dans un ensemble de petite taille. De plus, nous pouvons constater que la méthode du LASSO nous a permis de sélectionner plus de variables que la méthode SFS. En effet, la méthode SFS sélectionne les variables en les ajoutant une par une, de manière itérative, à un ensemble de variables déjà sélectionnées. Au contraire, le LASSO est une méthode d'optimisation globale : il vise à faire émerger des sous-groupes de variables pertinentes et sélectionne, de ce fait, plus de variables que les méthodes itératives.

| Nombre de variables | Descripteur couleur | $\mathcal{R}$-signature | *HRT* |
|---|---|---|---|
| Sans sélection | 48 | 180 | 5760 |
| SFS | 11 | 7 | 79 |
| LASSO | 45 | 23 | 103 |

TABLE 7.4 – Nombre moyen de variables en fonction de la méthode de sélection de variables

Le tableau 7.5 montre l'impact de la méthode de sélection de variables sur la qualité de la classification. De façon à mesurer cet impact, une classification a été effectuée sur les caractéristiques visuelles avec notre modèle (noté mélange GM-Mult), et trois autres classificateurs : un classificateur SVM classique [Chang 01], un algorithme flou des $k$ plus proches voisins (noté FKNN) [Keller 85] et le modèle de mélange de lois multinomiales et Gaussiennes (noté GM-Mixture) [Blei 03], présenté dans le chapitre 4 Ces classificateurs ont été choisis pour leur aptitude à traiter à la fois les données discrètes et continues et leur efficacité en présence de grandes dimensions. Le modèle GM-Mixture présente les avantages supplémentaires d'être efficace en présence de données manquantes et de pouvoir être utilisé en annotation. De plus, il est très proche de notre modèle de mélange GM-Mult. Le modèle GM-Mixture a été utilisé sans segmentation des images : le descripteur de couleur et celui de forme ont été calculés sur les images entières, et les mots-clés sont également associés aux images entières. De plus, comme nous considérons, dans ce papier, un problème de classification supervisée, la variable discrète $z$, utilisée dans [Blei 03] pour représenter la classification jointe d'une image et de sa légende, n'est pas cachée pour les images des échantillons d'apprentissage. De même le nombre de clusters est connu. En fait, cette variable discrète $z$ correspond à notre variable classe « Classe ».

Nous avons comparé les taux de reconnaissance pour ces 4 classificateurs sans sélection de variables préalable et après la sélection d'un sous-ensemble de variables avec les méthodes SFS et LASSO. L'algorithme flou des $k$ plus proches voisins a été exécuté avec $k = 1$ et $k = m$, où $m$ désigne le nombre moyen d'images par classe dans l'échantillon d'apprentissage.

De plus, les résultats du tableau 7.5 montrent que la sélection de variables avec la méthode LASSO améliore le taux de reconnaissance de 1.8% en moyenne par rapport à celui obtenu sans sélection de variables préalable, et de 6.9% en moyenne comparé à celui obtenu après sélection de variables avec la méthode SFS. De plus, ce résultat est vérifié quel que soit le classificateur utilisé. En effet, les méthodes de rétrécissement comme le LASSO sont réputées pour être plus stables que les méthodes itératives (voir chapitre 6), pour sélectionner des variables dans un grand ensemble de variables mais avec peu d'exemples. Ainsi, la méthode du LASSO s'est montrée plus robuste expérimentalement, sur cette base d'images, que la méthode SFS. Enfin, nous pouvons remarquer que l'utilisation du LASSO a permis de réduire significativement les temps de calculs

(tableau 7.8). Ainsi, seules les variables sélectionnées avec la méthode du LASSO ont été utilisées dans la suite des expérimentations.

| Méthode de sélection | SVM | FKNN $k = 1$ | FKNN $k = m$ | GM-Mixture | **mélange GM-Mult** |
|---|---|---|---|---|---|
| Sans sélection | 32.2 | 43.2 | 39 | 36.1 | 40.7 |
| SFS | 30.5 | 33.7 | 35 | 32.5 | 33.8 |
| LASSO | 32.6 | 44.1 | 39.3 | 38.9 | 45.2 |

TABLE 7.5 – Taux de reconnaissance moyens (en %), en classification visuelle (histogramme de couleur et $\mathcal{R}$-signature $1D$), pour les classificateurs SVM, FKNN, GM-Mixture et le mélange GM-Mult,en fonction de la méthode de sélection de variables

Considérons maintenant le tableau 7.6. La notation « C + R »signifie que les descripteurs $\mathcal{R}$-signature $1D$ et l'histogramme de couleur (« C »pour couleur et « R »pour $\mathcal{R}$-signature $1D$) ont été combinés. La notation « C + R + KW »indique la combinaison des informations visuelles (couleur et $\mathcal{R}$-signature $1D$) et textuelles. Enfin, la notation « C + *HRT* + KW »indique la combinaison d'informations visuelles (apportées par l'histogramme de couleur et le descripteur *HRT*) et sémantique (apportée par les éventuels mots-clés).

Les taux de reconnaissance confirment que la combinaison des caractéristiques visuelles et sémantiques est toujours plus performante que l'utilisation d'un seul type d'information. En effet, on observe que la combinaison des caractéristiques visuelles et des mots-clés (quand ils sont disponibles) augmente le taux de reconnaissance de 38.6% en moyenne comparé aux résultats obtenus avec le descripteur couleur seul (colonne *C*), de 58.3% en moyenne, comparé à la classification basée sur un seul descripteur de forme (colonnes *R* et *HRT*) et de 37% par rapport à la classification utilisant uniquement l'information textuelle. De plus, on peut noter que pour toutes les expérimentations, combiner deux descripteurs visuels (couleur et $\mathcal{R}$-signature $1D$ ou couleur et descripteur *HRT*) apporte en moyenne une amélioration de 16% du taux de reconnaissance, comparé à l'utilisation d'un seul. Enfin, la classification visuo-textuelle montre une amélioration de 32.4% en moyenne, en terme de taux de reconnaissance, par rapport à la classification basée sur l'information visuelle seule. Enfin, on constate que l'utilisation du descripteur *HRT* à la place de la $\mathcal{R}$-signature $1D$ améliore le taux de reconnaissance de 0.1% en moyenne.

| Spécifications | | C | R | HRT | Mots-clés | C + R | C + *HRT* | C + R + KW | C + *HRT* + KW |
|---|---|---|---|---|---|---|---|---|---|
| app | test | | | | | | | | |
| 25% | 75% | 35 | 17.8 | 17.9 | 36.6 | 39.4 | 39.5 | 69.7 | 69,9 |
| 35% | 65% | 36.9 | 18.1 | 18.1 | 38.9 | 42.2 | 42.2 | 74.4 | 74,4 |
| 50% | 50% | 38.7 | 18.5 | 18.9 | 41.1 | 45 | 45.3 | 79.1 | 79,7 |
| 65% | 35% | 41.1 | 20.6 | 20.5 | 41.5 | 46.6 | 46.6 | 81.7 | 81,8 |
| 75% | 25% | 43.5 | 21.8 | 21.9 | 45.1 | 52.9 | 53 | 82.9 | 83,2 |

TABLE 7.6 – Taux de reconnaissance (en %) de la classification visuelle vs. classification visuo-textuelle (avec mélange GM-Mult)

Ensuite, le tableau 7.7 montre l'efficacité de notre approche (mélange GM-Mult) comparée aux classificateurs SVM, FKNN et GM-Mixture. Les résultats ont été obtenus en utilisant le descripteur de couleur et la $\mathcal{R}$-signature $1D$, comme caractéristiques visuelles, et les éventuels mots-clés associés, pour constituer l'information sémantique. Il apparaît que les résultats du

mélange GM-Mult sont meilleurs que ceux du SVM, du FKNN et du GM-Mixture. Plus précisément, le mélange GM-Mult se montre sensiblement supérieur aux classificateurs SVM et FKNN. Ces résultats ne sont pas surprenants car les SVM et le FKNN sont peu adaptés au traitement des données manquantes. Par contre, les résultats de notre modèle de mélange GM-Mult sont très proches de ceux obtenus par le modèle GM-Mixture. Ceci est dû à la similarité de ces deux modèles. En effet, dans notre mélange GM-Mult, un mélange de Gaussiennes multivariées est utilisé pour estimer la distribution des caractéristiques visuelles, là où le modèle GM-Mixture utilise une Gaussienne multivariée. Ceci explique aussi la légère supériorité de notre modèle, plus précis. Cette différence de précision se révèle plus significative dans l'extension d'annotations.

| Spécifications | | SVM | FKNN $k = 1$ | FKNN $k = m$ | GM-Mixture | **mélange GM-Mult** |
|---|---|---|---|---|---|---|
| apprentissage | test | | | | | |
| 25% | 75% | 38.3 | 59.1 | 58.5 | 68.1 | 69.7 |
| 35% | 65% | 41.3 | 62.3 | 58.3 | 73.9 | 74.4 |
| 50% | 50% | 39.9 | 68.2 | 58.2 | 75.9 | 79.1 |
| 65% | 35% | 40.5 | 72.9 | 67 | 80.7 | 81.7 |
| 75% | 25% | 41.9 | 73.2 | 69.3 | 81 | 82.9 |

TABLE 7.7 – Taux de reconnaissance (en %) des classificateurs SVM, FKNN et GM-Mixture vs. notre mélange GM-M

Enfin, le tableau 7.8 montre les temps CPU du modèle GM-Mixture comparés à ceux de notre modèle de mélange GM-Mult, pour les phases d'apprentissage et de test, dans les mêmes conditions expérimentales que dans le tableau 7.7. Les expérimentations ont été menées sur PC doté d'un processeur Intel Core 2 Duo 2,40 GHz, 2 Go RAM, Windows OS. Les deux classificateurs ont été exécutés avec Matlab©. Le temps CPU est plus élevé pour le modèle de mélange GM-Mult, car il dépend du nombre de Gaussiennes (dans ce cas, 2) et de la précision de l'algorithme EM, mais il reste faible de l'ordre de 0.04s par image en phase de classification hors apprentissage. Le modèle GM-Mixture est plus rapide car il n'utilise qu'une Gaussienne multivariée. Cependant, la différence en temps de calcul entre les deux méthodes est négligeable (inférieure à 0.015s par image).

| Apprentissage | GM-Mixture sans SV | | **GM-Mult** sans SV | | GM-Mixture avec SV | | **GM-Mult** avec SV | |
|---|---|---|---|---|---|---|---|---|
| | app | test | app | test | app | test | app | test |
| 25% | 91.1 | 86.5 | 137.3 | 145 | 80.8 | 53.7 | 101.5 | 77 |
| 35% | 127.2 | 80.6 | 178.4 | 128.3 | 106 | 47.9 | 134.3 | 71.7 |
| 50% | 177 | 56.7 | 284.3 | 113 | 156.5 | 34.9 | 202.8 | 55.5 |
| 65% | 255.9 | 40.5 | 388.4 | 75.3 | 209.2 | 25.1 | 277.7 | 37.4 |
| 75% | 263.4 | 30.3 | 460.7 | 58.5 | 227.6 | 15 | 300 | 26.5 |

TABLE 7.8 – Temps CPU (en secondes) du modèle GM-Mixture vs. modèle mélange GM-Mult, avec ou sans sélection de variables avec le LASSO (SV). Les temps CPU sont donnés pour la classification visuo-textuelle de toutes les images test

**Extension d'annotations**

Le tableau 7.9 compare les taux de bonnes annotations obtenus par notre approche (mé-

lange GM-Mult) par rapport au modèle de mélange de lois multinomiales et Gaussiennes (GM-Mixture). On observe que notre modèle est sensiblement meilleur que le modèle GM-Mixture. Comme en classification, cette supériorité est due au fait que notre modèle utilise un mélange de Gaussiennes multivariées pour estimer la distribution des caractéristiques visuelles, là où le modèle GM-Mixture utilise une Gaussienne multivariée.

| GM-Mixture | **mélange GM-Mult** |
|---|---|
| 36 | 77.5 |

TABLE 7.9 – Taux moyens de bonnes annotations (en %), obtenus par le modèle GM-Mixture vs. notre modèle mélange GM-Mult

De plus, le tableau 7.10 montre les temps CPU du modèle GM-Mixture comparés à ceux de notre modèle de mélange GM-Mult, pour les phases d'apprentissage et de test, dans les mêmes conditions expérimentales que dans le tableau 7.9. Les deux classificateurs ont été exécutés avec Matlab©. Le temps CPU de notre modèle GM-Mult est environ 2 fois supérieur à celui du modèle GM-Mixture, pour les mêmes raisons que celles évoquées précédemment (nombre de Gaussiennes et précision dans l'algorithme EM) mais il reste inférieur à 0.04s par image. Compte tenu de l'écart significatif entre les taux de bonnes annotations des deux méthodes, il nous semble que le compromis entre précision d'annotation et temps de calcul est meilleur pour notre modèle de mélange GM-Mult.

| GM-Mixture | | **mélange GM-Mult** | |
|---|---|---|---|
| apprentissage | test | apprentissage | test |
| 121.3 | 27.6 | 231.5 | 54 |

TABLE 7.10 – Temps CPU (en secondes) du modèle GM-Mixture vs. modèle mélange GM-Mult. Les temps CPU sont donnés pour l'extension d'annotations de toutes les images test

Enfin, des annotations ont été ajoutées automatiquement à toutes les images de la base de façon à ce que chacune soit annotée par 3 mots-clés. Puis, afin d'évaluer la qualité de cette extension d'annotations, la classification visuo-textuelle a été répétée avec les mêmes spécifications que dans le tableau 7.6. Le tableau 7.11 montre l'efficacité de notre extension automatique d'annotations. En effet, les taux de reconnaissance après l'extension d'annotations sont toujours meilleurs qu'avant. De plus, l'extension automatique d'annotations améliore le taux de reconnaissance de 6.8% en moyenne.

| Spécifications | | Avant extension d'annotations | Après extension |
|---|---|---|---|
| apprentissage | test | | |
| 25% | 75% | 69.7 | 77 |
| 35% | 65% | 74.4 | 79.3 |
| 50% | 50% | 79.1 | 85.4 |
| 65% | 35% | 81.7 | 87.6 |
| 75% | 25% | 82.9 | 92.7 |

TABLE 7.11 – Taux de reconnaissance (en %) de la classification visuo-textuelle (avec mélange GM-M) avant et après extension automatique d'annotations

#### 7.7.1.4 Conclusion

Nous avons proposé un modèle efficace permettant de combiner l'information visuelle et textuelle, de traiter les données manquantes et d'étendre des annotations existantes à d'autres images. Afin de diminuer la complexité de notre méthode, nous avons adapté une méthode de sélection de caractéristiques, qui a montré expérimentalement son efficacité. Nos expérimentations ont été effectuées sur une base d'images partiellement annotées provenant d'Internet. Les résultats montrent que la classification visuo-textuelle a amélioré le taux de reconnaissance comparée à la classification basée sur l'information visuelle seule. De plus, notre réseau Bayésien a été utilisé pour étendre des annotations à d'autres images, ce qui a encore amélioré le taux de reconnaissance. Enfin, la méthode proposée s'est montrée compétitive par rapport à des classificateurs classiques, aussi bien en classification qu'en extension automatique d'annotations.

D'autre part, ce modèle donne des résultats proches, mais supérieurs à ceux du modèle GM-Mixture. Nous verrons, dans la section suivante, qu'avec le nouveau modèle GM-B, l'écart avec le modèle GM-Mixture (en terme de taux de reconnaissance et de taux de bonnes annotations) se creuse encore.

### 7.7.2 Modèle de mélange GM-B

#### 7.7.2.1 Données

Dans cette section, nous présentons une évaluation de notre modèle sur plus de 30000 images provenant de la librairie d'images COREL, et fournies par Vasconcelos and al. [Carneiro 07]. Ces images sont réparties en 306 classes. Cette base est partiellement annotée. La connaissance d'un mot-clé pour une image de détermine pas la valeur de sa classe. En effet, un même terme peut apparaître dans l'annotation d'images de différentes classes. Par exemple, 4 images de classes différentes, avec le mot-clé « duck »en commun, sont données dans le tableau 7.12. De plus, certaines classes se chevauchent, *i. e.* que plusieurs images de la base sont dans deux classes différentes.

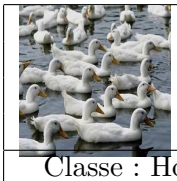
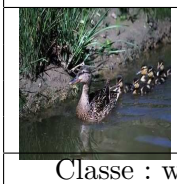
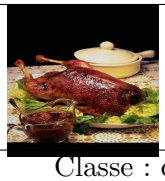
| | | | |
|---|---|---|---|
|  | water<br>duck<br>reflection<br>flock |  | bird<br>duck<br>water<br>close-up |
| Classe : Hong Kong | | Classe : african birds | |
|  | bird<br>duck<br>mallard<br>baby |  | duck<br>food<br>cuisine<br>meal |
| Classe : waterfowl | | Classe : cuisine | |

TABLE 7.12 – Exemples d'images, avec leurs classes et éventuels mots-clés

Enfin, toutes les images n'ont pas le même nombre de mots-clés. 72% des images de la base sont annotées par 4 mots-clés, 23% par 3 mots-clés, 4% par 2 mots-clés et 0.5% par 1 mot-clé (*i. e.* 99.5% des images de la base sont annotées par au moins 1 mot-clé), en utilisant un vocabulaire de 1036 mots-clés. Par conséquent, dans cette base, les images annotées par moins de 4 mots-clés sont considérées comme étant partiellement annotées. De plus, toutes les classes n'ont pas le même nombre d'images annotées. Enfin, la base ne contient pas que des images d'animaux.

Par exemple, elle contient, sans être exhaustifs, des images de paysages, de lieux publics, de personnes de différents pays, des scènes de sport, d'objets, d'aliments, *etc.*.

Les caractéristiques visuelles sont les mêmes que celles utilisées dans la section 7.7.1, à savoir un histogramme des composantes RGB et un descripteur de forme : la $\mathcal{R}$-signature $1D$. De même, la méthode du LASSO a été utilisée pour réduire le nombre de variables initiales.

#### 7.7.2.2 Protocole expérimental

Avant d'étudier les méthodes d'évaluation de la classification et de l'annotation, il faut établir les dépendances entre mots-clés. Ces dépendances sont établies à partir du vocabulaire. On définit une relation de dépendance entre deux mots-clés appartenant au même « synset »(groupe sémantique), comme défini dans Wordnet [Fellbaum 98].

De plus, comme pour le modèle GM-Mult, le nombre de Gaussiennes a fixé à deux. Ce nombre a été déterminé expérimentalement, de façon à réaliser le meilleur compromis entre temps de calcul et taux de reconnaissance. Pour ce faire, nous avons effectué les mêmes tests que dans la figure 6.10 de la section 6.7.

**Classification**

Notre méthode a été évaluée en effectuant 6 validations croisées, dont chaque proportion de l'échantillon d'apprentissage est fixée à 25%, 35%, 50%, 65%, 75% et 90% de la base. Les 75%, 65%, 50%, 35%, 25% et 10% respectivement restants sont retenus pour l'échantillon de test. Dans chaque cas, les tests ont été répétés 10 fois, de façon à ce que chaque observation ait été utilisée au moins une fois pour l'apprentissage et les tests. Ici encore le mode d'évaluation choisi est le taux de reconnaissance. Pour chacune des 6 tailles de l'échantillon d'apprentissage, on calcule le taux de reconnaissance moyen en effectuant la moyenne des taux de reconnaissance obtenus pour les 10 tests. Pour chaque test, le taux de reconnaissance correspond au ratio entre le nombre d'images bien classifiées et le nombre d'images de l'échantillon de test. Dans tous les tests, notre modèle de mélange de lois de Bernoulli et de mélanges de Gaussiennes (noté GM-B), a été appris et exécuté avec des mélanges de 2 Gaussiennes et des matrices de covariance diagonales.

**Extension d'annotations**

Considérons maintenant le problème d'extension d'annotations. Il est nécessaire que chaque annotation comprenne au moins un mot-clé pour comparer les annotations après l'extension automatique d'annotations à la vérité terrain. 99.5% de la base d'images, annotée par au moins 1 mot-clé, a donc été sélectionnée comme vérité terrain. Comme pour la classification, 6 validations croisées ont été effectuées. Les tests sont répétés 10 fois de sorte que chaque image soit utilisée pour l'apprentissage et les tests. Pour chaque test, l'annotation des images a été étendue automatiquement jusqu'à 4 mots-clés. Comme dans la section 8.2.2, le mode d'évaluation utilisé est le taux de bonnes annotations. Ce taux est calculé de le même façon que pour le modèle GM-Mult (voir section 7.7.1).

Enfin, le seuil utilisé pour l'annotation a été fixé à 0.5, comme dans le modèle GM-Mult (voir section 7.7.1). Cette valeur a été déterminée expérimentalement, de façon à réaliser le meilleur compromis entre le nombre de mots-clés et la qualité de l'annotation (voir la figure 7.7, dans la section 7.7.1).

#### 7.7.2.3 Résultats

**Classification**

Considérons le tableau 7.13. Notre modèle de mélange de lois de Bernoulli et de mélanges de Gaussiennes a été utilisé pour combiner différents types d'information. La notation « C + S »signifie que des descripteurs de couleur et de forme (« C »pour couleur, « S »pour forme (shape)) ont été combinés et « C + S + KW »intègre l'information textuelle (« KW »pour keywords *i. e.* mots-clés). Les taux de reconnaissance confirment que la combinaison de caractéristiques visuelles et sémantiques entraîne une meilleure classification que l'utilisation d'un seul type d'information.

| proportion apprentissage | C | S | KW | C + S | C + S + KW |
|---|---|---|---|---|---|
| 25% | 20.6 | 16.5 | 48.3 | 23.6 | 58.5 |
| 35% | 22.8 | 16.8 | 54.5 | 24 | 59 |
| 50% | 23.4 | 18.4 | 61.4 | 24.3 | 64.2 |
| 65% | 24.1 | 19.1 | 62.4 | 26 | 65.6 |
| 75% | 26 | 19.9 | 67.8 | 26.4 | 69.8 |
| 90% | 26 | 24 | 69.2 | 28.8 | 76 |

TABLE 7.13 – Taux de reconnaissance moyens (en %) de notre modèle GM-B sans relations sémantiques

Le tableau 7.14 montre les taux de reconnaissance obtenus avec notre modèle GM-B, en prenant en compte les relations sémantiques entre mots-clés pré-établies (colonne « avec RS », RS pour relations sémantiques), ou non (colonne « sans »). Ces résultats montrent que la prise en compte des relations sémantiques entre mots-clés améliore le taux de reconnaissance de 10.5%. De plus, le tableau 7.14 montre l'efficacité de notre approche (modèle GM-B) comparée au modèle de mélange de Gaussiennes et de lois multinomiales (GM-mixture) [Blei 03]. Le modèle GM-Mixture a été utilisé dans segmentation des images, comme dans dans la section précédente.

Les résultats ont été obtenus en utilisant les caractéristiques visuelles des images et leurs éventuels mots-clés associés. Il apparaît qu'en prenant en compte les relations sémantiques entre mots-clés, notre modèle GM-B a de meilleurs taux de reconnaissance que le modèle GM-Mixture. De plus, pour chaque taille d'échantillon d'apprentissage, un test de Student pour échantillons appariés [Feller 68] a été utilisé pour comparer les taux de reconnaissance moyens (sur les 10 tests de la validation croisée) du modèle GM-Mixture et de notre modèle GM-B. Quelle que soit la taille de l'échantillon d'apprentissage, la valeur $t$ (voir les écarts-types des différences et les valeurs de $t$ dans la tableau 7.15) montre que la moyenne des taux de reconnaissances obtenus avec notre modèle GM-B, avec relations sémantiques, est statistiquement différente que celle du modèle GM-Mixture, avec un risque inférieur à 1% et un degré de liberté de 9.

| Apprentissage | GM-Mixture | GM-B | |
|---|---|---|---|
| | | sans | avec RS |
| 25% | 61 | 58.5 | 68.7 |
| 35% | 62.4 | 59 | 69.5 |
| 50% | 67.2 | 64.2 | 76.2 |
| 65% | 67.7 | 65.6 | 75.4 |
| 75% | 72.2 | 69.8 | 80.4 |
| 90% | 78.6 | 76 | 86 |

TABLE 7.14 – Taux de reconnaissance moyens (en %) du modèle GM-Mixture vs. notre modèle GM-B

| Apprentissage | écart-type de la différence | valeur de $t$ |
|---|---|---|
| 25% | 0.33 | 22.96 |
| 35% | 0.2 | 35.5 |
| 50% | 0.095 | 94.3 |
| 65% | 0.098 | 78.99 |
| 75% | 0.16 | 50.86 |
| 90% | 0.11 | 64.06 |

TABLE 7.15 – Test de Student pour la comparaison de moyennes des taux de reconnaissance du modèle GM-Mixture vs. notre modèle GM-B avec RS en classification visuo-textuelle

**Extension d'annotations** Le tableau 7.16 compare les taux de bonnes annotations obtenus avec et sans prise en compte des relations sémantiques entre mots-clés. On peut observer que la prise en compte des relations sémantiques améliore les taux de bonnes annotations de 6.9% en moyenne. Le tableau 7.16 compare aussi les taux de bonnes annotations obtenus avec le modèle GM-Mixture et notre modèle GM-B. On peut voir que notre modèle est meilleur que le modèle GM-Mixture, même sans prendre en compte les relations sémantiques entre mots-clés.

Enfin, en utilisant le même test que dans le tableau 7.15, on peut remarquer que le taux moyen de bonnes annotations obtenu avec notre modèle GM-B, avec relations sémantiques, est statistiquement différent que celui du modèle GM-Mixture (voir valeurs de $t$ dans le tableau 7.17).

| Apprentissage | GM-Mixture | GM-B | |
|---|---|---|---|
| | | sans | avec RS |
| 25% | 40 | 52 | 71 |
| 35% | 56.2 | 72.6 | 78.9 |
| 50% | 60 | 72.8 | 79.6 |
| 65% | 61.7 | 77.1 | 79.7 |
| 75% | 66 | 78.9 | 82.3 |
| 90% | 68.7 | 79 | 82.4 |

TABLE 7.16 – Taux moyens (in %) de bonnes annotations du modèle GM-Mixture vs. notre modèle GM-B

| Apprentissage | écart-type de la différence | valeur de $t$ |
|---|---|---|
| 25% | 0.58 | 53.32 |
| 35% | 0.14 | 155.85 |
| 50% | 0.14 | 138.47 |
| 65% | 0.23 | 81.26 |
| 75% | 0.2 | 77.71 |
| 90% | 0.13 | 100.67 |

TABLE 7.17 – Test de Student pour la comparaison de moyennes des taux de bonnes annotations du modèle GM-Mixture vs. notre modèle GM-B avec RS en extension automatique d'annotations

## 7.8 Conclusion

Nous venons de proposer deux approches de modélisation, classification et extension d'images partiellement annotées. Contrairement au modèle GM-Mult, le modèle GM-B a l'avantage de prendre en compte des relations sémantiques entre les termes annotant les images. Les résultats expérimentaux obtenus en classification visuo-textuelle ont montré que la représentation des relations sémantiques améliore le taux de reconnaissance. De plus, notre approche a particulièrement montré de bonnes performances en extension automatique d'annotations. Enfin, l'évaluation, sur plus de 30000 images, a montré que nos modèles sont compétitifs face aux modèles de l'état de l'art.

# Chapitre 8

# Recherche d'images de scènes naturelles

## 8.1 Contexte

Dans l'introduction générale (chapitre 1), nous avons vu qu'une fois les images indexées, différents besoins concernant l'accès à ces images apparaissaient, correspondant chacun à une tâche différente : le parcours séquentiel d'un ensemble d'images dans la cas où l'utilisateur n'a pas vraiment d'idée de ce qu'il recherche ; la recherche d'images, quand l'utilisateur sait précisément ce qu'il recherche et qu'il est capable de l'exprimer à partir de mots-clés ou d'une image exemple (requête) ; et enfin la classification d'images, qui permet de regrouper entre elles des images ayant des thématiques proches et fournit ainsi une représentation simplifiée et ordonnée d'un ensemble d'images, qui permettra une manipulation et un accès à l'information faciles et rapides dans de grandes bases d'images.

Dans le chapitre précédent, nous avons proposé différentes approches au problème de classification d'images. Ici, on va s'intéresser à la recherche d'images à partir d'exemples. Le contexte est le suivant : à partir d'une base d'images partiellement annotées, on souhaite, à partir d'une image requête annotée ou non, rechercher les images les plus proches. Des bases d'images partiellement annotées sont considérées, car, comme on a pu le voir dans l'état de l'art, les annotations constituent de l'information sémantique utile à la recherche d'images. Cependant, on a vu aussi que les annotations obtenues manuellement sont les plus pertinentes et donc les plus efficaces. Or, l'annotation manuelle, surtout dans le cas de grandes bases d'images, constitue un travail fastidieux. Pour réaliser le meilleur compromis possible entre le coût de l'annotation manuelle préalable et l'utilité des annotations, on souhaite proposer une méthode efficace sur des bases d'images partiellement annotées, c'est-à-dire où seul un sous-ensemble d'images de la base dispose d'annotations et où le nombre de mots-clés peut différer suivant les images. Enfin, afin de retourner à l'utilisateur les images correspondant le plus à ses besoins, on souhaite modéliser ses préférences grâce à un processus de retour de pertinence avec exemples positifs et négatifs. En effet, comme nous avons pu le voir dans l'état de l'art, les processus de retour de pertinence, même s'ils restent coûteux pour l'utilisateur, améliorent grandement la qualité des résultats de recherche, d'autant plus si les exemples négatifs et positifs sont pris en compte.

## 8.2 Recherche visuo-textuelle avec des modèles graphiques probabilistes

Le problème de recherche d'images peut être vu comme un problème de classification non supervisé où la variable classe est « cachée », *i. e.*. En effet, retrouver les images proches d'une image requête revient à retourner, de manière ordonnée, les images provenant de la même classe que l'image requête. Ainsi, comme dans les approches de classification visuo-textuelle proposées dans le chapitre 7, notre choix s'est porté vers les modèles graphiques probabilistes, afin de résoudre notre problème de recherche d'images. En effet, en plus de permettre de modéliser des données de différents types (caractéristiques visuelles et information sémantique, continues ou discrètes), ils fournissent aussi un aperçu des relations entre les différents éléments inhérents au processus de recherche d'images visuo-textuelle avec processus de retour de pertinence et surtout ils permettent d'effectuer efficacement la recherche.

Ce chapitre est donc organisé de la façon suivante : dans la section 8.3, nous présentons notre modèle graphique probabiliste pour la recherche d'images, intégrant un processus de retour de pertinence, afin d'évaluer ce modèle dans la section 8.4. Enfin, la section 8.5, nous concluons en donnant les avantages et inconvénients de ce modèle.

## 8.3 Modèle pour la recherche d'images avec retour de pertinence avec exemples positifs et négatifs

Nous proposons un modèle hiérarchique probabiliste de données de différents types : images et mots-clés associés. Ce modèle a le double avantage de permettre de modéliser de telles données et d'être utilisé pour rechercher des images, à partir d'une image accompagnée éventuellement de mots-clés, dans de grandes bases d'images partiellement annotées. On rappelle qu'une image est considérée comme partiellement annotée si elle comporte moins de mots-clés que le nombre maximal de mots-clés disponibles dans la vérité-terrain pour une image.

### 8.3.1 Définition du modèle proposé

Cette section décrit comment chaque élément utilisé pour la recherche va être modélisé. Quelques définitions des variables utilisées pour représenter les différents éléments sont nécessaires. Considérons les préférences de l'utilisateur comme deux images qu'il pourra « pointer »parmi celles retournées comme résultat d'une requête. La première image pointée par l'utilisateur est une image qu'il considérera comme étant proche de la requête, et va constituer un exemple dit positif (car cette image répond positivement à la requête) dans le processus de retour de pertinence. La deuxième image pointée par l'utilisateur, au contraire, sera une image que ce dernier considérera comme éloignée de la requête, et va représenter un exemple dit négatif (dans le sens où cette image ne répond pas à la requête) dans le processus de retour de pertinence.

Pour caractériser ces deux images, les mêmes caractéristiques visuelles que celles utilisées pour représenter l'image requête sont utilisées. Par contre elles ne sont pas accompagnées de mots-clés, car seules les images sont désignées par l'utilisateur lors du processus de retour de pertinence.

Les caractéristiques visuelles sont considérées comme des variables continues. En effet, les descripteurs d'images utilisés fournissent des vecteurs de caractéristiques continues. Les éventuels mots-clés associés aux images sont considérés comme des variables discrètes suivant une loi multinomiale. En effet, pour chaque annotation possible est considérée comme une variable

discrète prenant ses valeurs dans le vocabulaire utilisé. De plus, l'observation de la distribution sur les différents histogrammes des variables caractéristiques nous a conduits à considérer que les caractéristiques visuelles peuvent être estimées par des densités à mélanges de Gaussiennes.

Considérons alors trois variables discrètes latentes, chacune utilisée pour représenter la classe virtuelle d'appartenance, respectivement d'une image requête, de l'exemple positif et de l'exemple négatif. On a donc un modèle de mélange de Gaussiennes (GMM) par classe, pour chaque variable latente classe. Or un GMM peut être représenté par le modèle graphique probabiliste de la figure 8.1, dont les explications ont déjà été données dans le chapitre 6.

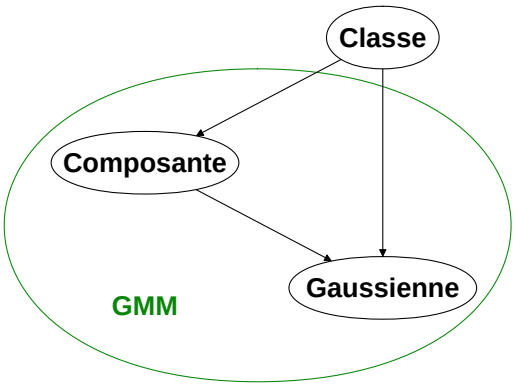



FIGURE 8.1 – GMMs représentés par un modèle graphique probabiliste

Ainsi le modèle proposé inclut 3 occurrences de ce modèle graphique : une pour représenter les exemples positifs (notée GMMs 1), une pour les exemples négatifs (GMMs 2) et une pour l'image requête (GMMs 3). Ces sous-modèles sont entourés en vert dans le modèle complet donné dans la figure 8.2 où :

- les nœuds CL-PI, CL-NI et CL-QI correspondent respectivement aux variables latentes « Classe »des exemples positifs (PI), exemples négatifs (NI) et à l'image requête (QI),
- les nœuds C-M1, C-M2 and C-M3 correspondent respectivement aux variables latentes « Composante »du modèle de mélange de Gaussiennes associé aux exemples positifs, aux exemples négatifs et à l'image requête,
- les nœuds G-M1, G-M2 et G-M3 correspondent respectivement aux variables continues « Gaussienne »du modèle de mélange de Gaussiennes associé aux exemples positifs, aux exemples négatifs et à l'image requête.

Maintenant le modèle peut être complété par les variables discrètes correspondant aux éventuels mots-clés associés à l'image requête. Ces variables discrètes, notées $KW\ 1, \ldots, KW\ n$, sont supposées avoir une distribution de probabilité multinomiale sur le vocabulaire de mots-clés.

Des *a priori* de Dirichlet[Robert 97], ont été utilisés pour l'estimation des probabilités des variables mots-clés. Comme les variables continues correspondant aux caractéristiques de l'image requête, les variables discrètes correspondant aux mots-clés associés aux images requêtes sont inclues dans la graphe en les connectant à la variable latente classe de l'image requête.

Enfin, une variable discrète latente est nécessaire pour représenter la classe virtuelle d'appartenance de l'image pertinente retournée. La classe des images pertinentes (proches de la requête) est supposée avoir des dépendances avec les autre variables latentes classes. Ainsi la variable représentant la classe des images pertinentes, notée $RI$ dans la figure 8.2, est la racine du graphe et est reliée aux trois autres variables latentes classes.

La variable latente $\alpha$ montre qu'un *a priori* de Dirichlet est utilisé. Le cadre autour de la

variable $KW$ indique $n$ répétitions de la variable $KW$, pour chaque mot-clé.

Le sous-graphe *Requete*, entouré en rouge, signifie que chaque image et ses mots-clés sont supposés avoir été générés conditionnellement à la même classe virtuelle.

Par conséquent les paramètres du mélange de lois multinomiales et de mélanges de Gaussiennes résultant doivent correspondre. Concrètement si une image requête, représentée par des caractéristiques visuelles, a une grande probabilité étant donnée une classe virtuelle, alors ses mots-clés doivent aussi avoir une grande probabilité dans cette même classe virtuelle.

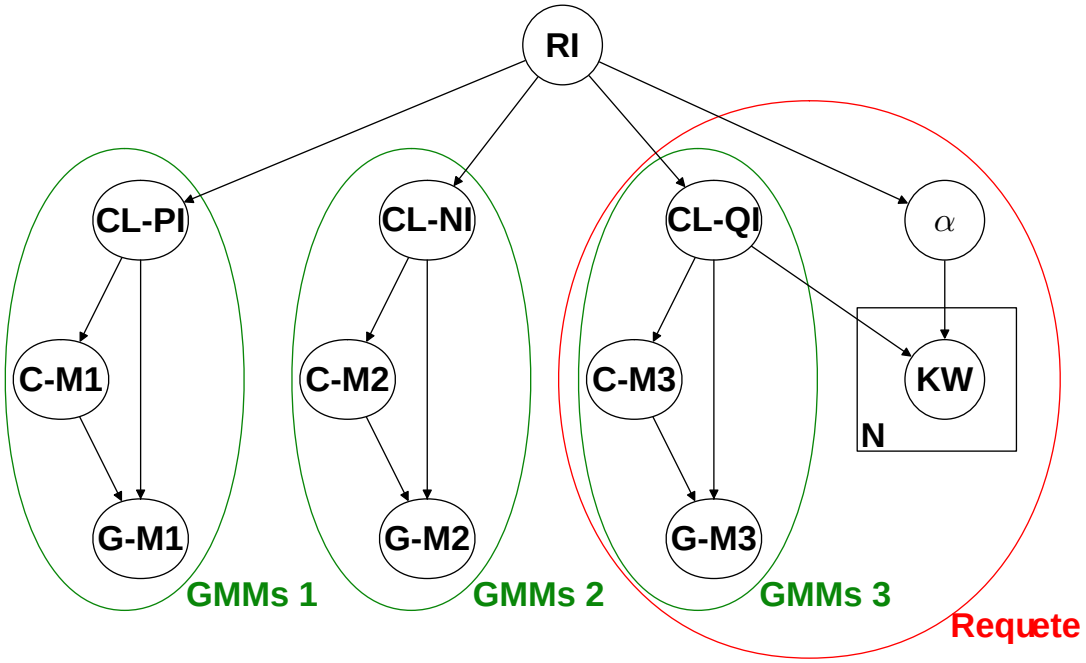



FIGURE 8.2 – Le modèle de recherche d'images proposé

### 8.3.2 Recherche d'images

Afin de rechercher les images les plus proches d'une image requête $f_j$, le nœud classe de l'image requête $CL-QI$ est inféré grâce à l'algorithme de passage de message. Cette image, caractérisée par ses caractéristiques visuelles continues $v_{j_1}, \ldots, v_{j_m}$ et ses éventuels mots-clés $KW\ 1_j, \ldots, KW\ n_j$ est considérée comme une « évidence »représentée par :

$$P(f_j) = P(v_{j_1}, \ldots, v_{j_m}, KW\ 1_j, \ldots, KW\ n_j) = 1$$

quand le réseau est évalué. Grâce à l'algorithme d'inférence, les probabilités de chaque nœud sont mises à jour en fonction de cette évidence. Après la propagation de croyance, on connaît, $\forall i \in \{1, \ldots, k\}$, la probabilité *a posteriori* :

$$P(c_i|f_j) = P(c_i|v_{j_1}, \ldots, v_{j_m}, KW\ 1_j, \ldots, KW\ n_j)$$

Si l'on dispose d'une image exemple positive pour le retour de pertinence, la classe cachée de cette image $CL-PI$ est inférée de la même manière.

Enfin, si l'on dispose d'une image exemple négative pour le retour de pertinence, la classe cachée de cette image $CL-NI$ est inférée de la même manière.

Il est parfaitement possible d'intégrer plusieurs exemples positifs et négatifs. Pour ce faire, il suffit d'entrer plusieurs évidences, chaque évidence correspondant à l'observation des caractéristiques visuelles sur une image exemple.

La classe cachée des images pertinentes est alors inférée.

On infère alors la classe de chaque image de la base de recherche, comme s'il s'agissait d'une image requête seule (non accompagnée d'images exemples positifs ou négatifs).

Finalement, les images considérées comme les $k$ plus proches de la requête sont les images de la base dont la classe inférée est la même que la classe inférée pour les images pertinentes ($RI$), avec les $k$ plus grandes probabilités d'appartenance à cette classe.

## 8.4 Évaluation et résultats

### 8.4.1 Données

La base de données est la même que celle utilisée dans la section 7.7.1. On rappelle qu'il s'agit d'une base de plus de 3000 images provenant d'Internet.

Les caractéristiques utilisées sont un descripteur de couleur (histogramme des composantes RGB) et un descripteur de forme : la $\mathcal{R}$-signature $1D$, tous deux utilisés dans le chapitre 7.

### 8.4.2 Protocole expérimental

Comme dans les chapitres 6 et 7, la méthode du LASSO a été utilisée pour réduire le nombre de variables initiales.

De plus, le nombre de Gaussiennes a fixé à deux. Ce nombre a été déterminé expérimentalement, de façon à réaliser le meilleur compromis entre temps de calcul et taux de reconnaissance. Pour ce faire, nous avons effectué les mêmes tests que dans la figure 6.10 de la section 6.7.

Ensuite, les expérimentations ont été menées en considérant successivement chaque image de la base comme image requête. Pour chaque requête, les résultats sont retournés sous la forme d'une liste ordonnée de 30 images. Ce nombre a été limité à 30 en fonction des principes d'élaboration d'Interfaces Homme-Machine ergonomes. En effet, il faut limiter le nombre d'informations affichées dans une page pour permettre aux utilisateurs de s'y retrouver.

Comme nous l'a vu dans la section 8.3, les images considérées comme étant pertinentes pour une requête sont celles de la même classe (on rappelle que cette variable est cachée/latente). Le processus de retour de pertinence avec exemple positifs a été exécuté de la façon suivante : une image de la même classe que l'image requête, choisie de façon aléatoire dans la classe, est considérée comme exemple positif. Concernant le processus de retour de pertinence avec exemple négatifs, une image non pertinente, choisie également de façon aléatoire parmi les classes différentes de la classe de l'image requête, sont considérées comme des exemples négatifs. Le processus de retour de pertinence a été testé avec 1 et 10 itérations. C'est-à-dire que dans le cas des 10 itérations, une même image a été présentée 10 fois comme image requête, et à chaque résultat, une image pertinente pour la requête était choisie (parmi la liste résultat) comme exemple positif, et une image non pertinente comme exemple négatif, avant de relancer le processus de recherche. Les itérations sont faites grâce à l'algorithme. Notre algorithme permet de présenter plusieurs exemples positifs et négatifs. Cependant, nous avons fait le choix de ne préciser qu'un exemple positif et un exemple négatif, pour minimiser l'intervention de l'utilisateur.

Pour évaluer la qualité de la recherche, une mesure de précision, fonction du rang d'une image dans la liste retournée, a été utilisée. Cette précision moyenne, notée $P$, utilisée dans le tableau 8.1, est exprimée en %. Soit $n$ la taille de la base (*i. e.* le nombre d'images qu'elle contient). Soit

$QI_j, \forall j \in \{1, \ldots, n\}$, une image requête. Alors les $k, \forall k \in \{1, \ldots, 30\}$ précisions de recherche de l'image $QI_j$ sont définies par :

$$P_{j_k} = \frac{\#\text{ images pertinentes parmi les k premières images de la liste}}{\text{k}}$$

et $P = \frac{\sum_{j=1}^{n} \frac{\sum_{k=1}^{30} P_{j_k}}{30}}{n} \times 100$.

### 8.4.3 Résultats

Considérons le tableau 8.1. La notation « VF »signifie que les caractéristiques visuelles ont été utilisées. La notation « KW »indique que l'information textuelle a été utilisée. Quant à la notation « PRF »(respectivement « RF »), elle indique l'utilisation du processus de retour de pertinence avec exemples positifs seulement (respectivement avec des exemples positifs et négatifs).

Les résultats confirment que la combinaison de caractéristiques visuelles et sémantiques avec retour de pertinence améliore la précision de la recherche. En effet, on observe que la combinaison de caractéristiques visuelles et d'éventuels mots-clés améliore la précision de recherche de 26.4% en moyenne comparée à la recherche d'images par le contenu. De plus, on peut noter que le processus de retour de pertinence, avec exemples positifs seulement, améliore la précision de recherche de 10.1% en moyenne. L'ajout d'exemples négatifs dans le processus de retour de pertinence améliore encore la précision de 8%.

Pour résumer, on a atteint une amélioration de 44.5% en moyenne en procédant à une recherche visuo-textuelle associée à un processus de retour de pertinence avec exemples positifs et négatifs, comparée à une recherche d'image par le contenu.

| VF | VF + KW | VF + KW + PRF | | VF + KW + RF | |
|---|---|---|---|---|---|
| | | 1 itération | 10 itérations | 1 itération | 10 itérations |
| 44.3 | 70.7 | 79.3 | 82.4 | 86.7 | 91 |

TABLE 8.1 – Précisions moyennes de recherche ($P$ en %)

Considérons maintenant la figure 8.3, représentant les précisions moyennes en fonction du rang $k$ dans la liste résultat, dans les mêmes conditions que dans le tableau 8.1. Le processus de retour de pertinence a été testé avec 10 itérations (courbes bleue et rose).

La courbe rouge (VF = caractéristiques visuelles uniquement), toujours en dessous des 3 autres, montre la robustesse de la recherche visuo-textuelle. De plus, on remarque que les courbes rouge et jaune, obtenues sans processus de retour de pertinence, décroissent continuellement.

Au contraire, les 2 autres courbes croissent à partir de 25 images environ, grâce au retour de pertinence. Ceci peut s'expliquer par le fait qu'une grande valeur de $k$ laisse plus de variabilité quant à la « reformulation »de la requête : une grande valeur de $k$ laissera plus de chances à de nouvelles images (dont des images pertinentes) de rentrer dans le top $k$, à chaque itération de retour de pertinence. Ceci montre que la valeur maximale de $k$, fixée à 30 dans un soucis d'ergonomie, pourrait encore être augmentée, pour permettre encore plus de variabilité au top $k$ et améliorer encore les résultats. Augmenter le nombre d'itérations permet aussi d'améliorer les résultats de recherche. Nous pouvons le constater sur le tableau 8.1. Enfin, pour fixer la valeur maximale de $k$, il faut trouver un compromis entre précision de recherche et ergonomie

de l'application. En effet, une petite valeur de $k$ offre une meilleur ergonomie mais nécessite plus d'itérations pour atteindre une bonne précision.

De plus, la courbe rose, toujours au dessus de la bleue, montre l'intérêt d'utiliser les exemples négatifs, en plus des positifs, dans le processus de retour de pertinence.

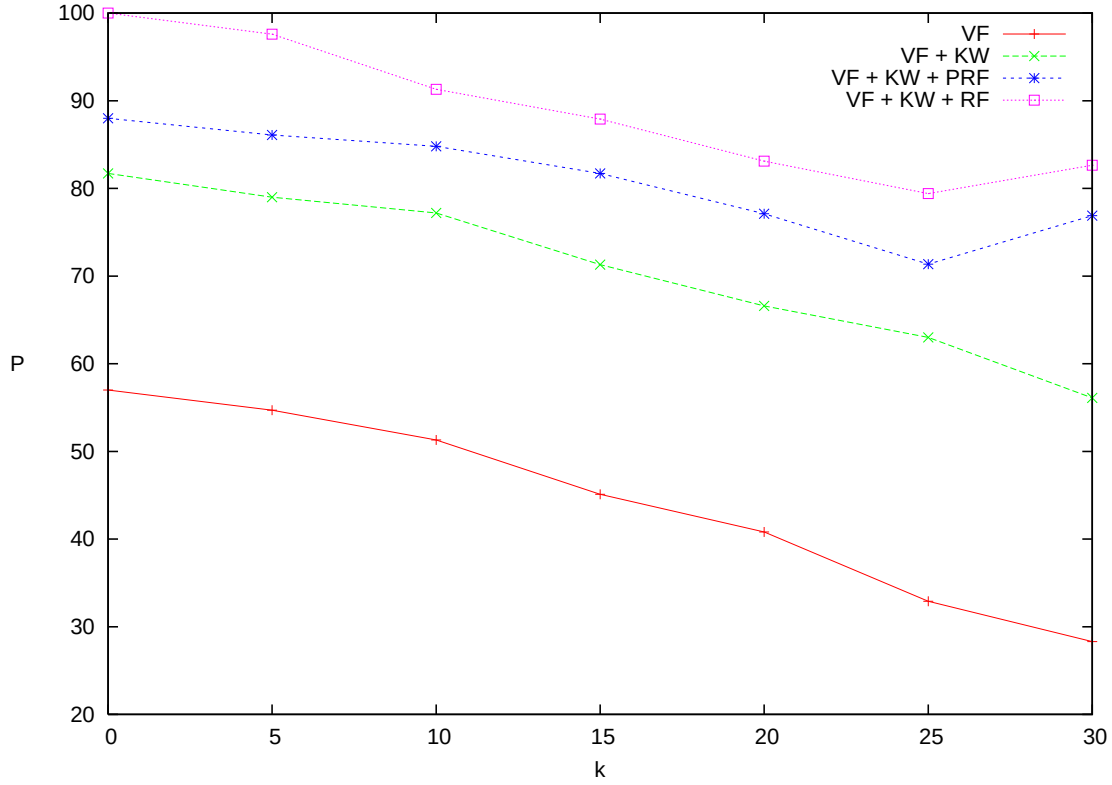



FIGURE 8.3 – Précision $P$ en fonction du rang $k$

D'autre part, le tableau 8.2 montre une image requête (1ère colonne) et les 5 premières images retrouvées pour cette requête. La première ligne montre les images retrouvées sans retour de pertinence. La deuxième ligne montre les images retrouvées suite à une itération du processus de retour de pertinence, avec un exemple positif et un exemple négatif.

Sur la première ligne, sans retour de pertinence, on constate que la dernière image n'est pas pertinente. Cette erreur a eu lieu car cette image, représentant des aliments sur une table, est annotée non seulement par le terme « nourriture »mais aussi par le mot-clé « fleur ». Cette image a été utilisée en exemple négatif pour le processus de retour de pertinence. Et c'est la première image qui est utilisée en exemple positif.

Enfin, sur la deuxième ligne, après une itération de retour de pertinence, toutes les images retrouvées sont pertinentes. Le système a finalement appris que si une image est annotée à la fois par « nourriture »et « fleur »elle n'est pas pertinente pour une image requête annotée par le seul mot « fleur ».

| requête « fleur » | Images retrouvées | | | | |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
|  |  |  |  |  |  |

TABLE 8.2 – Exemple d'une image requête et des images retrouvées par recherche visuo-textuelle

Enfin, le tableau 8.3 les 5 premières images retrouvées pour une requête sur le mot-clé « forêt ». La première ligne montre les images retrouvées sans retour de pertinence. La deuxième ligne montre les images retrouvées suite à une itération du processus de retour de pertinence, avec un exemple positif et un exemple négatif.

Sur la première ligne, sans retour de pertinence, la quatrième image est désignée comme non pertinente. Cette erreur provient du fait que cette image est annotée à la fois par les termes « forêt »et « voiture ». Il en est d'ailleurs de même pour la deuxième image. La première image est désignée comme exemple positif.

Suite à une itération de retour de pertinence, on peut observer les nouveaux résultats sur la deuxième ligne. Le système a finalement appris que si une image est annotée à la fois par « forêt »et « voiture »elle n'est pas pertinente pour une image requête annotée par le seul mot « forêt ». Cependant, dans ce résultat, on pourrait considérer la deuxième image, la troisième, voire la dernière comme non pertinentes. En effet, sur la deuxième image on peut observer des cyclistes sur une route de forêt. La pertinence ou non de cette image pour la requête « forêt »va dépendre de l'utilisateur. Il est est de même pour la troisième image sur laquelle on peut observer un pont, et pour la dernière image sur laquelle on peut observer des gens. Enfin, même si la pertinence de ces images n'est pas claire, elles ont toutes été renvoyées par le système car elles sont annotées par au moins un mot-clé « forêt ».

| Requête | Images retrouvées | | | | |
|---|---|---|---|---|---|
| « forêt » |  |  |  |  |  |
| « forêt » |  |  |  |  |  |

TABLE 8.3 – Exemple d'images retrouvées pour une recherche visuo-textuelle sur le mot-clé « forêt »

## 8.5 Conclusion

Dans ce chapitre, nous avons présenté les résultats d'un modèle probabiliste permettant de rechercher des images sur la base d'informations visuelles et/ou textuelles extraites des images. Ce modèle autorise en plus un processus de retour de pertinence, avec exemples positifs et négatifs. Les résultats ont montré que l'utilisation d'information sémantique, contenue dans les mots-clés annotant certaines images de la base, améliore la précision de recherche obtenue seulement en prenant en compte les caractéristiques visuelles extraite des images.

Par contre, notre modèle présente l'inconvénient d'être très lourd en terme de temps de calcul, d'autant plus qu'il y a d'itérations dans le processus de retour de pertinence. En effet, le temps CPU moyen de réponse pour une image (avec retour de pertinence avec un exemple positif et un négatif, et 10 itérations) est de 6.5 minutes (soit 39 secondes pour une itération), sachant que l'apprentissage des paramètres initiaux du réseau, réalisé hors ligne à partir de toutes les images de la base, a duré 3 heures et 50 minutes. Ces expérimentations ont été menées avec Matlab©, sur un PC avec un processeur Intel Core 2 Duo 2,40 GHz 2,40 Ghz, 2 Go RAM, Windows.

# 9

# Conclusions et projet de recherche

## 9.1 Rappel des objectifs

Il est temps maintenant de conclure. Nous constatons que la démocratisation du multimédia a entraîné une surabondance de données multimédia, et d'images en particulier. L'information est donc disponible, mais il est devenu difficile d'y accéder, d'où la nécessité d'indexer les images et de disposer de techniques pour y accéder efficacement (dans le sens où l'utilisateur est satisfait du résultat) et rapidement.

Les images peuvent être indexées de différentes façons :

- la première, utilisée par la plupart des moteurs de recherche sur Internet, consiste à décrire les images à l'aide de mots-clés (indexation textuelle),
- la deuxième utilise des vecteurs caractéristiques ou des représentations graphiques, extraits des image en analysant leurs formes, couleurs, texture, ... (indexation visuelle)

Le problème commun à ces deux types de méthodes d'indexation est qu'elles contraignent l'utilisateur dans sa façon d'exprimer ses besoins. En effet, dans le cas de l'indexation textuelle, l'utilisateur devra décrire avec des mots-clés l'image qu'il recherche. Au contraire, dans le cas de l'indexation visuelle, l'utilisateur doit disposer d'une image pour exprimer ses besoins. Il ne peut les exprimer sous forme de mots-clés.

Une solution semble donc être de combiner plusieurs informations visuelles et textuelles (indexation visuo-textuelle).

Le défi était de montrer que la combinaison d'informations, et en particulier la combinaison d'informations textuelles et visuelles, améliore la reconnaissance. Pour ce faire, nous avons proposé des techniques de modélisation, recherche et de classification d'images dédiées à la reconnaissance de formes.

## 9.2 Conclusion sur les apports

Compte tenu de la problématique établie, nous avons proposé, dans cette thèse, d'utiliser une approche à base de modèles graphiques probabilistes.

Nous avons proposé de nouveaux modèles permettant de représenter, rechercher, classer des images, dans de grandes bases d'images généralistes ou plus spécialisées (symboles). Ces modèles permettent de représenter des images sur la base de plusieurs caractéristiques visuelles et/ou textuelles, ce qui les rend plus souples pour les utilisateurs. De plus, nous avons montré que la combinaison d'informations visuelles, et la combinaison d'informations visuelles et textuelles, permettent d'améliorer le taux de reconnaissance. Afin d'éviter un travail d'indexation textuelle

trop coûteux pour l'utilisateur, nos modèles ne nécessitent pas que toutes les images soient indexées textuellement : ils sont efficaces sur des bases partiellement annotées.Enfin, nos modèles permettent de traiter des données de différents types : il est possible, dans un même modèle, d'utiliser à la fois des caractéristiques discrètes et continues. Cet aspect nous permettra d'intégrer facilement de nouvelles caractéristiques dans les modèles.

D'autre part, les modèles que nous avons proposés peuvent être utilisés afin d'annoter automatiquement des images ou de compléter les annotations d'images partiellement annotées. Ceci permet, d'une part, de remplacer les données manquantes correspondant aux annotations manquantes. D'autre part, l'annotation permet de réduire le fossé sémantique. Enfin, nos modèles ce sont montrés compétitifs par rapport à des modèles et classificateurs de l'état de l'art, lors de nos évaluations sur de grandes bases d'images.

De plus, notre modèle de recherche d'images intègre un processus de retour de pertinence avec exemples positifs et négatifs. Ce processus a l'avantage d'être inhérent au modèle. L'utilisation du processus de retour de pertinence avec les deux types d'exemples a permis d'améliorer la précision de recherche. Cependant, le modèle proposé est coûteux en temps. Ce modèle offre donc des résultats prometteurs mais des améliorations sont à prévoir (*cf.* section 9.3.1).

Enfin, nous avons proposé un nouveau descripteur de forme, *HRT*, qui s'est révélé, en général, plus performant que des descripteurs couramment utilisés sur des bases de formes variées. Par contre, du fait de sa structure matricielle, *HRT* était moins adapté à nos modèles que des descripteurs fournissant des vecteurs caractéristiques, comme la $\mathcal{R}$-signature $1D$, par exemple. C'est la raison pour laquelle la différence de performance entre ces deux descripteurs, dans nos modèles, n'était pas plus conséquente.

## 9.3 Projet de recherche

### 9.3.1 Projet à court terme

A court terme, nous souhaiterions apporter les améliorations suivantes à nos modèles :

- automatiser l'extraction, à partir de Wordnet, des dépendances entre les termes de nos vocabulaires, représentant des relations sémantiques entre ces termes. En effet, actuellement, cette étape est faîte « à la main »,
- adapter le descripteur *HRT* de façon à ce qu'il s'intègre mieux dans nos modèles. En effet, actuellement, nous concaténons chaque colonne de la matrice *HRT*, *i. e.* chaque histogramme de la matrice de Radon, de façon à obtenir un vecteur caractéristique. Le problème est que, de cette façon, nous perdons l'information portée par la structure matricielle et le descripteur, dans sa représentation vectorielle, devient moins efficace. Afin de pallier ce problème, nous pensons extraire de la matrice *HRT* plusieurs vecteur caractéristiques, un par colonne de la matrice. Chaque vecteur caractéristique sera représenté par un nœud dans le graphe d'un modèle donné. De plus, afin de ne pas perdre l'information portée par la structure matricielle, nous allons tenter de représenter cette structure dans nos modèles : on ajoutera un arc entre deux noeuds correspondant à deux vecteurs colonnes adjacents dans la matrice. On conservera ainsi la dimension angulaire présente dans la matrice *HRT* (chaque colonne correspond à un angle $\theta$ de projection de la transformée de Radon).

### 9.3.2 Projet à long terme

A plus long terme, nous souhaitons mettre en place un nouveau modèle de recherche d'images avec processus de retour de pertinence. En effet, le modèle que nous avons proposé offre des résultats prometteurs et montre que le processus d'inférence associé aux modèles graphiques probabilistes permet d'intégrer facilement, dans de tels modèles, les processus de retour de pertinence. Par contre, notre modèle a montré une complexité en temps élevée, que nous souhaiterions diminuer. A cet effet, nous avons pensé aux réseaux Bayésiens dynamiques, car ils permettent de représenter le facteur temps qui joue en rôle très important dans les processus de retour de pertinence (avec ces processus, le taux de reconnaissance, normalement, croît dans le temps car le modèle apprend de mieux en mieux car il a de plus en plus de données pour apprendre).

Enfin, une autre idée consisterait à mettre en œuvre notre système, de façon différente, en le combinant à un système de recherche d'images ayant une faible complexité. À partir d'une image requête, nous utiliserions ce système pour retrouver les images similaires à la requête, et les ordonner. Ensuite, nous pourrions utiliser notre propre modèle pour annoter l'image requête grâce aux annotations des images retrouvées. Cette combinaison de deux systèmes permettrait une annotation automatique plus rapide et plus précise, sur de plus grandes bases.

# Bibliographie


[Adankon 09] M. M. Adankon & M. Cheriet. *Model selection for the LS-SVM. Application to handwriting recognition.* Pattern Recognition, vol. 42, no. 12, pages 3264–3270, 2009.

[Ahonen 09] T. Ahonen, J. Matas, C. He & M. Pietikainen. *Rotation Invariant Image Description with Local Binary Pattern Histogram Fourier Features.* In SCIA' 09, pages 61–70, 2009.

[Albatal 09] R. Albatal, P. Mulhem, Y. Chiaramella & T. J. Chin. *Comparing image segmentation algorithms for content based image retrieval systems.* In SinFra' 09, 2009.

[Allwein 00] E. L. Allwein, R. E. Schapire, Y. Singer & P. Kaelbling. *Reducing Multiclass to Binary : A Unifying Approach for Margin Classifiers.* Journal of Machine Learning Research, vol. 1, pages 113–141, 2000.

[Andra 05] S. Andra & Y. J. Wu. *Multiresolution Histograms for SVM-Based Texture Classification.* In ICIAR' 05, pages 754–761, 2005.

[Anelli 07] M. Anelli, L. Cinque & Enver Sangineto. *Deformation tolerant generalized Hough transform for sketch-based image retrieval in complex scenes.* Image and Vision Computing, vol. 25, no. 11, pages 1802–1813, 2007.

[Angiulli 07] F. Angiulli. *Condensed Nearest Neighbor Data Domain Description.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 10, pages 1746–1758, 2007.

[Arzhaeva 09] Y. Arzhaeva, D. M. J. Tax & B. van Ginneken. *Dissimilarity-based classification in the absence of local ground truth : Application to the diagnostic interpretation of chest radiographs.* Pattern Recognition, vol. 42, no. 9, pages 1768–1776, 2009.

[Ayed 08] I. Ben Ayed & A. Mitiche. *A Region Merging Prior for Variational Level Set Image Segmentation.* IEEE Trans. Image Processing, vol. 17, no. 12, pages 2301–2311, 2008.

[Azimi Sadjadi 09] M. R. Azimi Sadjadi, J. Salazar & S. Srinivasan. *An Adaptable Image Retrieval System With Relevance Feedback Using Kernel Machines and Selective Sampling.* IEEE Trans. Image Processing, vol. 18, no. 7, pages 1645–1659, 2009.

[Baccini 01] A. Baccini, H. Caussinus & A. Ruiz-Gazen. *Apprentissage Progressif en Analyse Discriminante.* Revue Statistique Appliquée XLIX (4), vol. 49, pages 87–99, 2001.

[Backes 09] A.R. Backes & O.M. Bruno. *A Graph-Based Approach for Shape Skeleton Analysis*. In Image Analysis and Processing – ICIAP 2009, volume 5716, pages 731–738. Springer Berlin / Heidelberg, 2009.

[Ballan 08] L. Ballan, M. Bertini & A. Jain. *A system for automatic detection and recognition of advertising trademarks in sports videos*. In ACM Multimedia, pages 991–992, 2008.

[Barnard 01] K. Barnard & D. Forsyth. *Learning the Semantics of Words and Pictures*. In ICCV' 01, volume 2, pages 408–415, 2001.

[Barnard 03] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei & M. I. Jordan. *Matching words and pictures*. Journal of Machine Learning Research, vol. 3, no. 6, pages 1107–1135, 2003.

[Bayer 72] Rudolf Bayer & E. McCreight. *Organization and maintenance of large ordered indexes*. Acta Informatica, vol. 1, pages 173–189, 1972.

[Bayes 63] T. Bayes. *A essay toward solving a problem in the doctrine of chance*. Philosophical Transactions of the Royal Society, vol. 53, pages 370–418, 1763.

[Belkin 92] N. J. Belkin & W. B. Croft. *Information filtering and information retrieval : two sides of the same coin ?* Commun. ACM, vol. 35, no. 12, pages 29–38, 1992.

[Belongie 02] S. Belongie, J. Malik & J. Puzicha. *Shape Matching and Object Recognition Using Shape Contexts*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 4, pages 509–522, 2002.

[Bennett 07] G. Bennett, F. Scholer & A. Uitdenbogerd. *A comparative study of probabilistic and language models for information retrieval*. In ADC' 08, pages 65–74, 2007.

[Berchtold 96] S. Berchtold, D. A. Keim & H. P. Kriegel. *The X-Tree : An Index Structure for High-Dimensional Data*. In Proceedings of the 22nd International Conference on Very Large Databases, pages 28–39, 1996.

[Berkhin 06] P. Berkhin. *A Survey of Clustering Data Mining Techniques*. In Grouping Multidimensional Data, pages 25–71. Springer, 2006.

[Berrani 04] S. A. Berrani. Recherche aprocimative de plus proches voisins avec contrôle probabiliste de la précision ; application à la recherche d'images par le contenu. Master's thesis, Université de Rennes, février 2004.

[Berretti 00] S. Berretti, A. Del Bimbo & P. Pala. *Retrieval by Shape Similarity with Perceptual Distance and Effective Indexing*. IEEE Trans. Multimedia, vol. 2, no. 4, pages 225–239, 2000.

[Berry 99] M. W. Berry & M. Browne. Understanding search engines : mathematical modeling and text retrieval. Society for Industrial and Applied Mathematics, 1999.

[Bin 08] T. J. Bin, A. Lei, C. Jiwen, K. Wenjing & L. Dandan. *Subpixel edge location based on orthogonal Fourier-Mellin moments*. Image Vision Comput., vol. 26, no. 4, pages 563–569, 2008.

[Bishop 95] Christopher M. Bishop. Neural networks for pattern recognition. Oxford University Press, 1995.

[Bishop 06] C. M. Bishop. Pattern recognition and machine learning. Springer, 2006.

[Blanzieri 08] E. Blanzieri & F. Melgani. *Nearest Neighbor Classification of Remote Sensing Images With the Maximal Margin Principle*. GeoRS, vol. 46, no. 6, pages 1804–1811, 2008.

[Blei 03] David M. Blei & Michael I. Jordan. *Modeling annotated data*. In SIGIR '03, pages 127–134, 2003.

[Boubekeur-Amirouche 08] F. Boubekeur-Amirouche. Contribution a la definition de modeles de recherche d'information flexibles bases sur les cp-nets. Master's thesis, Université Toulouse III - Paul Sabatier, Juillet 2008.

[Boucher 05] A. Boucher & T.L. Le. *Comment extraire la sémantique d'une image ?* In SETIT' 05, 2005.

[Bratkova 09] M. Bratkova, S. Boulos & P. Shirley. *oRGB : A Practical Opponent Color Space for Computer Graphics*. IEEE Computer Graphics and Applications, vol. 29, no. 1, pages 42–55, 2009.

[Breiman 84] L. Breiman, J. Friedman, R. Olshen & C. Stone. Classification and regression trees. Wadsworth and Brooks, 1984.

[Breiman 01] L. Breiman. *Random Forests*. Machine Learning, vol. 45, no. 1, pages 5–32, 2001.

[Buckley 94] C. Buckley, G. Salton & J. Allan. *The effect of adding relevance information in a relevance feedback environment*. In SIGIR' 94, pages 292–300, 1994.

[Buntine 96] W. Buntine. *A guide to the literature on learning probabilistic networks from data*. IEEE Trans. Knowledge And Data Engineering, vol. 8, pages 195–210, 1996.

[Cai 05] D. Cai & C.J. Van Rijsbergen. *Semantic Relations and Information Discovery*. In Intelligent Data Mining, volume 5, pages 79–102, 2005.

[Callan 92] J.P. Callan, W.B. Croft & S.M. Harding. *The Inquery Retrieval System*. In Proceedings of the Third International Conference on Database and Expert Systems Applications, pages 78–83, 1992.

[Carneiro 07] G. Carneiro, A. B. Chan, P. J. Moreno & N. Vasconcelos. *Supervised Learning of Semantic Classes for Image Annotation and Retrieval*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 3, pages 394–410, 2007.

[Carterette 05] B. Carterette & F. Can. *Comparing inverted files and signature files for searching a large lexicon*. Inf. Process. Manage., vol. 41, no. 3, pages 613–633, 2005.

[Chang 01] C. C. Chang & C. J. Lin. *LIBSVM : a library for support vector machines*, 2001.

[Chang 03] Edward Y. Chang, Kingshy Goh, Gerard Sychay & Gang Wu. *CBSA : content-based soft annotation for multimodal image retrieval using Bayes point machines*. IEEE Trans. Circuits Syst. Video Techn., vol. 13, no. 1, pages 26–38, 2003.

[Chehel Amirani 09] M. Chehel Amirani & A.A. Beheshti Shirazi. *A New Approach to Rotation Invariant Texture Analysis Based on Radon Transform.* IEICE, vol. E92-D, no. 9, pages 1736–1744, 2009.

[Chen 06] S.M. Chen, H.C. Lin & Y.C. Chang. *A new method for query reweighting for document retrieval based on neural networks.* International journal of information and management sciences, vol. 17, no. 4, pages 95–110, 2006.

[Chen 09] C. S. Chen, C. W. Yeh & P. Y. Yin. *A novel Fourier descriptor based image alignment algorithm for automatic optical inspection.* J. Vis. Comun. Image Represent., vol. 20, no. 3, pages 178–189, 2009.

[Chiang 09] C. C. Chiang, Y. P. Hung, H. Yang & G. C. Lee. *Region-based image retrieval using color-size features of watershed regions.* J. Vis. Comun. Image Represent., vol. 20, no. 3, pages 167–177, 2009.

[Choi 99] Yong S. Choi & Suk I. Yoo. *Multi-agent learning approach to WWW information retrieval using neural network.* In IUI' 99, pages 23–30, 1999.

[Choy 08] S.K. Choy & C.S. Tong. *Statistical Properties of Bit-Plane Probability Model and Its Application in Supervised Texture Classification.* IEEE Trans. Image Processing, vol. 17, no. 8, pages 1399–1405, 2008.

[Clough 07] P. D. Clough, M. Grubinger, T. Deselaers, A. Hanbury & H. Müller. *Overview of the ImageCLEF 2007 Photographic Retrieval Task.* In CLEF Workshop 2007, volume 5152 of *LNCS*, pages 433–444. Springer, 2007.

[Colot 04] O. Colot, C. Olivier, P. Courtellemont, A. El-Matouat & D. de Brucq. *Information criteria and abrupt changes in probability laws.* Signal Processing VII : Theories and Applications, vol. 1, pages 387–391, 2004.

[Cooper 70] W.S. Cooper. *The Potential Usefulness of Catalog Access Points Other Than Author, Title, and Subject.* Journal of the American Society for Information Science, vol. 21, pages 112–127, 1970.

[Cooper 88] W. S. Cooper. *Getting beyond Boole.* Inf. Process. Manage., vol. 24, no. 3, pages 243–248, 1988.

[Cortes 95] C. Cortes & V. Vapnik. *Support-Vector Networks.* In Machine Learning, pages 273–297, 1995.

[Coustaty 08] M. Coustaty, S. Guillas, M. Visani, K. Bertet & J. M. Ogier. *On the Joint Use of a Structural Signature and a Galois Lattice Classifier for Symbol Recognition.* In Graphics Recognition. Recent Advances and New Opportunities, pages 61–70. Springer-Verlag, 2008.

[Cusano 04] C. Cusano, G. Ciocca & R. Schettini. *Image annotation using SVM.* In Proceedings of SPIE, volume 5304, pages 330–338, 2004.

[Dang 09] E. K. F. Dang, R. W. P. Luk, D. L. Lee, K. S. Ho & S. C. F. Chan. *Optimal Combination of Nested Clusters by a Greedy Approximation Algorithm.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 11, pages 2083–2087, 2009.

[Dash 97] M. Dash & H. Liu. *Feature Selection for Classification.* Intelligent Data Analysis, vol. 1, pages 131–156, 1997.

[Data 96] R. Kohavi Data & R. Kohavi. *Error-Based and Entropy-Based Discretization of Continuous Features.* In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pages 114–119, 1996.

[Dechter 98] R. Dechter. *Bucket elimination : a unifying framework for probabilistic inference.* In Proceedings of the NATO Advanced Study Institute on Learning in graphical models, pages 75–104, 1998.

[Deerwester 90] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas & R. A. Harshman. *Indexing by Latent Semantic Analysis.* Journal of the American Society of Information Science, vol. 41, no. 6, pages 391–407, 1990.

[Demartini 09] G. Demartini, J. Gaugaz & W. Nejdl. *A Vector Space Model for Ranking Entities and Its Application to Expert Search.* In ECIR, volume 5478, pages 189–201, 2009.

[Dempster 77] A. P. Dempster, N. M. Laird & D. B. Rubin. *Maximum Likelihood from Incomplete Data via the EM Algorithm.* Journal of the Royal Statistical Society. Series B (Methodological), vol. 39, no. 1, pages 1–38, 1977.

[Deng 01] Y. Deng & B. S. Manjunath. *Unsupervised segmentation of color-texture regions in images and video.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 8, pages 800–810, 2001.

[Denoeux 07] T. Denoeux & M.-H. Masson. *Dimensionality reduction and visualization of interval and fuzzy data : a survey.* In ISI' 07, 2007.

[Devijver 82] P. A. Devijver & J. Kittler. Pattern recognition : A statistical approach. Prentice Hall, 1982.

[Domingos 96] P. Domingos & M. J. Pazzani. *Beyond Independence : Conditions for the Optimality of the Simple Bayesian Classifier.* In International Conference on Machine Learning, pages 105–112, 1996.

[Duda 73] R. O. Duda, P. E. Hart & D. G. Stork. Pattern classification and scene analysis. Wiley New York, 1973.

[Duda 01] R. O. Duda, P. E. Hart & D. G. Stork. Pattern classification, second edition. Wiley-Interscience, 2001.

[Dudek 97] G. Dudek & J. K. Tsotsos. *Shape representation and recognition from multiscale curvature.* Comput. Vis. Image Underst., vol. 68, no. 2, pages 170–189, 1997.

[Duygulu 02] P. Duygulu, K. Barnard, J. F. G. de Freitas & D. A. Forsyth. *Object Recognition as Machine Translation : Learning a Lexicon for a Fixed Image Vocabulary.* In ECCV' 02, pages 97–112, 2002.

[Efron 04] B. Efron, T. Hastie, I. Johnstone & R. Tibshirani. *Least Angle Regression.* Annals of Statitics, vol. 32, no. 1, pages 407–451, 2004.

[El-Bakry 07] H. M. El-Bakry & N. Mastorakis. *New fast normalized neural networks for pattern detection.* Image Vision Comput., vol. 25, no. 11, pages 1767–1784, 2007.

[Estrada 09] F. J. Estrada & A. D. Jepson. *Benchmarking Image Segmentation Algorithms.* Int. J. Comput. Vision, vol. 85, no. 2, pages 167–181, 2009.

[Everingham 09a] M. Everingham, J. Sivic & A. Zisserman. *Taking the bite out of automated naming of characters in TV video.* Image Vision Comput., vol. 27, no. 5, pages 545–559, 2009.

[Everingham 09b] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn & A. Zisserman. *The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results.* http ://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html, 2009.

[Faloutsos 92] C. Faloutsos. *Signature Files.* In Information Retrieval : Data Structures & Algorithms, pages 44–65. Prentice-Hall, 1992.

[Fan 06] L. Fan & B. Li. *A Hybrid Model of Image Retrieval Based on Ontology Technology and Probabilistic Ranking.* In Web Intelligence, pages 477–480, 2006.

[Fan 08] J. P. Fan, Y. Gao & H. Z. Luo. *Integrating Concept Ontology and Multitask Learning to Achieve More Effective Classifier Training for Multilevel Image Annotation.* IEEE Trans. Image Processing, vol. 17, no. 3, pages 407–426, 2008.

[Fan 09] J.C. Fan, M. Han & J. Wang. *Single point iterative weighted fuzzy C-means clustering algorithm for remote sensing image segmentation.* Pattern Recognition, vol. 42, no. 11, pages 2527–2540, 2009.

[Fang 08] Y. C. Fang & B. W. Wu. *Prediction of the Thermal Imaging Minimum Resolvable (Circle) Temperature Difference with Neural Network Application.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 12, pages 2218–2228, 2008.

[Fellbaum 98] C. Fellbaum, editeur. Wordnet - an electronic lexical database (language, speech, and communication). The MIT Press, 1998.

[Feller 68] W. Feller. An introduction to probability theory and its applications, volume 1. Wiley, 1968.

[Feng 04] S.L. Feng, R. Manmatha & V. Lavrenko. *Multiple Bernoulli relevance models for image and video annotation.* CVPR '04, vol. 2, pages 1002–1009, 2004.

[Fernandez-Maloigne 08] C. Fernandez-Maloigne, N. Richard & A.-S. Capelle-Laizé. *Fuzzy Color Image Segmentation via multi-scale edge/texture based watershed.* In Annual Meeting of TTLA, 2008.

[Filippone 08] M. Filippone, F. Camastra, F. Masulli & S. Rovetta. *A survey of kernel and spectral methods for clustering.* Pattern Recognition, vol. 41, no. 1, pages 176–190, 2008.

[Flanagan 01] J. A. Flanagan. *Self-organization in the one-dimensional SOM with a decreasing neighborhood.* Neural Networks, vol. 14, no. 10, pages 1405–1417, 2001.

[Foltz 92] P. W. Foltz & S. T. Dumais. *Personalized information delivery : an analysis of information filtering methods.* Commun. ACM, vol. 35, no. 12, pages 51–60, 1992.

[Frakes 92] W. B. Frakes & R. A. Baeza-Yates, editeurs. Information retrieval : Data structures & algorithms. Prentice-Hall, 1992.

[François 04] O. François & P. Leray. *Etude Comparative d'Algorithmes d'Apprentissage de Structure dans les Réseaux Bayésiens.* Journal électronique d'intelligence artificielle, vol. 5, no. 39, pages 1–19, 2004.

[François 06] O. François. *De l'identification de structure de réseaux bayésiens à la reconnaissance de formes à partir d'informations complètes ou incomplètes.* PhD thesis, Institut national des sciences appliquées de Rouen, 2006.

[Freixenet 02] J. Freixenet, X. Muñoz, D. Raba, J. Martí & X. Cufí. *Yet Another Survey on Image Segmentation : Region and Boundary Information Integration.* In ECCV' 02, pages 408–422, 2002.

[Friedman 97a] J. Friedman. *On Bias, Variance, 0/1Loss, and the Curse-of-Dimensionality.* Data Mining and Knowledge Discovery, vol. 1, no. 1, pages 55–77, 1997.

[Friedman 97b] N. Friedman, D. Geiger & M. Goldszmidt. *Bayesian Network Classifiers.* Mach. Learn., vol. 29, no. 2-3, pages 131–163, 1997.

[Friedman 98] N. Friedman & M. Goldszmidt. *Learning Bayesian networks with local structure.* In Proceedings of the NATO Advanced Study Institute on Learning in graphical models, pages 421–459, 1998.

[Gao 06] Y. Gao, J. Fan, X. Xue & R. Jain. *Automatic image annotation by incorporating feature hierarchy and boosting to scale up SVM classifiers.* In ACM MULTIMEDIA '06, pages 901–910, 2006.

[Gavilan 08] D. Gavilan, S. Saito & M. Nakajima. *Query-by-Sketch Based Image Synthesis.* IEICE, vol. E91-D, no. 9, pages 2341–2352, 2008.

[Geiger 96] D. Geiger & D. Heckerman. *Knowledge representation and inference in similarity networks and Bayesian multinets.* Artif. Intell., vol. 82, no. 1-2, pages 45–74, 1996.

[Geurts 06] P. Geurts, D. Ernst & L. Wehenkel. *Extremely Randomized Trees.* Machine Learning, vol. 36, no. 1, pages 3–42, 2006.

[Ghosh 06] A. K. Ghosh. *On optimum choice of k in nearest neighbor classification.* Computational Statistics & Data Analysis, vol. 50, no. 11, pages 3113–3123, 2006.

[Gilks-Thomas 94] A. Gilks-Thomas & D. Spiegelhalter. *A language and program for complex Bayesian modelling.* The Statistician, vol. 43, pages 169–178, 1994.

[Goldberger 08] J. Goldberger & T. Tassa. *A hierarchical clustering algorithm based on the Hungarian method.* PRL, vol. 29, no. 11, pages 1632–1638, 2008.

[Grangier 08] D. Grangier & S. Bengio. *A Discriminative Kernel-Based Approach to Rank Images from Text Queries.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 8, pages 1371–1384, 2008.

[Grosky 01] W. I. Grosky & R. Zhao. *Negotiating the Semantic Gap : From Feature Maps to Semantic Landscapes.* In SOFSEM '01, pages 33–52, 2001.

[Grossman 04] D. A. Grossman & O. Frieder. Information retrieval : Algorithms and heuristics. Springer, 2004.

[Gunes 03] V. Gunes, M. Ménard, P. Loonis & S. Petit-Renaud. *Combination, Cooperation And Selection Of Classifiers : A State Of The Art.* IJPRAI, vol. 17, no. 8, pages 1303–1324, 2003.

[Hamarneh 09] Ghassan Hamarneh & Xiaoxing Li. *Watershed segmentation using prior shape and appearance knowledge.* Image Vision Comput., vol. 27, no. 1-2, pages 59–68, 2009.

[Hanbury 08] A. Hanbury. *A survey of methods for image annotation.* J. Vis. Lang. Comput., vol. 19, no. 5, pages 617–627, 2008.

[Haralick 79] R. M. Haralick. *Statistical and structural approaches to texture.* Proceedings of the IEEE, vol. 67, no. 5, pages 786–804, 1979.

[Harman 92] D. Harman, E. A. Fox, R. A. Baeza-Yates & W. C. Lee. *Inverted Files.* In Information Retrieval : Data Structures & Algorithms, pages 28–43. Prentice-Hall, 1992.

[Harrison 09] Robert F. Harrison & Kitsuchart Pasupa. *Sparse multinomial kernel discriminant analysis (sMKDA).* Pattern Recogn., vol. 42, no. 9, pages 1795–1802, 2009.

[Hastie 01] T. Hastie, R. Tibshirani & J. H. Friedman. The elements of statistical learning. Springer, 2001.

[Haykin 98] S. Haykin. Neural networks : A comprehensive foundation (2nd edition). Prentice Hall, 2 edition, 1998.

[Heckermann 91] D. Heckermann. Probabilistic similarity networks. MIT Press, 1991.

[Henrich 89] A. Henrich, H. W. Six & P. Widmayer. *The LSD tree : spatial access to multidimensional and non-point objects.* In VLDB' 89, pages 45–53, 1989.

[Henrion 90] M. Henrion. *An Introduction to Algorithms for Inference in Belief Nets.* In UAI' 89, pages 129–138, 1990.

[Herbert 06] B. Herbert, T. Tuytelaars & L. Van Gool. *SURF : Speeded Up Robust Features.* In 9th European Conference on Computer Vision, 2006.

[Hjouj 08] F. Hjouj & D. W. Kammler. *Identification of Reflected, Scaled, Translated, and Rotated Objects From Their Radon Projections.* IEEE Trans. Image Processing, vol. 17, no. 3, pages 301–310, 2008.

[Hodneland 09] E. Hodneland, X. C Tai & H. H Gerdes. *Four-Color Theorem and Level Set Methods for Watershed Segmentation.* Int. J. Comput. Vision, vol. 82, no. 3, pages 264–283, 2009.

[Holmes 08] Dawn E. Holmes & Lakhmi C. Jain, editeurs. Innovations in bayesian networks : Theory and applications, volume 156 of *Studies in Computational Intelligence.* Springer, 2008.

[Hong 08] G. Hong & Y. Zhang. *Wavelet-based image registration technique for high-resolution remote sensing images.* Comput. Geosci., vol. 34, no. 12, pages 1708–1720, 2008.

[Hsu 02] C. W. Hsu & C. J. Lin. *A comparison of methods for multiclass support vector machines.* IEEE Trans. Neural Networks, vol. 13, no. 2, pages 415–425, 2002.

[Huang 03] J. Huang, J. Lu & C. X. Ling. *Comparing Naive Bayes, Decision Trees, and SVM with AUC and Accuracy.* IEEE International Conf. Data Mining, vol. 0, page 553, 2003.

[Humphreys 09] James Humphreys & Andrew Hunter. *Multiple object tracking using a neural cost function.* Image Vision Comput., vol. 27, no. 4, pages 417–424, 2009.

[Hurtut 08] Thomas Hurtut, Yann Gousseau & Francis Schmitt. *Adaptive image retrieval based on the spatial organization of colors.* Comput. Vis. Image Underst., vol. 112, no. 2, pages 101–113, 2008.

[Jacquemin 02] C. Jacquemin, B. Daille, J. Royanté & X. Polanco. *In vitro evaluation of a program for machine-aided indexing.* Inf. Process. Manage., vol. 38, no. 6, pages 765–792, 2002.

[Jain 00] Anil K. Jain, Robert P. W. Duin & Jianchang Mao. *Statistical Pattern Recognition : A Review.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 1, pages 4–37, 2000.

[Jensen 90] F.V. Jensen, S.L. Lauritzen & K.G. Olesen. *Bayesian updating in recursive graphical models by local computations.* Computational Statistical Quaterly, vol. 4, pages 269–282, 1990.

[Jeon 03] J. Jeon, V. Lavrenko & R. Manmatha. *Automatic image annotation and retrieval using cross-media relevance models.* In SIGIR' 03, pages 119–126, 2003.

[Jin 04] R. Jin, J. Y. Chai & L. Si. *Effective automatic image annotation via a coherent language model and active learning.* In MULTIMEDIA' 04, pages 892–899, 2004.

[Johnson 67] S. Johnson. *Hierarchical clustering schemes.* Psychometrika, vol. 32, no. 3, pages 241–254, 1967.

[Jojic 05] N. Jojic. *A Comparison of Algorithms for Inference and Learning in Probabilistic Graphical Models.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 9, pages 1392–1416, 2005.

[Jones 00] K. Sparck Jones, S. Walker & S. E. Robertson. *A probabilistic model of information retrieval : development and comparative experiments.* Inf. Process. Manage., vol. 36, no. 6, pages 779–808, 2000.

[Jordan 99] M. I. Jordan, editeur. Learning in graphical models. MIT Press, 1999.

[Jordan 02] M.I. Jordan & Y. Weiss. *Graphical models : probabilistic inference.* In Handbook of Neural Networks and Brain Theory. 2nd edition. MIT Press, 2002.

[Jordan 03] M. I. Jordan. *Graphical Models.* Machine Learning, vol. 19, 2003.

[Kang 09] D.K. Kang & K. Sohn. *Learning decision trees with taxonomy of propositionalized attributes.* Pattern Recognition, vol. 42, no. 1, pages 84–92, 2009.

[Kanungo 00] T. Kanungo, R. Haralick, H. Baird, W. Stuezle & D. Madigan. *A Statistical, Nonparametric Methodology for Document Degradation Model Validation*. IEEE Trans. Pattern Anal. Mach. Intell, vol. 22, no. 11, pages 1209–1223, 2000.

[Kaufman 90] L. Kaufman & P.J. Rousseeuw. Finding groups in data an introduction to cluster analysis. Wiley Interscience, 1990.

[Keller 85] J.M. Keller, M.R. Gray & J.A. Givens. *A fuzzy k-nearest neighbor algorithm*. IEEE Trans. Systems, Man, and Cybernetics, vol. 15, no. 4, pages 580–585, 1985.

[Kent 90] A. J. Kent, R. Sacks-Davis & K. Ramamohanarao. *A signature file scheme based on multiple organizations for indexing very large text databases*. JASIS, vol. 41, no. 7, pages 508–534, 1990.

[Kherfi 04] M. L. Kherfi, D. Brahmi & D. Ziou. *Combining Visual Features with Semantics for a More Effective Image Retrieval*. In ICPR '04, volume 2, pages 961–964, 2004.

[Kim 83] J. H. Kim & J. Pearl. *A computational model for combined causal and diagnostic reasoning in inference systems*. In IJCAI-83, pages 190–193, 1983.

[Kim 99] W. Y. Kim & Y. S. Kim. *A new region-based shape descriptor*. Rapport technique, Hanyang University and Konan Technology, 1999.

[Kim 00] H.-K. Kim, J.-D. Kim, D.-G. Sim & D.-Il Oh. *A modified Zernike moment shape descriptor invariant to translation, rotation and scale for similarity-based image retrieval*. IEEE International Conf. Multimedia And Expo, vol. 1, pages 307–310, 2000.

[Kim 04] S.Y. Kim, S.J. Park & M.W. Kim. *Image Classification into Object / Non-object Classes*. In CIVR04, pages 393–400, 2004.

[Kim 07] Soo C. Kim & Tae J. Kang. *Texture classification and segmentation using wavelet packet frame and Gaussian mixture model*. Pattern Recognition, vol. 40, no. 4, pages 1207–1221, 2007.

[Kiranyaz 08] Serkan Kiranyaz, M. Ferreira & Moncef Gabbouj. *A Generic Shape/Texture Descriptor Over Multiscale Edge Field : 2-D Walking Ant Histogram*. IEEE Trans. on Image Processing, vol. 17, no. 3, pages 377–391, 2008.

[Kitamoto 99] Asanobu Kitamoto & Mikio Takagi. *Image Classification Using Probabilistic Models that Reflect the Internal Structure of Mixels*. Pattern Analysis and Applications, vol. 2, no. 1, pages 31–43, 04 1999.

[Kohonen 98] T. Kohonen. *The self-organizing map*. Neurocomputing, vol. 21, no. 1-3, pages 1–6, 1998.

[Kokkinos 09] Iasonas Kokkinos, Georgios Evangelopoulos & Petros Maragos. *Texture Analysis and Segmentation Using Modulation Features, Generative Models, and Weighted Curve Evolution*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 1, pages 142–157, 2009.

[Kondo 09] K. Kondo & S. Hotta. *Color Image Classification Using Block Matching and Learning*. IEICE, vol. E92-D, no. 7, pages 1484–1487, 2009.

[Kotsia 08] I. Kotsia, S. Zafeiriou & I. Pitas. *Texture and shape information fusion for facial expression and facial action unit recognition.* Pattern Recognition, vol. 41, no. 3, pages 833–851, 2008.

[Kotsiantis 07] S. B. Kotsiantis. *Supervised Machine Learning : A Review of Classification Techniques.* Informatica, vol. 31, pages 249–268, 2007.

[Kudo 00] M. Kudo & J. Sklansky. *Comparison of algorithms that select features for pattern classifiers.* Pattern Recognition, vol. 33, no. 1, pages 25–41, 2000.

[Lashkari 09] A. Habibi Lashkari, F. Mahdavi & V. Ghomi. *A Boolean Model in Information Retrieval for Search Engines.* International Conf. Information Management and Engineering„ pages 385–389, 2009.

[LaViola 07] J. LaViola & R. C. Zeleznik. *A Practical Approach for Writer-Dependent Symbol Recognition Using a Writer-Independent Symbol Recognizer.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 11, pages 1917–1926, 2007.

[Lavrenko 03] V. Lavrenko, R. Manmatha & J. Jeon. *A model for learning the semantics of pictures.* In NIPS' 03, 2003.

[Le Borgne 07] H. Le Borgne, A. Guerin-Dugue & N.E. O'Connor. *Learning Mid-level Image Features for Natural Scene and Texture Classification.* CirSysVideo, vol. 17, no. 3, pages 286–297, 2007.

[Leon 07] T. Leon, P. Zuccarello, G. Ayala, E. de Ves & J. Domingo. *Applying logistic regression to relevance feedback in image retrieval systems.* Pattern Recognition, vol. 40, no. 10, pages 2621–2632, 2007.

[Letsche 97] T. A. Letsche & M. W. Berry. *Large-scale information retrieval with latent semantic indexing.* Inf. Sci., vol. 100, no. 1-4, pages 105–137, 1997.

[Lew 06] M.S. Lew, N. Sebe, C. Djerba & R. Jain. *Content-Based Multimedia Information Retrieval : State-of-the-Art and Challenges.* ACM Trans. Multimedia Comput. Commun. Appl., vol. 2, no. 1, pages 1–19, 2006.

[Li 03] J. Li & J. Z. Wang. *Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 9, pages 1075–1088, 2003.

[Li 09a] J. Li & B. L. Lu. *An adaptive image Euclidean distance.* Pattern Recognition, vol. 42, no. 3, pages 349–357, 2009.

[Li 09b] P.J. Li, T. Cheng & J.C. Guo. *Multivariate Image Texture by Multivariate Variogram for Multispectral Image Classification.* PhEngRS, vol. 75, no. 2, pages 147–158, 2009.

[Li 09c] Z. Li, D. Lin & X. Tang. *Nonparametric Discriminant Analysis for Face Recognition.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 4, pages 755–761, 2009.

[Liao 09] S. Liao, M. W. K. Law & A. C. S. Chung. *Dominant Local Binary Patterns for Texture Classification.* IEEE Trans. Image Processing, vol. 18, no. 5, pages 1107–1118, 2009.

[Likforman-Sulem 08] L. Likforman-Sulem & M. Sigelle. *Recognition of degraded characters using dynamic Bayesian networks*. Pattern Recognition, vol. 41, no. 10, pages 3092–3103, 2008.

[Lillholm 09] M. Lillholm & L. D. Griffin. *Statistics and category systems for the shape index descriptor of local 2nd order natural image structure*. Image Vision Comput., vol. 27, no. 6, pages 771–781, 2009.

[Lin 91] X. Lin, D. Soergel & G. Marchionini. *A self-organizing semantic map for information retrieval*. In SIGIR' 91, pages 262–269, 1991.

[Lin 08] X. Lin & X. Wen. *Watershed-Based Texture Image Retrieval*. In IITA' 08, pages 1073–1077, 2008.

[Lin 09a] C. H. Lin, R. T. Chen & Y. K. Chan. *A smart content-based image retrieval system based on color and texture feature*. Image Vision Comput., vol. 27, no. 6, pages 658–665, 2009.

[Lin 09b] L. Lin, T. Wu, J. Porway & Z. Xu. *A stochastic graph grammar for compositional object representation and recognition*. Pattern Recognition, vol. 42, no. 7, pages 1297–1307, 2009.

[Lioma 08] C. Lioma & I. Ounis. *A syntactically-based query reformulation technique for information retrieval*. Inf. Process. Manage., vol. 44, no. 1, pages 143–162, 2008.

[Liu 97] H. Liu & R. Setiono. *Feature Selection via Discretization*. IEEE Trans. Knowledge and Data Engineering, vol. 9, no. 4, pages 642–645, 1997.

[Liu 06a] J. Liu, M. Li, W. Y. Ma, Q. Liu & H. Lu. *An adaptive graph model for automatic image annotation*. In MIR' 06, pages 61–70, 2006.

[Liu 06b] Y. Liu, Z. You & L. Cao. *A novel and quick SVM-based multi-class classifier*. Pattern Recognition, vol. 39, no. 11, pages 2258–2264, 2006.

[Liu 08a] J. S. Liu. Monte carlo strategies in scientific computing. Springer Publishing Company, Incorporated, 2008.

[Liu 08b] R. Liu, Y. Wang, T. Baba, D. Masumoto & S. Nagata. *SVM-based active feedback in image retrieval using clustering and unlabeled data*. Pattern Recognition, vol. 41, no. 8, pages 2645–2655, 2008.

[Liu 08c] Y. Liu, D. Zhang & G. Lu. *Region-based image retrieval with high-level semantics using decision tree learning*. Pattern Recognition, vol. 41, no. 8, pages 2554–2570, 2008.

[Liu 09a] G. Liu, Z. Lin & Y. Yu. *Radon Representation-Based Feature Descriptor for Texture Classification*. IEEE Trans. Image Processing, vol. 18, no. 5, pages 921–928, 2009.

[Liu 09b] J. Liu, M. Li, Q. Liu, H. Lu & S. Ma. *Image annotation via graph learning*. Pattern Recognition, vol. 42, no. 2, pages 218–228, 2009.

[Liu 09c] Y. Liu, X. Chen, C. Zhang & A. Sprague. *Semantic clustering for region-based image retrieval*. J. Vis. Comun. Image Represent., vol. 20, no. 2, pages 157–166, 2009.

[Lladós 01] J. Lladós, E. Martí & J.J. Villanueva. *Symbol Recognition by Error-Tolerant Subgraph Matching between Region Adjacency Graphs.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 10, pages 1137–1143, 2001.

[Lladós 02] J. Lladós, E. Valveny, G. Sanchez & E. Marti. *Symbol Recognition : Current Advances and Perspectives.* In GREC 2003, LNCS 3088, pages 104–128, 2002.

[Lloyd 82] S. Lloyd. *Least squares quantization in PCM.* IEEE Trans. Information Theory, vol. 28, no. 2, pages 129–137, 1982.

[López 08] F. López, J. Miguel Valiente, J. Manuel Prats & A. Ferrer. *Performance evaluation of soft color texture descriptors for surface grading using experimental design and logistic regression.* Pattern Recognition, vol. 41, no. 5, pages 1761–1772, 2008.

[Losada 08] D. E. Losada & L. Azzopardi. *Assessing multivariate Bernoulli models for information retrieval.* ACM Trans. Inf. Syst., vol. 26, no. 3, pages 1–46, 2008.

[Lucas 07] P. Lucas, J. A. Gmez & A. Salmern. Advances in probabilistic graphical models. Springer Publishing Company, Incorporated, 2007.

[Lukasiewicz 63] J. Lukasiewicz. Elements of mathematical logic. Pergamon Press, 1963.

[Luo 08] B. Luo, J. F. Aujol, Y. Gousseau & S. Ladjal. *Indexing of Satellite Images With Different Resolutions by Wavelet Features.* IEEE Trans. Image Processing, vol. 17, no. 8, pages 1465–1472, 2008.

[Luqman 09] M.M. Luqman, T. Brouard & J.Y. Ramel. *Graphic Symbol Recognition Using Graph Based Signature and Bayesian Network Classifier.* In ICDAR' 09, pages 1325–1329, 2009.

[Macqueen 67] J. B. Macqueen. *Some methods of classification and analysis of multivariate observations.* In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pages 281–297, 1967.

[Madjid 04] I. Madjid. Les systèmes de recherche d'informations : modèles conceptuels (traité des sciences et techniques de l'information), chapitre 3. Hermès Science publications, 2004.

[Magalhaes 07] J. Magalhaes & S. Rüger. *Information-theoretic semantic multimedia indexing.* In CIVR '07, pages 619–626, 2007.

[Malki 99] J. Malki, N. Boujemaa, C. Nastar & A. Winter. *Region Queries without Segmentation for Image Retrieval by Content.* In VISUAL' 99, pages 115–122, 1999.

[Manjunath 01] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan & A. Yamada. *Color and texture descriptors.* IEEE Trans. Circuits and Systems for Video Technology, vol. 11, no. 6, pages 703–715, 2001.

[Manjunath 02] B. S. Manjunath, P. Salembier & T. Sikora, editeurs. Introduction to mpeg-7 : Multimedia content description language. Wiley, 2002.

[Marcus 91] R.S. Marcus. *Computer and Human Understanding in Intelligent Retrieval Assistance.* In Proceedings of the ASIS Annual Meeting, volume 28, pages 49–59, 1991.

[Maron 60] M. E. Maron & J. L. Kuhns. *On Relevance, Probabilistic Indexing and Information Retrieval.* J. ACM, vol. 7, no. 3, pages 216–244, 1960.

[Martinez 98] A.M. Martinez & R. Benavente. *The AR face database.* Rapport technique 24, CVC, 1998.

[Mas 07] J. Mas, G. Sanchez, J. Llados & B. Lamiroy. *An Incremental On-line Parsing Algorithm for Recognizing Sketching Diagrams.* In ICDAR' 07, pages 452–456, 2007.

[Mejdoub 09] M. Mejdoub, L. Fonteles, C. BenAmar & M. Antonini. *Embedded lattices tree : An efficient indexing scheme for content based retrieval on image databases.* J. Vis. Comun. Image Represent., vol. 20, no. 2, pages 145–156, 2009.

[Metzler 04] D. Metzler & R. Manmatha. *An Inference Network Approach to Image Retrieval.* In CIVR, pages 42–50, 2004.

[Mezghani 08] N. Mezghani, A. Mitiche & M. Cheriet. *Bayes Classification of On-line Arabic Characters by Gibbs Modeling of Class Conditional Densities.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 7, pages 1121–1131, 2008.

[Mignotte 08] M. Mignotte. *Segmentation by Fusion of Histogram-Based K-Means Clusters in Different Color Spaces.* IEEE Trans. Image Processing, vol. 17, no. 5, pages 780–787, 2008.

[Mille 09] J. Mille. *Narrow band region-based active contours and surfaces for 2D and 3D segmentation.* Comput. Vis. Image Underst., vol. 113, no. 9, pages 946–965, 2009.

[Min 09] R. Min & H. D. Cheng. *Effective image retrieval using dominant color descriptor and fuzzy support vector machine.* Pattern Recognition, vol. 42, no. 1, pages 147–157, 2009.

[Mitchell 96] W. J. T. Mitchell. *Word and Image.* In Critical Terms for Art History. University of Chicago Press, 1996.

[Mitchell 97] T. M. Mitchell. Machine learning, chapitre 6. McGraw-Hill, 1997.

[Müller 08] H. Müller, J. Kalpathy-Cramer, C. E. Kahn, W. Hatt, S. Bedrick & W. R. Hersh. *Overview of the ImageCLEFmed 2008 Medical Image Retrieval Task.* In CLEF, volume 5706 of *Lecture Notes in Computer Science*, pages 512–522. Springer, 2008.

[Moffat 96] A. Moffat & J. Zobel. *Self-indexing inverted files for fast text retrieval.* ACM Trans. Inf. Syst., vol. 14, no. 4, pages 349–379, 1996.

[Moosmann 08] F. Moosmann, E. Nowak & F. Jurie. *Randomized Clustering Forests for Image Classification.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 9, pages 1632–1646, 2008.

[Mori 99] Y. Mori, H. Takahashi & R. Oka. *Image-to-word transformation based on dividing and vector quantizing images with words.* In MISRM' 99, 1999.

[Mothe 94] J. Mothe. *Search mechanisms using neural network model, comparison with vector space model.* In 4th RIAO Intelligent Multimedia Information Retrieval Systems and Management, pages 275–294, 1994.

[Mulhem 06] P. Mulhem & E. Debanne. *A framework for Mixed Symbolic-based and Feature-based Query by Example Image Retrieval.* International Journal for Information Technology, vol. 12, pages 74–98, 2006.

[Murphy 02] K.P. Murphy. *Dynamic Bayesian networks : representation, inference and learning.* PhD thesis, University of California, Berkeley, 2002.

[Myaeng 98] S.H. Myaeng, D-H. Jang, M-S. Kim & Z-C. Zhoo. *A flexible model for retrieval of SGML documents.* In SIGIR' 98, pages 138–145, 1998.

[Navigli 03] R. Navigli & P. Velardi. *An Analysis of Ontology-based Query Expansion Strategies.* In ECML' 03, pages 42–49, 2003.

[Ni 09] K. Ni, X. Bresson, T. Chan & S. Esedoglu. *Local Histogram Based Segmentation Using the Wasserstein Distance.* Int. J. Comput. Vision, vol. 84, no. 1, pages 97–111, 2009.

[Nielsen 09] J. D. Nielsen, R. Rumí & A. Salmerón. *Supervised classification using probabilistic decision graphs.* Computational Statistics and Data Analysis, vol. 53, no. 4, pages 1299–1311, 2009.

[Noda 07] H. Noda & M. Niimi. *Colorization in YCbCr color space and its application to JPEG images.* Pattern Recognition, vol. 40, no. 12, pages 3714–3720, 2007.

[Oja 02] M. Oja, S. Kaski & T. Kohonen. *Bibliography of Self-Organizing Map (SOM) Papers : 1998-2001 Addendum.* Neural Computing Surveys, vol. 1, pages 1–176, 2002.

[Ojala 01] T. Ojala, K. Valkealahti, E. Oja & M. Pietikainen. *Texture discrimination with multidimensional distributions of signed gray-level differences.* Pattern Recognition, vol. 34, no. 3, pages 727–739, 2001.

[Oussalah 08] M. Oussalah. *Content Based Image Retrieval : Review of State of Art and Future Directions.* In IPTA' 08, pages 1–10, 2008.

[Ouyang 09] J. Ouyang, N. Patel & I. Sethi. *Induction of multiclass multifeature split decision trees from distributed data.* Pattern Recognition, vol. 42, no. 9, pages 1786–1794, 2009.

[Paek 00] S. Paek & S. F. Chang. *A knowledge engineering approach for image classification based on probabilistic reasoning systems.* In IEEE International Conf. Multimedia and Expo, pages 1133–1136, 2000.

[Pan 04] J. Y. Pan, H. J. Yang, C. Faloutsos & P. Duygulu. *Automatic multimedia cross-modal correlation discovery.* In KDD' 04, pages 653–658, 2004.

[Pan 09] W. Pan, K. Qin & Y. Chen. *An Adaptable-Multilayer Fractional Fourier Transform Approach for Image Registration.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 3, pages 400–414, 2009.

[Papari 08] G. Papari & N. Petkov. *Adaptive pseudo-dilation for Gestalt edge grouping and contour detection.* IEEE Trans. Image Processing, vol. 17, no. 10, pages 1950–1962, 2008.

[Patil 07] P. M. Patil & T. R. Sontakke. *Rotation, scale and translation invariant handwritten Devanagari numeral character recognition using general fuzzy neural network.* Pattern Recognition, vol. 40, no. 7, pages 2110–2117, 2007.

[Pcekalska 09] E. Pcekalska & B. Haasdonk. *Kernel Discriminant Analysis for Positive Definite and Indefinite Kernels.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 6, pages 1017–1032, 2009.

[Pearl 88] J. Pearl. Probabilistic reasoning in intelligent systems : networks of plausible inference. Morgan Kaufmann Publishers Inc., 1988.

[Pedersen 04] T. Pedersen & S. Patwardhan. *Wordnet : similarity - measuring the relatedness of concepts.* In Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI' 04), pages 1024–1025, 2004.

[Pelleg 00] D. Pelleg & A. W. Moore. *X-means : Extending K-means with Efficient Estimation of the Number of Clusters.* In ICML' 00, pages 727–734, 2000.

[Pena 05] J.M. Pena, J. Bjorkegren & J. Tegner. *Learning dynamic Bayesian network models via cross-validation.* PRL, vol. 26, no. 14, pages 2295–2308, 2005.

[Peyré 09] Gabriel Peyré. *Sparse Modeling of Textures.* J. Math. Imaging Vis., vol. 34, no. 1, pages 17–31, 2009.

[Pham 09] T. T. Pham, L. Maisonnasse, P. Mulhem & E. Gaussier. *Modèle de langue visuel pour la reconnaissance de scènes.* In CORIA' 09, 2009.

[Plaza 09] J. Plaza, A. Plaza, R. Perez & P. Martinez. *On the use of small training sets for neural network-based characterization of mixed pixels in remotely sensed hyperspectral images.* Pattern Recognition, vol. 42, no. 11, pages 3032–3045, 2009.

[Ponte 98] J.M. Ponte & W.B. Croft. *A language modeling approach to information retrieval.* In SIGIR' 98, pages 275–281, 1998.

[Pudil 94] P. Pudil, J. Novovičová & J. Kittler. *Floating search methods in feature selection.* Pattern Recognition Letters, vol. 15, no. 11, pages 1119–1125, 1994.

[Qiu 93] Y. Qiu & H-P. Frei. *Concept based query expansion.* In SIGIR' 93, pages 160–169, 1993.

[Quinlan 93] J. Ross Quinlan. C4.5 : Programs for machine learning (morgan kaufmann series in machine learning). Morgan Kaufmann, 1 edition, 1993.

[Rabiner 90] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. Morgan Kaufmann Publishers Inc., 1990.

[Rahmani 08] R. Rahmani, S. A. Goldman, H. Zhang, S. R. Cholleti & J. E. Fritts. *Localized Content-Based Image Retrieval*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 11, pages 1902–1912, 2008.

[Ramirez 09] J. Ramirez, J.M. Gorriz, R. Chaves, M. Lopez, D. Salas-Gonzales, I. Alvarez & F. Segovia. *SPECT image classification using random forests*. Electronics letters, vol. 45, no. 12, pages 604–605, 2009.

[Ramos-Terrades 08] O. Ramos-Terrades, E. Valveny & S. Tabbone. *On the Combination of Ridgelets Descriptors for Symbol Recognition*. In Graphics Recognition. Recent Advances and New Opportunities, pages 40–50, 2008.

[Ramos-Terrades 09] O. Ramos-Terrades, E. Valveny & S. Tabbone. *Optimal Classifier Fusion in a Non-Bayesian Probabilistic Framework*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 9, pages 1630–1644, 2009.

[Rasiwasia 07] N. Rasiwasia, P. J. Moreno & N. Vasconcelos. *Bridging the Gap : Query by Semantic Example*. IEEE Trans. Multimedia, vol. 9, no. 5, pages 923–938, 2007.

[Rasiwasia 08] N. Rasiwasia & N. Vasconcelos. *Image retrieval using query by contextual example*. In MIR' 08, pages 164–171, 2008.

[Revaud 09] J. Revaud, G. Lavoué & A. Baskurt. *Improving Zernike Moments Comparison for Optimal Similarity and Rotation Angle Retrieval*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 4, pages 627–636, 2009.

[Ribeiro 96] B.A.N. Ribeiro & R. Muntz. *A belief network model for IR*. In SIGIR' 96, pages 253–260, 1996.

[Robert 97] C. Robert. A decision-theoretic motivation. Springer-Verlag, 1997.

[Robert 05] C. Robert & G. Casella. Monte carlo statistical methods (springer texts in statistics). Springer-Verlag New York, Inc., 2005.

[Robertson 76] S.E. Robertson & K. Sparck Jones. *Relevance weighting of search terms*. Journal of the American Society for Information Science, vol. 27, no. 3, pages 129–146, 1976.

[Robertson 77] S. E. Robertson. *The Probability Ranking Principle in IR*. Journal of Documentation, vol. 33, no. 4, pages 294–304, 1977.

[Robertson 09] Stephen Robertson, Milan Vojnovic & Ingmar Weber. *Rethinking the ESP game*. In CHI EA' 09, pages 3937–3942, 2009.

[Robinson 81] J. T. Robinson. *The K-D-B-tree : a search structure for large multidimensional dynamic indexes*. In SIGMOD' 81, pages 10–18, 1981.

[Rocchio 71] J. Rocchio. *Relevance Feedback in Information Retrieval*. In The SMART Retrieval System, pages 313–323. Prentice-Hall, 1971.

[Rodriguez 06] J. J. Rodriguez, L. I. Kuncheva & C. J. Alonso. *Rotation Forest : A New Classifier Ensemble Method*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 10, pages 1619–1630, 2006.

[Romeu 06] J. Mas Romeu, B. Lamiroy, G. Sanchez & J. Llados. *Automatic Adjacency Grammar Generator from User Drawn Sketches*. In Proceedings of 18th International Conference on Pattern Recognition, volume 2, pages 1026–1029, 2006.

[Rosin 99] P. L. Rosin. *Measuring rectangularity.* Mach. Vision Appl., vol. 11, no. 4, pages 191–196, 1999.

[Rosten 06] E. Rosten & T. Drummond. *Machine learning for high-speed corner detection.* In European Conference on Computer Vision, pages 430–443, 2006.

[Roudet 07] C. Roudet, F. Dupont & A.BASKURT. *Multiresolution mesh segmentation based on surface roughness and wavelet analysis.* In Proceedings of SPIE, the International Society for Optical Engineering, 2007.

[Rue 05] H. Rue & L. Held. Gaussian Markov random fields : Theory and applications, volume 104 of *Monographs on Statistics and Applied Probability.* Chapman & Hall, 2005.

[Rui 07] X. Rui, M. L., Z. Li, W.Y. Ma & N. Yu. *Bipartite graph reinforcement model for web image annotation.* In ACM MULTIMEDIA ' 07, pages 585–594, 2007.

[Russell 08] B. C. Russell, A. Torralba, K. P. Murphy & W. T. Freeman. *LabelMe : A Database and Web-Based Tool for Image Annotation.* Int. J. Comput. Vision, vol. 77, no. 1-3, pages 157–173, 2008.

[Safavian 91] S.R. Safavian & D. Landgrebe. *A survey of decision tree classifier methodology.* IEEE Trans. Systems, Man, and Cybernetics, vol. 21, no. 3, pages 660–674, 1991.

[Salton 68a] G. Salton. Automatic information organization and retrieval. McGraw Hill Text, 1968.

[Salton 68b] G. Salton & M. E. Lesk. *Computer Evaluation of Indexing and Text Processing.* J. ACM, vol. 15, no. 1, pages 8–36, 1968.

[Salton 71] G. Salton. The smart retrieval system - experiments in automatic document processing. Prentice-Hall, Inc., 1971.

[Salton 83] G. Salton, E.A. Fox & H. Wu. *Extended Boolean information retrieval.* Communications of ACM, vol. 26, no. 11, pages 1022–1036, 1983.

[Salton 86] G. Salton & M.J. McGill. Introduction to modern information retrieval. McGraw-Hill, Inc., 1986.

[Salton 88] G. Salton. *Syntactic approaches to automatic book indexing.* In Proceedings of the 26th annual meeting on Association for Computational Linguistics, pages 204–210, 1988.

[Samet 05] H. Samet. Foundations of multidimensional and metric data structures (the morgan kaufmann series in computer graphics and geometric modeling). Morgan Kaufmann Publishers Inc., 2005.

[Samet 08] H. Samet. *K-Nearest Neighbor Finding Using MaxNearestDist.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 2, pages 243–252, 2008.

[Sánchez 08] V. G. C. Sánchez, O. O. V. Villegas, G. R Salgado & H. De Jesús Ochoa Domínguez. *Quality inspection of textile artificial textures using a neuro-symbolic hybrid system methodology.* WSEAS. Trans. on Comp., vol. 7, no. 12, pages 1896–1905, 2008.

[Savoy 05] J. Savoy. *Indexation manuelle et automatique : une évaluation comparative basée sur un corpus en langue française.* In CORIA' 05, pages 9–24, 2005.

[Scarpa 09] G. Scarpa, R. Gaetano, M. Haindl & J. Zerubia. *Hierarchical Multiple Markov Chain Model for Unsupervised Texture Segmentation.* IEEE Trans. Image Processing, vol. 18, no. 8, pages 1830–1843, 2009.

[Schettini 01] R. Schettini, G. Ciocca & S. Zuffi. *A survey on methods for colour image indexing and retrieval in image databases.* In Color Imaging Science : Exploiting Digital, 2001.

[Shahrokni 09] A. Shahrokni, T. Drummond, F. Fleuret & P. Fua. *Classification-Based Probabilistic Modeling of Texture Transition for Fast Line Search Tracking and Delineation.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 3, pages 570–576, 2009.

[Shannon 51] C.E. Shannon. *Prediction and Entropy of Printed English.* The Bell System Technical Journal, vol. 30, pages 50–64, 1951.

[Sheng 09] W. Sheng, G. Howells, M. C. Fairhurst, F. Deravi & K. Harmer. *Consensus fingerprint matching with genetically optimised approach.* Pattern Recognition, vol. 42, no. 7, pages 1399–1407, 2009.

[Shi 07] X. Shi & R. Manduchi. *On the Bayes fusion of visual features.* Image Vision Comput., vol. 25, no. 11, pages 1748–1758, 2007.

[Shotton 09] J.D.J. Shotton, J. Winn, C. Rother & A. Criminisi. *TextonBoost for Image Understanding : Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context.* International journal of computer vision, vol. 81, no. 1, pages 2–23, 2009.

[Singhal 09] N. Singhal, Y. Y. Lee, C. S. Kim & S. U. Lee. *Robust image watermarking using local Zernike moments.* J. Vis. Comun. Image Represent., vol. 20, no. 6, pages 408–419, 2009.

[Siskind 07] J. M. Siskind, Jr J. Sherman, I. Pollak, M. P. Harper & C. A. Bouman. *Spatial Random Tree Grammars for Modeling Hierarchal Structure in Images with Regions of Arbitrary Shape.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 9, pages 1504–1519, 2007.

[Smeaton 09] A. F. Smeaton, P. Over & W. Kraaij. *High-Level Feature Detection from Video in TRECVid : a 5-Year Retrospective of Achievements.* In Multimedia Content Analysis, Theory and Applications, pages 151–174. Springer Verlag, 2009.

[Smeulders 00] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta & R. Jain. *Content-Based Image Retrieval at the End of the Early Years.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 12, pages 1349–1380, 2000.

[Smith 90] N. S. Smith, T. W. A. Whitfield & T. J. Wiltshire. *Comparison of the munsell, NCS, DIN, and coloroid colour order systems using the OSA-UCS model.* Color Research and Application, vol. 15, no. 6, pages 327–337, 1990.

[Snyder 02] W. E. Snyder. *NC State University Image Analysis Laboratory Database*, 2002. `http://www.ece.ncsu.edu/imaging/Archives/ImageDatabase/index.html`.

[Stearns 76] S. D Stearns. *On selecting features for pattern classifiers.* In Proceedings of the third international conference on pattern recognition, pages 71–75, 1976.

[Swain 91] M. J. Swain & D. H. Ballard. *Color indexing.* Int. J. Comput. Vision, vol. 7, no. 1, pages 11–32, 1991.

[Syu 94] I. Syu & S. D. Lang. *A competition-based connectionist model for information retrieval using a merged thesaurus.* In CIKM' 94, pages 164–170, 1994.

[Tabbone 02] S. Tabbone & L. Wendling. *Technical Symbols Recognition Using the Two-dimensional Radon Transform.* In ICPR' 02, volume 2, pages 200–203, 2002.

[Tabbone 08] S. Tabbone, O. Ramos-Terrades & S. Barrat. *Histogram of radon transform. A useful descriptor for shape retrieval.* In ICPR' 08, pages 1–4, 2008.

[Teague 79] M. R. Teague. *Image Analysis via the General Theory of Moments.* Journal of the Optical Society of America, vol. 70, no. 8, pages 920–930, 1979.

[Terrades 07] O.R. Terrades, S. Tabbone & E. Valveny. *A Review of Shape Descriptors for Document Analysis.* International Conference on Document Analysis and Recognition, vol. 1, pages 227–231, 2007.

[Tibshirani 96] R. Tibshirani. *Regression Shrinkage and Selection Via the Lasso.* Journal of the Royal Statistical Society. Series B (Methodological), vol. 58, no. 1, pages 267–288, 1996.

[Todorovic 08] S. Todorovic & N. Ahuja. *Region-Based Hierarchical Image Matching.* Int. J. Comput. Vision, vol. 78, no. 1, pages 47–66, 2008.

[Toews 09] M. Toews & T. Arbel. *Detection, Localization, and Sex Classification of Faces from Arbitrary Viewpoints and under Occlusion.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 9, pages 1567–1581, 2009.

[Tola 08] E. Tola, V. Lepetit & P. Fua. *A fast local descriptor for dense matching.* In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8, 2008.

[Torres 09] R. S. Torres, A/ X. Falcão, M. A. Gonçalves, J. P. Papa, B. Zhang, W. Fan & E. A. Fox. *A genetic programming framework for content-based image retrieval.* Pattern Recognition, vol. 42, no. 2, pages 283–292, 2009.

[Turtle 91] H. Turtle & W.B. Croft. *Evaluation of an inference network-based retrieval model.* ACM Trans. Inf. Syst., vol. 9, no. 3, pages 187–222, 1991.

[Turtle 92] H. Turtle & W. B. Croft. *A comparison of text retrieval models.* Comput. J., vol. 35, no. 3, pages 279–290, 1992.

[Tzimiropoulos 09] G. Tzimiropoulos, N. Mitianoudis & T. Stathaki. *A Unifying Approach to Moment-Based Shape Orientation and Symmetry Classification*. IEEE Trans. Image Processing, vol. 18, no. 1, pages 125–139, 2009.

[Umarani 07] C. Umarani, L. Ganesan & S. Radhakrishnan. *Analysis of skin images using texture descriptor by a combined statistical and structural approach*. International Journal of Imaging Systems and Technology, vol. 17, no. 6, pages 359–366, 2007.

[Urban 06] Jana Urban, Joemon M. Jose & Cornelis J. Rijsbergen. *An adaptive technique for content-based image retrieval*. Multimedia Tools Appl., vol. 31, no. 1, pages 1–28, 2006.

[Uz 09] T. Uz, G. Bebis, A. Erol & S. Prabhakar. *Minutiae-based template synthesis and matching for fingerprint authentication*. Comput. Vis. Image Underst., vol. 113, no. 9, pages 979–992, 2009.

[Valveny 04] E. Valveny & P. Dosch. *Symbol Recognition Contest : A Synthesis*. In Graphic Recognition, volume 3088 of Lecture Notes in Computer Science, pages 368–385, 2004.

[Valveny 08a] E. Valveny, P. Dosch, A. Fornés & S. Escalera. Graphics recognition. recent advances and new opportunities, chapitre Report on the Third Contest on Symbol Recognition, pages 321–328. Springer-Verlag, 2008.

[Valveny 08b] E. Valveny, S. Tabbone, O. Ramos-Terrades & E. Philippot. *Performance Characterization of Shape Descriptors for Symbol Representation*. In Graphics Recognition. Recent Advances and New Opportunities, volume 5046, pages 278–287. Springer Berlin / Heidelberg, 2008.

[Vapnik 95] V. Vapnik. The nature of statistical learning theory. Springer-Verlag New York, Inc., 1995.

[Vapnik 06] V. Vapnik. Estimation of dependences based on empirical data : Empirical inference science (information science and statistics). Springer, 2006.

[Vartiainen 08] J. Vartiainen, A. Sadovnikov, J. K. Kamarainen, L. Lensu & H. Kalviainen. *Detection of irregularities in regular patterns*. Mach. Vision Appl., vol. 19, no. 4, pages 249–259, 2008.

[Visani 05] M. Visani, C. Garcia & J. M. Jolion. *Bilinear Discriminant Analysis for Face Recognition*. In ICAPR (2), pages 247–256, 2005.

[Vivaracho-Pascual 09] C. Vivaracho-Pascual, M. Faundez-Zanuy & J. M. Pascual. *An efficient low cost approach for on-line signature recognition based on length normalization and fractional distances*. Pattern Recognition, vol. 42, no. 1, pages 183–193, 2009.

[von Ahn 04] Luis von Ahn & Laura Dabbish. *Labeling images with a computer game*. In CHI' 04, pages 319–326, New York, NY, USA, 2004.

[Von Ahn 06] L. Von Ahn. *Games with a Purpose*. Computer, vol. 39, no. 6, pages 92–94, 2006.

[Wainwright 08] M. J. Wainwright & M. I. Jordan. Graphical models, exponential families, and variational inference. Now Publishers Inc., 2008.

[Wang 06] L. Wang & L. Khan. *Automatic image annotation and retrieval using weighted feature selection.* Multimedia Tools Appl., vol. 29, no. 1, pages 55–71, 2006.

[Wang 07] X. Wang, B. Xiao, J. F. Ma & X. L. Bi. *Scaling and rotation invariant analysis approach to object recognition based on Radon and Fourier-Mellin transforms.* Pattern Recognition, vol. 40, no. 12, pages 3503–3508, 2007.

[Wang 08] X. J. Wang, L. Zhang, X. Li & W. Y. Ma. *Annotating Images by Mining Image Search Results.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 11, pages 1919–1932, 2008.

[Wang 09a] A. Wang, W. Yuan, J. Liu, Z. Yu & H. Li. *A novel pattern recognition algorithm : Combining ART network with SVM to reconstruct a multi-class classifier.* Comput. Math. Appl., vol. 57, no. 11-12, pages 1908–1914, 2009.

[Wang 09b] Y. Wang, T. Mei, S. Gong & X. S. Hua. *Combining global, regional and contextual features for automatic image annotation.* Pattern Recognition, vol. 42, no. 2, pages 259–266, 2009.

[Watt 99] A. H. Watt. 3d computer graphics (3rd edition). Addison-Wesley Longman Publishing Co., Inc., 1999.

[Wei 09] C. H. Wei, Y. Li, W. Y. Chau & C. T. Li. *Trademark image retrieval using synthetic features for describing global shape and interior structure.* Pattern Recognition, vol. 42, no. 3, pages 386–394, 2009.

[Wendling 07] L. Wendling & J. Rendek. *Symbol Recognition Using a 2-class Hierarchical Model of Choquet Integrals.* In ICDAR' 07, pages 634–638, 2007.

[Wenyin 07] L. Wenyin, W. Zhang & L. Yan. *An interactive example-driven approach to graphics recognition in engineering drawings.* Int. J. Doc. Anal. Recognit., vol. 9, no. 1, pages 13–29, 2007.

[White 96] D. A. White & R. Jain. *Similarity Indexing with the SS-tree.* In ICDE' 96, pages 516–523, 1996.

[Wong 08] R. C. F. Wong & C. H. C. Leung. *Automatic Semantic Annotation of Real-World Web Images.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 11, pages 1933–1944, 2008.

[Wu 08] Y.C. Wu, Y.S. Lee & J.C. Yang. *Robust and efficient multiclass SVM models for phrase pattern recognition.* Pattern Recognition, vol. 41, no. 9, pages 2874–2889, 2008.

[Xiao 08] Y. Xiao, H. Dong, W. Wu, M. Xiong, W. Wang & B. Shi. *Structure-based graph distance measures of high degree of precision.* Pattern Recognition, vol. 41, no. 12, pages 3547–3561, 2008.

[Xu 09] Y. Xu, H. Ji & C. Fermüller. *Viewpoint Invariant Texture Description Using Fractal Analysis.* Int. J. Comput. Vision, vol. 83, no. 1, pages 85–100, 2009.

[Yang 05] S. Yang. *Symbol Recognition via Statistical Integration of Pixel-Level Constraint Histograms : A New Descriptor.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 2, pages 278–281, 2005.

[Yang 08] N. C. Yang, W. H. Chang, C. M. Kuo & T. H. Li. *A fast MPEG-7 dominant color extraction with new similarity measure for image retrieval.* J. Vis. Comun. Image Represent., vol. 19, no. 2, pages 92–105, 2008.

[Yavlinsky 05] A. Yavlinsky, E. J. Schofield & S. Rüger. *Automated Image Annotation Using Global Features and Robust Nonparametric Density Estimation.* In CIVR' 05, 2005.

[Yin 09] F. Yin & C. L. Liu. *Handwritten Chinese text line segmentation by clustering with distance metric learning.* Pattern Recognition, vol. 42, no. 12, pages 3146–3157, 2009.

[Ying 09] Z. Ying, L. Guangyao, S. Xiehua & Z. Xinmin. *Geometric active contours without re-initialization for image segmentation.* Pattern Recognition, vol. 42, no. 9, pages 1970–1976, 2009.

[Zadeh 65] L.A. Zadeh. *Fuzzy Sets.* Information Control, vol. 8, pages 338–353, 1965.

[Zambon 06] M. Zambon, R. Lawrence, A. Bunn & S. Powell. *Effect of Alternative Splitting Rules on Image Processing Using Classification Tree Analysis.* PhEngRS, vol. 72, no. 1, pages 25–31, 2006.

[Zhang 00] G. P. Zhang. *Neural networks for classification : a survey.* IEEE Trans. on Systems, Man, and Cybernetics, vol. 30, no. 4, pages 451–462, 2000.

[Zhang 02a] D. S. Zhang & G. Lu. *Shape-based image retrieval using General Fourier Descriptor.* Signal Processing : Image Communication, vol. 17, no. 10, pages 825–848, 2002.

[Zhang 02b] R. Zhang & A. I. Rudnicky. *A large scale clustering scheme for kernel k-means.* In ICPR' 02, pages IV : 289–292, 2002.

[Zhang 03] D. Zhang & G. Lu. *Evaluation of MPEG-7 shape descriptors against other shape descriptors.* Multimedia Syst., vol. 9, no. 1, pages 15–30, 2003.

[Zhang 04a] D. Zhang & G. Lu. *Review of shape representation and description techniques.* Pattern Recognition, vol. 37, no. 1, pages 1–19, 2004.

[Zhang 04b] M. Zhang & Y. Jia. *Probabilistic Classification Based Image Regions Labeling.* In ICIG' 04, pages 100–103, 2004.

[Zhang 05] R. Zhang, Z. Zhang, M. Li, W. Y. Ma & H. J. Zhang. *A Probabilistic Semantic Model for Image Annotation and Multi-Modal Image Retrieva.* In ICCV' 05, pages 846–851, 2005.

[Zhang 06] W. Zhang, L. Wenyin & K. Zhang. *Symbol Recognition with Kernel Density Matching.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 12, pages 2020–2024, 2006.

[Zhang 07] K. Zhang, H. Lu, Z. Wang, Q. Zhao & M. Duan. *A Fuzzy Segmentation of Salient Region of Interest in Low Depth of Field Image.*

In International Multimedia Modeling Conference, pages 782–791, 2007.

[Zhang 08] H. Zhang, Z. Yi & L. Zhang. *Continuous attractors of a class of recurrent neural networks.* Comput. Math. Appl., vol. 56, no. 12, pages 3130–3137, 2008.

[Zhang 09a] F. Zhang, S. Q. Liu, D. B. Wang & W. Guan. *Review article : Aircraft recognition in infrared image using wavelet moment invariants.* Image Vision Comput., vol. 27, no. 4, pages 313–318, 2009.

[Zhang 09b] S. Zhang, B. Li & X. Xue. *Semi-automatic dynamic auxiliary-tag-aided image annotation.* Pattern Recognition, vol. 43, pages 470–477, 2009.

[Zhang 09c] Z. Zhang, S. Ma, H. Liu & Y. Gong. *An edge detection approach based on directional wavelet transform.* Comput. Math. Appl., vol. 57, no. 8, pages 1265–1271, 2009.

[Zhou 08] H. Zhou, Y. Yuan & C. Shi. *Object tracking using SIFT features and mean shift.* Computer Vision and Image Understanding, vol. 113, no. 3, pages 345–352, 2008.

[Ziou 09] D. Ziou, T. Hamri & S. Boutemedjet. *A hybrid probabilistic framework for content-based image retrieval with feature weighting.* Pattern Recognition, vol. 42, no. 7, pages 1511–1519, 2009.

[Zobel 98] J Zobel, A Moffat & K Ramamohanarao. *Inverted files versus signature files for text indexing.* ACM Trans. Database Syst., vol. 23, no. 4, pages 453–490, 1998.