UNIVERSITE JOSEPH FOURIER – GRENOBLE 1

U.F.R. INFORMATIQUE ET MATHEMATIQUES APPLIQUEES

THESE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE JOSEPH FOURIER – GRENOBLE 1

Discipline : Informatique

Présentée et soutenue publiquement le 13 octobre 2005 par

Stéphane BISSOL

TITRE

Indexation symbolique d'images : une approche basée sur l'apprentissage non supervisé de régularités.

Directeurs de thèse : Yves Chiaramella et Philippe Mulhem

Composition du jury :

Président :	M. Jean CAELEN
Rapporteurs :	M. Jacques LE MAITRE
	M. José MARTINEZ
Examinateurs :	M. Yves CHIARAMELLA
	M. Philippe MULHEM

Thèse préparée dans l'équipe MRIM du laboratoire CLIPS-IMAG

(Communication Langagière et Interaction Personne-Système)

Université Joseph Fourier – Grenoble 1

Remerciements :

Tout d'abord, je tiens à remercier M. Jean Caelen pour avoir accepté de présider ma soutenance, mais surtout pour ses commentaires encourageants sur le *fond* de mon travail.

Je remercie vivement M. Jacques le Maître pour avoir accepté de juger mes travaux. Le *feedback* que j'ai reçu de lui, à la fois rigoureux, critique mais enveloppé de bienveillance, s'est révélé précieux.

Je remercie vivement M. José Martinez pour avoir accepté de juger mes travaux. Nous avons été impressionnés par la rigueur et la précision de son rapport. Les questions et objections qu'il a posées constituent par conséquent des directions de recherche privilégiées.

Je tiens à exprimer ici ma gratitude à Philippe Mulhem pour… tout : son investissement à mon égard, la liberté qu'il m'a laissée dans mon travail, sa grande disponibilité, sa rapidité, son aide inconditionnelle et ce, quel que soit le domaine (scientifique ou non). Bref, en un mot : merci Philippe !

Je remercie vivement Yves Chiaramella, également disponible lorsque je le sollicitais. Son expérience, son recul se sont révélés particulièrement utiles pour se concentrer sur l'essentiel, pour faire le tri entre les détails et les aspects réellement importants.

Je remercie Dr Mohan ainsi que tous les membres du projet DIVA qui m'ont accueilli à Singapour. Mes années là-bas furent parmi les meilleures et certainement les plus enrichissantes. Alab, Karthik, Subu, RK, Paula, Nisha : vous ne lirez probablement jamais cette page (de toute façon vous ne lisez pas le français) mais : je ne vous oublierai jamais, et je viendrai vous voir dès que j'en aurais l'occasion.

Je remercie les membres de l'équipes pour leurs questions et commentaires (et disponibilité) lors des réunions, pré-soutenances, etc.

Quant à mes amis au labo : Bérange, Caro, Domi, Duke, Jean, StephA, je suis chanceux de vous connaître.

Merci Coloc d'écouter attentivement mes élucubrations philosophiquo-scientifiques quasi-journalières ;)

Merci Aurélie pour avoir été si patiente avec moi, tu vois c'est fini !

Je réserve le dernier merci à ma famille, le plus gros. Merci pour tout… le soutien, la confiance, et… les gènes ! MERCI !

Suis-je bête ! J'ai failli oublier de remercier  , ce qui aurait été un embarrassant manque de gratitude de ma part. Merci pour !

Résumé

Ce travail porte sur l'indexation automatique de photographies personnelles par des concepts visuels de haut niveau d'abstraction. Nous argumentons en faveur d'une approche basée sur l'apprentissage non supervisé, en mettant en avant les limites de l'apprentissage supervisé. Nous proposons un paradigme d'apprentissage non supervisé basé sur deux types de régularités, correspondant respectivement aux notions de *structure* et de *similarité*. Ces régularités sont apprises à partir d'un flux d'informations visuelles et constituent les nœuds d'un réseau grandissant. Les données d'apprentissage sont recodées en termes des connaissances déjà acquises. Des expérimentations sur des données réelles et synthétisées montrent que notre approche permet de créer une représentation des données pertinente, engendrant une indexation de meilleure qualité. Ces expérimentations très prometteuses permettent d'esquisser des perspectives ambitieuses.

Abstract

This work deals with automatic indexing of personal photographs by highly abstract visual concepts. We make a case for unsupervised learning, by putting forward arguments against the use of supervised learning alone. We introduce a new unsupervised learning paradigm based on two kinds of regularities, implementing respectively to the notion of *structure* and the notion of *contextual similarity*. These regularities are inducted from a stream of visual data and are stored as new nodes in a growing network. New data is systematically recoded in terms of previously acquired knowledge, thus continuously changing the lens through which the data is seen. Experiments on real visual data, as well as synthesized data, show that our approach creates 'relevant' recoding features, yielding better indexing results. From these very promising results, we draw a number of ambitious directions for future works.

Table des matières

I Introduction & Etat de l'art 1

Chapitre 1 La recherche d'images 3

1.1 Recherche d'images : les besoins des utilisateurs 6

1.2 Les difficultés de l'indexation d'images 9

1.2.1 Diversité des niveaux d'interprétation 9

1.2.2 Le fossé sensoriel 12

1.2.3 Le fossé sémantique 14

1.2.4 Interfaces 15

1.2.5 La recherche personnalisée d'images 17

1.2.6 Récapitulatif 20

1.3 Objectifs 21

1.4 Notre approche 22

1.5 Plan 24

Chapitre 2 Les Systèmes de Recherche d'Images par le Contenu 27

2.1 Introduction 27

2.2 Composants d'un SRIC 29

2.3 Représentation des images dans un SRIC 34

2.3.1 Un besoin de simplification 34

2.3.2 Représentation par les couleurs 37

2.3.3 Représentation par les textures 38

2.3.4 Discussion 43

2.4 Une taxonomie des approches 44

2.4.1 Taxonomies dans la littérature 44

2.4.2 Taxonomie basée sur les niveaux d'abstraction 46

2.5 Les systèmes à faible niveau d'abstraction 47

2.5.1 Définition 47

2.5.2 QBIC : le pionnier 48

2.5.3 VisualSEEK 50

2.5.4 Discussion 51

2.6 Les systèmes à moyen niveau d'abstraction 52

2.6.1 Définition 52

2.6.2 FourEyes : Une extension de PhotoBook 53
2.6.2.1 Description globale 53
2.6.2.2 Calcul des groupements 54
2.6.2.3 Apprentissage de concepts 55
2.6.2.4 Résultats 56
2.6.2.5 Discussion 57
2.6.3 Visual Keywords 59
2.6.4 Discussion 60
2.7 Les systèmes à haut niveau d'abstraction 61
2.7.1 Définition 61
2.7.2 Quelques systèmes 62
2.7.2.1 Kansei 63
2.7.2.2 Méta-données temporelles 64
2.7.3 Discussion 64
2.8 Conclusion 65
Chapitre 3 Techniques d'apprentissage automatique pour l'indexation 67
3.1 Introduction 67
3.2 Complexité des problèmes d'apprentissage 69
3.2.1 Problèmes d'apprentissage de type-1 70
3.2.2 Problèmes d'apprentissage de type-2 72
3.2.3 Discussion 73
3.3 Caractéristiques des techniques d'apprentissage 75
3.3.1 Introduction 75
3.3.2 Type du critère de décision 75
3.3.3 Le Biais Inductif 77
3.3.4 Réactivité de l'apprentissage 78
3.3.5 Stabilité de l'apprentissage 78
3.3.6 Résistance au bruit 79
3.3.7 Actif / Passif 80
3.4 Apprentissage automatique : les approches standards 80
3.4.1 Méthode des plus proches voisins 81
3.4.1.1 La méthode 81
3.4.1.2 Application à la recherche d'images 83
3.4.1.3 Les kNN faces aux problèmes de type-2 85
3.4.2 Le « Clustering » 85
3.4.2.1 La méthode 85
3.4.2.2 Applications à la recherche d'image 88
3.4.2.3 Le clustering face aux problèmes de type-2 89
3.4.3 Les réseaux de neurones 90

3.4.3.1 La méthode ... 90
3.4.3.2 Applications à la recherche d'images ... 91
3.4.3.3 Les réseaux de neurones faces aux problèmes de type-2 ... 92
Algorithmes et programmation génétique ... 93
3.4.3.4 La méthode ... 93
3.4.3.5 Application à la recherche d'images. ... 94
3.4.3.6 L'approche génétique face aux problèmes de type-2 ... 96
3.4.4 Les machines à vecteurs de support ... 97
3.4.4.1 La méthode ... 97
3.4.4.2 Adaptation aux problèmes de type-2 ... 98
3.4.4.3 Applications à la recherche d'images ... 99
3.4.5 Récapitulatif ... 100
3.5 Apprentissage automatique : Approches relationnelles ... 102
3.5.1 Apprentissage de CNF : [Moo95] ... 103
3.5.2 Many-Layered Learning : [Utg02] ... 104
3.5.3 Skewing Theory : [Ros05] ... 106
3.5.4 Discussion ... 107
3.6 Conclusion ... 107

II Proposition ... 111

Chapitre 4 Apprentissage constructif hiérarchique de régularités ... 113
4.1 Difficulté de l'indexation ... 114
4.2 Limitations de l'apprentissage supervisé ... 115
4.2.1 Apprentissage supervisé ... 115
4.2.2 Apprentissage semi supervisé ... 117
4.2.3 Inspirations biologiques ... 118
4.2.4 Limitations ... 121
4.2.5 Vers un couplage séquentiel *non supervisé* → *supervisé* ... 123
4.3 Apprentissage non supervisé de régularités ... 125
4.3.1 Régularités et concepts ... 126
4.3.2 Types de régularité ... 128
4.3.3 Disjonction & abstraction ... 133
4.3.4 Un contexte est une régularité ... 135
4.3.5 Définition de la similarité locale ... 136
4.3.6 Renforcement de notre biais inductif ... 139
4.3.6.1 Contrainte 1 : Apprentissage par agglomération ... 140
4.3.6.2 Contrainte 2 : « Le tout remplace les parties » ... 141

4.3.7 Résolution des ambiguïtés 143
4.3.7.1 Contrainte 3 : Préférer les configurations fréquentes 145
4.3.8 Apprendre à différents niveaux d'abstraction 145
4.4 Apprentissage supervisé basé sur des régularités 147
4.5 Classification 149
4.5.1 Classification stricte 149
4.5.2 Classification *souple* 150
4.5.2.1 Degré d'activation : type 'Et' : 152
4.5.2.2 Degré d'activation : type 'Ou' : 153
4.5.2.3 Degrés d'activation & Logique Floue 153
4.6 Intérêt pour la personnalisation 155
4.7 Résumé 157
Chapitre 5 Une instantiation de l'approche 161
5.1 Structure de travail 161
5.1.1 Aperçu 162
5.1.2 Les différents éléments 163
5.1.2.1 Type 164
5.1.2.2 Activation 164
5.1.2.3 Contraintes 164
5.1.2.4 Blocage 165
5.1.2.5 Liens de cooccurrence 165
5.1.2.6 Entrées & sorties 166
5.2 Apprentissage non supervisé 166
5.2.1 Etat initial 167
5.2.2 Apprentissage 167
5.2.2.1 Phase de propagation 167
5.2.2.2 Phase de renforcement 170
5.3 Apprentissage supervisé 174
5.3.1 Traduction des vecteurs 175
5.3.2 Corrélations régularités/concepts 176
5.4 Classification 177
5.5 Conclusion 178
III Expérimentations 179
Chapitre 6 Expérimentations 181
6.1 Rappel des objectifs 181
6.2 Traits de bas niveau & segmentation 182

6.3 Evaluation de l'apprentissage non supervisé 184
6.3.1 Résultats de l'algorithme 1-NN 185
6.3.2 Sans apprentissage non supervisé 186
6.3.3 Avec apprentissage supervisé 188
6.3.4 Conclusion 194
6.4 Evaluation de la réactivité 194
6.4.1 Segmentation & Collection d'images 195
Segmentation : 195
Collection : 195
6.4.2 Protocole 196
6.4.3 Résultats 198
6.4.4 Conclusion 201
6.5 Apprentissage et variabilité intra classe 201
6.5.1 Effets de la variabilité 202
6.5.2 Différentes topologies des réseaux de régularités 204
6.5.3 Conclusion 208
Plasticité : 208
Pertinence des régularités : 208
6.6 Données artificielles 208
6.6.1 Protocole 209
6.6.2 Résultats 209
6.6.3 Conclusion 211
6.7 Discussion et conclusion 211

IV Conclusion 211

Chapitre 7 Conclusion & Perspectives 215
7.1 Synthèse & Contributions 215
7.1.1 La recherche d'images 215
7.1.2 L'apprentissage 215
7.1.3 Limitations des approches classiques 216
7.1.4 Un autre point de vue 216
7.1.5 Hypothèses 217
7.1.6 Biais inductif de notre approche 218
7.1.7 Algorithmes 218
7.1.8 Résultats 219

7.2 Perspectives 220
7.2.1 Expérimenter 220
7.2.2 Utiliser la prévision comme évaluation 221
7.2.3 Eliminer la notion de traits arbitraires 221
7.2.4 Approfondir la réflexion sur notre biais inductif 222
7.2.5 Autres types de données 222

Références bibliographiques 224

Première partie

Introduction & Etat de l'art

Chapitre 1

La recherche d'images

La numérisation

La numérisation, née de l'informatique, aurait probablement paru inconcevable il y a quelques décennies. Elle permet la représentation unifiée d'entités physiques diverses (paysages, sons, phénomènes physique) ou immatérielles (idées, pensées) sous la forme d'information. Cette représentation, libérée de ses contraintes physiques, peut alors être multipliée, transportée, échangée ou modifiée à loisir par l'outil informatique, et tout cela, de manière fiable (Figure 1-1). Le processus inverse, la réification, permet de créer une entité physique similaire à l'original, à partir d'une représentation, *via* des machines (imprimantes, haut-parleurs, moniteurs, robots). Les phénomènes visuels ou auditifs que nous percevons étant absolument essentiels, le fait qu'il soit possible de les capturer de plus en plus fidèlement puis de les dupliquer, modifier, partager, ou transporter à loisir est une révolution.

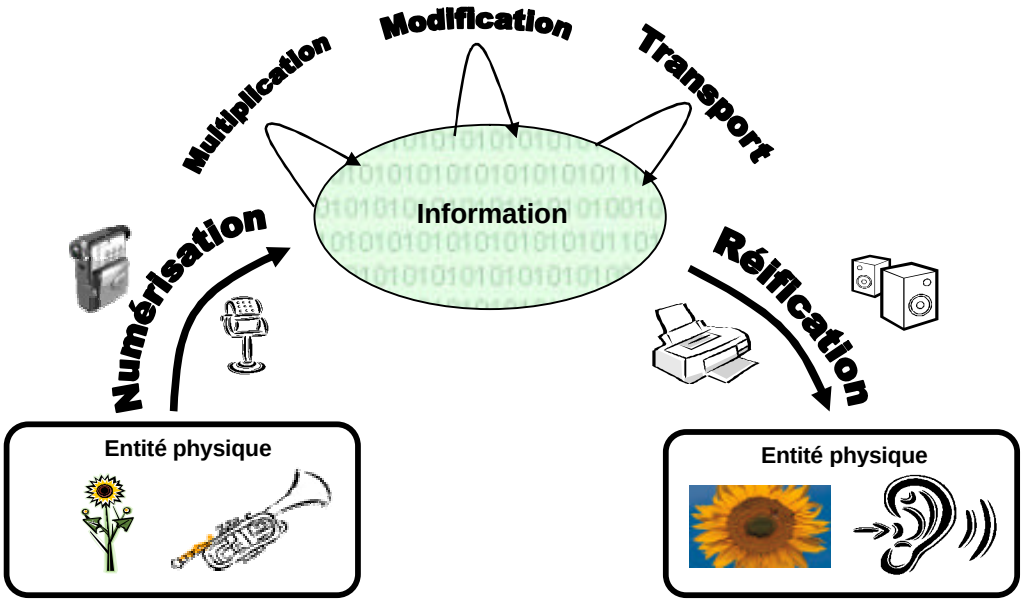


Figure 1-1 La technologie de l'information : entrées (numérisation) et sorties (réification).

La prolifération de l'information

La conséquence de la numérisation croissante et de l'omniprésence de l'outil informatique est que l'information est devenue facile à fabriquer et à manipuler. De plus, pouvant être dupliquée et stockée à faible coût, l'information a quantitativement explosé. Cette disponibilité de l'information, associée à sa masse, provoque des changements dans le comportement des individus (comme par exemple le rejet de l'information qui n'est pas immédiatement ou superficiellement perçue comme pertinente ou plaisante) mais entraîne également de nouveaux besoins. Un de ces besoins, parmi les plus évidents, est à l'origine de la branche de l'informatique à laquelle ont trait les travaux présentés ici : la recherche d'information.

La recherche d'information

Un système de recherche d'informations (SRI) a pour objectif de satisfaire les besoins d'informations d'un utilisateur et son rôle principal est de sélectionner les documents les plus pertinents pour l'utilisateur par rapport à ces besoins. La définition d'un SRI est donc centrée sur la notion de pertinence c'est-à-dire l'adéquation entre le contenu effectif des documents et l'information recherchée par un utilisateur. Pour calculer cette pertinence, il faut traduire les documents et les besoins de l'utilisateur dans un langage commun : c'est respectivement l'indexation et la formulation de requêtes. Il faut ensuite comparer requêtes et documents *via* des mesures de similarité afin de pouvoir présenter à l'utilisateur les documents les plus pertinents. C'est à l'indexation que nous nous intéressons ici : nous pensons en effet que c'est dans ce domaine que le plus reste à faire, particulièrement lorsqu'il s'agit d'images. Par rapport aux documents textuels ou vocaux, l'information pertinente contenue dans les images est en effet difficilement interprétable. L'indexation a pour but de représenter un document, sous la forme d'informations pertinentes (celles sur lesquelles pourront porter les requêtes de l'utilisateur), qui conservent l'essentiel du contenu du document tout en le simplifiant. Par conséquent, pour savoir comment indexer une image, il faut préalablement savoir quelles sont les informations pertinentes contenues dans l'image ou, quelles sont les requêtes susceptibles d'être formulées. Sachant comment on souhaite indexer l'image, il convient de se poser la question de la faisabilité de cette indexation ou, en d'autres termes, s'il est possible d'indexer pertinemment l'image à partir des informations brutes qu'elle contient.

Application aux images

Le processus de numérisation est le principal créateur d'images. La photographie capture, en un point donné, une partie de l'information lumineuse émise par une scène. Selon le type d'appareil utilisé, l'information lumineuse capturée est plus ou moins importante mais dans tous les cas, la perte d'information par rapport à la scène originale est énorme. On perd, entre autres, l'information relative aux volumes et la localisation des objets de la scène dans l'espace. En travaillant sur les images numériques, nous sommes donc condamnés à nous satisfaire d'une très pâle et approximative représentation de la réalité. De plus, les images auxquelles nous nous intéressons dans ce travail, issues de collections personnelles et privées, contiennent une grande quantité d'entités physiques, dont la nature ne peut être prévue à l'avance, ce qui rend difficile l'utilisation de connaissances *a priori*. Toutefois, l'image numérique a l'avantage (sur les photographies « papier ») d'être potentiellement indexable automatiquement, nous discutons plus loin à quel point ce potentiel est exploité.

L'apprentissage

La majorité des êtres humains n'ont pas l'impression, lorsqu'ils regardent leurs photographies, d'effectuer des calculs immensément compliqués pour reconnaître les objets familiers, les personnes qu'ils connaissent ou les lieux où ils sont allés. Ils ont tort, car ces « calculs », prenant en entrée quelques millions d'informations « unitaires » en provenance des cellules rétiniennes et convergeant, en « sortie » vers un ensemble de concepts, de sensations et d'émotions, sont nécessairement complexes. Cette reconnaissance des personnes, objets ou lieux n'est possible que grâce à l'apprentissage[1] qui commence dès le début de la vie. Cet apprentissage est lui-même rendu possible par le fait que le monde, étant contraint et structuré par des 'lois' physiques, présente des *régularités* qui induisent une forte *prévisibilité*. L'apprentissage se basant sur ces régularités, permet l'acquisition progressive de la notion de *similarité*, indispensable à la reconnaissance. En effet, lorsque l'on

[1] Et éventuellement certaines prédispositions génétiques, mais c'est une question particulièrement sujette à débats.

reconnaît une personne par exemple, on ne l'a en réalité *jamais* perçue sous cette forme exacte, de nombreux paramètres ont changé, comme la lumière ambiante, la position, la distance, et la personne elle-même. Pourtant, le fait que cette personne soit reconnue montre bien que nous avons été capables, à partir de millions de « points lumineux », de *reconstruire* le concept de cette personne.

Nous pensons que le projet ambitieux d'imiter mécaniquement ce processus de reconnaissance, pour permettre au final la recherche d'informations, requiert *nécessairement* le recours à des méthodes artificielles d'apprentissage. Sans cet apprentissage, l'unique alternative consisterait à caractériser « manuellement » les différents concepts que l'on souhaiterait voir reconnus, c'est-à-dire expliciter chacun des concepts, sous la forme de règles *si…alors*. Or, non seulement il faudrait probablement écrire une quantité énorme de telles règles, mais surtout, on ne saurait comment les écrire, la raison étant simplement que nous n'avons pas un accès conscient à notre propre mécanisme de perception. Il faut donc que la machine apprenne, au gré de confrontations avec un environnement (la présentation d'images par exemple), à mettre en *relation*, *hiérarchiquement*, un nombre grandissant d'unités d'information qui, seules (non considérées de manière relationnelle), ne contiennent presque pas d'information. Si énormément d'approches ont été développées dans le domaine de l'apprentissage automatique, bien peu s'intéressent à cet aspect relationnel, se reposant plutôt sur des *traits de bas niveau* qui ne caractérisent que quelques types particuliers de relations parmi celles dont nous avons conscience.

Cette introduction se poursuit de la manière suivante : nous nous intéressons en 1.1 aux besoins des utilisateurs d'images, nous essaierons de comprendre en 1.2 pourquoi ces besoins ne sont pas satisfaits à ce jour. Enfin, nous préciserons en 1.3 nos objectifs ainsi qu'un aperçu des moyens mis en œuvre pour les atteindre en 1.4.

1.1 Recherche d'images : les besoins des utilisateurs

Les besoins des chercheurs d'images dépendent évidemment du domaine dans lequel s'effectue la recherche (entre autres) mais on peut tout de même discerner

des tendances, telles qu'elles ressortent dans les études sur le comportement d'utilisateurs lors d'une recherche d'image.

Dans [Orn97] par exemple, l'auteur s'est intéressé à la manière dont les journalistes recherchent des images, et distingue les comportements d'utilisateur suivants.

- Celui qui cherche une image qu'il connaît
- Celui qui cherche des images par navigation. N'ayant qu'une idée vague de ce qu'il cherche, il a besoin de voir des images avant de faire son choix
- Celui qui cherche des images pour accompagner une histoire
- Celui qui cherche des images pour illustrer un document
- Celui qui cherche des images pour leur esthétisme

Cela suggère que des motivations diverses poussent les utilisateurs à chercher des images mais ne dit pas sur quelles caractéristiques de celles-ci portent leurs requêtes. C'est pourtant ce qu'il est nécessaire de prendre en compte pour satisfaire les requêtes en question, et donc pour orienter une stratégie d'indexation.

En nous rapprochant davantage de l'utilisation de systèmes informatiques pour la recherche d'images, quatre types de requêtes journalistiques sont distingués dans [Mar98] :

- Objets concrets (personnes, bâtiments, etc.)
- Thèmes ou abstractions interprétables à partir des photographies (par exemple « Image d'une réunion de travail » ou encore « Photographie de pêcheur de baleines »)
- Images liées à un événement, une actualité ou un film (« Photographies de la dernière élection présidentielle », « Dégâts provoqués par le cyclone El Niño »)
- Photographies connues (« Le Baiser de l'Hôtel de Ville, de Doisneau »)

Remarquons tout de suite que ces requêtes sont hautement abstraites et qu'il n'apparaît nulle part une demande d'images selon leurs textures ou couleurs. Ces

investigations ne concernant que le champ journalistique, peut-être en est-il différemment dans d'autres contextes ?

Dans [McC95], une étude sur la manière dont est recherchée l'information dans les musées (intéressante car les utilisateurs ne se sentant pas limités par un système automatique, expriment vraiment ce qu'ils cherchent) les auteurs ont examiné plus de mille requêtes provenant d'une centaine de musées. Une requête consiste ici en une question posée au personnel d'un musée. Il s'avère que 30 % des requêtes portaient sur des objets, 18 % portaient sur les artistes (et autres informations relatives), 13 % des requêtes concernaient des origines ou techniques de fabrication. Ce contexte est très lié à celui de la recherche d'image puisque les objets recherchés sont perçus visuellement, sans toutefois l'intermédiaire d'un écran. Il ressort de cet exemple que dans le cas de requêtes « libres » portant sur les éléments d'un musée, les recherches reposent sur des caractéristiques sémantiques.

On voit à travers ces exemples que les besoins sont très variés mais que, comme on pouvait s'y attendre, les requêtes des utilisateurs sont majoritairement abstraites. Si l'on peut regretter dans un premier temps que les utilisateurs ne s'intéressent pas aux couleurs et textures de l'image plutôt qu'à sa sémantique, il faut toutefois en tirer la conclusion qu'un système adapté à ces utilisateurs doit être capable d'abstraction. S'il est sûr que les requêtes sont trop diverses pour être complètement prévues, on peut toutefois affirmer que celles-ci sont principalement symboliques. De plus, comme le remarquent les auteurs de [Eak99], les études ci-dessus (comme d'autres) sont biaisées car les utilisateurs se mettent (consciemment ou non) au niveau des systèmes (ils ont une idée de ce dont les systèmes sont capables ou non) : on ne peut donc pas exactement savoir quels sont leurs vrais besoins et l'idée que l'on s'en fait est sans doute sous-estimée.

Notons l'existence de cas dans lesquels l'absence de sémantique dans un système n'empêche toutefois pas la formulation de requêtes sémantiques. Par exemple, le système QBIC est utilisé pour la recherche d'œuvres d'art au State Hermitage Museum de Saint Petersburg. Un utilisateur formule une requête portant sur des couleurs recherchées ou un arrangement de formes géométriques colorées. Par rapport à la classification de [Orn97], on se situe ici dans le cas « Celui qui recherche une image qu'il connaît. ». Certes, cela présuppose la connaissance des couleurs ou

de l'organisation spatiale de formes géométriques apparaissant sur le tableau, mais au final, un certain niveau d'expression est atteint. En effet, il est possible de retrouver un tableau connu ou de rechercher les œuvres d'un artiste à partir d'un exemple. Cependant, une caractéristique importante des tableaux, par rapport aux photographies, est que les styles et techniques diffèrent beaucoup selon les peintres, donnant lieu à des couleurs et textures différentes selon l'artiste[2]. Il existe donc une corrélation entre des caractéristiques non sémantiques (comme les couleurs) et des caractéristiques sémantiques (comme l'auteur de l'œuvre). Malheureusement, ce type de corrélation est rare comme nous allons le voir, particulièrement dans le cas de collections de photographies hétérogènes.

En ce qui concerne les collections de photographies personnelles et leurs utilisateurs, il est très difficile d'énoncer des généralités sur les images, les requêtes et les besoins car, contrairement aux journalistes ou aux visiteurs de musées, il n'existe pas de cadre fixe, de limites à la variabilité des thèmes ou encore de modèle type d'un utilisateur. L'information contenue dans ces images est aussi variée que les activités, évènements, centres d'intérêt des personnes qui les ont photographiées, ce qui suggère l'impossibilité de l'existence d'un système capable de satisfaire toutes les requêtes susceptibles d'être formulées. C'est à cette problématique que nous allons nous intéresser maintenant.

1.2 Les difficultés de l'indexation d'images

1.2.1 Diversité des niveaux d'interprétation

Nous avons vu qu'un système de recherche d'images adapté aux besoins des utilisateurs doit être capable d'abstraction, c'est-à-dire capable d'extraire de la sémantique des images (abstraction par rapport aux simples pixels dont sont constituées les images). On peut définir cette sémantique selon différents niveaux d'abstraction :

1) Objets visibles de l'image (« visage », « bâtiment », « portière », « fenêtre ».)

[2] Beaucoup de tableaux de Van Gogh, par exemple, se reconnaissent du premier coup d'œil grâce à son 'coup de pinceau' très caractéristique, aucune information de plus haut niveau (contenu sémantique du tableau par exemple) n'est donc nécessaire pour identifier l'auteur dans ce cas précis.

2) Objets visibles nommés (« visage : visage de Mozart », « bâtiment : La cathédrale de Chartres »), c'est-à-dire instances de « Objets visibles »
3) Scène (« ville », « plage ») ou agrégat d'objets formant un autre objet (« voiture », « immeuble »)
4) Scène nommée (« photo de ville : Paris », « Photo d'intérieur : Chez ma Grand-mère ») ou agrégat d'objets formant un objet nommé (« ma voiture »)
5) Interprétation abstraite (« Manifestation », « Evénement politique »)
6) Interprétation abstraite nommée (« Manifestation de mai 1968 », « Signature du traité de Yalta »)
7) Interprétations émotionnelles, qualificatifs évoqués par l'image (« Emouvant », « Caricatural », « Révoltant »)

Ce découpage présente sept niveaux d'abstraction, ordonnés selon la distance qui les sépare de la simple perception visuelle[3]. Ainsi, le passage d'un niveau à l'autre requiert une interprétation, nécessairement basée sur des informations. En effet, afin d'extraire une quelconque sémantique d'un ensemble de « points colorés », il est indispensable de posséder des connaissances. Savoir quels traits pertinents extraire d'une image est une connaissance essentielle, toutefois on peut en distinguer bien d'autres. Nous essayons d'identifier ci-dessous les connaissances utiles pour inférer, à partir d'une image, les sept catégories sémantiques présentées ci-dessus :

a) Connaissance sur l'apparence visuelle d'un objet
b) Connaissance fine sur l'apparence d'un objet, permettant d'en distinguer diverses instances (par exemple, faire la différence entre divers visages)
c) Connaissances sur les relations entre les objets (occurrences, cooccurrences, positions relatives) contenus dans une image (« ciel », « fleurs », etc.) et le type d'une image ou un agrégat (« voiture », « plage »)
d) Connaissances sur les relations entre les instances spécifiques d'objets contenues dans l'image et une scène nommée (« photo de ville : Paris ») ou un agrégat nommé (« voiture : ma voiture »)

[3] L'expression « Simple perception visuelle » est utilisée ici pour mettre en évidence le contraste entre différents niveaux d'abstractions ; elle ne suggère pas que la perception est un processus trivial. Il semble au contraire que la perception, loin d'être passive ou simple, soit influencée, conditionnée par l'expérience et les schémas mentaux.

e) Connaissances non visuelles sur les objets (date, appartenance, etc.)

Il est difficile de formaliser les relations existant entre chaque niveau d'abstraction d'une part, et les connaissances nécessaires pour les inférer d'autre part. On peut en effet toujours trouver des contre exemples, des cas où l'inférence est particulièrement facile ou difficile. Par conséquent, nous nous contentons de donner un exemple afin d'illustrer ces relations. La Figure 1-2 schématise les liens unissant une photographie, son interprétation à différents niveaux d'abstraction et les connaissances utilisées. Ainsi, le passage du contenu signal aux interprétations de niveau (1) (« Objet visible de l'image) nécessite des connaissances de type (a), c'est-à-dire des « connaissances sur l'apparence visuelle d'un objet ». Le passage du signal à l'interprétation « Fête nationale à Paris le 14 juillet » quant à lui, requiert l'utilisation hiérarchique de connaissances variées. Les flèches du shéma expriment les informations nécessaires pour le passage d'un niveau à l'autre.

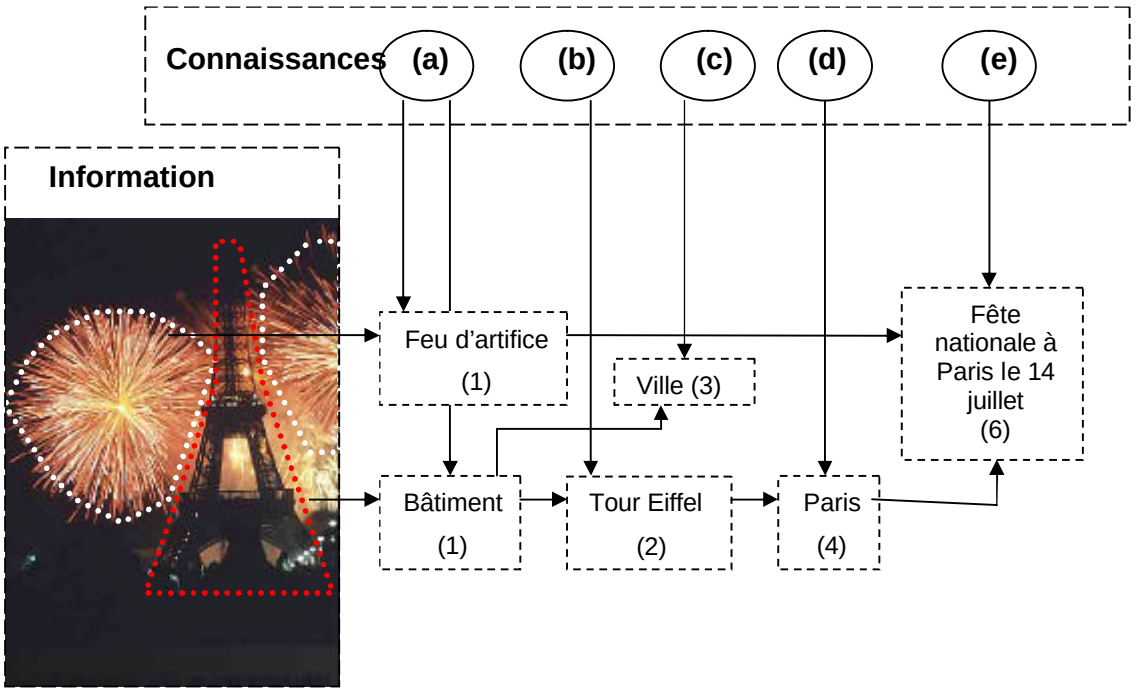


Figure 1-2 : L'extraction de différents niveaux de sémantique (1,2,3,4 et 6) et les connaissances nécessaires à leur inférence (a, b, c, d et e).

Pour se donner une idée de la difficulté du problème, remarquons que plusieurs décennies de recherche (intensive depuis une vingtaine d'années) n'ont pas suffit

pour « résoudre » de manière satisfaisante, ne serait-ce que le niveau (1) d'abstraction.

Les connaissances « non visuelles » sont requises pour beaucoup de niveaux sémantiques mais ne sont pas indispensables pour d'autres. Inversement, les informations visuelles seules ne permettent pas systématiquement d'accéder à la sémantique car une entité physique du monde réel peut avoir énormément d'apparences différentes dans une représentation photographique en deux dimensions. Ce problème est connu sous le nom de *fossé sensoriel*, notion que nous introduisons maintenant.

1.2.2 Le fossé sensoriel

Plaçons nous au niveau d'abstraction (1) décrit précédemment, c'est-à-dire « Objets visibles de l'image ». Comme nous l'avons vu, il faut au minimum le niveau (a) de connaissances pour atteindre ce niveau (c'est-à-dire « Connaissances sur l'apparence visuelle d'un objet »). Ces objets et leurs apparences respectives varient selon la collection d'images considérée. D'après [Sme00], la base d'images dans laquelle se situe un système de recherche d'images varie de manière graduelle entre le *domaine restreint* et le *domaine large*.

D'un *côté* nous avons le domaine restreint :

> ***Définition 1 :*** *Un domaine restreint présente une variabilité limitée et prévisible dans tous les aspects pertinents de son apparence.*

Dans ce domaine, la variabilité inter-images est faible. Généralement, il en est de même pour les conditions de prise de vue. Il peut s'agir d'une base d'images de visages dans laquelle chaque photographie représente un visage centré, sur un fond uniforme et dans des conditions constantes d'illumination. Il peut également s'agir d'images satellites ou de radiographies. Généralement, dans le domaine restreint, les conditions de prise de vue sont contrôlées. De plus, l'extraction d'éléments pertinents est facilitée par le fait que ceux-ci sont généralement connus. La corrélation entre l'apparence des objets et les concepts sémantiques correspondants est forte car ces concepts sont peu nombreux.

A l'autre extrémité se trouve le domaine large :

> ***Définition 2 :*** *Un domaine large présente une variabilité illimitée et imprédictible dans son apparence, même à sémantique égale.*

Dans le domaine large, les images sont polysémiques et leur sémantique n'est décrite que partiellement. Ce peut être le cas par exemple lorsque il existe dans l'image des objets de catégories inconnues ou pouvant donner lieu à différentes interprétations (une région bleue uniforme pourrait être interprétée comme 'ciel', 'lac', 'voiture', 'mur', etc.). Les bases d'images hétérogènes comme les bases de photographies personnelles se rapprochent du domaine large. Les images d'une collection personnelle proviennent en effet de lieux différents, donc présentent des objets différents ; les conditions de prise de vue ne sont soumises qu'à très peu de contraintes, les objets apparaissant sur les images prennent donc de multiples apparences (taille, luminosité, netteté, orientation varient). Prendre en compte le domaine et déterminer sa position entre restreint et large est utile pour déterminer le contexte applicatif du système à concevoir. Par exemple, beaucoup d'approches pour la reconnaissance de visages ([Sam92], [Che95]) utilisent des modèles géométriques et statistiques, ce qui n'est pas possible à l'heure actuelle pour un domaine large, dans lequel le nombre de paramètres requis serait gigantesque. Il existe donc une difficulté qui augmente proportionnellement avec la largeur du domaine, que Smeulders caractérise par le terme de *fossé sensoriel* :

> **Définition 3 :** *Le fossé sensoriel est la disparité qui existe entre un objet physique et sa représentation numérique tirée d'une scène.*

Le fossé sensoriel est tel que dans le cas général, il n'est pas possible de reconnaître de manière robuste les objets en ne se basant que sur leur apparence dans l'image. La variabilité de l'objet « table » par exemple est énorme dans le domaine large : une table peut être vue de près, de loin, sous différentes illuminations, différents angles, peut être constituée d'une multitude de matières différentes présentant toutes sortes de textures et de couleurs, peut être entière, occluse, floue, nette. Il faut donc soit se contenter d'une représentation très approximative de l'objet table, soit construire un modèle complexe de celui-ci, soit utiliser des connaissances contextuelles permettant d'inférer qu'il s'agit d'une table.

Le domaine dans lequel nous définissons et validons notre approche est celui des photographies personnelles, qui rentre dans le cadre du « domaine large ». La diversité des objets rencontrés dans les photographies personnelles, ainsi que la variabilité de leurs apparences est très importante. De plus, la taille des bases d'images personnelles, sans être énorme, est également importante (plusieurs milliers d'images à quelques dizaines de milliers), ce qui n'est pas la moindre des contraintes.

1.2.3 Le fossé sémantique

Toujours d'après l'étude décrite dans [Sme00], le fossé sémantique est la cause des désillusions qui ont succédé aux premiers systèmes de recherche d'images. Il est défini de la manière suivante :

> ***Définition 4 :*** *Le fossé sémantique est le manque de concordance entre l'information que l'on peut extraire des données visuelles et l'interprétation des mêmes données qu'en fait un utilisateur dans un contexte donné.*

Par exemple, un utilisateur recherche des images où apparaissent des personnes particulières (famille, amis) mais le système ne peut que détecter les visages « en général » (visage humain). L'existence du fossé sémantique est indirectement causée par le fossé sensoriel : le fossé sensoriel rend extrêmement difficile l'extraction de sémantique (objets sur l'images, activités, personnes, etc.) à partir des images, les SRIC[4] en sont donc également peu capables, or nous avons vu précédemment que c'est justement la sémantique contenue dans les images qui intéresse les utilisateurs plutôt que de simples caractéristiques visuelles[5].

La description linguistique d'une image est une tâche épineuse, peut-être même impossible : rien ne peut remplacer le fait de voir une image. Un utilisateur recherche une image selon les objets qu'elle contient mais également selon le message ou les connotations qu'elle véhicule. La description automatique des images, quant à elle,

---

[4] SRIC : **S**ystème de **R**echerche d'**I**mages par le **C**ontenu

[5] On peut d'ailleurs remarquer que pour la recherche de documents textuels, bien que le fossé sémantique existe également, il est bien moindre, car dans ce domaine le fossé sensoriel est réduit à des problèmes d'interprétation de mots polysémiques, de regroupement de mots en 'formes' ou unités sémantiques, etc.

est basée sur des caractéristiques visuelles et ces deux approches sont souvent complètement déconnectées. L'approche la plus classique pour tenter de résoudre ce problème est d'attacher aux images des mots clefs pour caractériser leur sémantique (soit à l'image entière, soit aux régions qui la composent). Malheureusement ces approches ne sont pas triviales et des problèmes émergent : attacher des mots clefs à des régions requiert un apprentissage préalable puis une classification et ces procédés peuvent être coûteux. De plus, il n'est pas possible de décrire exhaustivement l'image de cette manière ; il faut donc faire des choix quant au vocabulaire d'indexation utilisé. Une manière de pallier à ces problèmes nous paraît être la **personnalisation** du SRIC par l'utilisateur, ce dont nous allons parler en détails. Une alternative à cette approche est de caractériser la sémantique des images en utilisant des informations textuelles additionnelles (comme le texte qui entoure une image dans un document structuré). Des exemples de cette approche peuvent être trouvés dans [Cha97] ou [Rui98]. Naturellement, cette approche est attractive mais n'est applicable que lorsque des informations textuelles supplémentaires sont disponibles, ce qui n'est généralement pas le cas dans le cadre des photographies personnelles.

1.2.4 Interfaces

Les fossés que nous venons de décrire constituent autant de handicaps majeurs pour les systèmes de recherche d'images. Il existe une autre difficulté importante : la manière dont l'utilisateur et le système interagissent. D'un *côté*, il y a l'utilisateur, avec une définition mentale de ce qu'il recherche, constituée de toutes sortes de critères plus ou moins abstraits. De l'autre *côté* se trouve un système où chaque image est indexée de manière prédéterminée et fixée. L'interface se situe entre les deux et permet à l'utilisateur d'exprimer ses besoins sous une forme intelligible par le système. Nous distinguons trois types principaux d'interfaces, illustrées Figure 1-3 : interfaces par navigation, composition et spécification.

Une des solutions les plus courantes est de rechercher des images en donnant comme requête une « image exemple » de ce qu'on recherche ; le système propose alors un ensemble d'images similaires à l'exemple donné, puis l'utilisateur recommence l'opération en choisissant une des images présentées comme nouvel

exemple. C'est une interface de type **navigation**. Le problème, outre le fait que l'utilisateur n'a pas nécessairement d'exemple à sa disposition, est de savoir ce que l'on entend par « les images les plus similaires à l'exemple donné » puisque seul l'utilisateur sait ce qu'il cherche (un objet, une personne, un type d'image, une teinte globale, etc.). Certes, ce type d'approche, qui correspond à un bouclage de pertinence, aide à désambiguïser la requête. Mais le contenu d'une image est complexe et la représentation de cette même image très limitée, ce qui rend difficile la compréhension de la requête[6]. Par ailleurs, dans le domaine de la recherche de documents textuels, où le fossé sémantique est beaucoup moins profond, la technique de recherche à partir d'un « texte exemple » est quasi inexistante au niveau des modalités d'expression d'une requête.

Une alternative à « l'image exemple » est la **composition** directe par l'utilisateur de l'apparence visuelle de l'image cible, soit en choisissant des couleurs et des textures parmi des palettes, soit en dessinant une esquisse de l'image recherchée ([Fli95]). Cependant, l'idée qu'un utilisateur se fait d'une image recherchée est difficile à exprimer sous forme de caractéristiques visuelles grossières dans la mesure où la recherche repose généralement sur des caractéristiques sémantiques qui n'ont que très peu à voir avec les caractéristiques visuelles (sauf cas particuliers). Il existe cependant des approches (comme [Lim00]) où la composition est sémantique ; la palette de couleurs ou de textures est alors remplacée par une palette de termes sémantiques que l'utilisateur peut disposer spatialement pour formuler sa requête.

Une autre possibilité, qui requiert une indexation sémantique des images, est une interface dans laquelle l'utilisateur **spécifie** directement (sous forme textuelle par exemple) les concepts qu'il recherche (au niveau global ou local). Par exemple, dans [Vai01] ou [Tow00], les images sont indexées par la catégorie à laquelle elles appartiennent et peuvent donc être retrouvées directement en terme de catégories (par exemple « Image d'intérieur » ou « plage »).

[6] Combien d'images exemples faudrait-il avant qu'un système « comprenne » que l'utilisateur (un journaliste par exemple) recherche des images de ville contenant des fleurs ou de la verdure mais pas de voitures ?

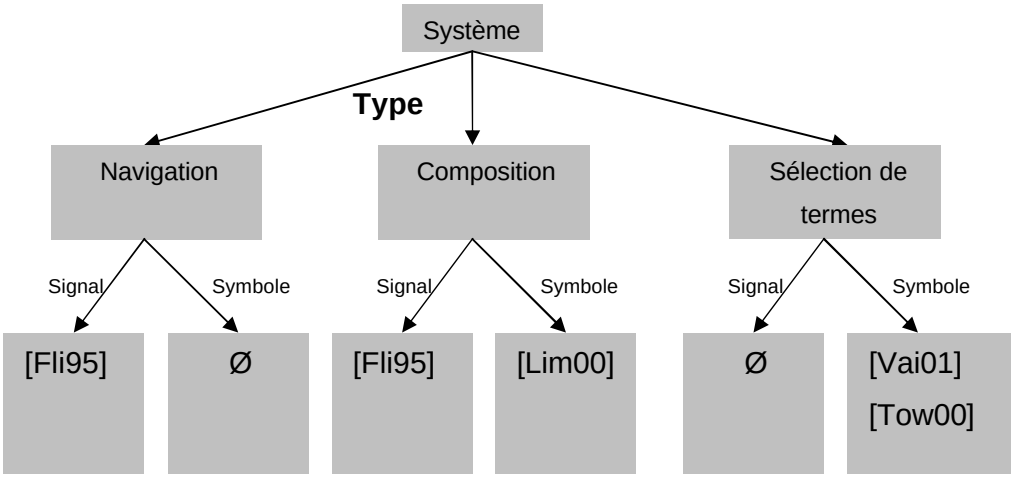


Figure 1-3 : Types d'interface pour la recherche d'images et exemples de systèmes.

1.2.5 La recherche personnalisée d'images

Les difficultés que nous venons d'aborder sont liées à la recherche et à l'indexation des images en général. Le contexte de la recherche personnalisée que nous avons choisi, tout en incluant ces difficultés, en implique de nouvelles. Nous détaillons maintenant ce contexte et les problèmes particuliers qui en découlent.

L'adaptation du système à l'utilisateur est au cœur de l'idée de personnalisation. Très (trop) souvent, c'est à l'utilisateur de s'adapter au système : le système indexe les images d'une manière arbitraire et fixée (donc non modifiable) et les requêtes qu'il autorise le sont également ; l'utilisateur n'a donc d'autre choix que de se plier à ces limitations, en convertissant, dans la mesure du possible (et au prix d'efforts cognitifs considérables), son besoin d'information en une ou plusieurs requêtes exprimables dans le système.

Nous souhaitons la situation inverse : l'utilisateur a des besoins d'information spécifiques, il a donc besoin d'une indexation spécifique de ses images et, naturellement, de requêtes spécifiques ; pour cela le système doit se plier à ces exigences. Cela nous emmène à la première difficulté de la recherche personnalisée des images :

> *<u>Difficulté 1</u> : En recherche personnalisée d'images, le système doit être capable d'apprendre (sous la supervision de l'utilisateur) à indexer les images selon des*

concepts définis par l'utilisateur d'une part, et inclure ces concepts comme nouvelles dimensions dans l'espace des requêtes formulables d'autre part.

L'utilisateur doit donc « faire apprendre » au système des concepts qu'il a en tête, comme « Ciel », « Mon pull-over vert et rouge » ou « Images prises dans mon jardin ». Nous ne formulons aucun *a priori* sur le « type » d'utilisateur, par conséquent et, par précautions, nous considérerons par défaut que celui-ci n'est expert ni en informatique, ni en SRIC, ni en photographie et qu'il peut commettre des erreurs ou se raviser sur des choix précédents. L'apprentissage, dans ces conditions, peut difficilement être explicite[7] et l'apprentissage par exemples est largement préférable. Nous pouvons maintenant formuler une seconde difficulté :

Difficulté 2 : L'utilisateur, supposé non expert en informatique ou en photographie, fait apprendre des concepts au système en lui fournissant des exemples. La pertinence de ces exemples (du point de vue des traitements informatiques dont ils seront l'objet) est considérée comme inconnue, donc potentiellement mauvaise. Il n'est pas exclu que certains de ces exemples soient faux et ne correspondent donc pas au concept qu'ils devaient illustrer.

De plus :

Difficulté 3 : Nous ne faisons aucune hypothèse particulière sur la patience de l'utilisateur ni sur sa tolérance et magnanimité envers le système, même si la tâche de ce dernier est difficile. Par conséquent, le système doit apprendre vite (i.e. peu d'exemples à fournir, rapidité des retours système) et indexer rapidement la collection.

Dû à la grande variabilité des lieux, des goûts et des activités, on ne peut prédire le contenu des collections d'images personnelles ; il existe certes des entités dont les occurrences sont fréquentes, comme le « Ciel », les « Personnes » ou les « Arbres », mais le contenu des collections reste imprévisible :

[7] Un apprentissage explicite consiste, par opposition à l'apprentissage par l'exemple, à modéliser un concept en en donnant explicitement toutes les caractéristiques. Par exemple, « Ciel » : région souvent bleue, parfois blanche, parfois rouge, dont la texture…

<u>Difficulté 4</u> : Le contenu sémantique d'une collection d'images personnelles est imprévisible, par conséquent, les concepts que l'utilisateur veut faire apprendre au système sont également imprévisibles.

Lorsqu'un utilisateur définit un concept (« Mon chat »), les exemples qu'il donne sont « similaires » car ils sont tous des instances d'un même concept, le défi pour le système est de trouver dans quelle mesure ils sont similaires. Plus important encore, le système **doit** être capable de généraliser efficacement, sans nécessiter des centaines d'exemples : sans généralisation, l'apprentissage est inutile. Pour généraliser correctement, il convient, au minimum, de « percevoir » et « repérer » les traits, les propriétés qui font que les instances d'un concept sont similaires ; ensuite, il s'agit pour le système de mesurer la similarité entre instances d'une manière cohérente par rapport à l'utilisateur. D'où :

<u>Difficulté 5</u> : La similarité (visuelle) entre entités visuelles perçue par l'utilisateur se base (au moins) sur sa perception (des couleurs, des formes et textures) et sur sa capacité à généraliser. La base de toute classification automatique étant justement la mesure de similarité, il est important que le système ait une « perception visuelle » et une « méthode de généralisation » se rapprochant le plus possible de celles de l'utilisateur.

L'apprentissage d'un concept, sa « qualité » et les capacités de généralisation sont liés à la taille de l'ensemble d'apprentissage disponible. Or, dans notre contexte où l'utilisateur définit les concepts « à partir de rien », il est probable que la qualité de l'apprentissage et les capacités de généralisations soient, au début, médiocres (i.e. mauvaises hypothèses)[8]. Il faut donc que le système ait la possibilité de remettre en cause ses hypothèses, de manière incrémentale :

*<u>Difficulté 6</u> : L'utilisateur fournit les données utiles à l'apprentissage (exemples) avec parcimonie et sporadiquement dans le temps (à l'opposé de l'apprentissage « offline »). Le système doit donc être capable d'apprendre incrémentalement **ET***

[8] On peut imaginer que les premiers exemples fournis par l'utilisateur pour faire apprendre le concept « Ciel » au système ne soient que des régions uniformément bleues. Dans ce cas, la couleur seule (= bleu) peut être suffisamment discriminante pour l'ensemble courant d'apprentissage, mais cette hypothèse ne permettra pas une généralisation satisfaisante : la remise en cause de cette hypothèse DOIT donc être possible.

de remettre en cause cet apprentissage qui peut s'avérer peu fiable (car basé sur un ensemble d'apprentissage trop réduit).

Cette difficulté réduit drastiquement le nombre d'algorithme envisageable pour mener à bien notre apprentissage ; bon nombre d'algorithmes sont en effet non incrémentaux (i.e. l'ajout d'un nouvel exemple nécessite de « refaire » l'apprentissage avec **tous** les exemples). Parmi les algorithmes incrémentaux, beaucoup nécessitent un réapprentissage complet afin de remettre en cause leur hypothèse ; l'ajout d'exemples ajuste l'hypothèse courante mais ne la remet pas en cause.

Nous terminerons en disant que dans notre contexte, les images d'une collection, représentant des mémoires personnelles, ont souvent énormément d'importance pour leur propriétaire. Les premières photographies d'un enfant, un mariage, des vacances exotiques, des photographies d'amis d'enfances, etc. ont, pour leur possesseur, une valeur inestimable et peuvent être très « émotionnellement chargées ».

1.2.6 Récapitulatif

Nous avons identifié divers types de niveaux sémantiques contenus dans les images, allant des objets qu'elles contiennent jusqu'aux émotions ou impressions qu'elles nous suggèrent. Puis, nous avons évoqué le problème majeur du fossé sensoriel qui nous dit qu'à partir d'une représentation pauvre et en deux dimensions (une photographie) d'une scène réelle, il est extrêmement difficile d'identifier les entités présentes dans la scène, cela à cause de la multitude des apparences possibles d'une même entité. Notons que ce problème se pose dès le premier des niveaux sémantiques que nous avons évoqués. La conséquence est que les systèmes existants peinent à atteindre le niveau le plus basique de sémantique, alors que de l'autre côté, les utilisateurs expriment leurs besoins à des niveaux sémantiques élevés : c'est le fossé sémantique. De manière évidente, le problème qui consiste à créer un système capable de comprendre globalement les images comme nous en

sommes capables n'est pas solvable aujourd'hui[9], puisque même des sous problèmes beaucoup plus simples (reconnaître des visages, des organes, des mots, etc.) ne sont toujours pas entièrement résolus aujourd'hui.

Nous avons finalement vu que lorsque nous contraignons notre contexte à celui de la recherche personnalisée d'images, de multiples difficultés supplémentaires surgissent, liées principalement à l'adaptation à l'utilisateur.

Malgré tous ces obstacles, nous pensons qu'il est toutefois possible d'assister de manière satisfaisante les utilisateurs dans leurs recherches, à condition d'explorer de nouvelles voies.

1.3 Objectifs

Nos travaux se focalisent principalement sur la tache d'indexation des images. L'indexation, en amont du processus de recherche d'images, est en effet indispensable et son automatisation le devient également.

Plus précisément, notre objectif est de proposer un modèle d'indexation personnalisée, c'est-à-dire dans lequel l'utilisateur peut lui-même définir les symboles et concepts utilisés pour l'indexation et la recherche.

Cet objectif requiert que le système proposé soit à même d'apprendre, au gré des interactions avec l'utilisateur. Cet apprentissage, de par la nature interactive de son contexte, doit posséder un certain nombre de bonnes propriétés, comme par exemple la réactivité. Partant du constat que les approches actuelles en apprentissage automatique ne possèdent pas toutes ces propriétés, nous nous proposons d'aborder une nouvelle approche.

[9] Certains sont toutefois plus optimistes [Li04] et classent automatiquement les images selon des catégories sémantiques telles que « France », « Christmas », « Modern », « Kenya », « Zimbabwe », « Historical » ou encore « occupation ».

1.4 Notre approche

Nous pensons que la recherche d'images assistée par ordinateur est plus utile dans un cadre sémantique (symboles, concepts) que dans un cadre purement « signal ». Cependant, le fossé sémantique rend la tâche très ardue, en particulier dans un domaine large comme celui des collections personnelles. La personnalisation, en réduisant l'étendue de ce domaine, est à notre avis une première contribution à la résolution du problème. Nous sommes également persuadés que l'accès d'un système à « la sémantique » requiert d'utiliser des techniques d'apprentissage automatique. La fusion, dans un même système, de la personnalisation et de capacités d'apprentissage adaptées permet à ses utilisateurs d'indexer et de retrouver leurs images selon leurs propres concepts.

Personnalisation :

Une première idée centrale à notre approche est de travailler dans un domaine limité, qui n'est autre que le domaine large, réduit aux intérêts de l'utilisateur. La connaissance d'un concept visuel par le système peut ainsi être réduite au sous-ensemble des quelques instances du concept qui apparaissent dans la collection d'images de l'utilisateur. Nous qualifions cette réduction par le terme de « personnalisation ». Cette manière d'envisager la recherche d'images n'autorise pas l'utilisation des paradigmes classiques dans lesquels le système est ajusté « offline » pour permettre la formulation et satisfaction d'un ensemble de requêtes dont le champ est connu et limité. Cette contribution repose donc sur la définition et la réalisation d'un processus d'indexation personnalisée, avec tous les problèmes nouveaux que cela engendre (un défi, d'après [Sme00]).[10]

Chaque entité physique (un objet, un être vivant, etc.) se décline en une multitude de variations et apparaît sur une image sous divers points de vue, éclairages et à diverses distances ; cela donne lieu à une grande variabilité dans son apparence visuelle. Il est donc impensable, à l'heure actuelle, de créer un détecteur robuste pour chaque entité[11]. Nous pensons que l'indexation personnalisée apporte une

[10] "The challenge for image search engines on a broad domain is to tailor the engine to the narrow domain the user has in mind *via* specification, examples and interaction."
[11] Après plusieurs décennies, le problème n'est toujours pas entièrement résolu pour l'entité « visage ».

solution réaliste aux problèmes que sont le fossé sensoriel et le fossé sémantique. L'idée est la suivante : s'il n'est pas possible de créer un « détecteur universel » pour une entité donnée (c'est-à-dire qui peut la détecter quelle que soit son apparence), alors créons un détecteur capable de reconnaître l'entité telle qu'elle apparaît dans la collection de l'utilisateur. En faisant apprendre au système les concepts qui l'intéressent, l'utilisateur rend le système adéquat à ses besoins, réduisant ainsi largement le problème du fossé sémantique.

D'autre part, en ne faisant apprendre au système que les concepts tels qu'ils apparaissent dans la base d'images, l'utilisateur contourne le fossé sensoriel. Par un apprentissage progressif (au fil des sessions), interactif (le système donne un retour instantané sur ce qu'il vient d'apprendre) et optimisé (le système tend à focaliser son apprentissage sur les points problématiques, il apprend à mesurer la similarité), les concepts appris doivent mieux s'adapter à ceux que l'utilisateur a en tête. Un Système de recherche d'images par le contenu (SRIC) personnalisé doit être en mesure de reconnaître des concepts généraux fortement représentés (statistiquement) dans les photographies et doit laisser à l'utilisateur la possibilité d'étendre le champ des concepts indexés selon ses besoins individuels.

Apprentissage :

La personnalisation d'un système de recherche d'image requiert que ce système soit capable d'apprendre à reconnaître des concepts définis par l'utilisateur sous la forme d'exemples. Cet apprentissage a pour objectif la *caractérisation* de ces concepts : qu'est-ce qui caractérise un concept ? Quelles sont les caractéristiques importantes, indispensables ? Cela revient à savoir comment mesurer la similarité entre deux instances de ce concept de manière pertinente. Notre hypothèse est qu'à partir de simples pixels, cette caractérisation ne peut être que relationnelle : les pixels considérés indépendamment ne peuvent pas caractériser un concept. Ce sont les relations entre pixels qui permettent cette caractérisation. L'utilisation de traits de bas niveau permet, nous le verrons, d'accéder à une partie de ces relations, partie que nous croyons mineure. Les algorithmes d'apprentissage utilisés aujourd'hui en recherche d'image ne tiennent pas, ou peu, compte des relations entre les traits utilisés pour décrire les images. En réalité, beaucoup s'intéressent plutôt aux relations entre les exemples d'apprentissage, ce qui est très différent. La contribution

principale de ce travail consiste en la proposition d'une approche relationnelle, constructive et hiérarchique de l'apprentissage. Les idées que nous défendons dans cette approche sont les suivantes :

- L'apprentissage doit concerner les **relations entre les attributs** qui constituent les exemples, et non les exemples eux-mêmes.
- L'apprentissage, au moins initialement, est nécessairement **non supervisé**. Un algorithme d'apprentissage non supervisé n'est pas nécessairement un algorithme de Clustering.
- L'apprentissage doit être **hiérarchique** et permettre le partage des représentations : ce qui est appris est mis à profit pour apprendre de nouvelles représentations.
- L'apprentissage a pour but l'appréhension de **régularités** dans les données présentées, nous considérons deux types de régularités :
    - **Conjonctives** : elles sont basées sur des cooccurrences et peuvent donc représenter des équivalences, des implications ou plus généralement des corrélations. Elles définissent une *structure* dans les représentations.
    - **Disjonctives** : elles caractérisent les informations qui sont similaires, ou interchangeables, dans un contexte particulier. Elles définissent la notion de similarité locale, c'est-à-dire certains degrés de liberté à chaque niveau de la structure.

Ces idées sont basées sur des hypothèses importantes que nous discuterons.

1.5 Plan

Dans un premier temps, nous posons les bases nécessaires à l'exposé de nos travaux. Le **chapitre 2** présente les idées et les approches principales dans le vaste domaine de la recherche d'images. Le portrait que nous dressons de ce domaine, ne pouvant être exhaustif, adopte un point de vue « en largeur d'abord ».

Le **chapitre 3** passe en revue les principales techniques utilisées en apprentissage automatique. Nous mettrons en avant leurs forces et leurs faiblesses, vis-à-vis de notre contexte particulier.

Les **chapitres 4 et 5** présentent notre approche. Les idées, motivations et hypothèses sont présentées au chapitre 4. Les structures de données et les contraintes principales sont exposées au chapitre 5. Finalement, une description de la mise en œuvre algorithmique de notre approche sera l'objet du sixième chapitre.

Le **chapitre 6** décrit les expérimentations que nous avons menées afin d'évaluer nos travaux.

Le **chapitre 7** conclut ce document, résumant d'abord notre approche et nos contributions puis esquissant nos perspectives.

Chapitre 2

Les Systèmes de Recherche d'Images par le Contenu : un état de l'art

2.1 Introduction

L'expression « recherche d'images par le contenu » (« Content-Based Image Retrieval, CBIR, en Anglais) remonte aux travaux de Kato en 1992. Son système, ART MUSEUM, permet de retrouver des images d'art par couleurs et contours. Le terme s'est étendu par la suite à tout procédé permettant de rechercher des images selon des traits, pouvant être de type « signal », comme la couleur et la forme, mais également symboliques. Comme le remarquent les auteurs d'un rapport important sur les systèmes de recherche d'image par le contenu [Eak99], retrouver des images indexées manuellement par des mots clefs n'est pas de la recherche par le contenu au sens où le terme est généralement compris, même si ces mots clefs décrivent le contenu effectif de l'image.

Les **applications** des systèmes de recherche d'images existants (et donc les collections d'images) sont variées. Elles incluent des applications judiciaires : les services de police possèdent de grandes collections d'indices visuels (visages, empreintes) exploitables par des systèmes de recherche d'images. Les applications militaires, bien que peu connues du grand public, sont sans doute les plus développées [Eak99] : reconnaissance d'engins ennemis *via* images radars, systèmes de guidage, identification de cibles *via* images satellites en sont des exemples connus. Le journalisme et la publicité sont également d'excellentes applications. Les agences de journalisme ou de publicité maintiennent en effet de grosses bases d'images afin d'illustrer leurs articles ou supports publicitaires. Cette communauté rassemble le plus grand nombre d'utilisateurs de recherche par le contenu (davantage pour les vidéos)

mais l'aide apportée par ces systèmes n'est absolument pas à la hauteur des espoirs initiaux ([Eak99]).

D'autres applications incluent : le diagnostic médical, les systèmes d'information géographique, la gestion d'œuvres d'art, les moteurs de recherche d'images sur Internet et la gestion de photos personnelles. C'est sur ce dernier thème que nous concentrons notre travail. Nous focalisons donc notre étude de l'état de l'art sur les approches dont la cible (utilisateurs, contenu des images) est peu spécialisée. Ces approches, devant nécessairement faire face aux problèmes dont nous avons parlé en introduction (fossés sensoriel et sémantique), sont par conséquent les plus pertinentes dans notre cadre.

Concevoir un système permettant d'assister des utilisateurs dans leurs tâches de recherche d'images pose des **problèmes** variés. Dans [Eak99] les difficultés suivantes sont identifiées :

1. Comprendre les utilisateurs d'images et leurs comportements : de quoi les utilisateurs ont-ils besoin ?
2. Identifier une manière « convenable » de décrire le contenu d'une image. C'est une tâche rendue difficile par la subjectivité intrinsèque aux images.
3. Extraire des « traits » des images brutes.
4. Pouvoir stocker de manière compacte un grand nombre d'images
5. Comparer requêtes et images stockées de manière à refléter les jugements de similarité humains
6. Accéder efficacement aux images par leur contenu
7. Fournir des interfaces utilisables

A cela, ajoutons la difficulté majeure, dont nous avons parlé dans l'introduction, connue sous le nom de « **fossé sémantique** » [Sme00].
Il convient donc d'abord de montrer quelles sont les approches existantes ainsi que leurs limitations. Nous commençons par rappeler les composants d'un système de recherche d'images par le contenu, puis nous présentons une

taxonomie des systèmes selon leur niveau d'abstraction en donnant systématiquement des exemples.

2.2 Composants d'un SRIC

Nous décrivons brièvement ici les caractéristiques communes à la plupart des approches : le traitement de la base d'images, les requêtes puis la mise en correspondance et la présentation des résultats. La Figure 2-1 illustre l'ordonnancement de ces étapes :

Dans un premier temps (2), des traits sont calculés à partir de chaque image de la collection (1), ils peuvent être de type signal ou/et symbolique (le vocabulaire d'indexation).Les données extraites (à présent représentatives du contenu de l'image du point de vue du système) constituent la base d'index (3). Les requêtes de l'utilisateur (4) sont alors transformées afin d'être comparables avec la base d'index (5) ; une mise en correspondance (6) entre la requête transformée et la base d'index permet ensuite de produire le résultat de la requête (7). Il se peut également que le système possède des composantes liées à la personnalisation, comme par exemple l'extraction, le stockage et l'utilisation d'un profil d'utilisateur.

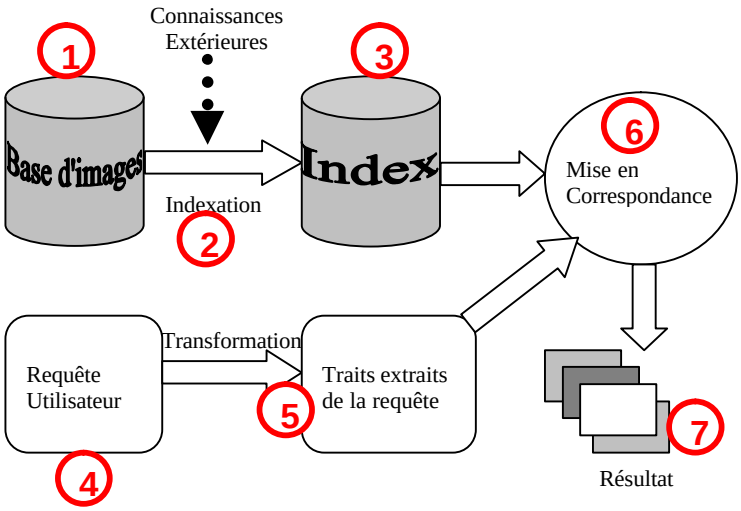


Figure 2-1 : ***Principaux composants d'un Système de Recherche d'Images par le Contenu***

(1) La base d'image

La collection (ou base) d'images est la donnée principale du système. Les bases d'images varient d'abord par leur **taille** : la majorité des systèmes est conçue pour des bases de quelques centaines ou milliers d'images ([Vai01], [Min96], [Li04a]). Ce nombre peut cependant s'approcher du milliard (880 millions d'images[12]) lorsque la base est constituée par les images collectées par des robots sur Internet. La taille de la base d'image impose des contraintes sur la complexité des traitements effectués sur chaque image. Il en résulte que la comparaison qualitative entre des systèmes travaillant sur des bases de tailles très différentes est peu pertinente.

Le **type d'image** composant la base varie également : des portraits en noir et blanc [Lew96], des peintures chinoises anciennes [Li04a], des images personnelles [Vai01], des images de tissus humains [Fel03], etc. Le type d'image influe fortement sur la conception globale du système, particulièrement sur les traits de bas niveau calculés. D'une manière générale, plus la variabilité intra et inter images est importante, plus le système doit être riche et précis (et plus le problème d'indexer/rechercher ces images est difficile).

Finalement, les collections diffèrent par leur **stabilité**, c'est-à-dire le taux de changements (ajouts d'images, retraits, etc.) en fonction du temps. Faible pour une collection d'images représentant les oeuvres d'un peintre ne créant plus, elle peut être très forte lorsque, par exemple, on s'intéresse aux images de la Toile ou à l'actualité.

(2) L'indexation

L'indexation est l'ensemble des processus aboutissant à la construction d'un index de l'image. Contrairement à d'autres types de données, comme le texte, il n'est pas possible d'utiliser les images *directement* dans un SRIC. Il faut caractériser les images par des informations à la fois discriminantes et invariables à certains paramètres (comme la taille de l'image, l'angle de la prise de vue, etc.). L'indexation peut être **fixe** ([Fli95] [Smi96]) : les traits calculés

[12] Google Image en 2004. (http://images.google.f)

sont toujours les mêmes. L'indexation peut aussi être **évolutive** ([Min96]) : les traits s'adaptent à l'utilisateur ou au contexte dans le temps, ce qui permet de renforcer l'adéquation système/utilisateur.

L'indexation peut être **générique** (indexation de photographies diverses dans [Fli95]), pouvant caractériser des collections hétérogènes, ou **spécifique** (indexation de peintures chinoises dans [Li04a]), adaptée à un type d'image particulier. Une collection hétérogène est par exemple constituée de photographies personnelles, mettant en scène diverses entités physiques dans des conditions de prise de vue variables. Indexer une telle collection impose l'usage de traits suffisamment génériques (la couleur par exemple), c'est-à-dire qui caractérisent une propriété discriminante applicable à la plupart des entités physiques. A l'inverse, indexer une collection d'images très spécifiques (des empreintes digitales par exemple) requiert l'utilisation de traits également très spécifiques qui, par ailleurs, ne conviendraient probablement pas à une collection hétérogène.

La phase d'indexation peut inclure une étape de **segmentation**, afin de caractériser des régions homogènes de l'image ([Smi96]) ou bien indexer l'image dans sa globalité [Vai01]. La segmentation de l'image précède généralement l'indexation individuelle des régions de l'image et cela permet, outre le fait d'accéder à des parties de l'image, de calculer des traits de 'forme'.

Enfin, l'indexation varie d'un système à l'autre par son **niveau d'abstraction** : extraire des histogrammes de couleurs est direct[13], alors que reconnaître des personnes ou des objets est beaucoup plus complexe et requiert un apprentissage préalable.

(3) La gestion des index

Elle concerne la manière dont sont gérés les index des images : stockage et accès. La gestion des index, anecdotique pour une collection de taille modeste, devient une préoccupation essentielle lorsque l'on travaille sur une base de

[13] Bien que le choix de l'espace de couleurs et de la taille des histogrammes soit important.

taille conséquente. La manière la plus basique de stocker les index est la liste séquentielle, que ce soit en mémoire ou dans un fichier. Cependant, lorsque le nombre d'images augmente, le temps d'accès à une image augmente linéairement et il est souvent nécessaire d'organiser les index de manière hiérarchique, sous forme d'arbres (organisés selon les attributs), ou de tables de 'hash-code' par exemple, afin d'accélérer l'accès à l'information.

(4) Les requêtes

Le type de requête proposé découle de choix faits en amont, au niveau de l'indexation. Dans des systèmes où seuls des traits de bas niveau sont extraits ([Fli95], [Smi96]), les requêtes ne peuvent être que de bas niveau : requête par « image exemple », par croquis ou par manipulation directe des traits de bas niveau. Dans ces systèmes, des traits sont extraits à partir de la requête (une image, un croquis…) et sont comparés aux traits calculés à partir des images de la base (les index des images).

A l'opposé, dans des systèmes proposant plus d'abstraction ([Vai01], [Tow00]), les requêtes peuvent être sémantiques (textuelles par exemple). Par exemple dans [Town00], les images sont indexées par des « catégories sémantiques visuelles », ce qui permet à un utilisateur de formuler des requêtes sémantiques (« Je veux des images prises à l'extérieur. »).

(5) Analyse de la requête

Cette étape a pour but de transformer la requête utilisateur pour la rendre comparable avec les index de la base d'images ; elle consiste donc généralement à extraire les mêmes types de traits que ceux extraits de la base d'image lors de l'indexation.

(6) Mise en correspondance requête / base

Il s'agit d'estimer dans quelle mesure une image (son index) satisfait une requête donnée. Dans le contexte de la recherche d'images, cela se ramène

souvent à calculer la similarité entre les caractéristiques extraites de la requête et les caractéristiques de chaque image dans la base. Cela aboutit généralement à une valeur de correspondance qui caractérise la pertinence (du point de vue du système) d'une image par rapport à la requête. Cette mise en correspondance peut être simple (comparaison d'histogrammes) ou complexe (comme dans [Smi96] par exemple, avec une mise en correspondance qui tient compte de l'arrangement spatial des régions).

La phase de mise en correspondance peut également inclure une pondération des traits (comme dans [Fli5] où chaque trait est pondéré par rapport à son pouvoir discriminant dans la base). Pondérer les traits permet d'éliminer une partie du bruit dans la mesure où les traits les moins pertinents voient leur influence diminuer dans l'évaluation de la similarité requête/image.

La mise en correspondance peut également inclure un bouclage de pertinence[14]. Le but est également d'éliminer le bruit (augmenter la précision[15]) en tentant de converger vers une précision maximale.

(7) La présentation des résultats

Dans la grande majorité des systèmes disponibles [Rem01], le résultat d'une requête est présenté sous la forme d'une liste d'images (réduites à des vignettes) ordonnées par pertinence décroissante. Parfois cette présentation prend d'autres formes, comme par exemple l'œil de poisson (FishEye View) [Gué03]. L'avantage des images par rapport aux documents textuels est qu'il est possible de visionner d'un coup d'œil l'intégralité du document, ce qui permet de visualiser un grand nombre de résultats et de les comparer plus rapidement. Comme indiqué plus haut, la présentation des résultats est souvent couplée avec une possibilité d'interaction, qui permet exemple de raffiner une requête en indiquant au système les résultats pertinents et ceux qui ne le sont

[14] Le bouclage de pertinence est une technique d'interaction utilisateur/système dans laquelle le jugement, par l'utilisateur, du résultat d'une requête est utilisé par le système pour proposer de nouveaux résultats. Idéalement, il y a convergence vers le résultat attendu par l'utilisateur.

[15] [Sal88] Précision : Le rapport entre le nombre de documents pertinents retrouvés et le nombre total de documents retrouvés. Rappel : Le rapport entre le nombre de documents pertinents retrouvés et le nombre total de documents pertinents.

pas (bouclage de pertinence), et de permettre ainsi une reformulation automatique de la requête.

2.3 Représentation des images dans un SRIC

Avant de décrire un certain nombre de systèmes, il nous paraît nécessaire de parler brièvement de la manière dont sont représentées les images dans ces systèmes, en particulier en termes de couleurs et de textures. Le lecteur familiarisé avec la notion de trait de bas niveau est invité à se rendre directement à la discussion (2.3.4).

2.3.1 Un besoin de simplification

Dans la vie de tous les jours, nous sommes soumis à une quantité astronomique d'informations brutes, sous forme d'images, de sons, etc. Pour pouvoir utiliser ces informations, il faut réduire de manière drastique cette quantité. C'est là le rôle de la perception : extraire d'une quantité énorme d'information brute un « résumé » utilisable ; et cela consiste essentiellement à trouver des régularités dans les données. Les régularités sont intéressantes car elles permettent de coder, représenter une information brute, de manière concise (particulièrement lorsqu'on ignore les détails). Prenons l'exemple de notre système visuel. Il existe toutes sortes de régularités visuelles et nous possédons un certain nombre de 'détecteurs' pour ces régularités. Il se trouve que les régularités que nous sommes capables de détecter nous permettent d'avoir suffisamment d'information pour identifier la plupart des objets physiques[16]. Dans ce chapitre, nous nous intéressons aux techniques utilisées en informatique pour reproduire[17] notre processus de perception. Nous appellerons ces régularités dans l'information visuelle des traits de bas niveau.

---

[16] Ce n'est probablement pas un hasard mais plutôt le fruit d'une sélection naturelle (où il vaut mieux savoir faire la différence entre un prédateur et un feuillage), allié à un apprentissage commençant dès la naissance.

[17] Il s'agit bien d'essayer de reproduire **notre propre perception**. Il existe probablement d'autres sortes de régularités dans l'information visuelle (c'est-à-dire que l'on pourrait coder sous une forme plus concise que l'information brute) que nous ne percevons pas et que nous ne tentons donc pas d'identifier et de reproduire. Toutefois, il n'y a pas, dans le fond, de raisons justifiant la seule utilisation de traits ressemblant à ceux que nous percevons, à part bien sûr le

Un trait de bas niveau est un ensemble de valeurs extraites directement et qui caractérisent l'image. L'extraction de traits de bas niveau représente une première abstraction par rapport à l'image brute. Elle constitue la *perception* du système, dans la mesure où les traits extraits sont la seule information conservée. Par opposition aux documents textuels par exemple, où l'extraction de mots porteurs de sens (même s'il est souvent ambigu) est directe, décrire une image par des traits de bas niveau est un problème difficile. La sémantique « contenue dans les pixels de l'image » n'est absolument pas accessible directement par une machine alors qu'elle apparaît de manière évidente à un humain qui voit simultanément l'ensemble des pixels. Associer une description à un ensemble de pixels requiert en effet de nombreux processus et surtout, une quantité énorme de connaissances. Par conséquent, les traits extraits d'une image ne sont pas porteurs de sémantique explicite mais tentent de capturer des propriétés visuelles intéressantes de l'image qui constitueront des indices suffisamment discriminants et invariables pour inférer de la sémantique.

Dans [Rui99], les auteurs distinguent deux types de traits de bas niveau : les traits génériques et ceux relatifs à un domaine particulier. Le premier type inclut la couleur, la texture et la forme alors que le second dépend du domaine investigué et peut inclure par exemple les visages, empreintes digitales ou une caractérisation du « coup de pinceau » (comme dans [Li04a]).

Un certain nombre de critères sont à prendre en compte lors de la sélection d'un trait. Le but est que ce trait soit **discriminant** par rapport aux entités visuelles que l'on cherche à caractériser. Les critères sont les suivants :

- Un trait de bas niveau doit être **pertinent** par rapport à un contexte donné. Le contexte repose principalement sur le type d'image traité : un trait pertinent pour la classification de visages ne l'est pas nécessairement pour la classification de photographies de peintures.
- Un trait de bas niveau pertinent doit être **cohérent** par rapport à l'entité visuelle qu'il caractérise. L'apparence visuelle d'une entité

---

fait que ces traits nous sont suggérés de manière évidente. Après tout, ordinateurs et cerveaux fonctionnent certainement de manières bien différentes.

physique varie de manière plus ou moins continue dans un intervalle donné selon son état (par exemple les différentes phases d'un coucher de soleil, les couleurs que revêt un arbre, etc.) et selon plusieurs axes. Un trait cohérent est un trait qui suit ces variations. Par exemple, un trait extrait à partir d'un espace de couleurs organisé en « Luminosité – teinte – saturation » est plus cohérent pour décrire l'apparence visuelle d'un arbre qu'un trait extrait d'un espace « Rouge – Vert – Bleu ». La raison est que l'axe « luminosité » suit plus ou moins les variations de luminosité au cours d'une journée, les axes « teinte » et « saturation » suivent à peu près les variations de couleurs au cours des saisons. Par conséquent, de faibles variations au niveau physique se traduisent par de faibles distances dans les caractéristiques calculées.

- Un trait de bas niveau, sauf cas particulier, devrait être **invariant** aux variations géométriques et aux variations d'illumination. La raison est que le but de ce trait est généralement de caractériser une entité visuelle parmi d'autres, et que celle-ci doit être identifiée quelle que soit la manière dont l'entité apparaît.
- Finalement, l'extraction d'un trait à partir d'une image doit être suffisamment **rapide** pour envisager son utilisation à grande échelle.

En résumé nous pouvons dire que dans un contexte donné, pour qu'un trait soit **discriminant**, il doit être à la fois **pertinent**, **cohérent** et **invariant**. De plus, dans un cadre interactif, ou si le nombre d'images à traiter est très grand, l'extraction de ces traits devrait être **rapide**. Nous présentons maintenant les trois grandes catégories de traits de bas niveau, tout en gardant à l'esprit que notre contexte est celui des photos personnelles, donc un contexte général.

2.3.2 Représentation par les couleurs

La couleur est le trait le plus utilisé en SRIC. Un trait couleur est généralement indépendant de la taille et l'orientation de la région caractérisée [Rui99]. Cependant, une collection d'images ou même une image seule contiennent généralement énormément de couleurs[18] distinctes et il n'est pas envisageable de toutes les considérer indépendamment. Un trait couleur repose sur deux choix : la sélection d'un espace de couleur et le choix d'une représentation. Voyons quels sont les moyens utilisés dans la littérature pour représenter la couleur des images.

Une première possibilité, qui semble naturelle, est de caractériser les couleurs comme nous le faisons chaque jour, par une palette de couleurs (rouge, bleu, jaune, etc.). En plus de réduire drastiquement le nombre de couleurs, cela permet de donner au système une représentation des couleurs similaire à celle des humains. Le système CIRES ([Iqb02]) illustre cette méthode. CIRES se base sur l'espace CIE LAB pour analyser les couleurs. Chaque couleur de l'image est alors associée à l'une des quinze couleurs d'une palette, l'information colorimétrique d'une image est donc représentée par un vecteur de quinze valeurs. Les auteurs justifient ce choix par le fait que la représentation habituelle par histogrammes ne tient pas compte du fait que peu de couleurs sont en fait nécessaires pour la discrimination visuelle et que des valeurs adjacentes dans un histogramme peuvent représenter finalement la « même » couleur.

Une généralisation de l'idée ci-dessus est de représenter l'information couleur sous la forme d'histogrammes. L'histogramme est une représentation efficace du contenu couleur de l'image, particulièrement lorsqu'un motif couleur (image ou région) est unique comparé au reste de la collection ([Lon02]). Un histogramme de couleurs est facile à calculer et efficace pour caractériser la distribution locale ou globale des couleurs. De plus, c'est une représentation insensible aux translations et rotations et peu sensible à l'échelle, aux occlusions, ou à l'angle de vue ([Lon02]). Ces qualités en font la manière de

[18] Sur 50 images (2 272 × 1 704 × 24BPP) choisies aléatoirement parmi une collection de l'auteur, une image contient en moyenne 250 832 couleurs distinctes.

caractériser la couleur la plus courante dans les SRIC. Ces histogrammes peuvent représenter l'information couleur de l'intégralité de l'image ([Fli95], [Ogl95] [Ing00]). Dans d'autres systèmes les histogrammes sont utilisés pour représenter l'information couleur de régions de l'image ([Ma99], [Car99]) ou l'information couleur de zones fixes (des carrés de taille arbitraire par exemple) de l'image ([Ort97]).

Les représentations dont nous venons de parler décrivent efficacement la présence des couleurs mais ne tiennent pas compte de leurs corrélations. Les '*color correlograms*' ([Hua97]) ont été proposés pour qualifier non seulement les couleurs mais aussi les corrélations spatiales de chaque paire de couleurs. Ces informations sont représentées sous la forme d'un histogramme à trois dimensions : les deux premières dimensions représentent les combinaisons possibles de paires de pixels, la troisième dimension représente leurs distances spatiales. Par rapport aux autres représentations, les '*color correlograms*' aboutissent aux meilleurs résultats en terme de pouvoir de discrimination, toutefois, leur calcul s'avère coûteux dû au nombre élevé de dimensions ([Lon02]).

L'information relative aux couleurs est particulièrement importante dans la caractérisation d'une image. Avant de sélectionner un type de description du contenu couleur, il convient de choisir un espace de couleurs. Si un objet physique évolue (déplacement, différence d'illumination, modifications liées au temps, etc.) alors cette évolution devrait se refléter proportionnellement dans un espace de couleur. C'est entre autres pour cela que nous choisissons de représenter les couleurs dans un espace dont les dimensions sont la teinte, la saturation et la luminosité : l'espace HSV.

2.3.3 Représentation par les textures

Au même titre que la couleur, la texture est une caractéristique fondamentale des images car elle concerne un élément important de la vision humaine. De nombreuses recherches ont été menées à la fois dans les domaines de l'analyse et de la synthèse de texture. Mais d'après [DEL99] une définition formelle de la texture est quasiment impossible. D'une manière générale, la

texture se traduit par un arrangement spatial des pixels que l'intensité ou la couleur seules ne suffisent pas à décrire. Elles peuvent consister en un placement structuré d'éléments mais peuvent aussi n'avoir aucun élément répétitif.

De nombreuses définitions ont été proposées, mais aucune ne convient parfaitement aux différents types de textures rencontrées. Dans une définition couramment citée [POL98], la texture est présentée comme une structure disposant de certaines propriétés spatiales homogènes et invariantes par translation. Cette définition stipule que la texture donne la même impression à l'observateur quelle que soit la position spatiale de la fenêtre à travers laquelle il observe cette texture. Par contre l'échelle d'observation doit être précisée. On peut le faire par exemple en précisant la taille de la fenêtre d'observation.

La notion de texture est liée à trois concepts principaux:

1. un certain ordre local qui se répète dans une région de taille assez grande,
2. cet ordre est défini par un arrangement structuré de ses constituants élémentaires,
3. ces constituants élémentaires représentent des entités uniformes qui se caractérisent par des dimensions semblables dans toute la région considérée.

Il existe un grand nombre de textures. On peut les séparer en deux classes: les textures structurées (macrotextures) et les textures aléatoires (microtextures). Une texture qualifiée de structurée est constituée par la répétition d'une primitive à intervalle régulier. On peut différencier dans cette classe les textures parfaitement périodiques (carrelage, damier, ...), les textures dont la primitives subit des déformations ou des changements d'orientation (mur de briques, grains de café, ...). Les textures qualifiées d'aléatoires se distinguent en général par un aspect plus fin (sable, herbe, ...). Contrairement aux textures de type structurel, les textures aléatoires ne comportent ni primitive isolable, ni fréquence de répétition. On ne peut donc pas extraire de ces textures une primitive qui se répète dans l'image mais plutôt un vecteur de paramètres statistiques homogènes à chaque texture.

Voyons maintenant à travers quelques exemples comment l'information relative à la texture est utilisée dans les systèmes de recherche d'images.

Les méthodes de caractérisation de la texture peuvent être classées selon la taxonomie suivante (*[Mir98]) :*

Les **méthodes statistiques** cherchent à caractériser des propriétés statistiques basées sur les occurrences de niveaux de gris de l'image. Parmi ces méthodes, la méthode des *matrices de cooccurrences* est sans doute la plus connue. Il s'agit d'une approche statistique de l'étude de la texture. Une matrice de co-occurrence P(i,j) est une matrice dans laquelle l'élément (i,j) décrit la fréquence d'occurrence de deux pixels (pour une distance d et une orientation o). Cette matrice décrit les régularités observables dans les niveaux de gris des pixels d'une région. Généralement, on ne se sert pas directement de la matrice de co-occurrence mais plutôt de valeurs calculées à partir de celle-ci (i.e. la moyenne, le contraste, l'homogénéité, l'entropie ou l'énergie). L'auteur précise que la méthode des matrices de co-occurrence est extrêmement coûteuse en temps de calcul.

Une approche différente est de considérer qu'une texture est un agencement de composants élémentaires. Cette approche, qualifiée de **géométrique** dans [Mir98] n'est pas adaptée au domaine large, particulièrement les scènes naturelles, qui ont des textures desquelles il est difficile d'extraire des composants élémentaires. Nous ne rentrerons donc pas dans des détails supplémentaires. De la même manière, les **méthodes basées sur des modèles**, qui tentent de calculer les paramètres d'un modèle de texture prédéfini, ne semblent pas adaptées aux textures naturelles [Mir98]. Elles sont par contre utilisées pour la génération de textures.

Les **méthodes issues du traitement du signal** semblent plus adaptées pour caractériser des textures naturelles. Les *filtres spatiaux* constituent probablement la technique la plus directe pour capturer les propriétés relatives aux textures de l'image. Les premières approches consistèrent à quantifier la densité de contour (« edge density per unit area ») : les textures fines ont une densité élevée en comparaison aux textures plus grossières. Ces filtres ont

pour but d'extraire la fréquence spatiale, qui fait référence à la fréquence de variation des différents tons qui apparaissent dans une image. Les régions d'une image où la texture est « rugueuse » sont les régions où les changements dans les tons sont abrupts; ces régions ont une fréquence spatiale élevée. Les régions « lisses » ont une variation des tons qui est plus graduelle sur plusieurs pixels; ces régions ont une fréquence spatiale faible. La méthode de filtrage spatial consiste à déplacer un filtre d'une dimension de quelques pixels (ex. : 3 sur 3, 5 sur 5, etc.) au-dessus de chaque pixel de l'image. Un filtre peut être uniforme ou réagir à des caractéristiques précises comme les coins, les jonctions, etc. Il est ainsi possible de caractériser grâce à cette méthode de nombreuses caractéristiques de la texture comme sa rugosité ou sa directionnalité.

Toujours dans le domaine du traitement du signal, les méthodes issues des *transformées de Fourier* sont sans doute les plus courantes en analyse de texture. L'analyse de Fourier est un outil largement utilisé en physique et en mathématiques. Le principe de la transformée de Fourier repose sur le fait que toute fonction périodique peut être représentée comme la somme d'une série de sinus et de cosinus dont on fait varier d'une part les amplitudes en les multipliant par des coefficients, et d'autre part les phases en les décalant de manière à ce qu'elles s'additionnent ou se compensent. Le problème est que cette représentation fréquentielle de l'image est globale. La caractérisation locale des textures utilise donc le principe des fenêtres de Fourier, qui « glissent » sur l'image en capturant ses propriétés locales. Les techniques dites « de Gabor » utilisent cette méthode avec une fenêtre Gaussienne ([Gos99]). Il est commun d'utiliser un banc de *filtres de Gabor* pour caractériser plus précisément l'information fréquentielle des régions de l'image selon des directions ou des distances particulières. Par leur nature fréquentielle, les méthodes issues des transformées de Fourier sont adaptées pour caractériser les textures régulière et uniformes (comme un grillage, un mur de brique ou un damier) et peu adaptées pour caractériser les textures aléatoires (un feuillage, une foule, un visage, une écorce, etc.) ou les textures régulières projetées sur des objets non plats ou vues sous un angle particulier.

Une étude très intéressante est menée dans [Sin02] sur les algorithmes de segmentation et les algorithmes d'extraction de texture. Au-delà des expérimentations menées dans leur travail, des faits remarquables sont rapportés.

Le premier de ces faits est qu'il n'existe aucun consensus sur la meilleure technique d'extraction de texture. De plus, les comparaisons entre techniques ne sont guère viables dans la mesure où les conditions expérimentales sont très différentes. Les résultats obtenus dépendent beaucoup des données utilisées pour les tests et celle-ci étant de faible taille, les résultats ne sont pas généralisables.

Un autre point très important est que les techniques d'extraction de texture sont très souvent évaluées sur des collections spécialisées telles que Brodatz, qui n'ont aucun rapport avec les photographies traditionnelles. Les auteurs rapportent qu'ils n'ont rencontré que très rarement des évaluations portant sur de « vraies scènes ». Or, les résultats obtenus sur des collections de textures ne sont pas reportables sur des images naturelles. Par exemple, toujours dans [Sin02], la méthode d'extraction "*Edge Frequency*" se révèle, en dépit de sa simplicité, parmi les plus efficaces pour la description de la texture de scènes naturelles alors que cette même méthode n'obtient que des résultats très modestes dans d'autres travaux validés sur des collections de textures. L'article passe également en revue onze articles qui testent diverses techniques d'extraction de textures. Il en ressort, outre le fait qu'il n'y a pas de consensus, le fait étonnant que les techniques basées sur les filtres de Gabor (très utilisées en SRIC) ne figurent jamais en tête.

En fait il ressort de cette étude que les méthodes simples semblent les plus efficaces pour les images naturelles ('Edge Frequency', matrices de cooccurrences, filtres spatiaux). Il ressort également que l'ajustement des paramètres est une tâche très difficile et qu'il est très rarement possible de l'automatiser. Enfin, un des points principaux est que l'efficacité d'un extracteur de texture est indissociable de l'algorithme de segmentation qui génère les régions à partir desquelles sont extraites les textures.

2.3.4 Discussion

Les traits de bas niveau constituent la base d'un SRIC, ils constituent le dispositif de perception du système et remplacent les images de manière interne. Par conséquent, les traits de bas niveau constituent une limite à la performance du système : quels que soient les techniques d'apprentissage, les mesures de similarités ou les tailles d'ensemble d'apprentissage utilisés, si les traits de bas niveau ne sont pas adaptés, le système entier en sera pénalisé.

Nous verrons dans ce qui suit que la manière dont les données sont représentées est fondamentale. En fait, la difficulté d'un problème d'apprentissage est principalement liée à la manière dont sont représentés les exemples d'apprentissage. Les traits de bas niveau sont généralement issus de tentatives de faciliter des problèmes d'apprentissage particuliers. Toutefois, quelques soient les traits considérés, ils ne sont jamais universels : ils sont en adéquation avec certains problèmes d'apprentissages mais ne sont pas adaptés aux autres. En fait, chaque problème d'apprentissage doit posséder son propre jeu idéal de traits de bas niveau. Puisqu'il n'est pas concevable de créer manuellement des traits adaptés à chaque problème, l'idéal serait que ceux-ci soient appris automatiquement.

Nous pensons qu'il n'existe pas de différence fondamentale entre la notion de trait de bas niveau et la notion de concept. Un trait de bas niveau a toujours pour origine un ensemble de *régularités* et son but est justement d'encoder ces régularités. Par conséquent, un trait encode également, quoique de manière plus implicite, une mesure de similarité « interne », ou locale. Par exemple un trait ayant pour but de détecter des angles, encode une mesure de similarité (spécifique au concept d'angle) qui permet de détecter des angles modulo un ensemble de variations possibles. Or, tout cela est également valable pour les concepts : un concept est une description générique, ou abstraite, qui recouvre un ensemble d'instances. Un concept possède également une mesure de similarité spécifique, permettant une certaine souplesse et donc une certaine abstraction. En outre, les concepts proviennent, tout comme les traits, de régularités. Nous pensons donc que traits et concepts se différencient principalement par leurs « ordre de grandeurs », les concepts étant

généralement plus « complexes » que les traits. Nous reviendrons sur ces questions lors de la présentation de notre approche.

2.4 Une taxonomie des approches

Malgré les limitations persistant dans le domaine de l'indexation d'images, de nombreux systèmes sont disponibles. Dans [Rem01] par exemple, les auteurs dénombrent et décrivent 43 systèmes parmi les plus connus. Étant donné le nombre et la diversité de ces systèmes, il est nécessaire de se donner une manière de les comparer ; afin de pouvoir mieux situer nos travaux et nos apports. C'est pourquoi nous nous intéressons maintenant à la définition d'une classification des SRIC.

2.4.1 Taxonomies dans la littérature

Ben Bradshaw [Brad00] passe en revue les systèmes actuels de recherche d'images par le contenu. Il classifie les systèmes en trois classes : première, seconde et troisième génération selon **le type de requête proposée**.

Ainsi, dans le groupe dit de première génération, l'auteur rassemble les systèmes dans lesquels les *requêtes se font par manipulation explicite des traits de bas niveau* (sélection de couleurs dans une palette par exemple). L'auteur remarque cependant que cette manipulation directe des traits par les utilisateurs n'est pas une méthode efficace pour les aider à trouver des images. Conformément à ce que nous avons dit précédemment, les utilisateurs trouvent en effet difficile de savoir quel trait est utile pour un besoin d'information donné et quelle valeur lui donner.

Les systèmes de deuxième génération proposent des *requêtes par l'exemple : rechercher un type d'image donné se fait en donnant au système une image exemple de ce type*. L'auteur note cependant qu'il existe un problème lorsque l'utilisateur ne dispose pas initialement d'une image ressemblant à ce qu'il cherche. Ajoutons qu'il existe un problème plus grave qui est de savoir *ce que recherche l'utilisateur*. Il peut en effet s'agir d'images ayant des couleurs

similaires à celle donnée en exemple, ou des textures similaires, les mêmes objets, la même personne, la même catégorie d'image, etc. Cela peut s'avérer difficile à déterminer, même en utilisant des techniques de bouclage de pertinence.

Les systèmes de « troisième génération » proposent des requêtes sémantiques plus abstraites. On cherche une image en utilisant des mots-clefs correspondant à des concepts sémantiques appris par le système. L'auteur souligne que ce type de requête est plus intelligible que dans les catégories 1 et 2.

Dans [Eak99], les auteurs organisent les systèmes en fonction de l'abstraction de leurs requêtes. Trois niveaux sont ainsi différenciés :

- **Niveau 1** : Les requêtes portent sur des traits primitifs de couleurs, textures ou formes. La requête la plus courante est : « Trouve une image qui ressemble à celle-là » (Requête par l'exemple).
- **Niveau 2** : Recherche par des 'traits' inférés à partir des traits primitifs. Il peut s'agir d'objets en général ou d'objet (ou personnes) nommés.
- **Niveau 3** : La recherche porte sur des attributs abstraits de l'image, impliquant un raisonnement de haut niveau sur le sens et le but des objets apparaissant dans la scène. Ce niveau est divisé en deux sous niveaux :
    - Evènements nommés ou types d'activités
    - Recherche par signification émotionnelle ou religieuse

Les deux taxonomies que nous venons de voir portent sur les requêtes proposées par le système, qui reflètent finalement les capacités intrinsèques du système vis-à-vis du fossé sémantique. Il apparaît dans ces taxonomies (ainsi que dans d'autres études) que le critère principal de catégorisation s'apparente à l'abstraction proposée par les systèmes, sans que celle-ci soit précisément

définie. C'est également le critère que nous retenons, cependant nous définissons plus formellement les différents niveaux d'abstractions considérés, tout en discutant les contraintes imposées par ces niveaux.

2.4.2 Taxonomie basée sur les niveaux d'abstraction

Dans notre cadre de recherche d'images dans des collections personnelles, les requêtes potentielles portent sur des concepts de différents niveaux d'abstraction: recherche de personnes, d'objets particuliers, d'événements, etc.

Comme nous l'avons vu au chapitre 1.2 portant sur la difficulté de l'indexation, atteindre différents niveaux d'abstraction (correspondant à différentes sémantiques) implique des processus de complexité différentes et également des connaissances différentes. C'est sur ces paramètres que nous basons notre découpage des systèmes en plusieurs niveaux d'abstraction.

Notre classification des systèmes de recherche d'images par le contenu est donc basée sur le critère **d'abstraction**. Nous verrons, en détaillant chaque niveau, que l'abstraction croissante implique des connaissances de plus en plus larges. Nous ne parlons pas ici de manipulation mais plutôt d'extraction d'information abstraite. Par exemple, dans un système comme EMIR2 [Mec95], la représentation des images est très riche et permet la formulation de requêtes complexes, cependant nous ne considèrerons le système capable d'abstraction que si la création de ces riches représentations est automatique.

Nous mettons délibérément de côté, comme critère d'organisation, les traits de bas niveau extraits, jugeant qu'ils ne constituent que la 'perception' du système. Le choix d'un trait est motivé par le type, le nombre d'images traitées, il influe sur la précision atteinte mais n'est en aucun cas à la base de la conception du système. De la même manière, notre classification n'est pas basée sur le type de requête proposé, celui-ci n'étant qu'une conséquence de choix situés en amont.

Nous avons noté que la principale différence entre les SRIC pionniers, les systèmes actuels et ceux que nous espérons voir émerger dans le futur est leur **degré d'adaptation aux besoins humains**. Un système comme QBIC [Fli95]

proposant de rechercher des images par sélection de couleurs et textures, par croquis ou image exemple n'est pas adapté aux recherches sémantiques. Lorsqu'un utilisateur recherche des photographies dans un album papier, la recherche porte rarement sur des *critères uniquement visuels* (« Où ai-je donc mis cette photo contenant 27 % de bleu et une texture modérément orientée ?»). Généralement c'est une recherche *sémantique* abstraite (« Où sont les photos du nouvel an en refuge ? »). Les critères d'une recherche sémantique incluent : les objets, lieux, personnes, actions mais aussi sans doute les ambiances, les impressions qu'évoquent une image ; ce sont des critères bien sûr désirables dans un SRIC.

Nous décrivons maintenant quelques systèmes, organisés selon trois niveaux d'abstraction (faible, moyen et haut). La description de chaque niveau d'abstraction sera suivie d'une discussion où nous analysons respectivement les faiblesses des systèmes à faible niveau d'abstraction et les difficultés rencontrées dans les niveaux à plus haute abstraction.

2.5 Les systèmes à faible niveau d'abstraction

2.5.1 Définition

Nous définissons un système à faible niveau d'abstraction de la manière suivante :

Définition 2-I*: Un système à faible niveau d'abstraction est un système dans lequel l'indexation et les requêtes ne portent que sur des traits de bas niveau. Il n'y a pas de manipulation explicite de sémantique. Toutefois, le système peut mettre en œuvre un bouclage de pertinence, une pondération des traits et de la segmentation.*

Les systèmes à faible niveau d'abstraction sont les systèmes les plus représentés à ce jour dans la littérature. Dans de tels systèmes, des traits sont d'abord extraits des images. Une interface (manipulation explicite des traits de bas niveau, requête par l'exemple, outils de croquis) permet de générer des traits de bas niveau qui sont comparés, *via* une mesure de similarité, aux

images de la collection. Les requêtes sont donc intrinsèquement limitées car elles portent sur l'apparence, plutôt que la sémantique des images. De plus, les requêtes peuvent être difficiles à formuler, soit par manque d'exemple soit par le fait qu'il faille « dessiner » ce que l'on cherche.

Malgré ces limitations, les systèmes appartenant à ce niveau d'abstraction sont nombreux, nous en discutons les raisons ci-dessous.

2.5.2 QBIC : le pionnier

Sans être exactement le premier système de recherche/indexation d'images, QBIC [Fli95] marque en 1995 le début des systèmes commerciaux. Il est aujourd'hui l'un des systèmes les plus cités et décrits de la littérature. Nous détaillons ici ses principales caractéristiques.

Extraction de traits :

QBIC extrait des traits de couleurs, textures et formes des images. Pour caractériser la **couleur**, QBIC calcule à partir d'une région ou d'une image entière des vecteurs couleurs tridimensionnels dans les espaces RGB, YIQ, Lab et Munsell ainsi qu'un histogramme à 256 dimensions des couleurs RGB quantifiées.

QBIC caractérise les **textures** en utilisant une version modifiée des traits de textures proposés par Tamura [Tam78]. Ceux-ci sont la rugosité, le contraste et la directionnalité.

En ce qui concerne la caractérisation des **formes**, QBIC calcule pour une zone donnée son aire, sa circularité, son excentricité ainsi que l'orientation de son axe principal. Ces deux derniers traits son calculés à partir de la matrice de covariance des pixels délimitant la zone : l'orientation étant la direction correspondant à la plus grande valeur propre, l'excentricité étant le rapport entre la plus petite et la plus grande des valeurs propres.

Requêtes et interface :

QBIC propose toutes les requêtes « classiques » pour un système de bas niveau d'abstraction, à savoir :

- Requête par l'exemple
- Manipulation directe des traits de bas niveau
- Requête par croquis
- Navigation par bouclage de pertinence

Les résultats des requêtes sont présentés à l'utilisateur, ordonnés selon une similarité décroissante par rapport à la requête.

Fonction de correspondance

Pour la **couleur**, la distance entre un objet requête et un objet dans la base d'image est une distance Euclidienne pondérée dans laquelle les poids sont inversement proportionnels à l'écart type de chaque composant dans l'ensemble de la base. Lors du calcul de distance entre deux histogrammes, deux mesures sont utilisées : la première (différence moyenne), calculable rapidement sert de filtre grossier sur toute la base. La seconde (distance quadratique), beaucoup plus coûteuse, n'est calculée que sur les images filtrées par la première.

Pour la **texture**, la distance Euclidienne est calculée pour les trois traits. Comme pour la couleur, chaque trait est pondéré en fonction de son écart-type sur la base.

En ce qui concerne les **formes**, QBIC compare également deux vecteurs de traits en utilisant une distance Euclidienne pondérée. Lorsque la requête est faite par croquis, celui-ci est réduit à une taille de 64×64 puis découpé en blocs de 8×8 pixels. La meilleure corrélation entre le bloc et une zone de recherche 16×16 dans chaque image de la base est calculée en déplaçant le bloc dans la zone de recherche. Le score final donné à une image de la base est la somme de toutes les corrélations calculées pour chaque bloc.

Personnalisation :

QBIC s'adapte dans une certaine mesure à l'utilisateur selon deux aspects. Le premier concerne le bouclage de pertinence qui constitue une adaptation, à court terme, du système aux besoins de l'utilisateur. Le deuxième aspect concerne les pondérations utilisées : celles-ci utilisent l'écart type d'un trait sur toute la base pour en évaluer l'importance. La pondération n'est donc pas fixée et s'adapte à la base d'images, mais pas directement à l'utilisateur car il ne s'agit pas d'une base d'images personnelle. Cependant, l'adaptation à l'utilisateur n'est globalement que très peu présente dans QBIC dans la mesure où elle n'existe qu'à très court terme.

2.5.3 VisualSEEK

Développé par l'université de Columbia (New York), VisualSEEK [Smi96] est un système de recherche d'image de bas niveau d'abstraction se démarquant des autres par son système de requête original, combinant relations spatiales et bouclage de pertinence.

Extraction de traits :

VisualSEEK segmente chaque image en régions homogènes par rapport à leur couleur dominante. Chaque région est alors caractérisée par ses propriétés signal et ses propriétés spatiales. VisualSEEK utilise une alternative aux histogrammes pour représenter la couleur, à savoir les « ensembles de couleur ».

Requêtes et interface :

Pour commencer une nouvelle requête, l'utilisateur dessine, positionne et dimensionne des régions sur une grille et sélectionne une couleur pour chacune de ces régions. L'utilisateur peut également indiquer des contraintes sur la taille ou les relations spatiales entre régions. Le système affiche les images retrouvées réduites et l'utilisateur peut continuer sa recherche en utilisant une de ces images comme requête.

Fonction de correspondance :

Lorsque la requête est constituée d'une région unique, les correspondances entre cette région et l'ensemble de couleurs, la taille de la région, sa position, et son orientation globale sont d'abord calculées. Une combinaison linéaire entre ces correspondances est calculée afin de déterminer la similarité requête/image, et donc le résultat final de la requête.

Si la requête consiste en plusieurs régions, chacune d'elle est considérée comme une requête indépendante et les résultats sont fusionnés en utilisant une représentation par chaîne 2D.

Personnalisation :

Nous considérons qu'il n'y a pas d'aspect personnalisation dans VisualSeek, c'est-à-dire que l'état du système demeure le même d'une session à l'autre.

2.5.4 Discussion

D'un point de vue pragmatique, les systèmes à faible niveau d'abstraction sont attractifs : le fait de ne pas chercher à caractériser la sémantique permet aux systèmes une grande souplesse, ils peuvent s'appliquer à tout type de base d'image, même hétérogènes et permettent de traiter de grandes quantités de données. Les systèmes dont nous venons de parler par exemple (QBIC et VisualSEEK) sont réellement utilisables et aboutis, ils sont rapides et peuvent gérer des bases d'images volumineuses. Il semble même, à la vue des expérimentations présentées, qu'ils soient capables de traiter certains cas de recherche par le contenu sémantique (la requête « Marguerite » retourne effectivement beaucoup d'images de marguerites). Cette impression, fausse, est due au fait que les requêtes testées concernent souvent des entités physiques (minoritaires) pour lesquelles il existe une forte corrélation entre apparence physique et sémantique (tigres, zèbres, couchers de soleil ou bus Londoniens par exemple).

Tout en remarquant que ces approches peuvent se révéler très utiles dans des contextes très précis, où l'apparence visuelle et la sémantique recherchée sont très liées (reconnaissance de visages, de caractères, de tumeurs, etc.), la voie empruntée par les paradigmes appartenant à la catégorie « faible niveau

d'abstraction » nous paraît être une voie sans issue dans le domaine des images personnelles, dans la mesure où elle est intrinsèquement limitée. Ce sont certes des approches souples, efficaces qui sont, d'un point de vue pragmatique, les mieux adaptées aujourd'hui mais le fait que ces SRIC ne soit en fait pas ou peu utilisés par les journalistes [Eak99] par exemple, suggère d'emprunter une autre voie. Dans la mesure où les personnes qui recherchent des images dans des collections plus hétérogènes souhaitent formuler des requêtes sémantiques, il faut que les systèmes soient capables d'extraire de la sémantique, du moins à un certain niveau utilisable.

Pour permettre une requête sémantique simple (disons « Animal »), le système doit être capable, d'une manière ou d'une autre, de voir comme similaires des apparences visuelles très différentes, mais qui correspondent à la requête (Par exemple « Chat » et « Poisson »). Dans ce cas, un niveau d'abstraction beaucoup plus élevé est requis. Evidemment, cette tâche requiert des techniques complètement différentes.

2.6 Les systèmes à moyen niveau d'abstraction

2.6.1 Définition

Nous définissons un système à moyen niveau d'abstraction de la manière suivante :

Définition 2-II ***: Un système à moyen niveau d'abstraction est un système dans lequel des informations sémantiques sont inférées uniquement à partir de caractéristiques visuelles des images. Le système est capable d'associer une ou des étiquettes à une région ou à l'image entière en se basant sur des traits de bas niveau extraits de celle-ci.***

Typiquement un système à moyen niveau d'abstraction utilise des techniques d'apprentissage automatique pour associer des vecteurs de traits à des étiquettes. Par exemple, un réseau de neurones peut être entraîné à reconnaître le concept « Ciel » : à partir d'instances de ce concept, le réseau apprend les multiples apparences du concept « Ciel ». Contrairement au niveau d'abstraction faible, on s'intéresse au concept « Ciel » et non pas à une seule de ses instances.

Théoriquement, les concepts indexés par un système à moyen niveau d'abstraction ne sont pas limités puisqu'il « suffit » de fournir un ensemble d'apprentissage constitué d'instances du concept. Toutefois, certains concepts d'abstraction élevée ne peuvent pas être identifiés robustement en ne se basant que sur des caractéristiques visuelles. Les concepts « Moyen-Âge », « Demoiselle charmante » ou « A Paris » ne peuvent être inférés que si l'on dispose (en plus de caractéristiques visuelles) d'informations allant au-delà du contenu brut de l'image. Comme indiqué précédemment, on touche ici au fossé sensoriel, qui rend la reconnaissance de concepts abstraits très ardue.

2.6.2 FourEyes : Une extension de PhotoBook

Décrivons brièvement Photobook [Pen94] avant de nous attarder sur son extension : FourEyes [Min96]. Photobook implémente trois approches différentes pour la représentation des images selon la catégorie à laquelle elles appartiennent : visages, formes 2D ou images de textures. Les deux premiers types de représentations sont similaires dans la mesure où les images sont caractérisées par une combinaison d'images prototypes (eigen-images), calculées à partir d'une matrice de covariance des images. Les textures quant à elles sont caractérisées selon trois axes : périodicité, aspect directionnel et aspect aléatoire. Les requêtes se font par sélection d'images exemples.

Dans notre taxonomie, Photobook appartient à la classe des systèmes à faible niveau d'abstraction ; cependant, augmenté de son extension FourEyes, le système appartient à la classe supérieure.

Etant donnée sa pertinence, nous décrirons FourEyes à un niveau de détails plus fin que pour les autres approches.

2.6.2.1 Description globale

FourEyes est un module ajouté au SRIC Photobook qui permet au système d'apprendre des concepts au fur et à mesure des interactions avec l'utilisateur. FourEyes repose sur la notion de « groupement ». Les groupements peuvent être intra images ou inter images et sont calculés lorsque le système n'est pas

utilisé (fonctionnement offline). Un groupement est un ensemble de régions similaires par rapport à un trait donné, la notion de groupement encode donc dans FourEyes la notion de similarité. L'utilisateur peut faire apprendre des concepts au système par le biais d'une interface graphique avec laquelle il fournit des exemples positifs et négatifs d'un concept donné. Un algorithme d'apprentissage tente ensuite d'isoler un ensemble de groupements contenant tous les exemples positifs mais aucun exemple négatif, ce qui permet alors de généraliser la sélection de l'utilisateur à toute l'image et aux autres images de la collection.

2.6.2.2 Calcul des groupements

La Figure 2-2 illustre la construction des groupements. FourEyes pré-calcule les groupements intra images pour chaque trait de bas niveau (couleur, texture, etc.) et pour chaque image. Les traits de bas niveau sont d'abord extraits et associés à chaque pixel de l'image (512×512) puis l'image est segmentée (16×16) et à chaque bloc est associé la moyenne et l'écart type local de chaque trait calculé précédemment. Un algorithme de clustering hiérarchique est exécuté pour chaque trait sur l'ensemble des blocs, générant des groupements. Chaque image est maintenant représentée uniquement par un ensemble de groupements.

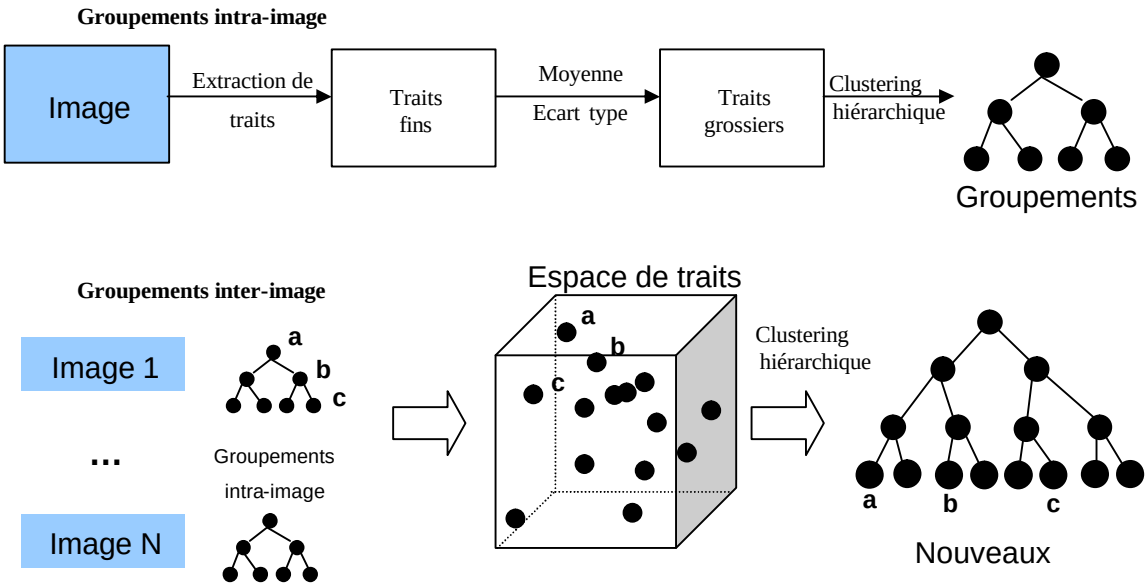


Figure 2-2 : Groupements intra et inter images dans FourEyes. Dans l'image 1, le groupement intra-image contient a, qui contient b, qui contient c. Projetés dans l'espace de traits, ils peuvent être dissimilaires et donc être regroupés différemment dans le clustering qui suit (ici a ressemble plus à b qu'à c)

Les groupements inter images sont le résultat d'un clustering hiérarchique de chaque trait sur l'ensemble des groupements intra image, comme indiqué sur la Figure 2-2.

2.6.2.3 Apprentissage de concepts

Une fois que l'ensemble des groupements intra et inter image a été calculé (avant toute interaction de l'utilisateur), il est possible de combiner certains groupements pour définir un concept. Pour cela, Photobook/FourEyes propose une interface graphique (Figure 2-3) à travers laquelle l'utilisateur peut indiquer (par des clics souris) des exemples positifs (pixels de l'image appartenant au concept cible) et des exemples négatifs.

Cet ensemble d'exemples constitue l'entrée de l'algorithme d'apprentissage, dont le but est de trouver un ensemble de **groupements tel qu'il inclut tous les exemples, mais aucun contre exemple**. L'algorithme d'apprentissage sélectionne à chaque étape un nouveau groupement tel que l'union des groupements contient tous les exemples positifs et tel que chaque groupement ne contient aucun exemple négatif. Dans les hiérarchies construites

précédemment, une feuille représente un groupement ne contenant qu'un élément. L'algorithme est donc capable d'apprendre avec n'importe quel ensemble d'exemples.

2.6.2.4 Résultats

Des expérimentations ont été menées sur une collection de scènes naturelles dont 25 ont été sélectionnées. Trois sujets humains ont segmenté les images et attribué une étiquette à chaque région appartenant à l'une des classes suivantes : bâtiment, voiture, herbe, personne, ciel, feuilles et eau. L'ensemble des régions étiquetées est ensuite réduit à des blocs carrés de seize pixels de côté, puisque c'est à cette résolution que les groupements ont été calculés.

Les régions sont données une par une à l'algorithme de classification, pour simuler une interaction avec l'utilisateur qui sélectionne les exemples positifs et négatifs un par un en fonction du retour donné par le système. Chaque nouvel exemple d'apprentissage fourni à l'algorithme (sauf le premier) est donc une région sur laquelle le système s'est trompé, ce qui est permet à l'algorithme de *se focaliser sur ses erreurs* (i.e. Active Learning).

Les résultats sont exprimés en fonction du nombre total de régions nécessaires pour atteindre un étiquetage parfait, puis donnant lieu à 25 % d'erreur. Parmi les divers tests, retenons celui qui donne les meilleurs résultats : 1 567 exemples (sur 4 546 disponibles) nécessaires pour atteindre 0 % d'erreur et 516 pour atteindre 25 % d'erreur. Les scores par concept ne sont pas précisés.



Figure 2-3 : ***L'interface graphique de Photobook/FourEyes permet à l'utilisateur de définir des exemples positifs et négatifs d'un concept à apprendre (ici ciel). Le système généralise par apprentissage et annote automatiquement le reste de l'image ainsi que les autres images.***

2.6.2.5 Discussion

FourEyes est, à notre connaissance, l'approche la plus similaire à la nôtre. En particulier, FourEyes est un système permettant une recherche d'images personnalisée et sémantique. Toutefois, il existe des différences importantes. En particulier, la notion de groupements correspondant à un trait de bas niveau

ne semble pas adaptée à la représentation des conjonctions sur les traits. Par exemple, si un concept « tigre » est caractérisé fortement par la présence de noir ET la présence d'orange ET une texture fortement directionnelle, on peut se demander comment FourEyes pourra généraliser ce concept. L'algorithme d'apprentissage sélectionnera d'abord (par exemple) le groupement couleur/noir ou texture/orientation forte, ou couleur/orange mais puisque aucun de ces couples attribut/valeur n'est **à lui seul** suffisant pour capturer le concept « tigre » (i.e. chacun des groupes contient de nombreux autres concepts), on peut légitimement s'attendre à ce que les groupements sélectionnés soit de très petite taille, voire unitaires. Les expérimentations présentées ne permettent pas de vérifier ce fait, mais on peut s'attendre à ce que le système généralise mal les concepts caractérisés pas une conjonction d'attributs. En effet, la notion de groupe repose sur une intervalle de valeurs d'un trait et ne peut donc pas représenter la conjonction de plusieurs traits. Toutefois, dans la réalité des applications, il est fréquent qu'il n'existe pas de trait unitairement discriminant mais plutôt, que la discrimination vienne d'une conjonction de traits. Or, la conjonction de groupes ne peut représenter cette conjonction, car il y a nécessairement des exemples négatifs dans ces groupes. Admettons par exemple que les groupes soit formés selon deux traits, couleur (C) et texture (T), donnant lieu à une hiérarchie de groupes du genre C1, C2, C3, composé de C11, C12, C13, composé de C131, C132, etc. (même chose pour la texture). Admettons que le concept cible soit « tigre » et que les exemples positifs se trouvent dans C2 (« oranges »), C5 (« noirs ») et T3 (« textures orientés »). Aucun de ces groupes ne peut représenter à lui seul le concept « tigre », car il contient des exemples négatifs (d'autres objets orange, noirs, dont la texture est orientée, etc.). L'algorithme devra alors considérer (par exemple) les sous-groupes C21, C23, C51, C54, T32 et T37, qui eux aussi contiendront des exemples négatifs et devront être divisés à nouveau, jusqu'à obtenir de très petits groupes (donc pas de généralisation). La raison est que seule une conjonction d'attributs est discriminante dans ce cas.

Il est tout à fait possible que l'utilisateur d'un tel système cherche à apprendre un concept ayant une forte variabilité dans son apparence, comme le concept « ciel » par exemple. On peut se demander comment réagira le système

lorsque les exemples fournis par l'utilisateur n'ont que peu de points communs. Là encore, on peut s'attendre à ce que tout groupement de taille importante contienne des exemples négatifs. En effet, plus l'utilisateur fournit d'exemples, plus les groupements capables de recouvrir les exemples positifs sans inclure d'exemples négatifs deviennent petits, en particulier lorsque les exemples positifs varient considérablement en apparence. Les auteurs rapportent d'ailleurs que l'utilisation des groupements inter images n'améliore pas les résultats, et ceux-ci ne sont pas utilisés dans les expérimentations présentées. Cela signifie que FourEyes (dans ce contexte) n'est pas capable de généraliser aux autres images de la collection, ce qui explique finalement le nombre relativement important d'exemples d'apprentissage nécessaire pour atteindre 25 % d'erreurs.

2.6.3 Visual Keywords

Une approche intéressante et proche de la nôtre sur certains points est décrite dans [Lim01a] et [Lim01b]. L'auteur propose un système d'indexation/recherche d'images basé sur le concept de « Visual Keyword » (VK). Un VK est une région carrée d'image (un bloc) associée à un concept sémantique donné (« visage », « ciel », « eau » sont des exemples). Le concepteur du système choisit, selon l'application, un ensemble de concepts et construit manuellement pour chacun d'eux un ensemble de VK, en découpant dans des images des blocs correspondants au concept. Cela correspond donc au vocabulaire d'indexation et à l'ensemble d'apprentissage.

Des traits de bas niveau sont calculés pour chaque VK, en l'occurrence un vecteur caractérisant la couleur (moyenne et écart-type pour chaque canal dans l'espace YIQ) et un vecteur caractérisant la texture (moyennes et écart-types des coefficients donnés par des filtres de Gabor pour cinq directions et six orientations).

Lorsque l'ensemble des VK est construit il est alors possible d'indexer de nouvelles images. Pour cela, l'image à indexer est divisée en blocs de taille fixe, et les traits de bas niveau sont extraits de chaque bloc. Tous les blocs de l'image sont alors comparés à tous les VK (*via* la distance « City Block »),

donnant lieu à une valeur d'appartenance à chaque concept. Toutes ces valeurs sont alors regroupées dans une représentation virtuelle sémantique de l'image, qui permet donc la formulation de requêtes sémantiques.

Les expérimentations menées concernent 2 500 photographies personnelles sur huit concepts sémantiques (visage, foule, ciel, sol, eau, verdure, montagne/rocher et bâtiment) d'une part, et 500 photographies issues d'une collection Corel sur six concepts (ciel, eau, montagne/plage, champ, arbre et montagne enneigée). L'auteur présente quatorze requêtes sémantiques/spatiales (on recherche tel concept à tel endroit dans l'image) et montre pour chacune d'elle les quinze premières images que le système renvoie. La plupart des images renvoyées sont pertinentes par rapport à la requête. Il est toutefois difficile de connaître, à partir de ces requêtes arbitraires, l'efficacité précise de l'approche.

L'approche dont nous venons de parler ne satisfait pas nos objectifs. Le vocabulaire d'indexation étant défini arbitrairement par le concepteur du système et non par l'utilisateur, il n'y a pas de personnalisation. L'abstraction atteignable par ce système nous paraît limitée dans la mesure où les « Visual Keyword » sont les entités les plus abstraites (pas de hiérarchisation) et par le fait qu'il n'y a pas de pondération des traits.

2.6.4 Discussion

Les systèmes dont nous venons de parler autorisent la formulation de requêtes sémantiques. Les résultats obtenus dépendent des concepts appris, et se dégradent avec l'augmentation de la variabilité visuelle de ceux-ci (« Herbe » est plus facile à apprendre que « Table »). Il y a également des problèmes lorsque plusieurs concepts on des apparences similaires, cela donne lieu à des confusions.

Les approches que nous appelons « moyen niveau d'abstraction » nous paraissent avoir une portée beaucoup plus grande que les approches à faible niveau d'abstraction. Elles permettent en effet de se rapprocher des vraies attentes des ‘ chercheurs d'images ’, c'est-à-dire une sémantique de haut

niveau. Toutefois, cette catégorie d'approches nous paraît également limitée : la représentation 2D d'une scène, même riche de plusieurs millions de pixels, a perdu presque toute l'information de la scène originale ; ce qui reste peut être vu comme un ensemble d'*indices*, qui nous (les humains) donne un point de départ pour inférer la scène. Cette inférence implique l'utilisation de connaissances diverses, difficiles à identifier mais qui incluent certainement des connaissances sur les relations existant entre les entités visuelles. Considérer une région de l'image, hors de son contexte (le reste de l'image), empêche l'utilisation des connaissances relatives aux relations entre entités et rend son identification très difficile, voire impossible (on se rappelle la difficulté de ces jeux dans lesquels il s'agit d'identifier un objet à partir du détail d'une photographie, et l'évidence de la bonne réponse lorsqu'on voit la totalité de l'image). Or, c'est exactement ce que font les systèmes appartenant à cette catégorie.

L'information visuelle en deux dimensions provenant des images, à elle seule, ne peut pas suffire à inférer de manière fiable et systématique leur contenu sémantique. Cette information visuelle pouvant être interprétée (à raison) de nombreuses façons, il convient donc, s'il on veut désambiguïser l'apparence visuelle (i.e. résoudre le problème du fossé sensoriel) d'utiliser également des informations autres que purement visuelles. L'ensemble de ces informations constitue ce que nous appelons le « *contexte* », qui peut être vu comme un ensemble de contraintes, ou de connaissances *a priori*.

2.7 Les systèmes à haut niveau d'abstraction

2.7.1 Définition

Nous définissons les systèmes à haut niveau d'abstraction de la manière suivante :

Définition 2-III*: Un système capable de combiner des informations visuelles et non visuelles pour inférer à partir des images une représentation sémantique de leur contenu. La sémantique extraite peut porter sur des abstractions allant des objets présents dans l'image à des notions telles que les lieux, le temps, les actions, les impressions.*

Le recours à ce type de système est indispensable lorsque les images doivent être indexées et retrouvées selon des critères ou des concepts *subjectif*, comme par exemple les impressions évoquées par l'image.

Si un concept est *subjectif*, alors par définition il n'existe pas dans l'image mais plutôt dans l'esprit de celui qui possède ce concept. L'image joue alors le rôle d'un « déclencheur ». Par conséquent, pour qu'un système soit capable de détecter ce concept, il doit non seulement utiliser des connaissances de type « visuelles », mais également des connaissances d'un autre type, puisque l'information contenue dans l'image n'est pas à elle seule suffisante.

Ce type de système peut également être utilisé lorsque les systèmes à moyen niveau d'abstraction échouent. Par exemple, il peut s'avérer impossible d'apprendre des concepts liés aux lieux ou évènements en utilisant seulement des exemples et contre exemples étiquetés (par exemple un lot d'images étiquetées « Paris » et un lot d'images étiquetées « non Paris »). Cet apprentissage peut malgré tout être rendu possible *via* l'utilisation de connaissances, extérieures aux images, comme un ensemble de règles. Ces règles peuvent utiliser l'image comme un ensemble d'indices à partir desquels une inférence peut conduire à la reconnaissance d'un concept complexe.

2.7.2 Quelques systèmes

Nous donnons ici quelques exemples de systèmes « à haut niveau d'abstraction ». Toutefois, dans la mesure où ces systèmes sont moins connus, plus hétéroclites et plus « expérimentaux », nous ne rentrerons pas dans les détails de leur implémentation.

2.7.2.1 Kansei

« Kansei » est un mot japonais qui signifie « impression intérieure » mais qui désigne également une « nouvelle branche » de la recherche d'images par le contenu.

Dans [Tan97], l'auteur explique que la recherche d'image traditionnelle, basée sur le contenu visuel de l'image, ne résout que partiellement le problème. En effet, si un artiste par exemple souhaite effectuer une recherche basée, non sur des objets ou des personnes, mais sur des impressions, les techniques traditionnelles ne peuvent répondre à cette attente. Cette incapacité est due au fait que les impressions sont subjectives. Dans ce domaine, dit « *Kansei research* », le focus est sur l'utilisateur plus que l'image, et les mesures de similarités portent sur les *impressions intérieures*, plutôt que sur des similarités visuelles.

Les systèmes de recherche par *Kansei* ([Tan97], [Hay97], ou [Shi99]), s'ils diffèrent radicalement des SRIC traditionnels par leur ambition, n'en demeurent pas moins très traditionnels dans leur architecture. Ces systèmes possèdent un module d'extraction de traits de bas niveau, un module d'apprentissage (réseaux de neurones, clustering) et une interface de formulation des requête et de visualisation des résultats. Nous plaçons ces systèmes dans la catégorie « haut niveau d'abstraction » car les concepts traités étant fortement subjectifs, ils ne sont pas apprenable en se basant sur les images : une partie de l'information indispensable pour apprendre ces concepts est détenue par la personne qui possède ces concepts.

Les résultats présentés dans ces travaux ne manqueront pas d'étonner. En effet, s'il n'est pas encore possible aujourd'hui d'indexer des images par les objets qu'elles contiennent, dans le domaine large, des systèmes comme dans [Hay97] sont toutefois capable d'indexer des images par des concepts comme *mystérieux, frais, sec, romantique ou élégant*, et cela avec un taux d'assignation correct moyen de plus de 80 %.

2.7.2.2 Méta-données temporelles

Dans certains cas, il existe des sources supplémentaires d'informations, c'est-à-dire autres que l'image, disponibles sans « coût » supplémentaire. Combiner ces informations avec celles provenant de l'image peut alors s'avérer plus performant. Cependant, combiner ces informations de types différents n'est pas un problème trivial, comme le savent les personnes travaillant sur la *fusion multimodale* dans le domaine de la vidéo.

Une source simple mais intéressante d'information est *la date et l'heure* de la prise de vue, que les appareils photo numériques stockent automatiquement. Dans [Mul03] par exemple, l'information temporelle associée à chaque photographie est utilisée pour construire une représentation hiérarchique de la base d'images. Parallèlement, ces images sont indexées par les mots clefs visuels que nous avons décrits en 2.6.3. Les résultats, sur 2 400 images et 26 requêtes montrent des performances accrues dues à l'utilisation d'informations temporelles. Cet accroissement des performances montre que l'utilisation d'information *extérieure* aux photographies peut s'avérer être une bonne stratégie.

2.7.3 Discussion

Utiliser toutes les sources d'information possibles, comme la date et l'heure de prise de vue, ou le texte qui entoure l'image dans un document est un moyen d'augmenter les performances d'un système. Dans le cadre de la recherche appliquée, cette stratégie est donc tout à fait pertinente.

Cependant, d'un point de vue plus fondamental, la prise en compte de méta informations (i.e. extérieures à l'image) n'augmente pas la compréhension du problème principal, à savoir extraire du sens à partir d'un ensemble de pixels. De plus, ces méta-informations n'existent que dans certains cas et les systèmes qui les utilisent ne sont donc pas nécessairement adaptés à des données de source différente.

2.8 Conclusion

Nous avons vu qu'il existe principalement deux manières d'aborder la recherche d'image. La première est de travailler au niveau du signal à tous les niveaux du système, que ce soit dans la représentation de l'image ou dans l'expression des requêtes. Nous choisissons de ne pas emprunter cette voie, principalement car les utilisateurs ont généralement des besoins *sémantiques*, que ces systèmes sont intrinsèquement incapables de satisfaire.

Une seconde catégorie de SRIC (qui inclut les systèmes à moyen et haut niveau d'abstraction) fonctionne à un niveau symbolique. A ce niveau, une certaine sémantique peut être extraite des images et manipulée explicitement par le système et l'utilisateur.

Cependant dans le domaine de l'image, par opposition au domaine textuel, il n'existe pas de sémantique directement accessible. Les images s'avèrent n'être que des sources de « signal ». Or, dans la majorité des cas, il n'est pas possible d'expliciter des règles capables de transformer ce signal en symboles. Le recours à des techniques d'apprentissage automatique est donc de rigueur dans la plupart des SRIC symboliques.

Nous allons maintenant voir quelles sont les principales techniques d'apprentissages utilisées dans ces systèmes, mais également, pourquoi nous pensons qu'elles ne sont pas adaptées à notre contexte.

Chapitre 3

Techniques d'apprentissage automatique pour l'indexation

3.1 Introduction

L'apprentissage est un processus de compression des observations et de l'expérience en une forme avantageuse pour le futur. Une hypothèse, ou une description générale est capable d'expliquer un grand nombre d'observations succinctement, et dans la mesure où elle utilise les régularités perçues dans le passé, elle est également capable de prévoir avec succès les événements futurs ([Ris79]). Pour atteindre ce but, de nombreuses solutions sont envisageables, allant de la simple *mémorisation* des expériences et du résultat attendu (souvent, il s'agit de couples *observation/étiquette*) à la *caractérisation* des expériences perçues qui implique une forme de compréhension (un recodage des données initiales qui recombine et/ou pondère les différent traits dont sont constituées les observations initiales de manière pertinente et qui permet une meilleure généralisation). Remarquons que dans la définition de l'apprentissage ci-dessus, il n'apparaît nulle part que celui-ci doive être supervisé, c'est-à-dire guidé par un but bien précis. Cependant, nous allons voir par la suite que la plupart des algorithmes d'apprentissage existants sont de type supervisé et nous discuterons en quoi cela nous paraît être une limitation très importante.

Dans le domaine de l'indexation des images, l'apprentissage peut jouer les rôles suivants :

- **Associer des traits de bas niveau à des symboles** : la tâche est de construire ou d'ajuster les paramètres d'une fonction capable d'associer avec justesse une observation (sous la forme de traits) à un symbole.

- **Sélectionner, pondérer les attributs discriminants** : il s'agit de découvrir automatiquement quels sont les attributs les plus utiles pour une tâche donnée, comme la classification.
- **S'adapter à un utilisateur** : le but est alors d'apprendre au fil des interactions quelles sont les préférences, les habitudes ou toute autre information concernant le comportement d'un utilisateur face au système de recherche d'image en question.

D'une manière générale, on utilise l'apprentissage automatique lorsque la modélisation d'un problème de classification est trop ardue pour être faite « à la main ». Par exemple, la reconnaissance de visages ou de caractères manuscrits, sont des problèmes complexes dont la modélisation sous forme de règles prédéfinies par le concepteur n'est généralement pas possible. Dans ce type de problèmes, il n'existe pas de relations directes entre les composants des données brutes (les pixels) et les concepts. En fait, il existe bien une relation entre les données et les concepts, mais celle-ci n'est accessible que si les données sont « recodées » sous formes de traits plus abstraits et plus pertinents. Cette relation existe bel et bien puisque les humains sont capables de reconnaître ces concepts à partir de l'image seule. Ces traits, appelés aussi motifs proviennent du fait que les données comportent des *régularités*. Une grande partie de l'apprentissage automatique s'intéresse à cette notion de *reconnaissance de formes*.

Comme le soulignent Anil K. Jain et al. dans [Jai00], en dépit de près de 50 ans de recherches intensives dans le domaine de la reconnaissance de formes, il n'existe pas d'approche générale capable de résoudre des problèmes complexes d'apprentissage (ironiquement, un enfant de cinq ans en est capable, et cela sans effort particulier). Par conséquent, de nombreuses approches ont été développées, en fonction de problèmes particuliers.

Nous commencerons ce chapitre en abordant la question de l'apprentissage sous l'angle des *problèmes d'apprentissages*, où nous distinguerons deux classes de problèmes. Nous verrons quelles sont les caractéristiques

principales des méthodes d'apprentissage et quelles sont celles qui nous intéressent particulièrement dans notre contexte. Finalement, nous parlerons de quelques techniques d'apprentissage courantes, sans rentrer dans leurs détails mais en essayant de montrer leurs limitations par rapport à notre contexte.

3.2 Complexité des problèmes d'apprentissage

Si énormément d'études ont porté sur les *techniques d'apprentissage*, bien peu se sont intéressées à la mesure de la *complexité des problèmes d'apprentissages*. Pourtant, il est généralement préférable d'avoir une idée de la difficulté d'un problème avant d'essayer de le résoudre. Il semble que la raison principale de ce vide théorique soit *l'impossibilité*, dans la grande majorité des cas, d'estimer *a priori* la difficulté d'un problème d'apprentissage.

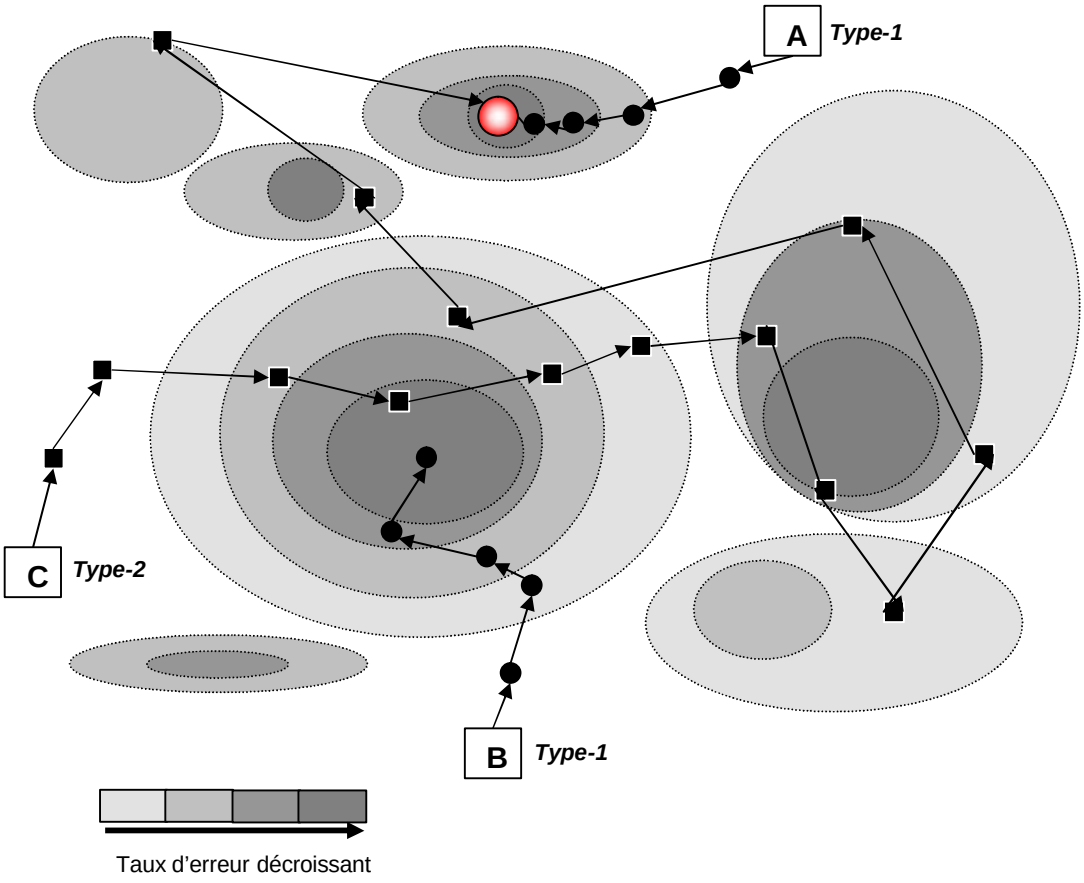


Figure 3-1 : Illustration informelle de la distinction entre problèmes d'apprentissage de type-1 et de type-2 : différents algorithmes explorent différemment l'espace des hypothèses.

Clark et Thornton ont introduit ([Cla97]) la notion de ***type-1*** et ***type-2*** comme mesure de la complexité des problèmes d'apprentissage. Cette distinction porte sur l'accessibilité de l'information pertinente pour résoudre le problème d'apprentissage donné. Dans certains problèmes, l'information pertinente est *relationnelle*, c'est-à-dire qu'elle ne devient visible que lorsque les attributs de base utilisés pour décrire les données sont recombinés d'une manière particulière (il pourrait s'agir des notes d'une mélodie par exemple). Ces problèmes difficiles sont dits de type-2. A l'opposé, il existe des problèmes où il n'est pas nécessaire de considérer les relations entre attributs. Dans ces problèmes, l'information pertinente est « à la surface », donc accessible sans effort.

Nous utiliserons la Figure 3-1 pour illustrer la distinction type-1/type-2. Cette figure symbolise un *espace d'hypothèses*, c'est-à-dire l'ensemble des hypothèses *représentables* par un algorithme donné. Chaque hypothèse, étant une fonction associant une classification (ou plus généralement une réponse) à une observation, peut être évaluée, par un taux d'erreur par exemple. Dans notre schéma, nous symbolisons ce taux d'erreur par des zones grisées. Plus la zone est sombre, plus le taux d'erreur est faible. Nous avons représenté le concept (ou hypothèse) cible par un cercle dégradé. Le but d'un algorithme d'apprentissage est alors de trouver ce concept cible sans rester bloqué dans les minima locaux. Cette recherche est symbolisée dans notre schéma par un chemin : chaque étape représente une modification de l'hypothèse courante (souvent, il s'agit de la prise en compte d'un attribut supplémentaire).

3.2.1 Problèmes d'apprentissage de type-1

Dans un problème d'apprentissage de type-1, le lien existant entre les données (l'ensemble d'apprentissage) et les sorties (des concepts à apprendre) est direct, ou encore *statistiquement visible*. Il est possible de savoir si un problème est de type-1 en examinant la matrice des probabilités conditionnelles créée à partir des données. Si le problème est de type-1, il existera dans cette matrice des probabilités fortes d'avoir détecté un (ou plusieurs) concept sachant

certaines entrées. A l'inverse, si le problème n'est pas de type-1[19], ces probabilités seront proches de l'aléatoire. Remarquons que s'il est possible de déterminer qu'un problème spécifique est de type-1, il est alors possible de le résoudre.

Lorsque des liens directs existent entre les entrées et les sorties (dans [Cla97], ou [Tho03] ces liens sont appelés *justifications*), leur découverte implique donc une étude statistique de probabilité. Ces justifications sont nombreuses, leur nombre est de l'ordre du nombre d'exemples différents représentables, chaque exemple pouvant être une hypothèse. Cette taille est donc généralement « grande » mais pas infinie. Par conséquent, des algorithmes peuvent explorer exhaustivement ou partiellement (guidés par des heuristiques) cet espace des justifications. Les problèmes d'apprentissage de type-1 ne sont donc pas les plus problématiques, ils sont résolus par de nombreuses techniques d'apprentissage.

La Figure 3-1 montre deux exemples de résolution de problèmes de type-1, notés A et B. Le chemin A, par exemple, montre les étapes intermédiaires d'apprentissage pour la résolution du problème A. La nature de ces étapes peut varier, mais souvent, à chaque nouvelle étape correspondra l'intégration d'un nouvel attribut dans la ou les hypothèses courantes. Le point important ici est qu'à la fin de chaque étape, l'algorithme s'est rapproché du but (ou au pire, ne s'en est pas éloigné) : c'est le concept de descente de gradient. En général, les algorithmes génèrent un ensemble de « prochains bouts de chemin » possibles et sélectionnent celui qui se rapproche le plus du concept cible. En ce sens, ils privilégient une « récompense » immédiate. Cette stratégie fonctionne parfaitement pour le problème A : le taux d'erreur diminue régulièrement jusqu'à atteindre un taux nul.

Pour le problème B, il en va autrement. Le chemin suivi est tel que le taux d'erreur diminue à chaque étape, cependant, le concept cible n'a pas été

[19] La notion de type-1 est liée à celle de séparabilité linéaire. Dans un problème linéairement séparable, toutes les variables sont numériques et les variables de sortie peuvent être obtenues par combinaison linéaire pondérée des variables d'entrée. Pour que cela soit possible, il faut que les entrées varient de manière monotone avec les sorties. Par conséquent, les problèmes linéairement séparables *sont* de type-1. La réciproque n'est pas vraie.

découvert. La question est de savoir pourquoi, sachant que les données contiennent suffisamment d'information pour résoudre parfaitement le problème. La réponse peut prendre plusieurs formes, on peut par exemple parler de *minimum local* : un minimum local est atteint lorsque toute modification de l'hypothèse courante aboutit à une augmentation du taux d'erreur. Plus généralement, on peut parler de *piège* dans l'espace des hypothèses utilisé par l'algorithme : l'algorithme est piégé lorsqu'il a utilisé toutes les données à sa disposition mais que le taux d'erreur est toujours élevé. Si cela arrive, c'est que les données n'ont pas été utilisées de manière pertinente. En particulier, c'est le cas lorsque des attributs contenant de l'information essentielle sont rejetés par l'algorithme (sélection ou pondération de traits, réduction de dimensions). En effet, lorsque les informations pertinentes sont relationnelles, il est possible que les constituants de ces relations, considérés indépendamment n'apportent aucun gain d'information pour l'apprentissage d'un concept. Nous allons voir maintenant que contourner ces pièges nécessite *un recodage des données*.

3.2.2 Problèmes d'apprentissage de type-2

A l'opposé, un problème d'apprentissage de type-2 est caractérisé par l'absence de liens directs entre entrées et sorties. Ces liens existent, c'est-à-dire que toute l'information nécessaire à l'inférence des sorties par rapport aux entrées est bien présente, mais ils sont indirects. La visibilité statistique de ces liens dépend d'un **recodage** systématique de l'espace des données. Or, l'espace des recodages possibles correspond à l'espace des machines de Turing applicables, qui est infini [Cla97].

Un exemple extrême de problème d'apprentissage de type-2 est celui de la parité. On dispose de vecteurs d'apprentissage composés de n binaires (les entrées). La sortie (binaire) vaut 1 si le nombre de '1' dans le vecteur d'entrée est impair, 0 sinon. Il est montré dans [Tho03] qu'une étude statistique des probabilités entre entrées et sortie échoue : quel que soit l'ordre considéré, les probabilités conditionnelles sont toutes de 0.5.

Les problèmes « réels » ne sont quasiment jamais purement de type-2 : il existe presque toujours certaines corrélations statistiques directes entre entrées et sorties. La nature du problème dépend bien sûr de la manière dont il est originellement représenté. Nous verrons d'ailleurs au chapitre 4 que le but implicite des traits de bas niveau est de transformer le problème initial en problème solvable de type-1.

La résolution d'un problème de type-2 est symbolisée dans la Figure 3-1 par le chemin C. On voit que ce chemin comporte beaucoup plus d'étapes que les chemins A et B. Plus important : il apparaît que le taux d'erreur ne diminue pas de manière monotone en fonction du nombre d'étapes accomplies : souvent, l'erreur augmente significativement. Une des idées que nous défendrons au chapitre 5 est que *les étapes intermédiaires du processus de recodage des données ne sont pas nécessairement immédiatement avantageuses par rapport au concept cible.* Cependant, comme illustré sur la figure, ces étapes sont *indispensables* pour éviter les pièges.

3.2.3 Discussion

Les auteurs de [Cla97] suggèrent que les problèmes d'apprentissages de type-2, loin d'être rares, sont en fait extrêmement communs. En particulier, parmi les problèmes couramment résolus par les êtres vivants, nombreux sont les problèmes de type-2. Par exemple, l'apprentissage de la langue, ou de la vision, est un problème dont la complexité est bien au-delà des problèmes de type-1.

Si ces problèmes sont de type-2, comment la « nature a-t-elle pu trouver le bon recodage » parmi l'infinité des recodages possibles ? Une hypothèse souvent avancée, notamment dans le domaine de l'acquisition de la langue, est celle de la génétique. Le mouvement, dit Nativiste, considère en effet que certaines fonctions complexes du cerveau sont rendues possibles par des prédispositions acquises génétiquement, permettant de réduire grandement les problèmes d'apprentissage subséquents. Par exemple, l'acquisition du langage nécessite, selon Noam Chomsky (voir discussion dans [Pal86]), une « grammaire universelle » innée. L'argument principal étayant cette hypothèse étant celui de

la « *pauvreté du stimulus* », c'est-à-dire que le « bain linguistique » (les entrées) dans lequel est plongé un apprenant ne contient pas suffisamment d'informations pour l'apprentissage des sorties (la maîtrise du langage). Remarquons que l'attitude nativiste ne fait que déplacer le problème : en admettant que le cerveau ne puisse apprendre une fonction complexe qu'avec l'aide d'un *biais génétique*, comment celui-ci est-il « appris » ?

Une autre alternative est de considérer qu'il existe un mécanisme d'apprentissage général, capable de surmonter la difficulté des problèmes de type-2. Un tel mécanisme ne peut être basé sur une recherche exhaustive dans l'espace des hypothèses, celui-ci étant infini. Si un tel mécanisme existe, il doit donc contenir des *biais*, un ensemble de contraintes capables de réduire l'espace des hypothèses. Ces contraintes doivent également être suffisamment génériques pour expliquer l'immense diversité des comportements appris.

Il existe, dans le monde scientifique, une idée récurrente, présente sous diverses formes mais qui peut se résumer par le concept de *simplicité*. L'idée fondamentale est que, face à une multitudes d'hypothèses capables d'expliquer un ensemble de faits, mieux vaut choisir la ou les plus simples. La formulation la plus célèbre de cette idée est sans doute le fameux « Occam's Razor » (ou *rasoir d'Occam*, voir [Mit97], pages 65 et 66) : *de toutes les hypothèses qui s'accordent avec les données, préférer la plus simple*.[20] Un des arguments les plus forts en faveur de ce principe est le suivant : il existe beaucoup plus d'hypothèses compliquées que d'hypothèses simples, par conséquent, la probabilité qu'une hypothèse simple soit en accord avec les données est beaucoup plus faible (il existe beaucoup d'hypothèses compliquées capables d'expliquer des données, peu de courtes), donc, si on en trouve une, elle doit être bonne. De plus, une hypothèse simple autorise plus de généralisation qu'une hypothèse compliquée. Un des problèmes avec cet argument est que la simplicité de la formulation d'une hypothèse ne tient pas compte de la taille de sa représentation interne. Par exemple, je peux formuler très succinctement une hypothèse, comme « Tout est relatif. » (3 mots), mais quelle est la taille, ou

[20] La version anglaise, beaucoup plus élégante, est : Prefer the simplest hypothesis that fits the data.

la quantité d'information, nécessaire au *support et à l'interprétation* de cette formulation ?

Une réponse à cette lacune dans la formulation de *l'idée de la simplicité* est donnée dans le *Principe de la description de taille minimum (Minimum Description Length Principle, ou MDL :* [Grü05]). Ce principe recommande de minimiser, non seulement la taille de l'expression des hypothèses, mais également la taille du codage de tout processus utilisé pour recoder (interpréter) les données. Ce principe implique que, s'il est nécessaire de recoder les données, ce recodage doit être effectué en maximisant l'utilisation (et la réutilisation) des *régularités* les plus fréquentes dans les données, afin de minimiser la taille du recodage et des hypothèses exprimées *via* ce recodage. Nous discuterons abondamment de la notion fondamentale de régularité au chapitre 5 et verrons que de cette notion émergent les concepts d'abstraction et de similarité.

3.3 Caractéristiques des techniques d'apprentissage

3.3.1 Introduction

Avant de détailler les techniques les plus utilisées en indexation et recherche d'images, nous considérons les caractéristiques dont il faut tenir compte lorsque l'on choisit l'une d'elle, ou lorsque l'on entreprend de créer une nouvelle méthode d'apprentissage. Chaque technique ayant des propriétés bien spécifiques il convient de choisir celle qui est la plus appropriée aux contraintes existantes (notamment le problème d'apprentissage lui-même). Si aucune technique ne semble appropriée, les faiblesses de ces méthodes pourront sans doute être clarifiées à la lumière des critères suivants.

3.3.2 Type du critère de décision

L'ensemble des informations utilisées par un algorithme pour s'orienter dans l'espace des hypothèses constitue le *critère de décision*. Il existe deux types principaux de critères qui créent une dichotomie forte entre algorithmes

d'apprentissage. Le premier type repose sur des mesures d'homogénéité, basées sur une notion de similarité. Le but de l'algorithme d'apprentissage est alors de créer une hypothèse qui rend compte des similarités entre les éléments de l'ensemble d'apprentissage. Cette hypothèse, ou représentation, doit être suffisamment concise (donc abstraite) pour permettre une marge de généralisation. Les algorithmes ayant ce type de critère sont généralement appelés *algorithmes d'apprentissage non supervisé*. Le résultat de ces algorithmes dépend en majorité de la mesure de similarité choisie. Le deuxième type de critère, à l'opposé du précédent, est extérieur aux données : il repose sur l'étiquetage des exemples d'apprentissages. Le but de l'apprentissage est alors de découvrir une hypothèse capable d'affecter la bonne étiquette à de nouvelles observations. Une manière d'atteindre ce but est de chercher une mesure de similarité en accord avec l'étiquetage des données.

En apprentissage supervisé, l'algorithme d'apprentissage utilise comme entrée un ensemble d'apprentissage où chaque élément est étiqueté. Le but de l'apprentissage est alors de construire (ou adapter) une fonction capable, non seulement de classer correctement les exemples d'apprentissage, c'est-à-dire de leur attribuer leur étiquette originale, mais également d'être capable de classer des observations inconnues. Le but de l'apprentissage supervisé est donc très clair, ainsi que l'évaluation du résultat.

En apprentissage non supervisé, les données ne sont pas étiquetées, ou si elles le sont ne sont pas utilisées. Le but de l'apprentissage non supervisé est donc nécessairement différent. En fait, il peut y avoir plusieurs buts différents ([Hin99]) : un des buts les plus populaires en vision par ordinateur est de réduire les redondances existants dans les données. Cette tâche ne nécessite pas de supervision car la notion abstraite de redondance est indépendante de tout contexte particulier. En supprimant les redondances, il devient possible de compresser l'information. Un autre but, lié au précédent d'une certaine manière, est d'utiliser l'apprentissage non supervisé pour regrouper les observations similaires. Cette application, sans doute la plus courante de l'apprentissage non supervisé, est appelée *Clustering*. Même si les groupements découverts par un algorithme de Clustering semblent « libres », c'est-à-dire dépendant

principalement des données, il est fondamental de garder à l'esprit que beaucoup d' « *a priori* » sont implicitement codés dans un algorithme de Clustering. En particulier, la mesure de similarité, définie *a priori*, détermine en grande partie les groupements obtenus. En pratique, on fait souvent l'hypothèse que les attributs représentants les observations sont indépendants, ou du moins que l'essentiel ne réside pas dans leurs relations.

3.3.3 Le Biais Inductif

Nous avons vu que bon nombre de problèmes d'apprentissage sont difficiles car l'espace des hypothèses est soit immense, soit infini. Explorer de tels espaces requiert donc une forme de *guide* car l'exploration exhaustive n'est pas envisageable. Ce *guide* est un ensemble de contraintes permettant de privilégier ou d'éliminer des hypothèses. En apprentissage automatique, cet ensemble de contraintes est dénommé le *biais inductif*. C'est un *biais* car il apporte une certaine subjectivité dans la manière dont sont considérées les hypothèses. Il est *inductif* car ce biais est utilisé pour induire des hypothèses abstraites (génériques) à partir d'un ensemble d'exemples concrets (ou spécifiques).

Mitchell [Mit97] définit le biais inductif de la manière suivante :

Le biais inductif d'un système apprenant est l'ensemble des hypothèses qui, combinées avec les exemples de l'ensemble d'apprentissage, permet la classification déductive de nouveaux exemples par le système.

En d'autres termes, le biais inductif représente la manière dont le système classifie de nouvelles instances en fonction de l'ensemble d'apprentissage. Par exemple, le biais inductif (approximatif) de l'algorithme ID3 (arbre de décision) est « Parmi toutes les hypothèses, préférer les hypothèses simples ». En effet, ID3 sélectionne le premier arbre acceptable dans son parcours de tous les arbres possibles, ce parcours étant de simple vers complexe. Cette manière de construire des hypothèses à partir des exemples d'apprentissage conditionne donc la classification de nouvelles instances.

Les biais inductifs peuvent prendre des formes complexes, par exemple, le biais inductif des réseaux de neurones à propagation arrière est très difficile à formuler. Toutefois, une *composante* courante pour un biais inductif est, d'une manière générale, une préférence pour la simplicité.

3.3.4 Réactivité de l'apprentissage

Par réactivité de l'apprentissage nous entendons la dérivée du rapport entre la qualité de l'apprentissage et la taille de l'ensemble d'apprentissage. En d'autre terme, l'apprentissage est réactif lorsque la qualité de l'apprentissage augmente rapidement par rapport à la taille de l'ensemble d'apprentissage.

La réactivité est une caractéristique de l'apprentissage dont il faut tenir compte particulièrement dans le cadre de l'apprentissage « en direct », lorsque existent des interactions avec l'utilisateur. Lorsque l'apprentissage est fait « off-line » ou une fois pour toutes, la qualité prend le pas sur la réactivité.

Dans notre travail, où l'utilisateur joue un rôle important dans ses interactions avec le système, la réactivité est un critère prépondérant. Dans le domaine de la théorie de l'apprentissage, il existe de nombreux travaux (rapportés dans [Mit97] s'intéressant à quantifier formellement la taille minimale d'un ensemble d'apprentissage pour obtenir un taux d'erreur donné. Il est donc souvent possible de donner une borne supérieure à l'ensemble d'apprentissage qui garantie, avec une probabilité donnée, un taux d'erreur donné. En règle générale, le nombre d'exemples requis dépend de la complexité de l'espace des hypothèses : plus on considère d'hypothèses, plus il faut d'exemples avant de sélectionner une hypothèse qui donne un faible taux d'erreur.

3.3.5 Stabilité de l'apprentissage

De la même nature que la réactivité, la stabilité qualifie le comportement de l'apprentissage face à l'ajout d'exemples. L'apprentissage sera dit stable si la qualité de l'apprentissage par rapport à la taille de l'ensemble d'apprentissage est une fonction croissante ou stable.

Par exemple, une des motivations qui a aboutit à la création des réseaux ART (Adaptive Resonance Theory) [Car03] est le manque de stabilité des réseaux de neurones. Dans un réseau de neurones, la fonction apprise est adaptée pour tenir compte de chaque exemple présenté. Ainsi, si le réseau a appris un certain motif avec succès, le fait de faire apprendre un autre motif au réseau va modifier la *fonction*, de sorte qu'elle tienne compte des deux motifs à la fois. La conséquence est une dégradation des performances pour la reconnaissance du premier motif.

Une caractéristique fondamentale des réseaux ART est la capacité de comparer différents motifs et, selon leur similarité, adapter un motif déjà appris, ou créer un motif différent. Plus précisément, lorsqu'un vecteur est présenté à un réseau ART, il est comparé à tous les motifs appris précédemment. S'il ressemble suffisamment à l'un de ces motifs, celui-ci sera adapté pour tenir compte du nouvel exemple présenté (comme dans un réseau de neurones). Par contre, si le vecteur ne ressemble pas suffisamment à *aucun* des motifs existants, un nouveau motif est créé, afin de ne pas « endommager » ce qui a déjà été appris par le réseau. Cependant, juger de la *nouveauté* d'un motif est loin d'être trivial, et nécessite généralement l'ajustement manuel d'un seuil (nommé *seuil de vigilance* par les auteurs de ART). Nous aborderons d'ailleurs cette notion lors de notre proposition, et nous verrons en quoi l'évaluation de la « nouveauté » requiert la connaissance d'un repère *absolu*.

3.3.6 Résistance au bruit

La résistance au bruit qualifie le comportement du système face à l'ajout d'observations incohérentes par rapport au concept cible. Dans notre contexte fortement interactif, il s'agit d'une propriété indispensable puisque nous considérons l'utilisateur comme une source potentielle de mauvais exemples d'apprentissage (bruit).

L'apprentissage par l'exemple ne dispose que de ces exemples pour inférer une fonction de classification. Il est donc important que ces exemples soient bons. Toutefois, il est souvent nécessaire qu'une certaine tolérance soit possible lorsque de « mauvais » exemples sont rencontrés. Un exemple peut

être mauvais pour plusieurs raisons. Il peut premièrement être mal étiqueté. Il se peut également que certains de ses attributs soient manquants ou que leurs valeurs soient fausses. Dans le premier cas, le bruit se situe au niveau des exemples, dans le second cas il s'agit de bruit dans les attributs.

3.3.7 Actif / Passif

Un algorithme d'apprentissage supervisé actif est un algorithme qui dispose d'un ensemble d'observations non supervisées et qui peut réclamer (à l'utilisateur d'un système, au concepteur) une étiquette pour une observation particulière.

L'idée derrière ce paradigme d'apprentissage est que les exemples fournis à l'algorithme durant l'apprentissage ne sont pas tous équivalents en terme d'intérêt pour l'apprentissage. Par exemple, un exemple qui se situe à la limite (selon la mesure de similarité utilisée) des exemples positifs et négatifs apporte plus d'information qu'un exemple « très positif » ou « très négatif ». Par conséquent, la nature des exemples influence la réactivité de l'apprentissage.

Les algorithmes non actifs sont dits passifs, ils utilisent indifféremment les données d'apprentissage données, et ne réclament jamais d'informations supplémentaires.

3.4 Apprentissage automatique : les approches standards

Dans le domaine de l'apprentissage automatique, comme dans tous les autres domaines, il existe un ensemble d'idées dominantes, autour desquelles gravitent de multiples clones améliorés ou adaptés à un sous domaine particulier. Nous présentons maintenant les approches centrales de l'apprentissage automatique tout en discutant de leur adaptation aux problèmes difficiles d'apprentissage, dits de type-2. Les approches sont présentées par ordre croissant de complexité « conceptuelle ». Le Tableau 3-2 introduit informellement les techniques dont nous allons parler.

	Apprendre, c'est…	*Reconnaître un exemple, c'est…*
k-NN	***Mémoriser***	***Le comparer à ceux déjà mémorisés***
Clustering	***Regrouper les observations similaires***	***Le comparer à des représentants de groupes d'exemples***
Réseaux de neurones	***Corriger ses erreurs en ajustant les poids d'un modèle prédéfini***	***Le propager dans le réseau appris (i.e. la fonction dont on a appris les poids)***
Algorithmes génétiques	***Considérer plusieurs hypothèses que l'on combine et modifie aléatoirement en ne conservant que les meilleures***	***Le confronter à la meilleure des hypothèses (i.e. une fonction dont on a optimisé les paramètres)***
Machines à vecteurs de support	***Complexifier les données afin de trouver une solution simple : un hyperplan séparateur***	***Le situer par rapport au plan séparateur (défini par quelques exemples d'apprentissage particuliers)***

Tableau 3-1 : Introduction à quelques techniques courantes d'apprentissage.

3.4.1 Méthode des plus proches voisins

3.4.1.1 La méthode

La méthode des plus proches voisins (k-Nearest Neighbor Learning, ou k-NN) appartient à la classe des algorithmes d'apprentissage basés sur l'exemple (Instance-Based Learning). Contrairement aux méthodes d'apprentissage qui construisent une représentation explicite du concept à apprendre (un arbre, un réseau, etc.) à partir d'exemples d'apprentissage, les algorithmes basés sur l'exemple se contentent de stocker ces exemples d'apprentissage. Généraliser au-delà de ces exemples ne se fait alors que lorsqu'un nouvel exemple est classé. C'est pour cette raison que ces algorithmes sont souvent qualifiés de « paresseux ».

Au lieu de calculer une représentation du concept cible une fois pour toute dans tout l'espace, ces algorithmes sont capables de représenter un concept localement et différemment pour chaque exemple à classer. C'est là un avantage clé de ces méthodes [Mit97- Chapitre 8]. Mitchell compare les

algorithmes d'apprentissage paresseux aux autres algorithmes d'apprentissage, que nous qualifierons de « avide » (traduction libre de « Eager ») :

La première chose est que l'apprentissage est généralement plus rapide dans le cas de l'algorithme paresseux ; par contre la classification prend plus de temps. Une deuxième différence fondamentale est la suivante :

- Les méthodes paresseuses peuvent tenir compte de l'exemple à classer lorsqu'elles généralisent au-delà de l'ensemble d'apprentissage,
- Les méthodes avides ne peuvent pas car dès lors que l'exemple à classer est observé, il est en fait déjà classé.

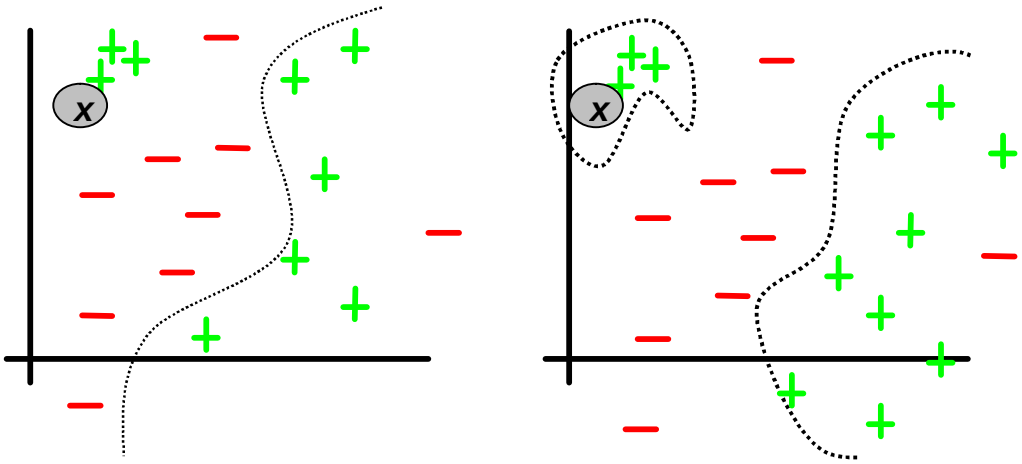


Figure 3-2 : A gauche, la représentation globale d'un concept (méthode avide). A droite, la méthode paresseuse autorise plusieurs représentations locales du concept.

En fait, et comme on peut le voir sur la Figure 3-2, les algorithmes paresseux ont la possibilité de représenter implicitement un concept global par un ensemble d'approximations locales alors que les autres types d'algorithmes doivent construire une unique approximation globale.

L'implication de ce fait pour un SRIC est qu'un algorithme de type k plus proches voisins est capable de représenter un concept (entité physique) selon de nombreuses approximations locales, correspondant chacune à une apparence particulière de l'entité physique. Il est donc possible de modéliser le

concept « Ciel », avec toutes ses nuances (clair, nuageux, gris, coucher de soleil, lever de soleil, de nuit, etc.), par un algorithme paresseux.

3.4.1.2 Application à la recherche d'images

Dans [Sin01], l'algorithme des k plus proches voisins est utilisé dans une tâche de compréhension d'images, c'est-à-dire l'étiquetage automatique de régions de l'image. Cinq étiquettes sont utilisées : arbres, herbes, ciel, reflet du ciel dans une rivière et reflet des arbres dans une rivière ; les images sont des photographies aériennes prises à basse altitude. Les auteurs montrent que ce classificateur est particulièrement bien adapté au domaine de la compréhension d'image car il est efficace et résistant au bruit. L'approche est comparée avec un autre classificateur, un réseau de neurones, qui donne lieu à des performances moindres.

Les auteurs de [Lon03] ont également sélectionné l'algorithme k-NN, leur choix étant motivé principalement par des critères de rapidité. En effet, leurs travaux portent sur la reconnaissance automatique de visages dans un album photographique familial et dans ce contexte fortement interactif, l'apprentissage des visages doit être réactif (peu d'exemples d'apprentissage), donner lieu à des temps de calcul peu élevés et être capable d'effectuer la reconnaissance dans des délais également courts. Les expériences menées sur une base de 1 700 images, où les quinze personnes apparaissant le plus fréquemment dans les photographies devaient être apprises, donnent lieu à des résultats satisfaisant en termes de reconnaissance mais également de temps de calcul. Lors de la reconnaissance, l'algorithme produit une liste ordonnée (par plausibilité décroissante) de candidats (un sous-ensemble des individus « appris »). Dans 64 % des cas, le premier candidat (sommet de la liste) est le visage à reconnaître, dans 87 % des cas le nom du visage à reconnaître est l'un des deux premiers candidats de la liste et dans 99 % des cas, il est l'un des cinq premiers candidats de la liste. Cette approche est intéressante car elle permet une personnalisation effective de la recherche dans un album photographique familiale électronique, ce qui est également notre but.

L'utilisation de la méthode des k plus proches voisins, appliquée à la personnalisation est une approche que nous avions d'ailleurs expérimentée lors de nos travaux antérieurs. Dans [Bis04], nous décrivons un système dans lequel un utilisateur peut lui-même définir les termes de l'indexation. Pour cela, une interface (Figure 3-3) permet de visualiser les images, segmentées en blocs, et permet également de définir de nouveaux concepts, en leur associant des exemples positifs et négatifs de blocs. Le système donne un retour *immédiat* à l'utilisateur en montrant sur l'image courante les blocs indexés par le concept en cours d'apprentissage, ainsi que les blocs qui ne le sont pas. L'utilisateur perçoit alors *visuellement* les erreurs commises par le système (faux positifs ou négatifs), ce qui lui permet de les corriger en ne donnant que des exemples d'apprentissage pertinents.

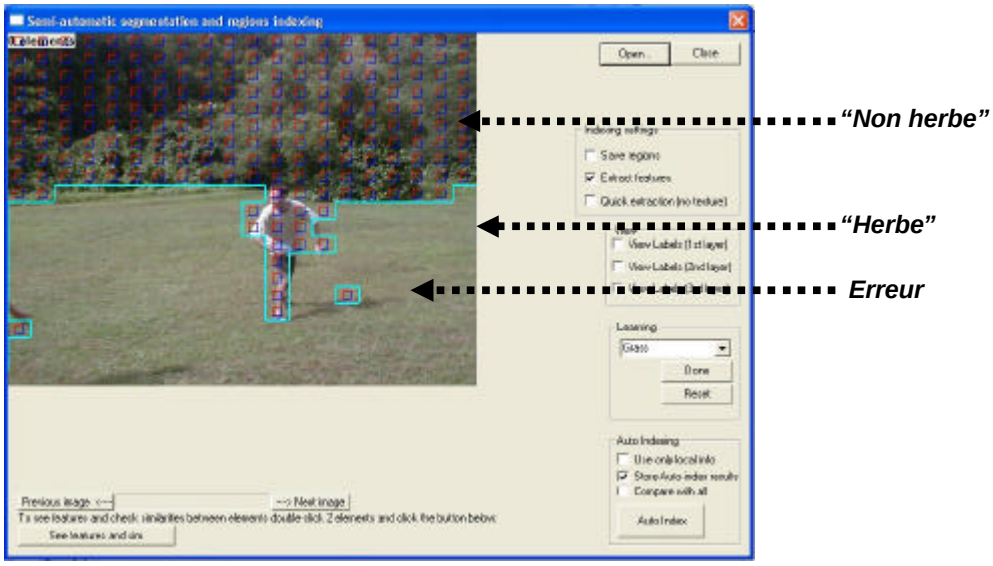


Figure 3-3 : Interface utilisées dans [Bis04] pour sélectionner les exemples positifs et négatifs utilisés par le k-NN pour indexer la base d'image selon le concept « Herbe ».

A chaque concept correspond un classifieur distinct, ce qui permet d'adapter la mesure de similarité au concept appris, par pondération des attributs. Les expériences que nous avions menées, bien que préliminaires, ont montré que l'approche est intéressante. En particulier, la réactivité du système s'est révélée satisfaisante : sur cent images et six concepts, le taux de reconnaissance variait de 61 % pour une moyenne de 4.3 exemples d'apprentissage par concept, à 70 % pour une moyenne de 14.2 exemples.

3.4.1.3 Les kNN faces aux problèmes de type-2

Clairement, la méthode des k plus proches voisins n'est d'aucun intérêt pour la résolution de problèmes de type-2. Les données sont utilisées dans leur forme d'origine et il n'y a aucune forme de recodage[21]. Sans recodage, un problème de type-2 n'est pas solvable par les méthodes statistiques connues.

Toutefois, il faut souligner un point important : bien que cette méthode soit parmi les plus simples (conceptuellement) que l'on puisse imaginer, elle n'en demeure pas moins très efficace. Les kNN sont, en moyenne, tout à fait compétitifs par rapport à d'autres techniques beaucoup plus complexes. En particulier, le 1-NN, qui se contente d'attribuer à une nouvelle observation la classe du point étiqueté le plus proche, se révèle être une technique efficace et surtout très fiable. Par ailleurs, il est recommandé dans [Jai00] (une étude très citée sur les techniques de reconnaissance de motifs) de comparer les résultats d'un nouvel algorithme d'apprentissage avec ceux du 1-NN, car ses performances moyennes, quel que soit le contexte, sont constantes et souvent très bonnes. De plus, mis à part la mesure de similarité entre observations, il n'y a pas de paramètres à ajuster.

3.4.2 Le « Clustering »

3.4.2.1 La méthode

Le clustering est une technique d'apprentissage couramment utilisée en traitement d'image. Le clustering peut être défini comme un processus qui organise des objets dans des groupes dont les membres se ressemblent selon des critères donnés. Par exemple, cette technique est souvent utilisée pour séparer les pixels d'une image en groupes homogènes, c'est la segmentation.

Il y a deux principaux types de clustering : le partitionnement, dans lequel chaque objet est assigné à un et un seul groupe (dit clustering « plat », voir

[21] D'un point de vue stockage, et representations des données, ce n'est pas très "Occam's Razor".

Figure 3-4) et le clustering hiérarchique dans lequel chaque groupe de taille supérieure à 1 est lui-même composé de sous-groupes.

Dans [FAS99], l'auteur dénote trois utilisations des algorithmes de clustering :

- classification avec connaissances *a priori*,
- classification pour tester une hypothèse,
- classification non supervisée.

La première classe concerne les algorithmes de clustering supervisés ou semi supervisés. Ces algorithmes utilisent des observations étiquetées afin de contraindre le processus de formation de groupes. L'utilisation de ces observations étiquetées peut se faire de plusieurs manières. Par exemple, on peut les utiliser comme « graines » et regrouper les autres observations autour ou les utiliser pour déterminer des mesures de similarités pertinentes entre observations. Par exemple, dans [Dem99], les auteurs utilisent simultanément des données étiquetées et non étiquetés, il s'agit d'apprentissage semi supervisé. Les données sont partitionnées par un algorithme de clustering intrinsèquement non supervisé. Toutefois, un schéma d'optimisation génétique utilisant une mesure de la qualité du clustering optimise celui-ci. Cette mesure de qualité tenant compte de la dispersion des éléments ayant la même étiquette, il s'agit bien d'un algorithme semi supervisé. Les auteurs concluent que l'utilisation de données non étiquetées en apprentissage augmente les capacités de généralisation.

Dans le second cas, on souhaite tester une hypothèse sur les données qui, si elle est vérifiée, donnera lieu à de « bons » clusters. Il existe plusieurs mesures de la qualité des clusters, selon le contexte du problème, utilisées pour déterminer si les clusters sont « bons ». Une mesure très classique prend en compte la distance intra classe (les observations contenues dans un cluster se ressemblent-elle ?) et interclasse. Dans nos travaux précédents [Bis03], nous utilisons le clustering dans ce cadre : nous cherchons à optimiser une mesure de similarité entre images et utilisons pour cela un algorithme génétique. La

fonction d'évaluation de cet algorithme est la qualité du clustering des images obtenue en utilisant la mesure de similarité à tester.

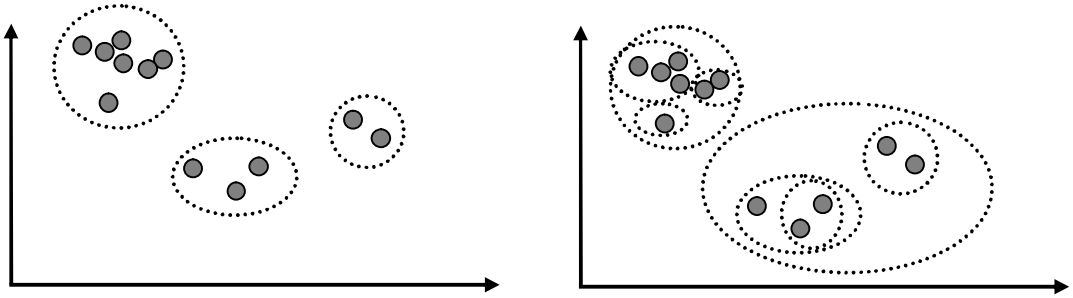


Figure 3-4 : Le clustering regroupe les observations similaires selon une mesure de similarité donnée.

Les travaux décrits dans [Cuc98] sont un autre exemple où le clustering est utilisé pour tester une hypothèse. L'auteur utilise un algorithme génétique dans lequel chaque élément de la population représente une distribution d'observations (des vecteurs de traits représentant des images) dans les clusters. La qualité du clustering (cette fois calculé selon la distance moyenne des observations d'un cluster à leur centre de gravité) est utilisée comme fonction d'évaluation de l'algorithme génétique.

Le troisième cas, le clustering non supervisé est utilisé pour la « fouille de données », c'est-à-dire pour découvrir des structures dans un ensemble de données sur lequel on n'a « aucun » *a priori*.

L'algorithme de clustering typique s'appelle l'algorithme (k) moyennes mobiles (de l'anglais *k-means*). Son principe est simple : former k groupements d'observations par itérations successives. Au début, k observations sont sélectionnées aléatoirement et constituent des « graines » autour desquelles vont s'agglomérer les observations restantes : chaque observation est affectée au groupe ayant la graine la plus proche. Lorsque toutes les affectations ont été faites, l'algorithme sélectionne un représentant pour chaque groupe, qui n'est plus la graine initiale mais le centre de gravité du groupe. Toutes les observations sont de nouveau affectées à un groupe selon leur distance aux représentants. L'algorithme itère jusqu'à stabilisation, c'est-à-dire lorsqu'il n'y a plus de changement de groupe.

3.4.2.2 Applications à la recherche d'image

Dans [Che03b], basés sur la conviction que les images partageant la même sémantique partagent également les mêmes clusters, les auteurs proposent une approche de la recherche d'images basée sur le clustering. Des traits sont tout d'abord calculés sur les images de la collection, puis les distances entre paires d'images sont calculées et stockées. Lorsqu'une image requête est présentée au système, les images proches (au sens d'une mesure de similarité non présentée) sont regroupées et les représentants de chaque cluster sont présentés à l'utilisateur, avec la possibilité de sélectionner un des clusters pour en voir le détail. Les auteurs rapportent une précision moyenne de 0.54, sur dix catégories et 60 000 images issues de la collection Corel. Parmi les catégories choisies, on trouve des chevaux (marron sur fond vert), des bus londoniens (rectangles rouge vif), des plages ou encore des bâtiments.

Les auteurs de [Kim03] partent de la constatation suivante : lorsqu'une requête est complexe, l'espace des traits utilisés pour représenter les images et la perception de l'utilisateur sont très différents. Par conséquent, représenter la requête dans l'espace des traits est difficile et implique non pas une, mais plusieurs régions de l'espace. Il est alors nécessaire que le système soit capable de supporter les requêtes dites disjonctives. La première étape est une requête de l'utilisateur : une image exemple. Un nombre k d'images similaires est sélectionné. L'utilisateur évalue la pertinence de ces k images, c'est la phase de bouclage de pertinence (i.e. *relevance feedback*). Ces images pertinentes constituent de nouvelles requêtes. Cependant, pour limiter le nombre d'images sélectionnées à chaque étape, un clustering est effectué sur l'ensemble des images pertinentes, puis une méthode de fusion de clusters permet de réduire le nombre de clusters. Les images proposées à l'utilisateur sont alors les représentants de ces clusters. Une comparaison avec le système MARS [Ort97] est faite et montre une supériorité de l'approche proposée, c'est-à-dire +33 % de précision par rapport à MARS et +34 % en rappel.

3.4.2.3 Le clustering face aux problèmes de type-2

Une hypothèse *implicite* souvent faite lorsque le clustering non supervisé est utilisé pour la recherche d'images est que les groupements obtenus *correspondent* dans une certaine mesure aux groupements qui seraient obtenus de manière supervisée. Par exemple, dans [Che03], cette hypothèse est rendue explicite : les auteurs considère que *les images de sémantique similaire tendent à être groupées dans les mêmes clusters*. Le problème de la vision par ordinateur étant, à notre connaissance, toujours ouvert, cette hypothèse ne doit pas être totalement juste.

En fait, la véracité de cette hypothèse est proportionnelle au « degré de type-1 » du problème. Si le problème est en grande partie de type-1, i.e. les données sont corrélées statistiquement et directement aux concepts cibles, alors, étant donnés une mesure de similarité et des paramètres bien choisis, les clusters obtenus seront similaires structurellement aux clusters « attendus ». Cela peut-être le cas par exemple si l'on clustérise des images de ciel bleu et de végétation verte : il existe un lien statistique direct entre *bleu et ciel* d'une part et *vert et végétation* d'autre part, qui donnera lieu à des clusters « propres », conformes aux attentes.

Lorsque le problème est de type-2, c'est-à-dire que les informations pertinentes sont situées dans les relations entre attributs, plutôt qu'entre exemples dans l'espace, il n'y a guère de chances pour que les clusters obtenus aient la sémantique attendue. Par exemple, s'il l'on clustérise des mélodies, représentées sous forme de vecteurs de notes, personne ne peut raisonnablement s'attendre à ce que les clusters formés correspondent, par exemples, aux différents style musicaux, ou que chaque cluster contiennent les compositions d'un auteur particulier ! Le clustering considère des relations simples *entre observations,* mais lorsque les données sont dites relationnelles, les relations sont *entre attributs*.

3.4.3 Les réseaux de neurones

3.4.3.1 La méthode

Ici nous allons quitter le « mode » symbolique, mode par lequel on considère que l'intelligence se réalise par des opérations sur des structures symboliques, pour présenter des approches neurologiques ou biologiques de l'apprentissage. Ces approches, aussi connues sous le nom de systèmes parallèles distribués (SPD), préconisent une absence de symboles et voient l'intelligence comme un comportement collectif d'un grand nombre de composantes simples qui interagissent entre elles. Le cerveau et le système nerveux sont des paradigmes de cette « architecture ». Ils sont composés d'un grand nombre de cellules nerveuses ou neurones. Isolé, un neurone seul n'a pas de fonction par lui-même.

Les réseaux de neurones, une famille d'architectures informatiques inspirées des cerveaux biologiques, sont un exemple typique de SPD. Il existe plusieurs types de réseaux de neurones. Les **réseaux à une couche** sont parmi les plus simples mais sont limités, ils ne peuvent représenter que des classes d'éléments séparables linéairement. Les **réseaux de neurones multicouches** par contre sont des machines de calcul complètes (ils peuvent potentiellement apprendre n'importe quelle fonction calculable). Les **réseaux de neurones récurrents** ont la particularité d'avoir des connexions orientées dans les deux sens (entrées$\rightarrow$ sorties et sorties $\rightarrow$ entrées) et peuvent donc contenir des cycles. Ces réseaux sont capables d'assimiler des structures temporelles en plus des structures spatiales, ils ont de la mémoire. Le type de réseau récurrent le plus connu est sans doute le réseau de Elman ([ELM90]).

La principale motivation pour les réseaux de neurones (et plus généralement les SPD) vient des problèmes rencontrés avec les approches symboliques. La plupart des systèmes symboliques souffrent de fragilité. Cette fragilité vient en partie de la nature binaire d'une logique à deux valeurs. Chez les humains, par exemple, la performance à résoudre un problème se dégrade au fur et à mesure que la difficulté augmente. Un système expert, en comparaison, fonctionnera parfaitement jusqu'à ne donner aucune réponse pour un cas trop

difficile, non traité. L'humain, lui, proposera une réponse raisonnable même à des problèmes très complexes. C'est cette robustesse qui est recherchée dans les SPD.

Un réseau de neurones est entraîné, plutôt que programmé comme un système expert. Les poids sur les arcs s'ajustent en fonction de l'expérience à classer un ensemble d'exemples d'apprentissage. L'apprentissage se fait en observant l'erreur du réseau en sortie. Pour les unités de sortie, l'erreur se calcule simplement comme la différence entre les valeurs calculées et les valeurs correctes. Il est toutefois plus difficile de déterminer l'erreur commise par un neurone interne et d'évaluer sa contribution sur l'erreur totale. L'algorithme d'apprentissage le plus utilisé est l'algorithme de propagation arrière. Cet algorithme partitionne la responsabilité de l'erreur sur un réseau multicouche. Les neurones dans un réseau de propagation arrière sont connectés en couches dont les unités de la couche de niveau *k* passent leurs valeurs uniquement aux unités de la couche de niveau *k*+1. Pour solutionner un problème, l'activation se passe des unités d'entrée à travers une ou plusieurs couches internes, appelées les unités cachées, jusqu'à la dernière couche puis à l'environnement. Le réseau peut calculer l'erreur des unités de sortie exactement comme pour un réseau simple à une seule couche. L'erreur d'un neurone au niveau *n* est une fonction des erreurs de tous les neurones au niveau *n*+1 qui utilisent sa sortie (dont la valeur est influencée par sa sortie).

L'utilisation des systèmes parallèles distribués est très courante dans le domaine de la recherche d'image. Ces systèmes sont en effet bien adaptés à ce domaine où le bruit est très présent et où la complexité des problèmes rencontrés empêche la construction explicite de modèles.

3.4.3.2 Applications à la recherche d'images

Dans [Tow00] par exemple, les auteurs utilisent des réseaux de neurones pour apprendre à classifier des régions d'images en catégories sémantiques (comme "Herbe", "Sable" ou "Ciel"). Chacune de ces catégories (au nombre de 11) est apprise par un réseau de neurones spécifique, à partir d'exemples étiquetés fournis par une segmentation automatique d'images. Les taux de classification

correcte sont entre 86 % et 96 % avec une moyenne de 92 %. L'apprentissage requis pour obtenir ces résultats est conséquent : 40 000 images segmentées et annotées manuellement (le nombre de régions reste inconnu). Il est donc hors de question ici de proposer un apprentissage interactif dans lequel l'utilisateur fournirait lui-même des exemples d'apprentissage. Les auteurs précisent, de plus, que les résultats obtenus à partir d'images personnelles donnaient de moins bons résultats (sans préciser à quel point).

Dans [Ike00], les auteurs décrivent une approche différente pour l'utilisation des réseaux neuronaux pour la recherche d'images. Ceux-ci sont utilisés pour associer l'esquisse d'un utilisateur (recherche par croquis) aux images pertinentes de la base. Lors de l'apprentissage, un réseau de neurones multicouche est entraîné avec les images de la base, dont la résolution a été fortement réduite (18 par 12 pixels). Lors de cet apprentissage (par propagation arrière), les croquis sont convertis au même format que les images et leurs pixels sont utilisés en entrée du réseau de neurone, la sortie est le numéro de l'image correspondant au croquis. Lors de la reconnaissance, un croquis est présenté au réseau et sa ressemblance avec une ou plusieurs images de la base se traduit par l'activation d'un ou plusieurs neurones de la couche de sortie. Les auteurs rapportent une précision d'environ 55 % pour l'image correspondant au neurone de sortie le plus activé, avec une durée d'apprentissage de quatre heures pour 100 images.

Les méthodes parallèles distribuées sont omniprésentes dans de nombreux domaines de l'informatique et en particulier en recherche d'image. Leur efficacité n'est plus à démontrer. Toutefois, ces techniques sont plus adaptées à un apprentissage de type « off-line » qu'à un apprentissage interactif. Cela est dû au fait que leur réactivité est faible et qu'un grand ensemble d'apprentissage est généralement nécessaire pour apprendre un concept.

3.4.3.3 Les réseaux de neurones faces aux problèmes de type-2

Les réseaux de neurone à couches cachées permettent un recodage des données, quoique limité ([Tho00]), dans la mesure où l'ensemble des connexions du réseau s'adapte progressivement aux régularités présentes

dans l'ensemble d'apprentissage. Le problème du recodage *ad hoc* des données, nécessaire à la résolution des problèmes de type-2, n'est cependant pas résolu, il y a en effet deux bémols. Le premier est que la structure du réseau, le nombre de couches, de nœuds, sont fixés et arbitraires. Par conséquent, le nombre de recodages possibles est également fixé, et faible. De plus, la densité des supports de l'information dans un réseau de neurones est uniforme, c'est-à-dire que des entrées non pertinentes sont supportées, à la base, par autant de nœuds et de connexions vers les couches cachées que des entrées très pertinentes. Le second bémol est que le recodage n'est pas « libre », il est contraint par la descente de gradient. Par conséquent, tout recodage des données découvert par le réseau sera consistant, à tous les stades de sa construction, avec les concepts cibles.

Cette consistance monotone limite énormément la taille de l'espace des recodages exploré. Ce problème est inhérent à toutes les approches supervisées et nous suspectons que c'est là une des raisons d'une certaine « stagnation » dans la performance des algorithmes. Parallèlement, cette stagnation est partiellement atténuée, ou plutôt cachée, par la conception de traits de bas niveau toujours plus performants et adaptés à des problèmes spécifiques.

3.4.4 Algorithmes et programmation génétique

3.4.4.1 La méthode

Les algorithmes génétiques (AG) sont des algorithmes inspirés des mécanismes de la sélection naturelle et de la génétique. Ils utilisent à la fois les principes de la survie des individus les mieux adaptés et ceux de la propagation du patrimoine génétique.

De façon très intuitive, on identifie le problème à un environnement donné et les solutions à des individus évoluant dans cet environnement. A chaque génération, on ne retient que les individus les mieux adaptés à cet environnement. Au bout d'un certain nombre de générations, les individus

restants sont normalement adaptés à l'environnement donné. On obtient donc dans ce cas des solutions très proches de la solution idéale du problème.

Les algorithmes génétiques constituent une excellente méthode pour trouver rapidement un maximum global ou une valeur minimale de manière approximative. La diversité de la population initiale d'individus, ainsi que leur croisement, permet l'exploration « parallèle » de nombreuses zones de l'espace des hypothèses. Les mutations permettent quant à elles le raffinement des solutions potentielles.

Une des difficultés de ce type d'algorithme est le fait qu'il faille convertir un problème de sorte qu'il tienne dans une séquence de bits. Cela peut être trivial dans des problèmes « simples » de recherche de maximum globaux mais impossible pour d'autres. Beaucoup d'imagination et de créativité peuvent être requises, ainsi que de l'expérience.

Un autre problème crucial est de déterminer une « bonne » fonction d'évaluation. Dans le cas idéal, si la solution est optimale au sens de la fonction d'optimisation, alors elle l'est aussi pour le problème lui-même.

Un problème pratique de cette catégorie de problèmes est de choisir convenablement la valeur des différents paramètres. Ceux-ci sont en effet nombreux : taux de mutation, taux de croisement ou encore de reproduction. Le nombre d'individus de la population dépend quant à lui de la taille et de la forme de l'espace de solutions. Le nombre de gènes dépend de la manière dont on souhaite coder le problème. Un critère d'arrêt est également à déterminer, il a généralement trait à la stabilisation de la population.

3.4.4.2 Application à la recherche d'images.

Les travaux décrits dans [Emr04] proposent un système de recherche d'images médicales. L'accent est mis sur la pertinence des résultats par rapport à la perception humaine (des experts) de la similarité entre images. Pour générer les données d'apprentissage, un ensemble de personnes évalue la similarité entre chaque paire d'images de la base (48 images) sur une échelle de quatre degrés de similarité. Le résultat de cette évaluation est une matrice de

similarité. Un algorithme génétique est alors utilisé pour optimiser les paramètre d'une fonction de mesure de similarité (basée sur la distance de Manhattan, ou City-Block). La fonction d'évaluation de l'algorithme génétique est basée sur la distance entre la matrice de similarité utilisateur et la matrice obtenue en utilisant la fonction de similarité à évaluer. Initialement, la corrélation entre les deux matrices est 0.56, après optimisation elle est de 0.73.

Ces travaux sont intéressants, dans la mesure où chercher à paramétrer une fonction de mesure de similarité de manière à minimiser la distance entre instances d'un même concept, tout en maximisant la distance entre instances de concepts différents, est en fait une des formes que peut prendre l'apprentissage de concepts. Une fonction définissant un concept *est* une fonction de similarité : concept et similarité sont deux mots qui expriment la même chose, d'un point de vue différent. Cependant, les auteurs ne comparent pas leur approche avec d'autres algorithmes. De plus, le nombre de dimensions utilisées pour représenter les images est faible (vingt, réduit à huit par Analyse en composantes principales).

Il est également possible d'utiliser l'approche génétique non pas pour optimiser les paramètres d'une fonction mais pour construire un programme. Dans ce cas, une représentation arborescente des solutions potentielles (individus) est utilisée, plutôt qu'une représentation « à plat », comme un tableau. Johnson dans [JOH94] utilise la programmation génétique pour résoudre un problème de localisation de points dans une image. Le but du système est de localiser la main droite et la main gauche d'une silhouette. Pour cela, ils disposent d'un certain nombre de primitives d'extraction d'information de l'image. Ces primitives forment les briques de base qu'il faut ensuite assembler pour former une « routine visuelle » capable de réaliser la tâche voulue. Voici un exemple de routine visuelle :

```
(RIGHTMOST-POINT
        (FIND-TOP-EDGE
    (POINT-BETWEEN
      ...
```

Chaque individu de la population est une routine visuelle (initialement construite au hasard). L'algorithme génétique crée alors de nouveaux individus, c'est-à-dire de nouveaux programmes et sélectionne les meilleurs. Il y a stabilisation au bout d'une cinquantaine de générations : les programmes ainsi obtenus sont capables de traiter correctement à peu près 90 % des nouvelles images qu'on leur présente, ce qui est mieux que le meilleur des programmes conçu manuellement par les auteurs.

3.4.4.3 L'approche génétique face aux problèmes de type-2

L'approche génétique propose un compromis entre la taille de l'espace des hypothèses exploré et la vitesse de convergence. L'idée d'explorer de manière parallèle plusieurs zones de l'espace des hypothèses (idée connue sous le nom de *Beam Search*), permet d'éviter les « pires minima locaux », et garantit donc généralement (selon la taille de la population) une solution acceptable. Toutefois, cette approche n'autorise pas de recodage des données comme c'est le cas, dans une certaine mesure, pour les réseaux de neurones ou les SVM (machines à vecteurs de support). Par conséquent, si le problème d'apprentissage est principalement « relationnel », comme le problème de la parité, cette approche n'est pas plus intéressante qu'une autre.

De plus, si l'on s'écarte temporairement de l'analogie entre l'algorithme et le phénomène naturel, on voit qu'un algorithme génétique n'est autre qu'une descente de gradient, comme tous les algorithmes d'apprentissage supervisé. La différence principale est qu'un algorithme génétique préfère explorer « bêtement » plusieurs régions de l'espace (en modifiant aléatoirement plusieurs hypothèses parallèles) plutôt que de modifier « intelligemment » une seule hypothèse (en utilisant une heuristique), comme le font par exemple les réseaux de neurones. Cette stratégie peut être avantageuse lorsque aucune connaissance *a priori* n'est disponible pour choisir la manière dont on modifie l'hypothèse courante. Cependant, si le nombre d'individus dans une population, c'est-à-dire le nombre de solutions potentielles explorées parallèlement, est limité par des contraintes calculatoires, la taille de l'espace des solutions augmente exponentiellement avec le nombre de dimensions considérées. Par

conséquent, l'intérêt de considérer parallèlement quelques solutions potentielles plutôt qu'une seule, s'amenuise avec l'augmentation des dimensions, jusqu'à disparaître.

3.4.5 Les machines à vecteurs de support

3.4.5.1 La méthode

Développées au début des années 90 par Vapnik, les machines à vecteur support (SVM) et plus généralement les méthodes à noyau ont depuis connu un développement rapide. Ils constituent actuellement un axe de recherche important et actif dans les vastes domaines de l'apprentissage des machines, les réseaux de neurones, l'optimisation et les statistiques. Les SVM sont la technique d'apprentissage la plus utilisée aujourd'hui.

Les données d'apprentissage, des vecteurs de réels étiquetés (positifs ou négatifs), sont représentés comme des vecteurs dans l'espace. Le but de l'algorithme SVM est essentiellement de découvrir, par optimisation itérative, un hyperplan séparateur (voir Figure 3-5) tels que les vecteurs positifs soient d'un *côté* du plan, et les vecteurs étiquetés négatifs de l'autre. Le plan et la marge sont définis en fonctions des vecteurs de support, c'est-à-dire les vecteurs qui définissent la marge. Une fois le modèle appris, seuls ces vecteurs sont utiles.

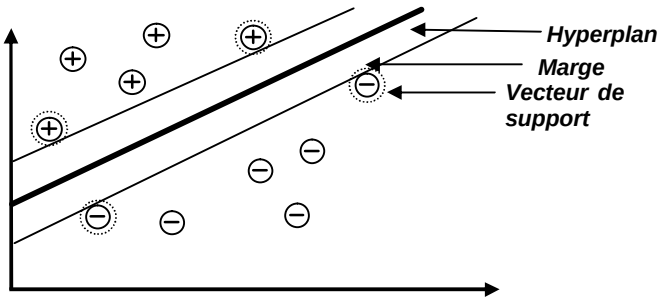


Figure 3-5 : SVM recherche un hyperplan qui maximise la marge. Ce plan est défini par les vecteurs de support.

Bien souvent, cependant, il n'existe pas de plan ayant la faculté de diviser les vecteurs de cette manière, du moins dans l'espace initial. La véritable force des SVM repose dans la technique utilisée pour résoudre ce problème : les

données sont projetées dans un espace de dimension supérieure (généralement très supérieure) où il existe un hyperplan séparateur et cette projection est effectuée *sans calcul explicite de la transformation de l'espace initial vers l'espace cible*. Cette technique est rendue possible par l'utilisation du « *kernel trick* » ([Aiz64]).

Le « kernel trick » repose sur l'utilisation de fonctions dites « noyau » ([Mül01]). En utilisant une fonction noyau, il est possible de transformer tout algorithme basé sur le calcul de produits scalaires entre couples de vecteurs : là où un produit scalaire est utilisé, on le remplace par une fonction noyau, et un algorithme linéaire peut être transformé en algorithme non linéaire. Puisque une fonction noyau est utilisée, il n'est pas nécessaire de calculer la transformation d'un vecteur d'un espace à l'autre, ce qui est souhaitable car il arrive que l'espace cible soit de dimension infinie (c'est le cas lorsque le noyau est dit Gaussien). D'autres algorithmes utilisent le *kernel trick,* comme l'analyse en composantes principales.

3.4.5.2 Adaptation aux problèmes de type-2

Etant donnée la popularité actuelle et croissante des Machines à Vecteurs de Supports (SVM), on peut penser que cette technique d'apprentissage constitue l'aboutissement actuel du paradigme de *fence-and-fill-learner*. Ce terme, introduit par Thorton ([Tho00]), représente la catégorie d'algorithmes d'apprentissage qui représentent les données dans un espace, où chaque dimension correspond à un attribut, et qui cherchent à compartimenter (*fence*) les exemples de manière à séparer les positifs des négatifs. La classification est faite en affectant à une nouvelle observation la catégorie correspondant au compartiment de l'espace où elle se trouve (*fill*). La version la plus basique (mais très efficace) de ce paradigme étant l'algorithme k-NN.

SVM apporte plusieurs améliorations à ce paradigme. La première est la notion de marge. Non seulement SVM découvre un hyperplan capable de séparer les exemples positifs des exemples négatifs, mais en plus, cet hyperplan maximise la marge entre le plan et les exemples. Cette maximisation a pour conséquence bénéfique une bonne capacité de généralisation. En effet, intuitivement, un

hyperplan « collé » aux exemples positifs aurait pour conséquence la classification « négative » de certains exemples, même s'ils sont très proches d'autres exemples positifs.

Une autre amélioration repose sur l'utilisation de fonctions dites « noyau ». Si les exemples ne sont pas séparables linéairement, c'est-à-dire qu'un hyperplan séparateur n'existe pas dans l'espace d'origine, une SVM permet le « recodage » des données dans des espaces différents et éventuellement de dimensions supérieures, où il *existe* un hyperplan séparateur. Nous avons vu qu'un recodage des données est indispensable pour la résolution des problèmes de type-2. Toutefois, parmi l'infinité des recodages possibles, SVM n'en propose que quelques-uns, comme : des noyaux gaussiens, polynomiaux ou RBF[22]. L'utilisateur a donc pour tâche de trouver, en fonction des paramètres du problème, le noyau le plus adapté.

3.4.5.3 Applications à la recherche d'images

Dans [Ton01], une approche de classification des images par concepts est proposée. Une première particularité de cette approche est que l'apprentissage est actif. Pendant l'apprentissage d'un concept, l'algorithme peut demander à l'utilisateur l'étiquette d'une image. Cette image n'est pas choisie au hasard mais selon la nature et la complexité du concept à apprendre. Pour caractériser cette complexité, les auteurs utilisent trois critères de mesure : la *rareté* qui caractérise la difficulté de l'algorithme à trouver des images similaires à l'exemple initial fournit par l'utilisateur pour initier l'apprentissage du concept. L'*isolation* caractérise la distance entre les images du concept courant et les autres images. Lorsque l'isolation est faible, la frontière entre exemples positifs et négatifs est difficile à trouver et le système cherche à éclaircir cette zone. Finalement le critère de *diversité* permet à l'algorithme de savoir si le concept est concentré dans une région de l'espace des traits ou plutôt éclaté. S'il est éclaté, l'algorithme privilégie une recherche large de l'espace. Les deux premiers critères servent à choisir les exemples, le dernier permet de décider le nombre d'exemples dont l'étiquette sera demandée à l'utilisateur. L'ensemble

[22] Radial Basis Function

des images étiquetées collectées est utilisé par un classificateur SVM pour décider de la pertinence des images par rapport à la requête. Les k images présentées à l'utilisateur comme résultat sont celles qui sont le plus éloignées de l'hyperplan séparateur, *côté* positif.

Dans [Jin03], les images sont d'abord segmentées. Puis un ensemble de traits est extrait de chacune d'entre elles. Plusieurs approches sont alors comparées pour calculer le résultat d'une requête avec bouclage de pertinence. La première consiste à mesurer la distance entre une image requête et les images de la base en utilisant la distance EMD[23] (Earth Mover Distance). La seconde méthode consiste à créer une image virtuelle, constituée des régions les plus pertinentes des images exemples. Le choix de ces régions et leurs pondérations respectives est déterminé par clustering : les régions similaires sont alors représentées par un élément du cluster, ce qui permet de tenir compte de toutes les régions dans une seule image virtuelle. La troisième méthode réside dans l'utilisation d'un SVM. Les exemples positifs et négatifs acquis par bouclage de pertinence sont utilisés pour découvrir un hyperplan séparateur. Les images retournées à l'utilisateur sont celles qui se situent le plus loin possible de cet hyperplan, du *côté* des exemples positifs.

3.4.6 Récapitulatif

Toutes les méthodes dont nous venons de parler sont valables et efficaces, pour des tâches particulières, chacune ayant des caractéristiques propres. Récapitulons maintenant les techniques évoquées ci-dessus, à la lumière des contraintes imposées par nos objectifs. Le Tableau 3-2 résume les caractéristiques des quelques méthodes d'apprentissages que nous avons abordées.

Etant donnée la nature hautement interactive de notre approche, une caractéristique indispensable de l'algorithme élu doit être une forte réactivité. Il doit être capable d'apprendre un concept rapidement, sans que l'utilisateur n'ait

[23] La distance EMD entre deux histogrammes est le *travail* minimal les rendre identiques en *transportant* le contenu des colonnes qui diffèrent d'un histogramme à l'autre. Le *travail* est proportionnel à la valeur des colonnes et à la distance à parcourir pour *transporter* cette valeur.

à fournir une multitude d'exemples et de contre exemples. Cela nous pousse à écarter les *réseaux de neurones*, connus pour nécessiter de grands ensembles d'apprentissage.

	Supervisé	Réactivité	Stabilité	Résistance au bruit	Adapté type-1 ?	Adapté type-2 ?	Adaptation à notre contexte
Algorithmes Génétiques	Oui	**	**	****	Oui	Non	*
Clustering	Non	****	****	****	-	-	**
Réseaux de neurones	Oui	*	***	*****	Oui	Très peu	**
SVM	Oui	****	****	****	Oui	Très peu	***
k-NN	Oui	*****	****	***	Oui	Non	***
ART Networks	Oui/Non	****	*****	****	Oui	Non	***
Arbres de décision	Oui	***	***	***	Oui	Non	**

Tableau 3-2 : Tableau comparatif de quelques techniques d'apprentissage

Nous écartons également les *algorithmes génétiques*, d'une part car notre problème d'apprentissage ne se modélise pas aisément sous la forme d'un algorithme génétique. En effet, le caractère évolutif et incrémental de notre contexte (ajout de concepts, en particulier), requièrent une grande souplesse de l'algorithme d'apprentissage utilisé. Les algorithmes génétiques nécessitant dans la pratique beaucoup de « réglages » manuels des différents paramètres, ils n'offrent pas la souplesse requise. D'autre part, leur réactivité et stabilité sont trop faibles pour être adaptées à un système centré sur l'utilisateur.

Le *Clustering* serait un bon candidat, en termes de réactivité, stabilité, résistance au bruit. Le problème vient du passage au supervisé (le clustering est, à la base, non supervisé). Si les clusters formés ne correspondent pas aux concepts cibles, l'apprentissage sera un échec. Puisque les concepts sont en partie définis par l'utilisateur, il est improbable que les clusters obtenus soient adaptés. Le même problème se pose pour les *réseaux ART*, qui sont à la base une méthode d'apprentissage non supervisée, dont la transformation en méthode supervisée n'est pas immédiate.

Les approches que nous n'avons pas éliminées, à savoir les SVM, kNN et arbres de décisions sont des candidats potentiels. Cependant, un problème

subsiste pour ces trois méthode : elles ne sont pas adaptées aux problèmes de type-2. Par conséquent, faces à un problème d'apprentissage difficile (de type-2), elles sont, en dépit de leurs différences internes énormes, similaires en termes de résultats. Toutes les trois sont capables d'exploiter les corrélations statistiques directes entre traits et concepts présents dans l'ensemble d'apprentissage, mais ne peuvent aller au-delà. En terme de performance de classification, nous avons remarqué que, en moyenne, dans l'ensemble des travaux que nous avons étudiés, les SVM arrivent en tête, suivis des kNN et arbres de décisions. Cependant, un point intéressant est que *l'écart moyen* entre les performances des différents algorithmes est en moyenne faible.

3.5 Approches relationnelles

Les approches que nous venons de décrire reposent souvent sur une hypothèse implicite d'indépendance des attributs, ou du moins sur l'hypothèse que les concepts cibles ne sont pas entièrement caractérisés par des relations entre les attributs. C'est cette hypothèse qui permet d'utiliser la technique de *descente de gradient*, qui peut prendre de nombreuses formes. Lors de la descente de gradient, certaines valeurs (absolues) de certains attributs sont identifiées comme ayant un lien de corrélation avec le ou les concepts cibles : on peut alors utiliser cette information pour améliorer l'hypothèse courante.

Que se passerait-il si au lieu d'être définis de manière *absolue*, les concepts étaient en fait caractérisés uniquement de manière *relative* ? Dans ce cas, il n'existerait pas de valeur absolue d'attribut permettant à elle seule d'apporter de l'information susceptible de progresser vers le concept cible. Dans ce cas, le concept ne serait alors pas défini par une *zone homogène, contiguë et précise de l'espace des hypothèses*, identifiable par un ensemble de coordonnées absolues. Apprendre à reconnaître des visages représentés par un ensemble de simples vecteurs de pixels est un problème de ce type : chaque pixel (donc chaque dimension) considéré indépendamment n'apporte aucune information discriminante. Par conséquent, les coordonnées absolues du point représentant un visage dans l'espace n'apportent pas d'information. L'information

discriminante est cachée dans les relations complexes qu'entretiennent ces pixels.

Pour répondre à ces problèmes, la communauté d'apprentissage automatique s'est intéressée aux approches d'apprentissage dites relationnelles. Ces approches, bien que très minoritaires par rapports aux approches classiques, sont en pleine expansion. Nous voyons deux raisons principales au fait que l'apprentissage relationnel ait longtemps été relégué au second plan. La première est que l'apprentissage relationnel implique l'utilisation de représentations des hypothèses beaucoup plus riches, ce qui fait littéralement exploser la taille de cet espace. La recherche de concepts dans cet espace est naturellement plus difficile. La deuxième explication est qu'il existe une alternative à l'apprentissage relationnel : au lieu d'apprendre les relation à partir des données, celles-ci sont introduites manuellement sous la forme de traits de bas niveau *ad hoc*, c'est-à-dire adaptés au domaine étudié.

Nous allons maintenant présenter quelques approches représentatives de ce domaine récent d'investigation.

3.5.1 Apprentissage de CNF

L'auteur introduit ce papier ([Moo95]) par une constatation : la majorité des approches en Induction Symbolique de Concepts (apprentissage par l'exemple où les attributs sont de type symbolique, souvent des booléens) utilisent des représentations sous forme d'arbres de décision. L'auteur note en particulier que très peu de recherche a été effectuée sur des algorithmes d'apprentissage de concepts représentés par des formes normales conjonctives, un formalisme pourtant riche, en particulier pour l'expression des relations mais ayant la réputation de ne pas être une représentation « naturelle » des concepts.

Les formes normales conjonctives (CNF) représentent les concepts sous la forme de conjonctions de disjonctions ($F_1 \wedge F_2 \wedge ... \wedge F_j$ où chaque F_i est une disjonction de littéraux). Une forme normale beaucoup plus courante est la forme normale disjonctive (DNF) qui est à l'inverse une disjonction de conjonctions ($F_1 \vee F_2 \vee ... \vee F_k$ où chaque F_i est une conjonction de littéraux).

Dans ces travaux, l'auteur compare l'apprentissage, sur des données naturelles, de concepts représentés sous les trois formes suivantes : arbres de décisions, DNF et CNF.

L'algorithme utilisé pour l'apprentissage de concepts sous la forme de CNF/DNF est appelé FOIL. Cet algorithme supervisé raffine itérativement une hypothèse, en la spécialisant de manière à ce qu'elle couvre un maximum d'exemples positifs sans couvrir d'exemple négatif. L'algorithme utilisé pour apprendre les concepts sous la forme d'un arbre de décision est l'algorithme ID3.

Ces trois algorithmes sont alors confrontés sur cinq jeux de données naturelles (comme des séquences de nucléotides, ou des données médicales). Les mesures effectuées portent à la fois sur le taux d'erreur des algorithmes de classification obtenus, ainsi que sur la taille des hypothèses construites. L'auteur souligne avoir été surpris par les résultats : la représentation sous forme CNF produit systématiquement des résultats meilleurs ou équivalent aux autres représentations. De plus, les hypothèses construites sous forme CNF sont quasiment toujours plus compactes que les autres.

En conclusion, l'auteur indique qu'il n'existe pas de travaux consacrés à l'élaborations d'algorithmes ayant la forme normale conjonctive comme moyen de représentation, et espère que les résultats reportés dans [Moo95] stimuleront la recherche sur cette voie. Une décennie plus tard, il ne nous semble pas, à notre connaissance, que son souhait ait été exaucé.

3.5.2 Many-Layered Learning

Pour Paul Utgoff ([Utg02]), l'apprentissage est un élément prépondérant de tout comportement intelligent. Toutefois, une connaissance particulière ne peut être apprise à n'importe quel moment, car il existe un ordre, une hiérarchie dans l'apprentissage. Les connaissances « prêtes » à être acquises sont dites *à la frontière de la réceptivité*. La question posée dans ces travaux est la suivante : comment un agent apprenant peut-il organiser en connaissances hiérarchiques un flux continu de données non structurées.

Pour tenter de répondre à cette question, l'auteur présente un algorithme d'apprentissage basé sur la notion de « couches de connaissances ». Son intuition est que l'empilement de couches capables uniquement d'apprentissage de type-1, permet l'apprentissage de type-2. Ces couches, empilées, sont telles qu'une couche peut utiliser les représentations apprises dans les couches inférieures, en les combinant de manière relationnelle, permettant ainsi le partage et la réutilisation des connaissances. L'algorithme d'apprentissage proposé, nommé STL (Stream-to-Layers), opère de la manière suivante. Un ensemble de buts (des fonctions) intermédiaires est créé (manuellement). Ces buts sont définis par une fonction d'évaluation mais doivent être appris. Lorsqu'un exemple est présenté à l'algorithme, celui-ci essaye d'apprendre parallèlement tous les buts. Cependant, les buts n'étant pas de difficulté identique, certains sont appris plus tôt que d'autres. Lorsqu'un but est atteint, c'est-à-dire qu'une fonction a été apprise, elle devient une entrée (une dimension supplémentaire) pour tous les autres buts à apprendre. Ce mécanisme permet ainsi un recodage progressif des données, faisant ainsi reculer progressivement la frontière de réceptivité.

L'algorithme STL est évalué dans le contexte d'un jeu de cartes, plus précisément, il s'agit d'apprendre à empiler des cartes selon certaines contraintes liées aux couleurs et aux figures. Les résultats montrent que l'algorithme est capable d'apprendre avec succès à empiler les cartes en respectant les règles du jeu. L'auteur compare parallèlement STL à divers réseaux de neurones, et montre que ces derniers ne sont pas capables d'apprendre ces règles avec succès.

Cette approche, bien qu'intéressante, présente un problème de taille : les fonctions intermédiaires nécessaires à l'apprentissage de concepts de haut niveau, sont définies au préalable, ce qui constitue plus que des « indices » : l'espace des hypothèses potentielles s'en trouve extrêmement réduit, et son exploration facilitée.

3.5.3 Skewing Theory

Ces travaux récents ([Ros05]) partent de la constatation que certaines catégories de fonctions booléennes ne sont pas apprenables par des arbres de décision standards. Ces fonctions difficiles incluent par exemple le problème de la parité, où la fonction est vraie si le vecteur d'entrée contient un nombre impair de booléens *vrais*. La raison à cette incapacité est que les attributs considérés séparément n'apportent pas d'information, or, un arbre de décision inclut les attributs un par un dans la construction de son hypothèse. Par conséquent, les arbres de décision *classiques* ne sont pas adaptés aux concepts définis de manière relationnelle. Ces travaux présentent une extension aux arbres de décision dont le but est de rendre possible l'apprentissage de concepts difficiles.

L'idée du *skewing* (biaiser) est de pré traiter les données de manière à faire émerger les variables discriminantes : les données originales sont alors biaisées avant d'être présentées à l'arbre de décision. Pour cela, une *préférence* est associée à chaque variable x (en l'occurrence, 1 ou 0), ainsi qu'un poids p compris entre ½ et 1. Chaque exemple d'apprentissage (chaque vecteur de booléens) est pondéré par p si la valeur de x correspond à sa *préférence*, mais pondéré pas 1-p s'il n'y a pas correspondance. Le poids final d'un exemple est le produit des poids associés à chaque variable, ce qui induit un biais dans la distribution initiale. Ce processus de pondération est répété plusieurs fois, avec des préférences différentes. Après pondération, le gain est calculé pour chaque variable, c'est-à-dire la quantité d'information qu'apporte cette variable pour décider si l'exemple est une instance du concept ou non. La variable exhibant le gain le plus élevé est choisie pour ajouter de nouvelles branches à l'arbre de décision.

Les auteurs soulignent que, bien que les expérimentations montrent clairement que cette technique permet un meilleur apprentissage des fonctions booléennes difficiles, la raison de ce succès n'est que très peu comprise. De plus, l'apprentissage ne semble pas être amélioré lorsque le nombre de variables dont est composé le concept cible est supérieur à 7.

3.5.4 Discussion

Les approches que nous venons d'évoquer sont autant de tentatives intéressantes de se mesurer au problème de l'apprentissage relationnel. Les problèmes d'apprentissage relationnels ont pour difficulté majeure la taille prohibitive de l'espace des hypothèses à considérer. Explorer cet espace nécessite donc un biais beaucoup plus fort que dans le cadre plus classique où l'hypothèse d'indépendance est présente. Dans [Utg02] par exemple, le biais utilisé est très fort puisque les représentations intermédiaires nécessaires à l'apprentissage sont directement suggérées à l'algorithme.

De plus, ces méthodes demeurent fondamentalement supervisées, ce qui les empêche, par construction, de considérer des représentations intermédiaires non directement utiles.

3.6 Conclusion

Nous avons vu que, dans la plupart des problèmes « d'intérêt », les données brutes, c'est-à-dire le résultat de la numérisation, ne présentent pas de liens statistiques directs avec les concepts abstraits que l'on voudrait faire apprendre à partir de ces données. Ces problèmes sont qualifiés de problèmes d'apprentissage de type-2.

Les algorithmes actuels ne sont pas conçus pour « affronter » des problèmes de type-2, ils ne peuvent résoudre que des problèmes de type-1 : on peut donc les qualifier *d'algorithmes d'apprentissage de type-1* (la Figure 3-6 illustre cette situation). Ils sont toutefois capables d'apprendre les corrélations superficielles entre données et concepts. Ces algorithmes diffèrent principalement dans leurs capacités de généralisation, la raison étant que leurs biais inductif sont différents. Par conséquent, en pratique, on sélectionne l'algorithme dont la manière de généraliser est la plus appropriée au problème. De plus, les algorithmes diffèrent dans leurs capacités de réactivité, de résistance au bruit et de stabilité.

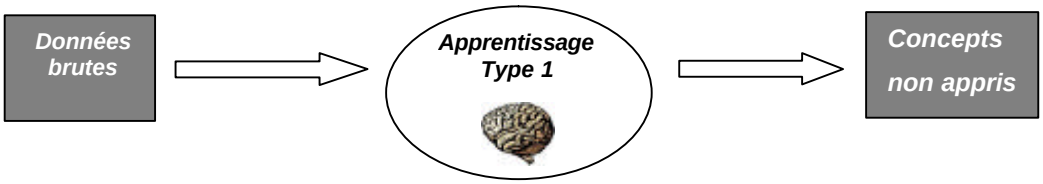


Figure 3-6 : L'apprentissage de type-1 ne peut pas apprendre de concepts à partir des données brutes.

Pour contourner ce problème, la recherche s'est intéressée à la *vision par ordinateur*, un domaine au carrefour de l'informatique, des mathématiques et de la psychologie cognitive, qui cherche (entre autres) à reproduire certaines caractéristiques de la perception humaine. Les traits permettant de caractériser et représenter les images proviennent principalement de ce domaine.

Lorsque les images ne sont plus représentées par des pixels mais par des traits qui capturent certaines « propriétés » de la perception humaine, le problème d'apprentissage est considérablement allégé. Dans ce cas, l'intelligence nécessaire à la résolution du problème d'apprentissage provient alors de deux sources distinctes (Figure 3-7). Le symbole de *gros* cerveau associé au recodage illustre notre point de vue selon lequel c'est cette phase, et non la suivante, qui nécessite le plus d'intelligence, et qui est la plus déterminante.

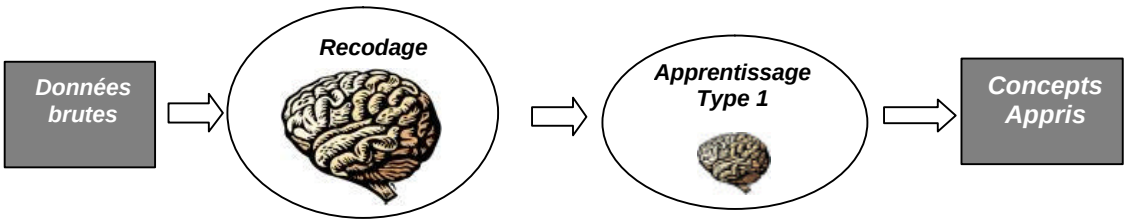


Figure 3-7 : L'apprentissage de concepts est parfois possible, lorsque les données sont représentées sous la forme de traits judicieusement conçus et sélectionnés par des humains.

La première source, en amont, est contenue dans les processus de traitement de l'image qui donnent lieu aux traits. Nous pensons que c'est dans cette première source que réside la plus grosse partie de l'intelligence nécessaire à la résolution du problème. La seconde source se situe dans le processus

d'apprentissage, en aval, et son but est de « finir le travail », c'est-à-dire d'associer les traits extraits de l'image, aux concepts que l'on souhaite apprendre. Nous pensons que cette source d'intelligence est très limitée et que le succès ou l'échec de l'apprentissage dépend principalement de la représentation de l'image calculée en amont.

Un algorithme d'apprentissage de type-2 devrait être capable d'accomplir simultanément la tâche de recodage et celle d'apprentissage, comme illustrée Figure 3-8.

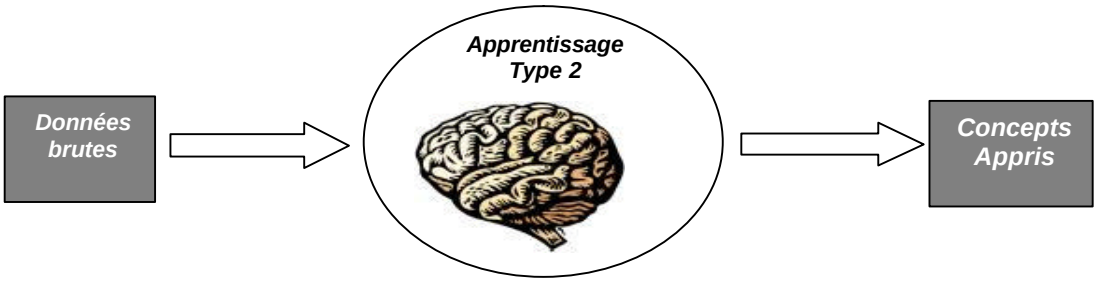


Figure 3-8 : Un algorithme d'apprentissage de type-2 inclut le recodage des données dans l'apprentissage.

Pour que l'apprentissage d'un recodage des données pertinent, sous forme de traits, soit possible automatiquement, il faut d'une part que les données soient suffisamment *riches* pour permettre cet apprentissage, et d'autre part que l'algorithme dispose d'un *biais inductif* lui permettant d'explorer l'espace des recodages de manière pertinente. Ce second point sera l'objet principal de la suite de notre exposé.

Deuxième partie

Proposition

Chapitre 4

Apprentissage constructif hiérarchique de régularités pour l'indexation

Ce chapitre introduit les motivations, les idées et concepts d'une nouvelle approche d'apprentissage automatique. Cette approche sera décrite plus concrètement au cours des deux chapitres suivants.

La notion d'apprentissage automatique constitue le cœur de notre travail. Ce n'est que par l'apprentissage qu'il est possible d'acquérir des concepts difficiles *via* des exemples, comme les concepts correspondant à des objets présents dans une collection d'images.

Nous rappelons dans la suite pourquoi l'apprentissage est un problème si difficile, en particulier dans notre contexte. Nous argumentons ensuite sur les limites de l'apprentissage *uniquement supervisé* avant de décrire notre approche de l'apprentissage. En s'inspirant d'exemples issus de l'apprentissage humain, nous proposons que l'apprentissage supervisé devrait laisser place au préalable à un apprentissage non supervisé dont le but est d'acquérir les 'briques de base' nécessaires à l'apprentissage de concepts.

Nous verrons que ces briques de base proviennent de régularités[24] ayant toutes comme source commune les lois physiques. Malheureusement, ces régularités sont souvent invisibles par un algorithme d'apprentissage supervisé parce qu'elles ne sont pas *immédiatement* utiles.

Ces concepts seront décrits plus formellement, nous verrons comment l'apprentissage de régularités s'apparente à l'apprentissage de formes normales disjonctives et conjonctives. Nous insisterons sur l'apprentissage de disjonctions,

[24] Ces régularités sont tellement omniprésentes qu'il est par exemple très difficile de générer de vrais nombres aléatoires.

quasi inexistant dans la littérature, qui nous mènera à la notion d'abstraction et qui nous fera aboutir à la notion de *similarité locale.*

Nous montrerons comment un apprentissage supervisé classique succède à l'apprentissage non supervisé, l'apprentissage supervisé ayant pour but de faire le lien entre les concepts cibles et les représentations apprises de manière non supervisée.

Finalement nous présenterons deux méthodes de classification, l'une *stricte* l'autre *souple* et discuterons du rapport étroit entre la classification souple et la logique floue.

4.1 Difficulté de l'indexation

Dans l'état de l'art et plus spécifiquement dans la partie dédiée à l'apprentissage, nous avons vu que le problème d'apprendre à indexer des images (à un niveau d'abstraction plus élevé que la simple extraction de traits) est difficile. L'apprentissage de concepts visuels n'est toutefois pas classifiable en problème de type-1 ou type-2 ([Cla97]). D'une part, de nombreux facteurs entrent en jeu, tels que le nombre de concepts indexés, la qualité des traits extraits, leur pouvoir d'expression ou les données utilisées. D'autre part, s'il est possible de montrer quelles sont les régularités de type-1 présentes dans un *corpus* d'apprentissage, il est en revanche impossible de savoir à l'avance (c'est-à-dire avant de les avoir trouvées) s'il existe des régularités de type-2. Par conséquent, un problème non solvable par l'exploitation unique des régularités de type-1 est « peut-être » de type-2 mais on ne peut le savoir[25].

Vu l'incapacité des systèmes actuels à apprendre des concepts visuels non triviaux (cf. état de l'art), nous faisons l'hypothèse que le problème d'apprendre des concepts visuels à partir de traits (dans le contexte que nous détaillerons) est un problème de type-2, c'est-à-dire que les régularités de type-1 ne suffiront pas pour l'apprentissage des concepts. Par conséquent, nous admettons que les relations existant entre traits et concepts sont en partie inaccessibles par des algorithmes d'apprentissage

[25] Notons que cette incertitude, bien que gênante, est inévitable dans la mesure où lorsque l'on peut déterminer de manière exacte la solvabilité d'un problème, on peut également le résoudre.

classiques. Nous allons montrer dans la suite que l'apprentissage uniquement supervisé ne peut être suffisant pour un problème de type-2 et que seul un apprentissage non supervisé préalable peut fournir les « briques de base » nécessaires à l'apprentissage de concepts complexes, c'est-à-dire caractérisés par des régularités non triviales de type-2.

4.2 Limitations de l'apprentissage supervisé

Parmi les nombreuses classifications que l'on peut appliquer aux algorithmes d'apprentissage, la dichotomie « supervisé / non supervisé » est sans doute la plus classique. En informatique, cette distinction est très forte : l'apprentissage non supervisé concerne la fouille de données et consiste généralement à grouper automatiquement des éléments selon une mesure de similarité (Clustering). L'apprentissage supervisé, quant à lui, est utilisé pour faire de la classification, c'est-à-dire associer des éléments à un ensemble de classes prédéfinies.

Nous argumentons maintenant sur les limites intrinsèques de l'apprentissage *uniquement* supervisé, et montrons que les approches dites *semi-supervisées* ne constituent pas une réponse à ces limites.

4.2.1 Apprentissage supervisé

Une particularité problématique de l'apprentissage supervisé, est que les algorithmes n'explorent qu'une infime sous partie de l'espace des hypothèses, car dans cette exploration, ils sont constamment guidés par leur but, c'est-à-dire un concept à apprendre. De ce fait, tout algorithme d'apprentissage incrémental, même hiérarchique, qui utilise une « vérité terrain » pour guider l'apprentissage peut être vu comme une descente de gradient (« hill climbing »).

L'hypothèse fondamentale de la descente de gradient est que les attributs qui sont écartés (ou dont l'influence est diminuée), car ils ne sont pas reliées directement (statistiquement) aux concepts cibles, ne l'auraient de toute manière jamais été. Pour paraphraser, on peut dire que cette hypothèse considère une forte indépendance des différentes variables ou attributs.

Pour visualiser intuitivement ce problème, on peut imaginer un « agent » désirant se rendre d'un lieu de départ D à un lieu d'arrivée A, le tout dans un environnement complexe. Un agent supervisé (par son but d'arriver à A), sans connaissances particulières, se dirigera tout droit vers A, et à moins de se trouver dans un lieu particulièrement désertique, se retrouvera rapidement bloqué par un obstacle quelconque. Un agent également supervisé, mais possédant des connaissances liées à son environnement, pourra se rendre en A, en évitant des obstacles et s'éloignant même parfois temporairement de son objectif. Même pendant ses détours, cet agent aura toutefois la certitude de se rapprocher à tout moment de son but. D'où viennent ces connaissances sur l'environnement et comment les acquérir est le centre du problème de l'apprentissage de type-2.

Dans les problèmes d'apprentissage de *type-2*, certains attributs peuvent sembler ne pas être corrélés aux concepts cibles alors qu'ils le sont mais d'une manière relationnelle. Une corrélation peut émerger dans un agrégat de variables de moins haut niveau, tout en étant invisible (statistiquement parlant) pour ces variables considérées isolément. Par conséquent, l'hypothèse implicitement faite lors de l'utilisation d'une des formes de la descente de gradient est généralement fausse et l'algorithme qui l'utilise converge vers un minimum (si l'on parle d'erreur de classification) local, en omettant les représentations pertinentes. Pour reprendre la métaphore précédente, résoudre un problème de type-2 (aller de D à A) requiert : soit d'explorer *au préalable* et sans but *précis* son environnement, soit d'être informé par une source extérieure sur la nature de cet environnement.

Par exemple, bon nombre de techniques d'apprentissage (SVM, k plus proches voisins, etc.) représentent les exemples d'apprentissage dans un espace et tentent de partitionner celui-ci de sorte à ce que les exemples positifs appartiennent à un même (ou plusieurs) « compartiment » dans lequel ne se trouve aucun exemple négatif. Tout exemple non étiqueté qui se trouve à l'intérieur du compartiment sera étiqueté comme appartenant au concept. L'hypothèse sous-jacente est que les points proches dans l'espace sont également proches conceptuellement. Cette hypothèse est généralement inexacte, en particulier lorsque les régularités en question n'existent pas entre les points de l'espace mais plutôt entre les attributs. Dans ce cas, un concept n'est pas représentable par une zone homogène de l'espace. Il est alors nécessaire de découvrir les relations pertinentes entre attributs

pour un concept donné. Il se peut que ces relations soient complexes, et que toute relation incomplète plus simple soit totalement décorrélée statistiquement du concept à apprendre.

Ainsi, nous pensons qu'utiliser trop tôt les étiquettes de l'ensemble d'apprentissage *pendant* l'apprentissage, annule la possibilité même de découvrir certaines régularités discriminantes. Nous pensons qu'un problème de *type-2* n'est solvable que si l'apprentissage est d'abord **non supervisé**. L'apprentissage non supervisé est *indifférent* vis-à-vis de tout concept et les régularités 'découvertes' ne sont pas nécessairement liées à un concept. Cependant, elles correspondent toute à des propriétés de l'ensemble d'apprentissage. L'apprentissage non supervisé n'est pas réellement sans but, il a pour objectif de construire des représentations pertinentes de son environnement, c'est-à-dire capables d'effectuer des prédictions. L'apprentissage non supervisé permet de transformer l'espace initial en un espace adapté aux données.

S'il est facile de critiquer l'apprentissage supervisé en général, il est aussi facile de comprendre pourquoi il demeure un paradigme omniprésent, sans rival sérieux. Nous avons vu que les régularités *relationnelles* qui caractérisent les problèmes de type-2, nécessitent un recodage systématique des données, afin d'être rendue statistiquement visibles (par les algorithmes d'apprentissage classiques). Or, l'ensemble des recodages que l'on peut appliquer aux données est l'ensemble des machines de Turing applicables, c'est-à-dire un ensemble *infini*. Cela signifie que sans contraintes additionnelles (apprentissage sans biais), l'espace des hypothèses à explorer est infini. L'apprentissage de type-2 nécessite donc l'introduction d'un biais fort.

4.2.2 Apprentissage semi supervisé

Les recherches s'intéressant à la fois à l'apprentissage supervisé et non supervisé sont quasi inexistantes. À notre connaissance, il n'existe pas de recherches dans lesquelles l'apprentissage supervisé et non supervisé sont réellement fusionnés, de sorte qu'ils s'appuient sur les mêmes structures sous-jacentes. Certes, il existe des approches qualifiées de « **semi supervisées** » qui pourraient faire penser que les deux formes d'apprentissage cohabitent et se complètent. En réalité, le terme « semi

supervisé » désigne les approches dans lesquelles une partie de l'ensemble d'apprentissage n'est pas étiquetée, ce qui a pour effet d'autoriser une marge de manœuvre (ou d'erreur) à l'algorithme d'apprentissage supervisé, et d'éviter ainsi le sur-apprentissage (overfitting[26]). Par exemple, dans [Kem04], les exemples non étiquetés sont utilisés pour inférer une structure (clusters hiérarchiques). L'algorithme supervisé modifie alors cette structure afin qu'elle classifie au mieux les exemples étiquetés. Ces deux phases sont bien distinctes et ne sont pas en interaction : la première travaille sur les exemples dans leur globalité (tous traits compris) alors que la seconde considère chaque trait indépendamment (en faisant en plus l'hypothèse que ceux-ci ne sont pas corrélés).

Parfois, l'apprentissage semi supervisé consiste à utiliser les données étiquetées pour un but, et les données non étiquetées pour un autre. Dans [Li04b] par exemple, les données étiquetées sont utilisées pour construire deux classificateurs distincts (correspondant à deux ensembles de traits de bas niveau très différents), puis les données non étiquetées sont classifiées par ces deux classificateurs et les exemples sur lesquels les classificateurs « sont d'accord » sont utilisés pour reconstruire les classificateurs.

Il est également très courant d'utiliser les données étiquetées pour mesurer la qualité d'une mesure de similarité utilisée pour le clustering des données non étiquetés, comme c'est le cas dans [Gri05].

Pour résumer, lorsque apprentissage supervisé et non supervisé sont utilisés au sein d'un même système, ils le sont séparément, pour des tâches différentes. Cela ne constitue donc pas une solution au problème du recodage des données.

4.2.3 Inspirations biologiques

Nous avons donc d'un *côté* des algorithmes non supervisés, qui cherchent des groupements dans de grandes masses de données, et d'un autre *côté*, des

[26] Il y a sur-apprentissage lorsque l'algorithme est capable de classifier les exemples d'apprentissage mais généralise mal à de nouveaux exemples en raison d'un apprentissage trop précis (presque par cœur), qui n'autorise aucune souplesse.

algorithmes supervisés, qui tentent *directement*[27] de lier leurs entrées (un ensemble de traits) à leurs sorties (un ensemble de concepts).

En apprentissage « biologique », c'est-à-dire l'apprentissage ayant le cerveau comme support, les choses ne sont bien sûr pas aussi simples. Premièrement, peut-on imaginer que l'apprentissage soit uniquement supervisé ? Il faudrait dans ce cas que beaucoup de connaissances de base soient « précablées ». Or, même ceux qui prennent le partie d'un Nativisme fort, [28]admettent qu'il existe au moins une part d'apprentissage non supervisé dont le but serait de spécifier, en fonction de l'environnement, des structures mentales prédéterminées. Apprentissage supervisé et non supervisé cohabitent et la frontière entre les deux n'est pas nette, si elle existe. Néanmoins, l'apprentissage initial est entièrement non supervisé, puisque la supervision elle-même consiste en une rétroaction complexe (interprétation de sons, d'expressions faciales, etc.) qui nécessite un apprentissage préalable.

Par exemple, l'humain apprend dès son plus jeune âge à « organiser » les sons qu'il perçoit, et ce de manière non supervisée. On peut par exemple citer les travaux de Kuhl, comme [Kuh00] où l'auteur montre que dès sa première année (bien avant de savoir parler), un enfant apprend les propriétés statistiques de la langue dans laquelle il baigne. Ces propriétés sont des régularités liées au fait qu'une langue est structurée, fortement contrainte et non aléatoire. Plus généralement, l'apprentissage non supervisé organise les millions « d'entrées » (visuelles, sonores, etc.) unitaires en provenance de l'environnement de manière à ce que l'apprentissage de concepts abstraits soit possible.

C'est cet apprentissage non supervisé qui permet l'acquisition des « briques de base » (traits) ou représentations intermédiaires à partir desquelles seront construites des représentations plus complexes. La présence de nombreux invariants dans l'ordre d'apprentissage suggère une nature fortement hiérarchique de l'apprentissage. En effet, si les représentations apprises étaient indépendantes, elles pourraient être apprises dans un ordre quelconque, ce qui n'est pas le cas. L'apprentissage supervisé devient possible lorsque les représentations créées lors

[27] C'est-à-dire sans réel recodage des données, ou, lorsqu'il y en a un (réseaux de neurones), il est superficiel.
[28] Comme Noam Chomsky, Jerry Fodor ou Steven Pinker.

de l'apprentissage non supervisées sont suffisantes pour les concepts cibles. Nous pensons que ce moment, où l'apprentissage supervisé d'un concept devient possible, correspond au passage du type-2 au type-1. Cela implique que ce qui est appris initialement (de manière non supervisé), est appris sans but particulier, sauf peut-être celui de représenter le plus concisément possible les perceptions. En fait, c'est la structure innée du cerveau qui contient implicitement le biais inductif qui guide l'apprentissage. Même si personne ne peut dire aujourd'hui en quoi consiste ce fantastique biais inductif, nous pensons qu'une possibilité probable est qu'il inclue une préférence pour la simplicité[29].

En apprentissage automatique, la partie non supervisée est donc le plus souvent omise. Lorsque l'apprentissage commence, il est immédiatement guidé par les données étiquetées. L'apprentissage non supervisé en informatique consiste à créer des groupements de données similaires mais n'est pas utilisé pour créer des *représentations* que l'apprentissage supervisé utilisera. Nous avons vu qu'en apprentissage biologique (le cas de l'humain ayant été le plus étudié), une phase d'apprentissage non supervisée était indispensable. Elle permet d'organiser les perceptions en termes de concepts, de *reconstruire* l'environnement de manière prédictive. Nous pensons qu'il en est de même en apprentissage automatique, même si ce n'est pas prouvé et en dépit du fait qu'il s'agit d'une pratique quasi inexistante. D'ailleurs, le résultat d'un apprentissage non supervisé **existe** indirectement dans presque tous les cas : il réside dans l'utilisation de traits (calculés au préalable), imaginés et conçus par des experts dans le but de capturer les régularités statistiques d'un domaine d'application particulier. L'incapacité des algorithmes d'apprentissage à découvrir un recodage des données capable de rendre les concepts cibles statistiquement visibles est compensée, quoique faiblement, par un recodage arbitraire d'origine humaine.

Par exemple, dans le domaine de l'image on utilise les traits dont nous avons parlé précédemment (couleurs, textures, etc.) plutôt que les pixels ; dans le domaine de la

[29] En général, la perception « comprise » d'une scène (scène décomposée en entités connues, comme des notes de musique ou des objets visuels) est plus agréable qu'une scène non comprise (comme la vision floue ou l'écoute de sons non structurés). Une scène comprise est perçue de manière concise, symbolique. Une scène non comprise (ou non reconstruite) ne peut être simplifiée car le cerveau ne peut en extraire des symboles : cela provoque une sensation désagréable que l'on cherche donc à éviter.

reconnaissance vocale, on utilise par exemple les MFCC[30] plutôt que le flux audio brut. Pourtant, l'information dont l'algorithme d'apprentissage a besoin est bien présente dans les pixels ou dans le flux audio brut. Il y a même beaucoup plus d'information que ce que les traits en retiennent. Toutefois, puisqu'il n'existe pas de corrélation évidente entre des concepts de haut niveau et les « unités d'informations » (comme les pixels), on crée et utilise des traits. L'avantage principal est la simplicité : d'une part la taille des données manipulées est énormément diminuée par rapport aux données brutes, d'autre part, on a la garantie d'obtenir un résultat, car beaucoup d'efforts ont été faits pour que les traits soient *directement corrélés* aux concepts à apprendre. La conception de traits de bas niveau *ad hoc* a en fait pour objectif de transformer un problème d'apprentissage de **type-2** en problème de **type-1**, en faisant apparaître des corrélations statistiques de surface entre traits et concepts.

4.2.4 Limitations

Bien que la « méthode » exposée ci-dessus, qui consiste à remplacer un apprentissage non supervisé par des traits, permette souvent d'obtenir des résultats satisfaisants, nous pensons que ces résultats sont intrinsèquement limités pour les raisons suivantes :

a) Une fois les corrélations statistiques exploitées (par exemple, « Beaucoup d'angles et de lignes » renforce le concept « Bâtiment » comme dans [Tan01]), si les résultats ne sont pas satisfaisant, il faut créer des traits de bas niveau davantage corrélés aux concepts à apprendre. Or, créer des traits de bas niveau discriminant et invariants à de nombreuses transformations est une tâche dont l'ampleur est énorme. Des traits trop grossiers (c'est-à-dire invariants à de trop nombreuses transformations) ne seront pas capables de différencier des concepts « proches ». Similairement, des traits très fins (et donc nécessairement nombreux, si l'on veut préserver l'information) seront capables de distinguer des fines variations mais perdront leurs corrélation avec les concepts à apprendre.

[30] Les « Mel-Frequency Cepstral Coefficients » ont été créés pour être représentatifs de la perception humaine.

b) Les traits utilisés ne conservent que très peu d'information par rapport aux concepts cibles. Comment alors pourrait-on apprendre des concepts visuels comme « France », « Noël » ou « Véhicule » avec des histogrammes de couleurs et de textures ?

c) L'apprentissage étant supervisé, on pondère (ce n'est pas systématique) les traits selon leur corrélation avec les concepts à apprendre : on risque donc de négliger les traits dont la corrélation n'est pas évidente, c'est-à-dire les régularités de type-2. Nous détaillons ce point plus loin.

d) Les hypothèses sur la nature du domaine, faites lors de la création des traits, sont différentes des hypothèses contenues implicitement dans l'algorithme d'apprentissage (i.e. le biais inductif). Par conséquent, l'adéquation entre les traits et les données n'est que partielle. En particulier, un trait contient implicitement une notion de similarité (l'ensemble des propriétés qui font apparaître ce trait sont considérées comme similaire). Or cette notion de similarité n'est pas issue des données sur lesquelles l'algorithme travaille mais plutôt de l'esprit de la personne qui a conçu les traits. Par conséquent, les traits utilisés ne correspondent pas nécessairement aux régularités présentes dans le *corpus* d'apprentissage.

Pour ces raisons, nous pensons que l'apprentissage supervisé seul n'est pas suffisant pour des problèmes d'apprentissage de type-2. La création de traits apporte une solution dans certains cas (reconnaître des caractères par exemple) mais devient rapidement hors de portée pour l'apprentissage de concepts abstraits car ce qui caractérise ces concepts ne peut être décrit concisément en termes de traits simples : on peut décrire ce qui caractérise le concept « Caractère 'A' » facilement en termes de segment, courbes, angles mais on ne peut pas décrire de la même manière le concept « Végétation ». On peut donc créer des traits discriminants et suffisants dans le premier cas, pas dans le second.

La difficulté majeure dans le processus de création des traits est qu'il est nécessaire de comprendre quels sont les traits que nous utilisons en tant qu'humains. Une compréhension superficielle de ces traits est possible, elle est même d'actualité. Par exemple, nous savons que dans la reconstruction mentale d'une scène visuelle, la

notion de « ligne » joue un rôle important, tout comme la notion de couleur ou de texture. Nous le savons car nous avons un accès *conscient* à ces caractéristiques. Mais quelles sont les *autres* caractéristiques, celles pour lesquelles nous n'avons même pas de mots, « combien »[31] y en a-t-il, et comment les découvrir ? De plus en plus d'approches mettent en relation des recherches informatiques et cognitives dans le but de faire avancer la vision par ordinateur (et d'autres thèmes liés à la perception en général). On peut se poser la question de la limite de ces approches, car l'exploration du monde de la pensée est l'une des tâches les plus difficiles qu'il soit, mais également de l'intérêt, pour l'informatique, de pousser ces recherches beaucoup plus loin. On peut en effet faire l'hypothèse légitime que les traits construits dépendent, en grande partie, des propriétés physiques des « capteurs » qui approvisionnent en information des structures d'apprentissage en aval, capteurs qui constituent la frontière entre l'environnement et la représentation de l'environnement. De plus, les traits utilisés par un individu dépendent certainement de l'environnement dans lequel celui-ci a été plongé, et ne sont pas nécessairement adapté à des environnements différents, comme c'est le cas pour les langues par exemple.

Nous nous inspirons de l'apprentissage « biologique » (du moins certains aspects) pour définir un modèle d'apprentissage qui, au lieu d'imiter artificiellement la perception humaine, commence par s'adapter à son environnement (i.e. l'ensemble d'apprentissage qu'on lui fournit), avant tout apprentissage supervisé de concepts.

4.2.5 Vers un couplage séquentiel *non supervisé → supervisé*

En résumé, nous pensons que l'apprentissage supervisé devrait être la dernière étape du processus d'apprentissage. Ce processus devrait au préalable construire, de manière non supervisée, les traits qui seront utilisés lors de la liaison avec les concepts. Ces traits devraient refléter les régularités présentes dans l'ensemble d'apprentissage.

[31] Il peut être difficile de dénombrer ce qui est entremêlé.

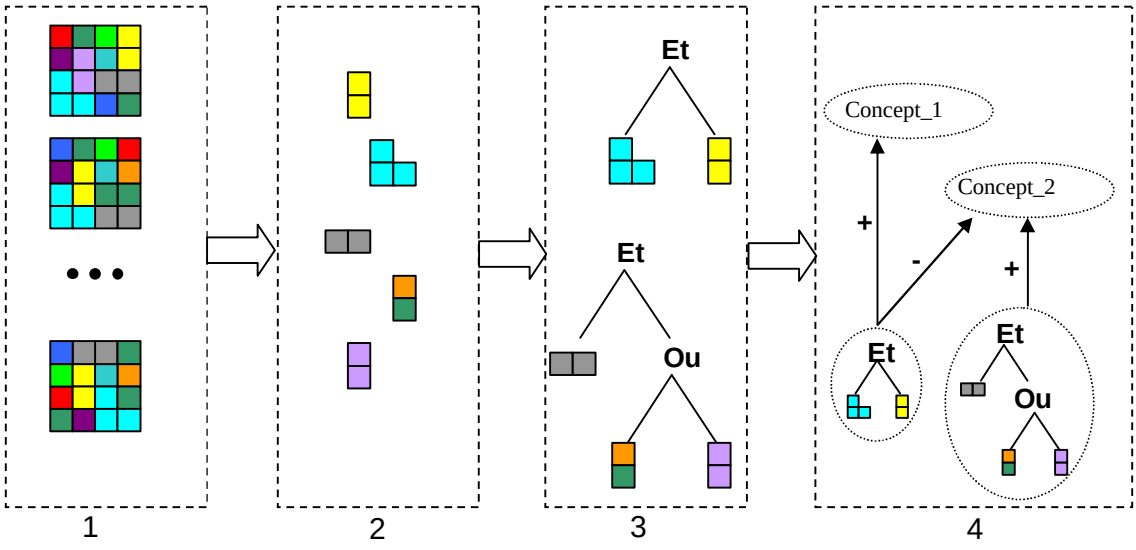


Figure 4-1 : Un processus d'apprentissage non supervisé, puis supervisé.

La Figure 4-1 schématise l'apprentissage tel que nous l'envisageons : l'apprenant est soumis à un flux (1) de données, ces données n'étant pas générées aléatoirement, elles possèdent certaines régularités que l'apprenant découvre : d'abord, des régularités simples (2) impliquant un petit nombre de traits, puis ces régularités sont utilisées pour construire des traits plus complexes, de manière hiérarchique (3). Lorsque l'apprenant possède les représentations nécessaires, l'apprentissage supervisé peut avoir lieu : les régularités perçues influent positivement ou négativement sur la détection de certains concepts (4). Par exemple, une flèche '+' signifie qu'une régularité est fréquemment présente dans les instances du concept. Par conséquent, découvrir cette régularité dans un exemple inconnu renforce la croyance que cet exemple est une instance de ce concept. A l'inverse, une flèche '-' signifie qu'une régularité n'est pas statistiquement liée à un concept.

L'idée centrale de cette vision de l'apprentissage est qu'il est préférable de construire l'espace de représentation des données *à partir des données*, plutôt que de chercher à *délimiter* ces données dans un espace arbitraire.

Cette vision de l'apprentissage réduit les limitations évoquées plus haut de la manière suivante :

a) Les traits utilisés ont divers niveaux de granularité, en fonction de l'exemple observé. Le niveau de détail varie donc en fonction des données observées.

b) Si les traits atomiques sont suffisamment nombreux, l'information nécessaire sera conservée. En même temps, le fait qu'il y ait de très nombreux traits atomiques n'est pas un problème car dans la plupart des cas, des régularités seront perçues et l'espace dans lequel sera vu l'exemple s'en trouvera d'autant diminué.

c) La pondération est inhérente à cette approche de l'apprentissage. Si un trait est important (i.e. très présent dans les données), il sera construit rapidement, et sera par conséquent beaucoup utilisé pour fabriquer d'autre traits. Donc, son importance relative moyenne (car il y a dépendance au contexte) sera élevée par rapport à des traits rares.

d) La construction des traits et l'algorithme d'apprentissage sont le **même processus**. De plus, les traits étant issus des régularités de l'ensemble d'apprentissage, ils sont nécessairement pertinents par rapport à celui-ci.

4.3 Apprentissage non supervisé de régularités

On peut définir une régularité comme l'existence d'un agencement de traits atomiques, dont la probabilité d'apparition est plus grande que la probabilité moyenne d'observer des agencements du même type (même structure, même nombre de traits atomiques impliqués). La notion de régularité est le fondement de l'apprentissage : la régularité est l'inverse du chaos, elle permet de compresser l'information, de faire des prévisions, de catégoriser et d'abstraire. Les régularités sont également à l'origine comme nous allons le voir à la notion de similarité.

Parallèlement, il nous semble fondamental de considérer la notion de régularité comme une entité hiérarchique. La hiérarchisation permet la représentation de concepts en sous-concepts, et réduit donc un problème en sous-problèmes. Les briques de base que constituent ces sous-concepts représentent des unités de connaissance pouvant être imbriqués en structures plus complexes. Ces unités de

connaissances peuvent être ré-utilisées, ce qui permet une représentation compacte des données.

D'autre part, les données et concepts sur lesquels porte l'apprentissage proviennent tous du même monde structuré. Des lois physiques régissent la manière dont sont générées ces données : les particules s'assemblent en systèmes, ces systèmes s'assemblent à leur tour pour former des entités plus complexes. Ces lois étant toujours les mêmes, toutes les entités partagent certaines propriétés qui se traduisent par des régularités, et ce quelle que soit la source de perception. Le but de l'apprentissage est alors de « reconstruire » cette hiérarchie de régularités.

4.3.1 Régularités et concepts

Nous voyons l'apprentissage comme un moyen d'acquérir des régularités et de les associer à des concepts. Mais les notions de régularité et de concept sont-elles si différentes ? Un concept est défini par le dictionnaire comme *une idée générale ou abstraite inférée ou dérivée d'exemples spécifiques*. Une régularité est définie comme *la qualité d'être caractérisé par une caractéristique fixée*. Ces deux définitions suggèrent que la différence principale entre un concept et une régularité est que le premier est *subjectif* (perçu par un humain) alors que la seconde est objective (caractéristique intrinsèque des données observées). D'ailleurs, les concepts portent généralement des noms, là où les régularités ne sont que perçues. Pourtant, dans les deux cas, concepts et régularités sont subjectifs, puisqu'ils sont observés par des humains, mais possède également une part d'objectivité dans la mesure ou ils permettent de faire des prédictions qui s'avèrent vérifiées, ce qui implique que ces concepts et régularités sont bien des propriétés intrinsèques ne dépendant pas de l'observateur.

Par conséquent, par la suite, nous considérons que concepts et régularités sont identiques, à la différence que les concepts sont des régularités générant de l'intérêt, et pouvant toujours être perçus consciemment. Les simples régularités quant à elles, sont similaires aux concepts mais, ne provoquant que peu ou pas d'intérêt, ne sont pas perçues nécessairement de manière consciente. Les concepts peuvent être vus comme les éléments de haut niveau dans une hiérarchie de régularités, bien qu'il puisse y avoir des exceptions (comme les concepts correspondant aux couleurs).

En pratique cependant, nous devons nous attendre à ce que les régularités qui correspondent à des concepts d'intérêt (complexes et abstraits) soient difficiles à apprendre. Nous aurons alors recours à l'apprentissage supervisé, dont le but sera de guider et d'accélérer l'apprentissage *vers* ces concepts.

Cette hypothèse, qui suppose que concepts et régularités sont des propriétés des données, que nous pourrions nommer « hypothèse de l'universalité des concepts », est très importante car de sa véracité dépend la possibilité d'une intelligence artificielle « forte ». L'intelligence artificielle, en effet, a généralement pour objet des concepts « humains ». Or, si la nature des concepts dépend principalement du *support* de l'apprentissage (machine, cerveau), il semble compromis que les concepts appris par une machine convergent vers ces concepts humains. A l'opposé, si la nature des concepts dépend principalement des données, la convergence est possible et ne dépend plus que des capacités d'apprentissage des supports que nous créons.

Si cette hypothèse est vérifiée, l'apprentissage non supervisé prend tout son sens car l'apprentissage de régularités conduit à l'apprentissage des concepts. Dans cette vision, *régularités et concepts sont avant tout des propriétés intrinsèques des données* et seuls les *noms* donnés aux concepts proviennent d'une source 'extérieure'. Dans la pratique cependant, on ne peut pas raisonnablement s'attendre à ce que les concepts soient appris directement et une phase d'apprentissage supervisée est nécessaire pour contraindre les liens entre régularités et concept.

Dans les approches traditionnelles de l'apprentissage, régularités et concepts sont considérés distinctement. Les régularités sont d'abord identifiées humainement, il s'agit souvent de similarités entre couleurs, de textures particulières, ou encore de formes. Des détecteurs spécialisés dans l'identification de ces régularités sont créés et utilisés pour recoder les données. Par conséquent, les régularités extraites proviennent d'une source extérieure aux données. Les concepts sont ensuite appris, généralement pas descente de gradient, à partir de ces régularités artificielles.

4.3.2 Types de régularité

Dans ce qui suit, en particulier les figures, nous utiliserons une notation arborescente pour décrire les régularités, cette représentation se prêtant particulièrement bien à la description hiérarchique de l'information.

Nous avons jusqu'à présent défini la notion de régularité de manière abstraite. Il est maintenant nécessaire de lui donner un sens concret. Pour cela, nous considérons un flux d'information F constitué d'une succession de vecteurs V_i, chaque vecteur étant lui-même constitué de n booléens. L'utilisation de booléens permet l'apprentissage de régularités sans qu'il soit nécessaire de définir ou de calculer de seuils. Si ces valeurs étaient, par exemple, des entiers, il faudrait définir une mesure de similarité qui dépende de la distribution de ces valeurs. Ce flux est issu de la numérisation d'un phénomène quelconque d'origine naturelle mais nous nous focalisons ici sur des données visuelles. Nous ne nous intéressons pas ici à la dimension temporelle du flux d'information, c'est-à-dire que l'ordre dans lequel sont ordonnés les vecteurs, même s'il peut influer sur les régularités observées, n'est pas notre objet d'étude.

Qu'est-ce qu'une régularité dans ce flux d'information ou, en d'autres termes, quel type d'information pourrait être utile pour effectuer des prédictions sur la valeur des attributs des vecteurs ? De la réponse à cette question dépend une partie importante du biais inductif de notre approche car les types de régularité identifiés seront le seul sous-ensemble représentable parmi l'infinité des régularités imaginables.

Etant donnés un flux de données visuelles (régions ou images) et un ensemble de traits atomiques, nous définissons les régularités suivantes, illustrées graphiquement Figure 4-2

- **Type Unaire** : Un trait atomique apparaît (a pour valeur 1, ou vrai) fréquemment. Cette fréquence permet d'effectuer des prédictions sur la valeur de l'attribut (dont la confiance est proportionnelle à cette fréquence). Dans le cas d'un flux de données infini, ou dont on ne connaît pas la taille, la fréquence est remplacée par un couple *compteur/seuil.*

- **Type 'Et'** : La valeur d'un trait atomique A est corrélée avec la valeur d'un autre trait atomique B. Cette corrélation se traduit par l'observation de cooccurrences. Selon la fréquence de ces cooccurrences, il peut s'agir d'une *implication* ($A \Rightarrow B$) ou plus généralement d'une *corrélation*. Par exemple, le nœud 'C' de la Figure 4-2est une régularité de type 'Et' présente lorsque les nœuds 'a' et 'b' sont actifs. Nous verrons que dans une représentation hiérarchique, une régularité de type 'Et' définit une **structure**, en ce sens qu'elle permet l'assemblage d'unités d'information apprises dans le passé.
- **Type 'Ou' :** Nous qualifions de régularité le fait que des traits atomiques puissent apparaître séparément, mais dans un contexte identique. Une régularité de ce type est donc formée d'un contexte et d'un ensemble de traits atomiques considérés comme équivalents dans ce contexte[32]. Par exemple, le nœud X de la Figure 4-2 est une régularité présente lorsque le contexte (nœud 'n') est présent, ainsi que l'un de nœuds 'e', 'f' ou 'g'. Les nœuds 'a', 'b' et 'c' quant à eux sont des régularités de type 'Et'. Nous verrons comment les régularités de ce type apportent des **degrés de liberté** dans les structures définies par les régularités de type 'Et'.

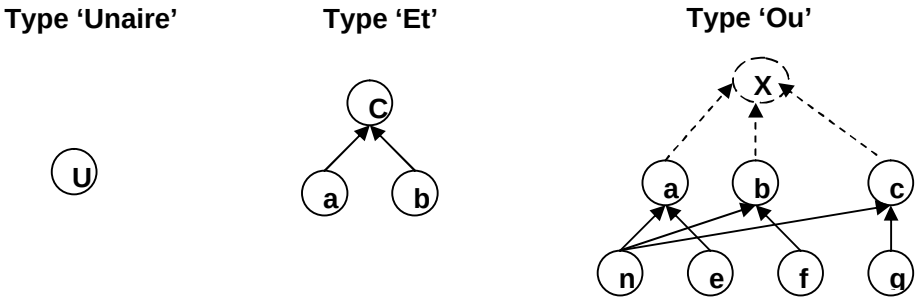


Figure 4-2 : C et X représentent trois types de traits : un trait de type unaire (U), une conjonction (c) et une disjonction (x)

[32] Nous définirons plus loin ce que nous entendons par contexte. Pour le moment, nous dirons simplement qu'il s'agit d'un invariant, d'un trait présent dans plusieurs conjonctions.

Un point important est que du point de vue du pouvoir d'expression (ce qui est représentable), les régularités de type 'Et' suffisent. En effet, une régularité de type 'Ou', ou contenant une régularité de type 'Ou' peut être représentée par un ensemble indépendant de régularités de type 'Et'[33]. Toutefois, il s'agit d'une forme d'énumération ne se prêtant pas à la généralisation. En introduisant les régularités de type 'Ou', une forme de généralisation est introduite. En effet, une fois apprise, une régularité de type 'Ou' peut être utilisée dans la construction de nouvelles régularité sans qu'il soit nécessaire de rencontrer lors de cet apprentissage tous les traits auxquels elle fait référence.

Il nous semble nécessaire, à ce point de l'exposé, de faire une distinction importante concernant la notion de disjonction. Il existe deux types de disjonctions très différentes que nous qualifierons de **disjonction énumérative** et de **disjonction constructive**.

Les disjonctions **énumératives** sont communes à tous les algorithmes d'apprentissage. Elles proviennent du fait qu'un concept est représenté par une disjonction de ses instances. Par exemple, il peut s'agir des branches d'un arbre de décision aboutissant au même concept, ou de plusieurs points dans l'espace correspondant à un même concept dans le cas de la classification par plus proches voisins. Ces disjonctions implicites sont nécessaires car, sauf dans des cas triviaux, les concepts appris ne peuvent pas être représentés par un unique point dans l'espace des hypothèses. La raison à cela est que généralement, la notion de distance dans « l'espace des concepts » est très différente de la notion de distance dans l'espace utilisé pour représenter les données. Nous qualifions ces disjonctions d'énumératives car leur but est d'énumérer des instances, ou groupes d'instances d'un concept, pour pallier au fait que ces instances sont trop dissimilaires (à cause d'une représentation mal adaptée par exemple) et ne peuvent être couvertes de manière unifiée.

Les **disjonctions constructives** à l'opposé sont une propriété intrinsèque de l'espace de représentation des données. Une disjonction de ce type exprime le fait qu'un ensemble de configurations *différentes* dans les données initiales doivent être

[33] Par exemple, la régularité (A et B et (C ou D)) peut être représentée par l'ensemble de régularités : {(A et B et C) , (A et B et D)}.

vues comme *équivalentes ou similaires* dans l'espace utilisé pour représenter ces données. Par exemple, on peut imaginer qu'une des dimensions dans l'espace de représentation corresponde à la valeur de sortie d'un détecteur de lignes horizontales. Ce détecteur peut produire des valeurs de sortie identiques, ou similaires, alors que dans les données initiales, les lignes en question ne sont pas situées au même endroit, n'ont pas la même couleur ou la même épaisseur. Nous qualifions ces disjonctions de constructives car elles introduisent une notion de *similarité* qui est absolument indispensable à la construction des hypothèses, et donc à l'apprentissage. Cependant, ce type de disjonction est généralement implicite, se cachant dans les traits de bas niveau ou les espaces de couleurs. Un de nos objectifs est de rendre explicite l'apprentissage de ces disjonctions.

Formes normales

Les régularités que nous venons d'évoquer peuvent être décrites plus formellement par les formes normales issues de la logique des prédicats. On distingue deux formes normales :

- DNF : forme normale disjonctive (« Disjonctive Normal Form »)
  - o $F_1 \vee F_2 \vee ... \vee F_k$ où chaque F_i est une conjonction de littéraux (des traits, dans notre cas)
- CNF : forme normale conjonctive (« Conjonctive Normal Form »)
  - o $F_1 \wedge F_2 \wedge ... \wedge F_j$ où chaque F_i est une disjonction de littéraux (des traits, dans notre cas)

Nous suggérons que les formes normales disjonctives sont implicitement la représentation courante des hypothèses dans la majorité des algorithmes d'apprentissage. On peut voir le $\vee$ des DNF comme une **disjonction énumérative**, c'est-à-dire que chaque F_i représente une des instances du concept représenté (c'est par exemple une branche dans un arbre de décision, ou un point dans l'espace).

Notre approche pour l'apprentissage des régularités utilise une représentation sous forme normale conjonctive. Les $\vee$ contenus dans les CNF correspondent à des

disjonctions constructives : chaque F_i représente une régularité de type 'Ou' qui contribue à l'expression du concept (la forme CNF totale). Nous pensons que l'intérêt principal de ce type de représentation est qu'il permet l'apprentissage explicite de la notion de similarité *à partir des données*. Nous détaillerons ce point plus loin.

Il existe une motivation supplémentaire justifiant la représentation par CNF. Il est démontré en théorie de l'apprentissage ([Mit97], page 213) qu'un concept est apprenable s'il est exprimé sous forme normale conjonctive, mais qu'il ne l'est pas s'il est exprimé par une forme normale disjonctive.[34] L'apprentissage d'un concept représenté par une DNF de k littéraux nécessite un nombre d'exemples d'apprentissage qui est polynomial par rapport à k mais le problème vient de la complexité calculatoire qui elle n'est pas polynomiale (car on peut montrer que ce problème est réductible à des problèmes NP-complets). Sous une forme CNF, le nombre d'exemples nécessaires, ainsi que la complexité calculatoires sont polynomiaux par rapport à la taille k du concept. Ce résultat a cela d'étonnant que la forme CNF est plus expressive que la forme DNF ; toute expression DNF peut en effet être exprimée par une expression CNF mais la réciproque n'est pas vraie.

Notons toutefois que nous n'utiliserons qu'une forme limitée de CNF, dans la mesure où nous ne considérons pas les négations. La motivation de ce choix est *sémantique* plus que formelle : l'utilisation d'une variable dans une formule logique traduit simplement l'existence d'une caractéristique particulière dans une observation (comme la présence de couleur rouge par exemple). L'utilisation de la négation de cette même variable porte une sémantique moins triviale que la simple observation d'une caractéristique. En effet, le fait de d'expliciter la non existence d'une caractéristique sous-entend une *attente* sur la valeur de cette caractéristique (sauf si la formule contient toutes les négations possibles). Cette attente sous-entend à son tour des connaissances préalables : connaissances *a priori* ou résultant d'un apprentissage. Par exemple, le fait d'utiliser « *non rouge* » dans une formule sous-entend que « *l'on s'attendait à du rouge* » et suppose donc déjà une certaine compréhension basée sur les autres caractéristiques. Inclure la notion de négation

[34] Par apprenable, on entend que le concept peut être appris avec une probabilité p arbitrairement grande avec un taux d'erreur e arbitrairement petit, en un temps polynomial.

impliquerait alors une scission du processus d'apprentissage, afin de gérer *a posteriori* les négations.

4.3.3 Disjonction & abstraction

Apprendre des descriptions plutôt que des énumérations requiert une capacité de généralisation, ou d'abstraction. Or, ce qui caractérise l'abstraction, ou la généralisation est une certaine indépendance aux détails, c'est-à-dire le fait de considérer certains ensembles de traits comme similaires, interchangeables, dans un certain contexte particulier. Or ce qui permet cela est la notion de disjonction. Classiquement, la généralisation est obtenue par simple élimination d'un attribut. Par exemple, si l'on supprime l'attribut « couleur » d'une instance du concept « mer », l'expression devient plus générale… mais fausse, car même si la mer peut revêtir de nombreuses teintes, beaucoup sont improbables ou impossibles. La généralisation est alors trop forte et sera la cause d'erreurs de classification. La généralisation obtenue par apprentissage de disjonctions est plus fine car elle permet une *certaine* indépendance vis-à-vis d'un attribut particulier.

Une disjonction représente un ensemble, elle est vraie lorsqu'au moins un élément de l'ensemble est vrai, nous l'avons vu pour les couleurs par exemple. De ce fait, elles permettent une abstraction qui simplifie l'expression d'un concept. Par exemple, on peut exprimer le concept « Mer » de deux manières :

1 : $Mer = F_{bleu} \wedge F_{Orienté}$

C'est la forme CNF où $F_{Bleu} = F_{Bleu1} \vee F_{Bleu2} \vee ... \vee F_{Bleu_k}$ est un ensemble de couleurs bleues spécifiques à la mer. C'est une régularité **constructive**. De la même manière, $F_{Orienté} = F_{Orienté1} \vee F_{Orienté2} \vee ... \vee F_{Orienté_j}$ représente un ensemble de textures orientées spécifiques à la mer qui est également une régularité constructive.

On peut également exprimer le concept « Mer » par une forme DNF :

2 : $Mer = (F_{Bleu1} \wedge F_{Orienté1}) \vee (F_{Bleu2} \wedge F_{Orienté1}) \vee (F_{...} \wedge F_{...}) \vee (F_{Bleu_k} \wedge F_{Orienté_j})$

Sous cette forme, les disjonctions sont dites **énumératives** car l'expression est construite en énumérant quelques unes des instances du concept « Mer » : celles rencontrées lors de l'apprentissage. Cette formulation du concept « Mer » ne contient donc aucune généralisation intrinsèque. Un algorithme « strict » de classification ne classifierait comme « Mer » que des exemples rencontrés lors de l'apprentissage. En pratique toutefois, l'algorithme n'est jamais strict, il classifie un nouvel exemple selon sa **similarité** avec les exemples appris et non selon une **égalité**. Cependant, cette notion de similarité est tout à fait arbitraire. Par exemple, la similarité entre F_{bleu1} et F_{bleu5} est toujours plus grande que la similarité entre F_{bleu1} et F_{vert18} : cela peut être vrai dans le contexte « Ciel » mais complètement faux dans d'autres, comme « Voiture » ou « Fleur ».

L'expression CNF est plus « constructive » que l'expression DNF. La raison est simplement que F_{bleu} (ou $F_{orient'e}$) peut probablement être utilisé pour la construction (description) d'autres concepts, ce qui permet un partage des représentations, alors que chaque expression entre parenthèses dans la deuxième expression est si spécifique, que la probabilité d'être réutilisée est faible.

De plus, la première expression permet de généraliser à des cas non vus, ce que ne peut pas la seconde[35]. En effet, l'expression DNF n'est vraie que si un cas déjà vu est rencontré, alors que l'expression CNF peut être vraie même si une combinaison de couleur de mer et de texture de mer non apprise est rencontrée. Dans l'expression DNF, un exemple inconnu ne sera reconnu comme instance du concept « Mer » que s'il est semblable (dans un sens arbitraire) à au moins une des instances apprises ; alors que dans l'expression DNF, un exemple sera reconnu s'il est constructible à partir de l'expression proposée. Il y a là également une notion de similarité, mais il s'agit d'une similarité apprise.

On peut légitimement se demander pourquoi on n'utilise pas directement des traits de bas niveau qui codent ces disjonctions très utiles (comme F_{bleu} et $F_{orienté}$). La première partie de la réponse est que c'est précisément ce qui se fait actuellement : espace HSV, Gabor, etc. Ces traits sont construits de manière *ad hoc* pour être les

[35] En pratique, on assiste tout de même à une généralisation, mais d'un autre ordre, car on utilise des mesures de similarité entre l'exemple inconnu à classifier et le modèle appris.

plus généraux possibles. Cependant, ils ne sont pas adaptés pour représenter la couleur de la mer comme F_{bleu}, car la mer a son propre ensemble de couleurs qui ne sont pas nécessairement connexes dans un espace de couleurs. En fait, le fait qu'une couleur soit équivalente à une autre dépend entièrement du **contexte**. Bleu et vert sont proches dans le contexte « Mer » mais très éloignés dans le contexte « Ciel ». La deuxième partie de la réponse est donc que l'on ne peut pas, à l'avance, trouver de traits qui permettent d'exprimer succinctement tous les concepts. **Cette notion de contexte est fondamentale car la notion même de similarité n'a pas de sens sans contexte.**

4.3.4 Un contexte est une régularité

C'est dans un contexte particulier qu'une régularité de type 'Ou' prend son sens, c'est bien entendu le cas dans notre contexte de recherche d'image mais c'est également vrai dans n'importe quel domaine. Une régularité de type 'Ou' représente un ensemble de traits similaires dans un contexte donné. Une régularité de type 'Et', est par contre indépendante de tout contexte : si deux traits co-occurrent souvent dans un ensemble d'observation, c'est une régularité, quoi qu'il arrive. Cela est dû au fait que les conjonctions décrivent une structure, une construction, alors que les formes disjonctives décrivent une généralisation dont le sens repose sur un contexte particulier. Par exemple, « Nuage ET ciel » représente une régularité, en dépit de tout contexte précis particulier[36], alors que « Vert OU orange OU rouge » n'est une régularité que dans un (ou plusieurs) contexte particulier comme « Feux de circulations » ; il n'y a aucune raison de considérer ces trois éléments comme interchangeables ou équivalents dans un contexte différent.

Il convient maintenant d'expliciter ce que nous entendons exactement par 'contexte' :

Toute régularité est un contexte pour les régularités qui lui sont associées de manière conjonctive.

[36] Quel que soit le contexte, la présence de nuages est fortement corrélée avec celle du ciel, et vice versa.

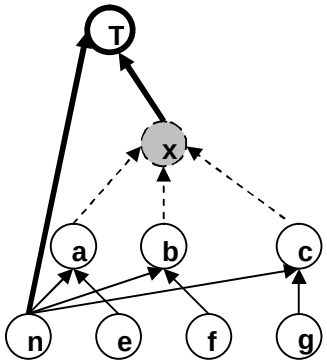


Figure 4-3 : x est une régularité de type 'Ou', a, b, c et T sont des régularités de type 'Et'. 'n' est le contexte dans lequel e,f et g sont similaires.

Dans la Figure 4-3, 'a' représente une régularité de type 'Et', c'est-à-dire que 'n' et 'e' occurrent souvent ensemble, 'b' représentent une régularité (cooccurrence de 'n' et 'f'), et c également. Ce qui est commun dans les traits 'a', 'b' et 'c', c'est la présence de 'n' : la signification est « en présence de 'n', on peut s'attendre à 'e', 'f', ou 'g' ». Ceci est une régularité, exprimée par le trait 'x' ; remarquons que 'x' n'est pas directement lié aux traits 'e', 'f' et 'g' car la régularité n'est valable qu'en présence de 'n'.

En fait, 'n' représente dans notre exemple le **contexte** dans lequel 'e', 'f' et 'g' peuvent être considérés comme similaires. Cela nous permet de définir la notion de similarité *locale*.

4.3.5 Définition de la similarité locale

Nous avons vu que, dans un flux de données non aléatoires, il est possible d'apprendre des régularités conjonctives et disjonctives. Le premier type représente la notion de cooccurrence, de composition, d'assemblage : ces régularités de type 'Et' permettent la construction hiérarchique de représentations complexes. Cependant, elles sont en réalité très strictes, car un assemblage hiérarchique de 'Et' imbriqués est dépendant de la moindre composante : si un seul composant de l'assemblage manque, c'est-à-dire n'est pas « activé », la régularité correspondant à cette construction n'est pas non plus activée. Dans ce scénario peu souple, il faudrait réapprendre une grande partie de cette construction pour représenter une régularité quasi-identique.

La souplesse est apportée par les régularités disjonctives, de type ‘Ou’. En fait, on peut voir les régularité conjonctives comme définissant une structure rigide, et les régularités de type ‘Ou’ comme des « degrés de liberté » présents à chaque niveau de la structure, et autorisant une certaine souplesse. Chaque niveau de la structure étant un contexte dans lequel une certaine variabilité, bien définie, est autorisée. Cela ressemble par exemple à la manière dont nous nous représentons le concept de « maison » : il existe une structure rigide imposant qu’une maison soit formée de murs, d’un toit, de portes et de fenêtres. Pour chacun des niveaux toutefois, il existe une certaine souplesse. Par exemple, le toit peut être constitué de tuiles ou de tôles, les murs peuvent être un assemblage de briques ou de planches de bois. La taille des éléments, leurs couleurs, peuvent varier, mais de manière restreinte. Toute cette souplesse, à chaque niveau d’une structure rigide, constitue la notion de similarité.

La similarité est à la base de la formation de concepts, des comparaisons, de l’abstraction et donc de l’apprentissage. Cette notion étant si fondamentale, on pourrait imaginer qu’elle est définie clairement et formellement. Ce n’est pas du tout le cas. Si l’on regarde dans différents dictionnaires quelles sont les définitions de la similarité, on trouve :

- « La qualité d’être similaire » - WordNet [www-Wordnet]
- « Fait d’être presque identique » - Cambridge Online Dictionaries [www-COD]
- « Fait de posséder une ou plusieurs caractéristiques communes » - Encarta [www-Encarta]

Ces définitions, comme bien d’autres non mentionnées ici, sont caractérisées par auto référence : elles définissent la similarité en terme d’elle-même ou de concepts similaires comme l’identité, la ressemblance, la distance, etc. Les deux premières définitions sont clairement définies par auto référence. La troisième également, mais de manière moins évidente : si deux objets possèdent des caractéristiques différentes et des caractéristiques communes, comment savoir *quelles caractéristiques comparer* ? Si l’on considère qu’une caractéristique n’est pas commune à deux objets, c’est que l’on sait déjà que ces caractéristiques sont

comparables, et donc similaires (d'un point de vue fonctionnel par exemple). Par exemple, considérer que deux visages se ressemblent car leurs forme est similaire implique, *a priori*, la détection de ces formes par rapport aux autres aspects d'un visage, qui implique une connaissance de la notion de similarité entre différents aspects.

En science (informatique, physique, mathématique), on utilise de nombreuses mesures de similarités basées sur un certain nombre de distances : la distance Euclidienne (ou 2-norme, la distance la plus intuitive), la distance de Mahalanobis, la distance de Hamming (en théorie de l'information), la distance EMD (Earth Mover Distance, pour les histogrammes), la distance de Manhattan (ou 1-norme), etc. Il existe potentiellement une infinité de manière de mesurer la similarité et on choisit habituellement la plus appropriée en fonction du contexte, et ce de manière plus ou moins intuitive.

En informatique, il est même fréquent que la mesure de similarité fasse l'objet de recherches poussées. Par exemple, dans beaucoup d'approches, une mesure de similarité est apprise par un algorithme d'apprentissage, de manière à ce que des objets soient groupés d'une manière cohérente par rapport aux humains.

Suite à notre discussion sur les régularités, nous faisons l'hypothèse suivante :

Lorsqu'un ensemble de régularités conjonctives possèdent toutes un constituant commun, celui-ci est le contexte dans lequel les autres constituants sont considérés comme similaires.

Dans notre approche de l'apprentissage, la notion de similarité est donc centrale : elle est apprise et représentée par des régularités disjonctives. Le fait de considérer que la similarité puisse être apprise à partir d'un flux de données, sans « intervention extérieure » est une position forte, car cela implique que la similarité est une propriété intrinsèque du monde, qui ne dépend pas de celui (ou ce) qui le perçoit mais plutôt de quelles parties de ce monde sont perçues et dans quel ordre. Nous ne détaillerons pas davantage ce point, qui dépasse largement notre cadre et dont la discussion ne serait pas ici à sa place.

Les similarités apprises sont *locales*, ce qui signifie qu'un ensemble de régularités ne sont considérées comme similaires que dans un contexte particulier. Cette localité permet une généralisation *contrôlée*, ce qui ne serait pas le cas si des constituants étaient considérés comme similaires de manière hors contexte.

Remarquons finalement qu'il existe nécessairement en apprentissage automatique, au niveau le plus bas, une forme de similarité « atomique », hors contexte, quelque part entre la numérisation de exemples d'apprentissage (photographie dans le cas des images), et la présentation des exemples à l'algorithme. Cette similarité est nécessaire pour effectuer le passage du continu au discret, elle est nécessaire au formatage des données. C'est d'ailleurs la même chose chez l'homme : chaque neurone produit une sortie binaire à partir d'entrées multiples (jusqu'à 10000 entrées) : la perception d'un l'environnement « continu », aboutit à la discrétisation de celui-ci, qui implique une notion de similarité. Cette similarité « matérielle » est contenue dans le seuil d'activation de chaque neurone : toutes les configurations d'entrées n'aboutissant pas à l'activation de la sortie sont, dans le contexte de ce neurone, similaires, puisqu'elles provoquent le même résultat. De cette similarité « originelle » découlent des similarités plus complexes.

4.3.6 Renforcement de notre biais inductif

Notre approche consiste, jusqu'à présent, à apprendre une structure hiérarchique de régularités conjonctives et disjonctives, de manière non supervisée. L'ensemble de ces choix constitue un *biais*, qui limite et guide l'exploration de l'espace des hypothèses représentables[37]. Nous allons voir maintenant que ce biais est insuffisant et allons suggérer une contrainte supplémentaire pour le renforcer, que nous justifions avec des arguments « anthropomorphiques ».

Considérons des données d'apprentissage constituées de vecteurs de n traits atomiques binaires. L'espace des données d'apprentissage a comme dimension 2^n. La taille de l'espace des hypothèses, considérant qu'une hypothèse est formée d'un nombre de traits atomiques quelconque entre 0 et n est également 2^n dans le cas de

[37] Dans la mesure où nous sommes dans un cadre non supervisé, une hypothèse ne correspond pas à une tentative de représentation d'un concept, mais plutôt à une tentative « d'explication des données ».

régularités uniquement conjonctives. Lorsque les disjonctions sont considérées, l'espace des hypothèses a pour borne supérieure 2^{2^n} , car chaque conjonctions peut avoir comme constituants des disjonctions (au nombre de 2^n).

Nous voyons qu'il est hors de question d'identifier les régularités présentes dans un ensemble d'apprentissage par la « force calculatoire », en détectant itérativement la présence ou l'absence de chaque régularité. Nous devons donc déterminer une heuristique pour nous guider dans cet immense espace d'hypothèses, qui définira partiellement le biais inductif de notre approche.

4.3.6.1 Contrainte 1 : Apprentissage par agglomération

Nous imposons la contrainte suivante :

Toute régularité apprise ne peut avoir comme constituants que des régularités apprises précédemment.

Cette contrainte impose une construction des hypothèses de type « de bas en haut », c'est-à-dire du spécifique au générique. Une bonne propriété induite par cette contrainte est que l'hypothèse courante est, par construction, toujours consistante avec les données. Ce choix impose également que les premières régularités apprises soient de type unaire. Cette contrainte, si elle paraît évidente, pourrait ne pas être tenue s'il existait des sources *externes* de régularités, comme des connaissances *a priori*, d'interactions avec l'utilisateur ou encore provenant d'un module de création de régularités partiellement aléatoires.

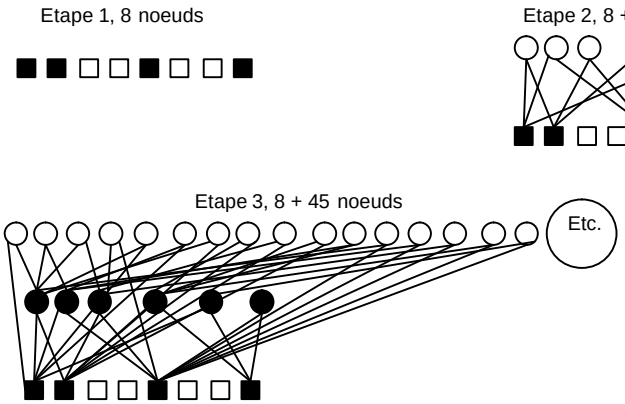


Figure 4-4 : Illustration du problème de l'explosion combinatoire.

Toutefois, cette contrainte n'est pas suffisante car le nombre de régularités, initialement faible, augmente très fortement avec le nombre de vecteurs d'entrée. En effet, les régularités sont initialement des combinaisons de traits atomiques. Puis, elles sont des combinaisons de traits atomiques et de traits non atomiques de « première génération », et ainsi de suite. Prenons un exemple : considérons un vecteur d'entrée de taille n, dont k valeurs sont égales à « vrai ». Dans un premier temps (après avoir présenté n fois le vecteur), les régularités apprises sont des conjonctions entre les nœuds de la couche d'entrée : il en existe $\frac{k \times (k-1)}{2}$. Nous avons alors $\frac{k \times (k-1)}{2} + k = \frac{k^2 + k}{2}$ nœuds. On présente le même vecteur n fois de plus : les régularités apprises sont des conjonctions entre les nœuds existants, il y en aura donc : $k^4 + 2k^3 - k^2 - 2k$. La Figure 4-4 illustre ce problème : un vecteur où k valeurs sont vraies génère d'abord six régularités, puis 45, puis 990, etc. Nous introduisons une contrainte supplémentaire nécessaire pour éviter une telle explosion combinatoire.

4.3.6.2 Contrainte 2 : « Le tout remplace les parties »

Nous renforçons la première contrainte avec celle-ci :

L'apprentissage ne touche que les régularités du plus haut niveau de la hiérarchie.

L'idée derrière ce choix est que les régularités apprises constituent les dimensions d'un nouvel espace de représentation des données, les constituants de ces régularités étant ignorés.

Outre le problème de l'explosion combinatoire, nous justifions également ce choix par un argument de type « anthropomorphique ». Lorsque nous percevons une scène visuelle, nous voyons généralement des objets, et non les « points » qui les composent. Par exemple, lorsque nous voyons un arbre, bien que l'arbre soit « complet » sur notre rétine[38], nous ne percevons pas consciemment chacune de ses feuilles. Lorsque nous regardons une personne, nous ne percevons pas consciemment chaque pore de la peau, ou chaque cheveu, alors que cette

[38] Notons tout de même que la densité d'information varie selon la zone du champ visuel.

information est présente sur la pupille. Seules les caractéristiques de haut niveau sont perçues consciemment, les autres servant de base à leur fabrication. Par exemple, lors de la lecture, les idées lues sont perçues et apprises, les mots également mais de manière moindre, les lettres ne sont peu ou pas perçues consciemment et les points les composants ne le sont pas du tout.

Ce qui est perçu dépend bien sûr de l'expérience : un joueur d'échec expérimenté perçoit certaines configurations de pièces comme un tout, là où un débutant y verrait un ensembles de pièces (Figure 4-5). D'ailleurs, il a été montré ([Gob01]) qu'un joueur d'échecs professionnel est capable de mémoriser en quelques secondes une partie jouée par des professionnels mais incapable de mémoriser une partie générée aléatoirement. Pour un débutant par contre, la mémorisation est la même, que la partie ait été jouée par des professionnels ou générée aléatoirement. Or, il est également connu que la mémoire à court terme est limitée à sept (plus ou moins deux) items (ou « chunks » en anglais) et ne varie que très peu selon les individus. Puisque le joueur professionnel mémorise (presque) parfaitement la configuration d'un échiquier entier (plus de neuf pièces), il doit nécessairement percevoir plusieurs pièces comme un « tout », comme un motif particulier.

D'un point de vue combinatoire, cette simplification est indispensable. Sans regroupements hiérarchiques des perceptions, il faudrait apprendre toutes les combinaisons possibles de tous les traits à tous les niveaux.

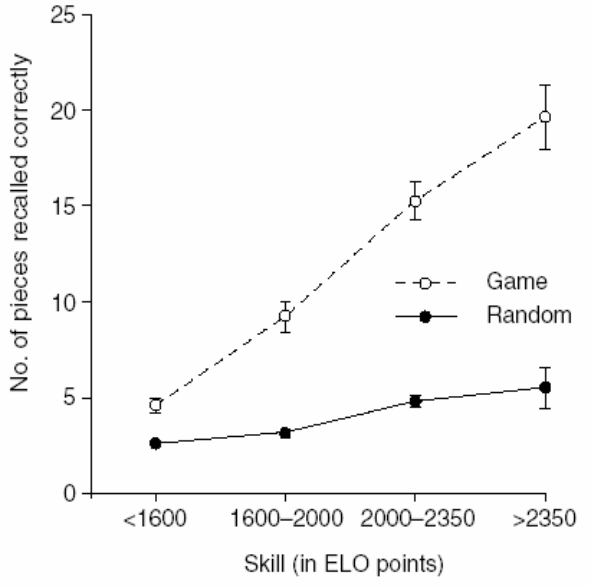


Figure 4-5 Au jeu d'échecs, la capacité de remémoration de la configuration d'une partie en cours est liée au niveau. (Graphique tiré de [Gob01])

De plus, ce choix fait l'hypothèse que l'apprentissage à un niveau inférieur dans la hiérarchie a déjà eu lieu, ce qui est raisonnable si l'apprentissage est suffisamment lent. Comme pour le joueur d'échec, lorsqu'un ensemble de traits atomiques sont combinés pour être perçus comme un tout, l'heure n'est plus à la remise en question et on ne considère plus de nouvelles cooccurrences éventuelles entre traits atomiques. Nous pensons que cette hypothèse est tout à fait raisonnable, tant que les données présentées à l'algorithme d'apprentissage sont générées par les mêmes règles et présentent des régularités semblables, construites à partir des mêmes briques de base. Si ce n'est pas le cas, les données d'un nouveau type seront perçues en fonction des régularités présentes dans les données antérieures (comme un joueur d'échec qui jouerait au Go, en essayant d'appliquer les même concepts), et les nouvelles données seront *déformées* à cause des représentations inadaptées.

4.3.7 Résolution des ambiguïtés

Lors de la « reconstruction » d'un vecteur de traits atomiques en termes de régularités, il se peut que plusieurs reconstructions de niveau hiérarchique identiques soient possibles. Pour illustrer ce problème de « multiples reconstruction », nous présentons, Figure 4-6, une illustration célèbre de double interprétation. A partir de quelques centaines de points noirs ou blancs, l'observateur aboutit à un « symbole »

unique qui peut être soit « jeune fille » soit « vieille dame ». Il semble que l'âge de l'observateur influence le perception : les personnes jeunes tendent à voir la jeune fille alors que des personnes plus âgées tendent à voir la vieille dame. Il s'agit d'un problème d'ambiguïté : plusieurs interprétation sont possibles, mais une seule est autorisée.



Figure 4-6 : Une illustration du problème des multiples reconstruction.

Le même phénomène se produit lors d'ambiguïtés linguistiques : la phrase « Le pilote ferme la porte. » est ambiguë car elle peut être reconstruite de deux manière aboutissant à deux sens différents, à savoir « (Le pilote) (ferme la porte) » et « (Le pilote ferme) (la) (porte) ». Sans contexte particulier, la plupart des gens comprendraient qu'il s'agit d'un pilote fermant une porte, probablement car l'autre sens (en particulier un « pilote ferme ») est beaucoup *moins fréquent*.

Pour revenir à notre problème, si un vecteur peut être interprété de plusieurs manières différentes mais que nous n'en voulons qu'une, il nous faut une méthode de décision. Typiquement, la classification d'un vecteur requiert qu'il soit interprété d'une manière unique.

Il nous semble que lorsque plusieurs reconstructions sont possibles, à un même niveau de hiérarchie, la plus « courante », ou plausible est prioritaire. Une solution pour réduire l'espace des hypothèses est donc, lorsque plusieurs reconstructions sont possibles, de ne garder que la plus fréquente. Nous allons donc ajouter une troisième contrainte à notre biais inductif.

4.3.7.1 Contrainte 3 : Préférer les configurations fréquentes

Lors de l'interprétation, ou recodage, d'un vecteur, si plusieurs alternatives se présentent, alors la configuration la plus fréquente est choisie.

Ce choix est tout à fait en accord avec le *principe de la description de longueur minimum* (voir [Grü05]) (Minimum Description Length Principle ou MDL), introduit par Rissanen en 1978. Cette nouvelle contrainte est toutefois à double tranchant. D'une part, elle permet une certaine *convergence* de l'apprentissage, car les configurations fréquentes étant choisies, leur fréquence en est renforcée. Cette propriété est souhaitable pour effectuer des prédictions (et compresser l'information). D'autre part, cette contrainte induit également une certaine *inertie* à l'apprentissage de « nouveautés », puisque ce qui est perçu est interprété en terme de ce qui est connu et « ce qui est connu » n'est pas nécessairement adapté pour représenter ce qui est nouveau.

Cependant, nous émettons l'hypothèse que cela est une propriété inévitable de tout système d'apprentissage non supervisé, car mesurer précisément la différence entre le connu et l'inconnu requiert la connaissance d'un « repère absolu », qui lui-même nécessite la connaissance de « tout ce qui peut arriver ».

4.3.8 Apprendre à différents niveaux d'abstraction

Nous allons montrer ici qu'il est nécessaire de représenter les exemples d'apprentissage à différents niveaux d'abstraction, car le niveau d'abstraction des concepts appris lors de l'apprentissage supervisé est inconnu. Ces données sont constituées d'un ensemble de *vecteurs* d'apprentissage.

Lorsque sont introduites les régularités de type 'Ou' (disjonctives), il devient possible d'exprimer un vecteur particulier selon plusieurs niveaux d'abstractions, c'est-à-dire de manière spécifique, générique ou intermédiaire. Un vecteur décrit de manière très générique, en n'utilisant que des régularités de type 'Ou' ne serait pas utile. Le niveau le plus spécifique consiste à exprimer le vecteur en termes de régularités de type 'Et', formées uniquement de régularités de type 'Et'. Dans ce cas, le recodage est un isomorphisme dans lequel aucune donnée n'est perdue, les traits atomiques étant simplement recombinés. Lorsque des disjonctions sont introduites dans

l'expression d'un vecteur, le recodage est plus abstrait et certains détails sont perdus. Dans la mesure où il n'est pas possible de savoir quel sera le recodage le plus avantageux pour l'apprentissage futur de concepts (de manière supervisée), nous faisons le choix de conserver les expressions d'un vecteur à tous les niveaux d'abstraction, du plus spécifique, au plus générique. Par conséquent, l'apprentissage « progresse parallèlement » à différents niveaux d'abstraction. Cependant, dans la mesure où une régularité abstraite peut être activée par plus de vecteurs différents qu'une régularité spécifique, les niveaux d'abstractions élevés progressent plus rapidement que les niveaux plus spécifiques.

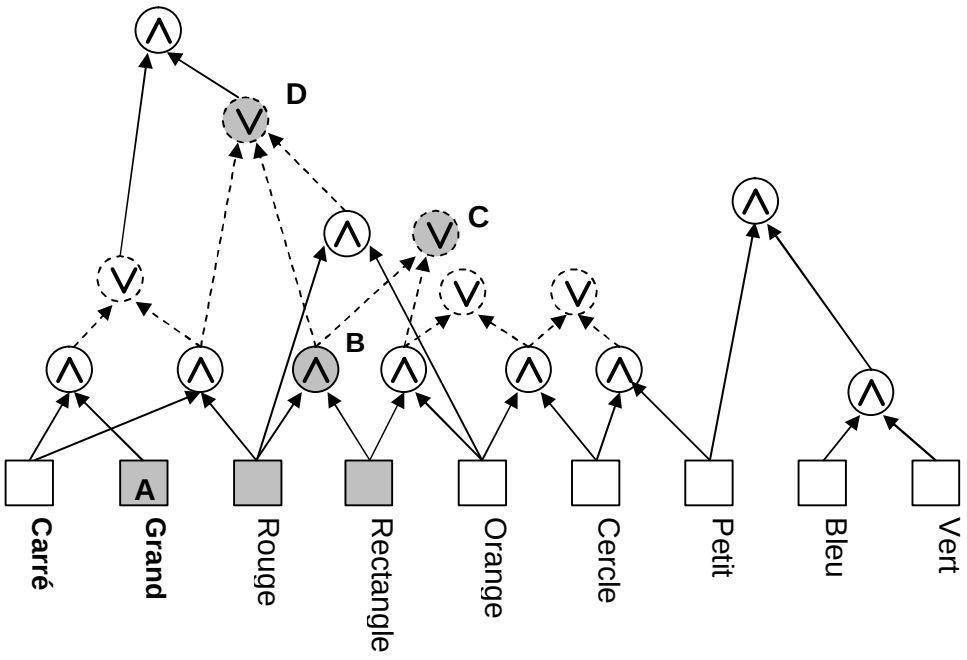


Figure 4-7 : Un vecteur (un grand rectangle rouge) présenté à un réseau de régularités

La Figure 4-7 montre un réseau de régularités en cours d'apprentissage. Ce réseau a appris des régularités sur un ensemble de formes colorées. On présente alors un nouveau vecteur V d'apprentissage : un grand rectangle rouge. Il existe plusieurs manières de représenter ce vecteur en fonction des régularités connues : on peut considérer que V est « un grand, rectangle rouge » ($A \wedge B$), que V est « un carré ou rectangle rouge, grand » ($A \wedge D$) ou que V est « un grand, rectangle rouge ou orange » ($A \wedge C$). Quelle est alors la meilleure représentation ? Il n'est pas possible de le deviner. Il se pourrait en effet que le fait qu'un rectangle soit rouge ou orange

n'ait pas d'importance. Il se pourrait aussi que la couleur rouge soit pertinente mais que la forme soit indifférente, tant qu'il s'agit d'un carré ou d'un rectangle mais pas d'un cercle. Dans ce dernier cas, l'expression $A \wedge D$ est préférable car ce concept générique est plus fréquent que ses instances spécifiques.

Un même vecteur peut donc être perçu à différents niveaux d'abstraction. Selon la manière dont il est perçu, différentes régularités seront activées, et par conséquent, l'apprentissage induit par ce vecteur renforcera les cooccurrences entre régularités activées. Ne sachant pas quel niveau d'abstraction conviendra lors de l'apprentissage supervisé, nous choisissons de conserver tous les niveaux d'abstractions.

4.4 Apprentissage supervisé basé sur des régularités

Après que l'apprentissage non supervisé a eu lieu, un ensemble de régularités a été créé. L'hypothèse est que l'expression d'un vecteur en termes de ces régularités doit être plus pertinente que de l'exprimer sous forme de traits atomiques. La raison est que des vecteurs, recodés sous forme de régularités, sont plus facilement comparable, car leurs similitudes et leurs différences sont mises en évidence. De plus, les caractéristiques importantes d'un vecteur sont accentuées et le « bruit » est ignoré. En effet, le bruit n'étant, par définition, pas régulier, il n'est pas appris lors de l'apprentissage non supervisé, et peut être ignoré lors de l'apprentissage supervisé. En résumé, nous espérons que grâce à l'apprentissage non supervisé, le problème d'apprentissage, à l'origine de type-2, est devenu de type-1, c'est-à-dire que les régularités et les concepts sont statistiquement liés de manière directe. Notre approche de l'apprentissage supervisé est simple, elle se rapproche fortement des techniques habituelles. L'idée est de découvrir des corrélations statistiques « superficielles » entre les traits et les concepts. On ne s'intéresse qu'aux corrélations directes car nous faisons l'hypothèse que la phase non supervisée a réduit la complexité du problème, en « traduisant » les données dans un langage adapté, celui des régularités.

La méthode est donc la suivante : chaque exemple étiqueté est d'abord « traduit » en termes de régularité conjonctives et disjonctives. Les composants d'une régularité conjonctive activée sont ignorés, conformément à notre discussion sur l'aspect

quantitatif. Dans le cas d'une régularité disjonctive, les composants sont conservés, afin de maintenir divers niveaux d'abstraction. Puis les cooccurrences entre concepts et régularités sont comptées. On peut noter que cette méthode permet de ne pas perdre d'information : les composants ignorés d'une régularité conjonctive sont implicitement pris en compte car ils sont impliqués par la présence de la régularité.

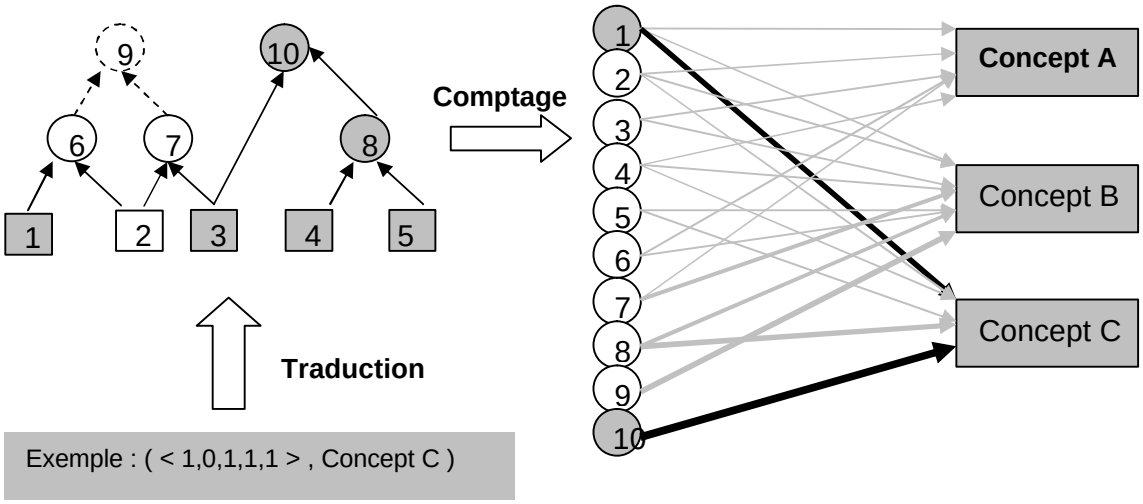


Figure 4-8 : Apprentissage supervisé : les vecteurs sont traduits en termes de régularités, puis les cooccurrences entre les concepts et les régularités sont comptées.

La Figure 4-8 illustre cette approche. L'apprentissage non supervisé a donné lieu à un ensemble de régularités (numérotées de 1 à 10) hiérarchiquement organisées (à gauche). La phase supervisée consiste alors à traduire des vecteurs étiquetés de l'ensemble d'apprentissage en termes de ces régularités, puis à compter les cooccurrences entre l'étiquette (qui correspond au concept) et les régularités qui apparaissent lorsque le vecteur étiqueté est présenté au réseau. Sur le schéma, l'épaisseur des liens entre concepts et régularités symbolise la pertinence d'une régularité pour un concept.

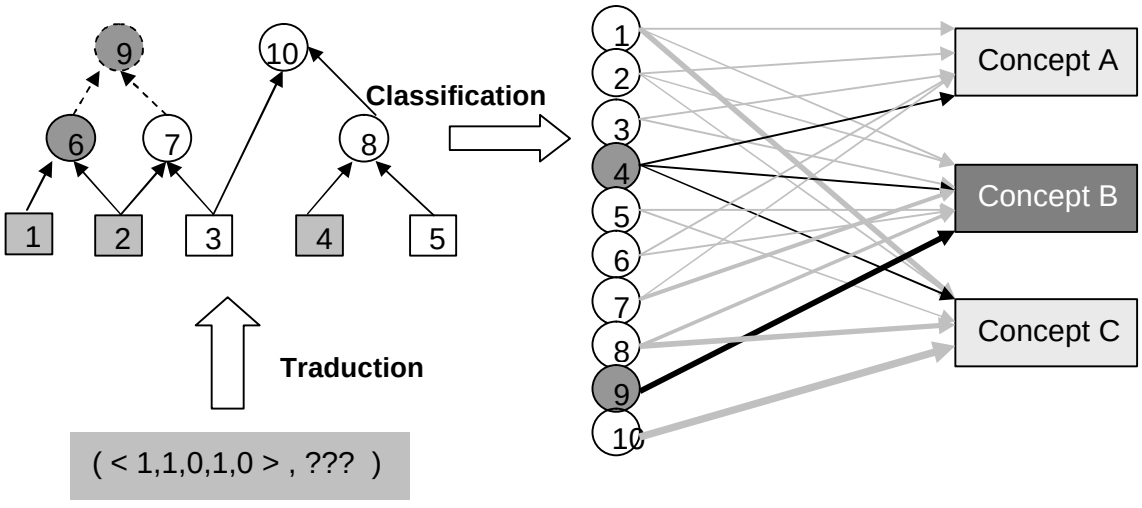
Pour qu'une régularité soit discriminante par rapport à un concept, il faut d'une part que la régularité soit fréquente parmi les instances du concept et d'autre part que cette régularité ne soit pas fréquente pour les autres concepts[39].

4.5 Classification

La dernière étape du processus est celle de la classification, qui consiste à déterminer quels sont les concepts dont sont instances les vecteurs non étiquetés présentés. Dans le cas où plus d'un concept a été appris, il existe deux alternatives pour effectuer la classification : la première consiste à prendre un vecteur en entrée et donner un concept en sortie. La deuxième alternative, plus « flexible » donne comme sortie l'ensemble des concepts, pondérés selon leur « degrés d'appartenance » au concepts. Cette deuxième solution, celle que nous retenons, autorise la mise en œuvre en aval, de processus complémentaires, visant par exemple à désambiguïser les résultats obtenus.

4.5.1 Classification stricte

Le principe de la classification est simple : le vecteur inconnu à classer est traduit en termes de régularités, puis chaque régularité détectée active les concepts proportionnellement au nombre de cooccurrences entre concept et régularité calculés lors de l'apprentissage supervisé.



[39] Cela se rapproche donc fortement du *tf.idf* ([Sal88]) utilisé en recherche d'information. Dans cette analogie, les régularités équivalent aux termes et les concepts équivalent aux documents.

Figure 4-9 : Classification d'un vecteur : le vecteur est traduit en termes de régularités, puis les régularités « activent » les concepts.

Comme lors de la classification supervisée, lorsqu'une régularité de type 'Et' est détectée, ses composants sont ignorés. Par exemple, dans la Figure 4-9, les régularités atomiques numérotés 1 et 2 sont ignorés lors de la classification car elles composent une régularité conjonctive activée : 6. On peut remarquer que si 1, 2 et 6 étaient pris en compte, 1 et 2 seraient « comptés » deux fois : leur poids relatif dans la classification, relativement au trait 4 par exemple, serait exagéré. Pour éviter cette redondance, seule la régularité 'Et' influe sur la classification, et non les constituants de celle-ci.

Considérons maintenant les régularités de type 'Ou' (disjonctives), par exemple la régularité numérotée 9 dans la Figure 4-9. La régularité 9 représente « le générique », ses constituants (ici, 6 et 7) activés (ici 6) constituent les spécifiques. Si l'on tient compte du générique **et** du spécifique lors de la classification, cela pose un problème de redondance qui modifie le poids relatif des régularités. C'est pour cette raison que, lors de la classification, seule la régularité la plus pertinente est prise en compte, c'est-à-dire soit la régularité générique, soit une des spécifiques. La pertinence d'une régularité est liée à l'information qu'elle apporte vis-à-vis de la classification. Une régularité liée de manière uniforme à tous les concepts n'apporte aucune information, alors qu'au contraire, une régularité liée à un unique concept est très informative. Par exemple, la régularité numérotée 9 dans la Figure 4-9 a pour composant activé la régularité 6. On peut donc considérer l'une des deux pour la classification. Mais puisque la régularité générique 9 est plus pertinente que la régularité 6 (par rapport aux trois concepts A, B et C), c'est celle qui est choisie pour la classification.

4.5.2 Classification *souple*

Tout ce que nous avons présenté jusqu'à maintenant, à savoir l'apprentissage non supervisé de régularités, l'apprentissage supervisé de concepts et la classification de vecteurs inconnus est en réalité très strict. Une régularité de type 'Et' n'est activée que si ses deux composants le sont aussi. Une régularité formée d'une multitude de régularités de type 'Et' ne sera pas activée si un seul des composants ne l'est pas.

Cela ne semble pas très réaliste dans la mesure où les données réelles comportent toujours une part de bruit, d'erreurs ou de données manquantes. Il nous faut donc introduire une certaine « souplesse » quelque part dans le processus. Il existe deux alternatives : assouplir l'apprentissage ou assouplir la classification.

Rendre plus souple signifie « accepter les approximations », par exemple, considérer qu'une régularité est présente même si certains de ses composants sont absents. Or, pour **une** régularité, il existe **beaucoup** d'approximations. Donc, si nous acceptions les approximations lors de l'apprentissage, il faudrait tenir compte, lorsqu'un vecteur est présenté, de toutes les approximations possibles de celui-ci : l'apprentissage des cooccurrences devrait être fait sur les régularités activées par le vecteur, mais également sur les régularités activées par toutes les approximations du vecteur. Cela nous conduirait immanquablement à une explosion combinatoire.

L'autre alternative consiste à rendre la classification plus souple, et heureusement, cela est beaucoup plus réaliste. L'idée est la même que pour la classification stricte, à la différence suivante : au lieu de considérer une régularité comme activée ou non activée (approche binaire et donc stricte), nous allons parler de *degrés d'activation*. La Figure 4-10 montre un exemple de classification souple où des régularités présentes *partiellement* influencent tout de même la classification. Graphiquement, nous représenterons l'activation des régularités par des niveaux de gris différents (noir : activation maximum, blanc : aucune activation).

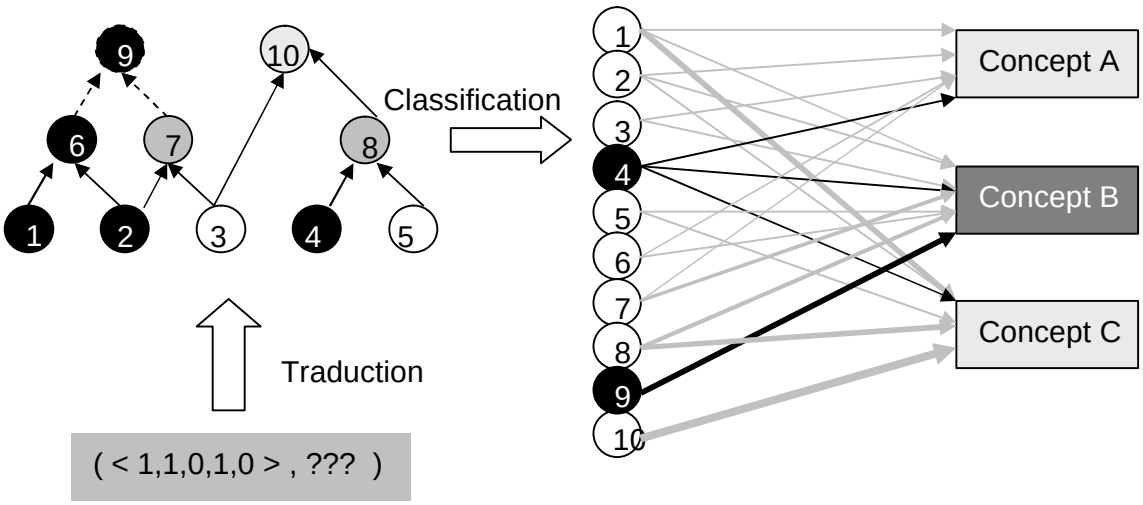


Figure 4-10 : Classification souple : les régularités sont toutes activées, mais à des degrés différents.

Il s'agit maintenant de définir ce qu'est un *degré d'activation* et comment le calculer. Tout d'abord, nous définissons un degré d'activation comme un réel compris entre 0 (la régularité n'est pas du tout présente) et 1 (la régularité est présente au sens strict).

4.5.2.1 Degré d'activation : type 'Et' :

Pour les régularités de type 'Et', nous posons que le degré d'activation est égal à la moyenne des degrés d'activation des constituants pondérés par le nombre de traits atomiques contenu dans l'ensemble de leurs descendants. Dans le premier exemple de la Figure 4-11, le degré d'activation du nœud A est égal à 0.5 car ses descendants atomiques sont au nombre de deux et un seul est activé, on a donc $\frac{1\times 1+1\times 0}{1+1}=0.5$. Dans le second exemple, le degré d'activation du nœud A est cette fois égal à $\frac{2\times 1+1\times 1}{2\times 2+1\times 1}=\frac{3}{5}=0.6$. Les exemples qui suivent illustrent le processus d'activation dans des réseaux plus complexes.

Cette pondération par le nombre de constituants atomiques vise à maintenir l'importance relative des éléments d'une conjonction. Pour comprendre intuitivement cette pondération, imaginons une régularité conjonctive C, constituée de M et b. M signifie « Maison », b signifie « bleu » et C signifie donc « Maison bleue ». Les constituants ne sont pas équivalents, ils n'ont pas le même poids : M est construit à partir d'un très grand nombre de traits atomiques et b n'en comporte qu'un. Si M n'est pas activé mais que b l'est (quelque chose bleu), a-t-on « presque » une maison bleue ? La réponse est non. Dans le cas inverse toutefois, M étant activé mais pas b, la similarité entre « Maison bleue » et « Maison non bleue » est beaucoup plus forte. L'hypothèse faite ici est donc que le poids d'une régularité est proportionnel au nombre de traits atomiques impliqués dans sa représentation.

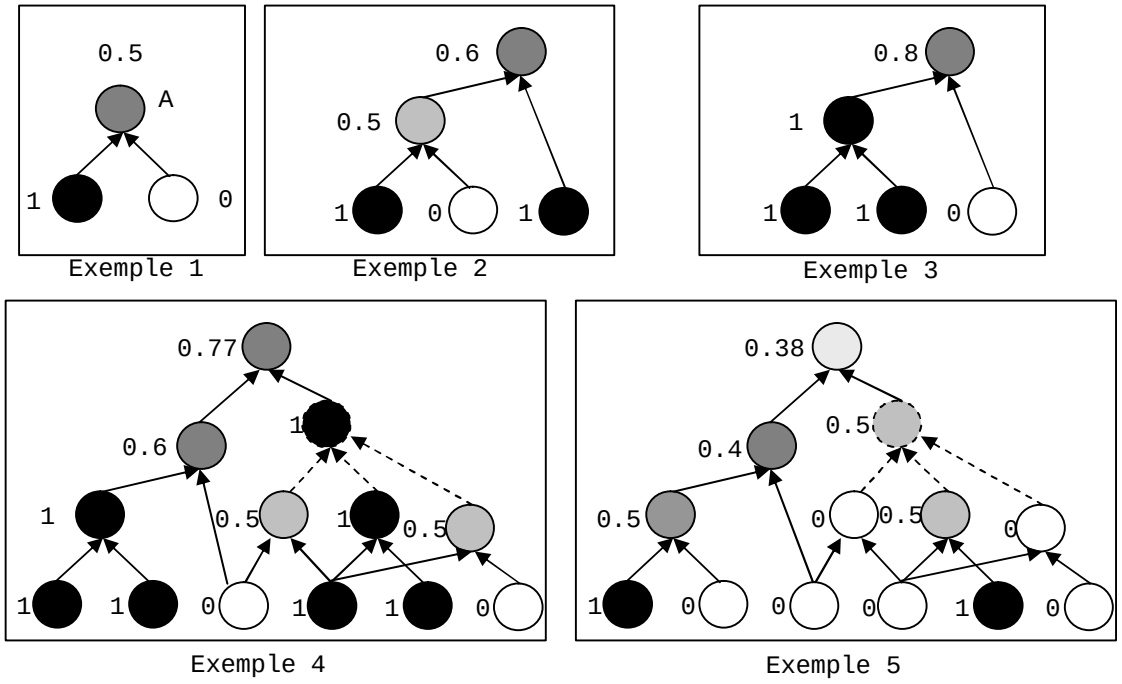


Figure 4-11 : Degrés d'activation des régularités selon différentes configurations.

4.5.2.2 Degré d'activation : type 'Ou' :

En ce qui concerne les régularités de type 'Ou', la même méthode n'est pas applicable. Au lieu de prendre en compte la moyenne des degrés d'activation des constituants, nous allons utiliser le constituant dont le degré d'activation est maximum. Les exemples 4 et 5 de la Figure 4-11 illustrent cette règle.

4.5.2.3 Degrés d'activation & Logique Floue

Le rapport entre notre classification stricte et notre classification souple s'apparente fortement au rapport entre la logique classique (clauses de Horn, logique des prédicats) et la logique floue. La logique floue est issue de la théorie des ensembles flous, initiée par Zadeh en 1965 [Zad65]. Le principe de base de la théorie des ensembles flous est que l'appartenance d'un objet à un ensemble n'est pas binaire : un objet est inclus dans un ensemble, avec *un certain degré d'appartenance*. Notons que le degré d'appartenance à un ensemble est très différent de la *probabilité*

d'appartenance à cet ensemble[40]. Lorsque les objets en question sont des propositions, les degrés d'appartenance deviennent des *degrés de vérité* : une proposition n'est plus vrai ou fausse, mais plus ou moins vrai.

C'est exactement ce qui se passe avec nos régularités : « La régularité R est présente dans le vecteur V » est une proposition vrai ou fausse dans le cadre de notre classification stricte, et une proposition admettant des *degrés de vérité* dans le cadre de notre classification souple. Puisque les régularités sont des assemblages hiérarchiques d'autres régularités, ayant elles-mêmes leurs degrés de vérités (que nous avons appelé *degré d'activation*), nous avons dû trouver des méthodes permettant de calculer un degré de vérité à partir d'autres degrés de vérité.

De la même manière, la logique floue, qui découle de la théorie des ensembles flous, permet d'effectuer des calculs logiques sur des valeurs continues, et non plus binaires. Des règles remplacent les tables de vérités dans le calcul des conjonctions, disjonctions, négations ou implications. En particulier, les règles standards de calcul des conjonctions et disjonctions sont les suivantes :

- Conjonction : $a \wedge b = \min(V(a), V(b))$
- Disjonction : $a \vee b = \max(V(a), V(b))$

Où V(x) représente le degré de vérité de x, compris entre 0 et 1.

La règle de calcul des disjonctions est similaire à la nôtre. Par contre, celle permettant de calculer le degré de vérité des conjonctions est très différente. Remarquons tout d'abord que si nous avions appliqué la règle ci-dessus, la classification n'en n'aurait pas été assouplie pour autant car nos traits atomiques étant binaires, si un '0' apparaissait il serait systématiquement transmis aux régularités hiérarchiquement supérieures, car '0' est toujours le minimum.

Remarquons ensuite que la règle de calcul des conjonctions ci-dessus paraît faire l'hypothèse que les constituants sont décorrélés les uns des autres. Lorsque des

[40] Un exemple très parlant, tiré de [Bez93] est le suivant : soit P l'ensemble flou contenant toutes les boissons potables. On dispose de deux boissons : A, qui a un degré d'appartenance à P de 0.9 et B qui a une probabilité d'appartenance à P de 0.9. La soif nous pousse à choisir l'une des deux. Laquelle ? Je choisirais A, qui est dans tous les cas « quasi potable », plutôt que B qui est très probablement « totalement potable » mais peut-être mortelle !

constituants ne sont pas corrélés, ils ne peuvent se *compenser mutuellement*, par conséquent, le choix du minimum paraît judicieux. Par exemple, si « Voyage_possible = passeport_valide ET billet_avion ET vaccins_à_jour », les constituants n'étant ni corrélés, ni du même type, ils ne peuvent pas se compenser mutuellement et si l'un d'eux est « faux », la conjonction est fausse. Par contre, si l'on considère l'exemple : « Octogone = segment1 ET segment2 ET … ET segment8 », les constituants sont du même type, de plus ils ne sont pas indépendants. Par conséquent, dans ce cas, le choix du minimum ne paraît pas judicieux, car le fait qu'un segment soit « peu présent » peut être compensé par la présence des autres segments.

Il nous semble donc que plus les composants d'une conjonction sont corrélés, moins la règle du minimum de la logique floue nous semble pertinente. Plus les composants sont corrélés, plus ils peuvent se « compenser » mutuellement, et plus la règle de la moyenne que nous avons adopté nous semble pertinente. Ajoutons à cela que les régularités conjonctives tendent, par construction, à avoir des constituants corrélés. En effet, s'ils ne l'étaient pas (traits indépendants), la probabilité de leur cooccurrence serait faible et leur cooccurrence ne constituerait pas une *régularité de l'ensemble d'apprentissage*.

Dernière différence : en logique floue classique, les constituants d'une conjonction ou d'une disjonction ont le même poids. Cela peut induire des résultats surprenants et non souhaité lorsque des constituants d'importances très différentes sont mélangés. Nous faisons le choix de pondérer les constituants par le nombre de traits atomiques qui les constituent. L'importance d'une régularité est proportionnelle à sa fréquence dans l'ensemble d'apprentissage. Une régularité fréquente sert davantage à la construction de régularités qu'une régularité plus rare. Par conséquent, l'importance et la taille (en termes de traits atomiques) d'une régularité sont fortement corrélées.

4.6 Intérêt pour la personnalisation

La vision de l'apprentissage que nous venons de présenter s'intègre parfaitement dans le cadre de la personnalisation. Elle possède en outre de bonnes propriétés en tant qu'algorithme d'apprentissage.

Tout d'abord, la phase d'apprentissage non supervisé s'adapte à la collection d'images de l'utilisateur, indépendamment de tout concept. L'algorithme d'apprentissage apprend à représenter ces images en termes de régularités statistiques présentes dans la collection. Par conséquent, lors de la phase supervisée, qui requiert l'intervention de l'utilisateur, le système « connaît déjà la collection » et, idéalement, l'apprentissage supervisé est de type-1 (donc rapide et nécessitant peu d'exemples d'apprentissage, ce qui lui confère une bonne réactivité).

Les traits construits durant la phase d'apprentissage non supervisée sont, par définition, adaptés à la collection de l'utilisateur. Ces traits sont donc plus susceptibles (par rapport aux traits génériques habituels) de permettre l'apprentissage de concepts propres à l'utilisateur et à sa collection. Habituellement, les traits utilisés dans les SRIC ne sont adaptés qu'à certains concepts car ils ont été conçus indépendamment de la collection d'images.

De plus, le fait d'apprendre des régularités à plusieurs niveaux d'abstraction, en parallèle, permet l'apprentissage supervisé de concepts dont le niveau d'abstraction varie de « très spécifique » à « très générique ».

Une approche basée sur la notion de régularité est résistante au bruit (provenant d'erreurs de l'utilisateur par exemple) dans la mesure où un exemple mal étiqueté ne possède normalement pas les mêmes régularités qu'un ensemble d'exemples, instances d'un même concept. Or, l'apprentissage reliant les régularités les plus fréquentes à un concept, l'influence du bruit est minimisée car par définition, le bruit n'est pas *régulier*.

Finalement, il faut noter que l'apprentissage non supervisé, le plus coûteux en temps de calcul (traduction + mise à jour des cooccurrences), ne requiert pas l'attention de l'utilisateur. Ce déplacement de la charge de travail vers l'apprentissage non supervisé est en faveur de l'apprentissage supervisé et de la classification, qui s'en trouvent considérablement allégés. Les processus qui requièrent l'attention de l'utilisateur sont traités plus rapidement que si *tout* l'apprentissage était effectué de manière supervisé.

4.7 Résumé

L'apprentissage, automatique ou biologique, est possible car les données (un ensemble de traits atomiques) sur lesquels il s'appuie ne sont pas aléatoires, c'est-à-dire qu'il existe des relations entre traits atomiques. Théoriquement, les relations possibles entre ces traits ne sont pas limitées, mais un algorithme d'apprentissage nécessite une représentation interne de ses hypothèses qui est nécessairement limitée. Nous limitons donc l'ensemble des relations possibles à celles détectables par observations de cooccurrences et représentables sous forme CNF.

Nous utilisons également la notion de *régularité disjonctive* : lorsqu'il existe un constituant commun à plusieurs conjonctions, celui-ci est appelé *contexte* et les éléments liés par conjonction à ce contexte sont considérés comme *similaires (*dans ce contexte uniquement). La présence d'un contexte *commun* d'une part, et de constituants *variables* d'autre part, induit dans notre modèle la notion *d'abstraction et de similarité*. Nous modélisons l'abstraction (et donc la notion de similarité) sous la forme de disjonctions : la disjonction constitue la représentation *générique*, ses constituants forment l'ensemble de ses instances *spécifiques*. Apprendre un ensemble de *similarités locales* permet de combiner *abstraction* (grâce à la notion de similarité) et *précision* (grâce à la localité, qui évite de sur-généraliser).

L'ensemble des conjonctions et des disjonctions apprises à partir des données sont appelées *régularités*. Les régularités sont apprises *hiérarchiquement*, c'est-à-dire qu'une régularité peut servir de constituant à une autre régularité.

Les régularités sont apprises de manière non supervisée. Ce choix est motivé d'une part par des arguments biologiques : chez les êtres vivants dotés de capacités d'apprentissage, l'apprentissage non supervisé précède un éventuel apprentissage supervisé. D'autre part, dans un problème d'apprentissage de type-2 (i.e. les problèmes intéressants et difficiles), les relations entre traits atomiques et concepts sont *statistiquement invisibles*, ce qui élimine toute chance d'utiliser une méthode basée sur une descente de gradient. Or, les techniques d'apprentissages supervisées utilisent d'une manière ou d'une autre une forme de descente de gradient pour guider la recherche d'une hypothèse satisfaisante dans un espace d'hypothèses généralement très grand. La solution retenue habituellement consiste à

remplacer les traits atomiques (par exemple les pixels des images) par des traits (comme les histogrammes de couleurs ou des détecteurs de textures) censés augmenter les corrélations statistiques entre entrées (traits) et sorties (concepts). Malheureusement, cette solution élimine une grande part de l'information initiale et n'est pas généralisable car un « jeu de traits » particulier ne peut convenir à tous les problèmes.

Le *biais inductif* de notre approche est défini par les contraintes suivantes :

- L'apprentissage porte sur les régularités présentes dans le flux de données sous forme de cooccurrences.
- Les régularités sont apprises par agrégation de régularités connues.
- L'apprentissage de nouvelles régularités porte sur une représentation des données traduite en termes des régularités activées du plus haut niveau de la hiérarchie.
- Les données sont traduites en termes des régularités les plus fréquentes.

L'apprentissage supervisé qui succède à celui non supervisé est très simple, les vecteurs étiquetés sont traduits, ou recodés, sous forme de régularités. Puis les cooccurrences entre les régularités et les concepts concernés sont comptabilisées.

La classification consiste alors à déterminer quels sont les concepts les plus « stimulés » par les régularités présentes dans un vecteur non étiqueté. Nous proposons une classification stricte, où les régularités présentes dans un vecteur doivent être identiques à celles apprises, ainsi qu'une classification « souple », où les régularités contenues dans un vecteur peuvent être partielles. Les opérateurs utilisés pour calculer le degré d'activation d'une régularité s'apparentent à ceux utilisés en logique floue et nous en avons discuté les différences.

Par rapport aux critères utilisés dans notre comparatif des méthodes d'apprentissage du chapitre 3 (voir Tableau 3-2), notre approche s'inscrit de la manière suivante. L'apprentissage est **non supervisé** et **supervisé**. La résistance au **bruit** est obtenue par l'utilisation de régularités (une régularité étant par définition l'opposé du bruit). L'apprentissage non supervisé étant constructif, il est aussi **stable**, car

l’apprentissage de données différentes résulte en de nouvelles régularités et ne modifie pas ce qui a déjà été appris. D’un point de vue de l’utilisateur, cette approche devrait se révéler **réactive**, car l’apprentissage non supervisé, qui est la phase nécessitant le plus de calculs, peut être effectuée sans recours à ce dernier. Les problèmes d’apprentissage de **type-1** sont gérés par la partie supervisée de l’approche, qui associe des traits à des concepts. Les problèmes de **type-2** sont gérés d’une part par l’apprentissage non supervisé, qui recode les données *en fonction des données*, et d’autre part l’apprentissage supervisé, qui associe ces représentations apprises aux concepts cibles.

Chapitre 5

Une instanciation de l'approche

Nous avons montré, quoique non encore démontré, que l'apprentissage supervisé seul ne suffit pas pour la résolution de problèmes d'apprentissage complexe. La solution que nous avons évoquée repose sur l'extraction non supervisée, constructive et hiérarchique de régularités, avant l'apprentissage supervisé de concepts. Ces régularités représentent les nouvelles dimensions d'un espace dans lequel l'apprentissage supervisé de concepts est supposé être facilité. Nous allons maintenant montrer comment implanter ce modèle d'apprentissage, en focalisant sur la partie non supervisée. La partie supervisée sera traitée plus succinctement car elle se résume, comme bon nombre d'algorithmes existants, à un traitement statistique superficiel.

Dans cette partie, nous commençons par définir une *structure de travail*, qui inclut toutes les données nécessaires à l'apprentissage, y compris les éléments temporaires ne servant que de structures intermédiaires pour l'apprentissage des régularités. Ensuite, nous décrivons la première phase de l'apprentissage non supervisé, appelée *phase de propagation*, dont le but est la traduction des vecteurs d'apprentissage en termes des régularités connues. Vient ensuite la seconde phase, appelée *phase de renforcement* qui renforce les cooccurrences entre régularités détectées, en créant éventuellement de nouvelles régularités.

Nous traiterons ensuite de la phase supervisée de l'apprentissage avant de spécifier le processus de classification.

5.1 Structure de travail

Nous décrivons ici les différentes structures de données, ainsi que la manière dont elles s'agencent.

5.1.1 Aperçu

Les régularités apprises sont représentées dans un réseau de forme arborescente. Elles sont apprises itérativement par présentations successives d'exemples, représentés par des vecteurs de traits atomiques binaires. La couche d'entrée d'un réseau d'apprentissage de régularités est formée de nœuds correspondant aux dimensions du vecteur d'entrée. Graphiquement, nous représentons les traits atomiques par des carrés et les régularités par des cercles.

Lorsqu'un vecteur est « présenté » au réseau, il **active** les nœuds de la couche d'entrée qui correspondent à ses dimensions égales à « vrai ». **Un** nœud **activé** sera **grisé**, un nœud non activé restera blanc. La sémantique liée à l'activation est la suivante :

- Un noeud activé représente une régularité, qui est présente dans le vecteur soumis à la couche d'entrée.
- Un noeud non activé représente une régularité qui n'est pas présente dans le vecteur présenté au réseau.

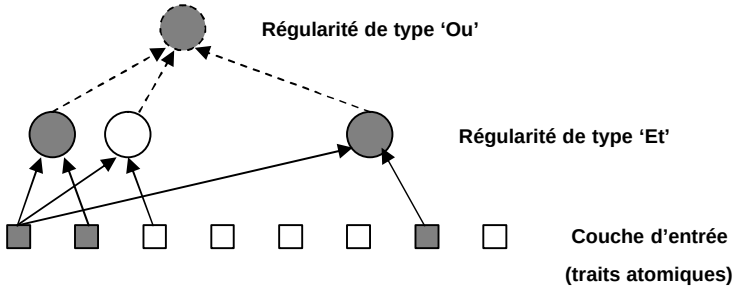


Figure 5-1 Les principaux éléments

La Figure 5-1 montre les principaux éléments : les traits atomiques qui forment la couche d'entrée (indiqués par des carrés), des régularités de type 'Et', une régularité de type 'Ou', ainsi que les connexions entre les divers éléments. Dans l'exemple, six nœuds sont activés, dont trois dans la couche d'entrée.

Un nœud représentant une régularité de type 'Et' possède toujours deux entrées (flèches arrivantes) et éventuellement des sorties (flèches sortantes). Une régularité

de type ‘Ou’ possède au moins deux entrées et éventuellement des sorties. Les flèches pointant vers un nœud ‘Ou’ sont, par convention, en pointillés.

Il existe un lien bidirectionnel entre chaque nœud et tous les autres nœuds, chaque lien stocke le nombre de cooccurrences entre deux nœuds. Pour des raisons de clarté, ces liens n’apparaîtront pas sur les figures mais nous les mentionnerons lorsque ce sera nécessaire.

Les données d’apprentissage, comme des régions d’images, sont représentées par des vecteurs de réels compris entre 0 et 1. Toutefois, ces valeurs doivent être converties en booléens car les entrées du réseau sont booléennes. Pour cela nous utiliserons des seuils d’activations, pouvant être différents selon que le réseau apprend ou reconnaît.

5.1.2 Les différents éléments

Un réseau est donc un arbre formé de nœuds et de connexions entre ces nœuds. Il existe trois types de nœuds (atomiques, nœuds ‘Et’, nœuds ‘Ou’) et deux types de connexions : les liens et les connexions de type entrée/sortie. Nous décrivons maintenant ces éléments.

Un noeud représente une régularité. Soit N l’ensemble des nœuds $N = \{n_i\}$. Les premiers nœuds ($i \in [1..k]$) sont les nœuds atomiques de la couche d’entrée. Ensuite, les nœuds sont numérotés par ordre de création. Par exemple, le premier noeud créé lors de l’apprentissage sera le noeud de rang k+1.

Les nœuds étant composés d’autres nœuds, nous définissons l’ensemble de compositions suivant :

$Comp \subseteq N \times N$

$(n_1, n_2) \in Comp$ Signifie que n2 est descendant de n1.

Nous définissons également l’ensemble des compositions directes :

$CompD \subseteq N \times N$

$(n_1, n_2) \in CompD$ Signifie que n2 est fils de n1.

5.1.2.1 Type

Les nœuds pouvant être de type atomique, ‘Et’ ou ‘Ou’, nous définissons une fonction « Type » ayant comme résultat le type d’un noeud donné :

$Type : N \rightarrow \{'Et', 'Ou', 'Ato'\}$

Tous les nœuds de la couche d’entrée sont des nœuds atomiques.

5.1.2.2 Activation

Un noeud, quel que soit son type, est soit activé soit inactivé. Nous utiliserons la fonction suivante pour représenter l’activation d’un noeud :

$Activé : N \rightarrow \{vrai, faux\}$

L’activation d’un noeud traduit la présence de la régularité associée à ce noeud dans le vecteur présenté à la couche d’entrée.

5.1.2.3 Contraintes

Soit la fonction « Fils » suivante, permettant de connaître les fils directs d’un noeud donné :

$Fils : N \rightarrow N^k$ où k est le nombre de fils

$Fils(n) = \{f_i | (n, f_i) \in CompD\}$

Nous ajoutons les contraintes suivantes :

$(Type(n) = 'Et') \Rightarrow |Fils(n) = 2|$

$(Type(n) = 'Ou') \Rightarrow |Fils(n) \geq 2|$

$$(Type(n) = 'Ato') \Rightarrow |Fils(n) = 0|$$

En d'autres termes : les nœuds atomiques n'ont pas de fils car ils ne sont pas constitués d'autres régularités, les nœuds 'Et' ont deux fils et les nœuds 'Ou' en ont au moins deux.

Le nombre de fils d'un nœud conjonctif est limité à deux afin de conserver le plus grand pouvoir d'expression possible : des conjonctions n-aires ne seraient pas à même de refléter précisément l'importance relative de chaque constituant.

Un noeud ne peut être composé que de nœuds « plus anciens » :

$$(n_i, n_j) \in Comp \Rightarrow i > j$$

5.1.2.4 Blocage

Au chapitre précédent, nous avons défini des contraintes qui ensemble forment notre biais inductif. Ces contraintes se concrétisent ici sous la forme de *règles de blocage* que nous détaillons plus loin. Nous utiliserons la fonction suivante pour représenter le blocage d'un noeud :

$$Bloqué : N \rightarrow \{vrai, faux\}$$

5.1.2.5 Liens de cooccurrence

Un lien est une connexion non orientée entre deux nœuds (non représenté dans les figures). La seule information contenue dans un lien est une « charge ». Le rôle du lien est de représenter la cooccurrence entre deux nœuds. Nous définissons ces liens de cooccurrence de la manière suivante :

$$Coo : N \times N \rightarrow N$$

Si l'on a $(n_i, n_j) \rightarrow c$, alors il existe un lien de cooccurrence entre le nœud n_i et le nœud n_j dont la valeur est c.

Lorsque la charge portée par un lien surpasse un seuil, un nouveau nœud de type 'Et' est créé pour représenter le fait qu'une cooccurrence fréquente est une régularité.

5.1.2.6 Entrées & sorties

Deux nœuds peuvent être reliés par une connexion de type entrée/sortie : la sortie d'un nœud est l'entrée de l'autre. C'est *via* ce type de connexion que les signaux d'activations sont propagés. Ces connexions sont représentées formellement par les éléments de l'ensemble CompD(n1, n2) : n1 étant le père de n2, n1 possède une sortie connectée à une entrée de n2.

5.2 Apprentissage non supervisé

Le réseau est soumis à un flux de vecteurs. Les nœuds de la couche d'entrée sont activés ou non selon les traits atomiques que présente le vecteur exemple. Les régularités apprises dans le passé, et présentes dans le vecteur sont activées (en gris sur la Figure 5-2). Néanmoins, seules certaines de ces régularités (en noir sur la Figure 5-2) sont sélectionnées pour l'apprentissage. Les nœuds noirs représentent la *traduction* du vecteur.

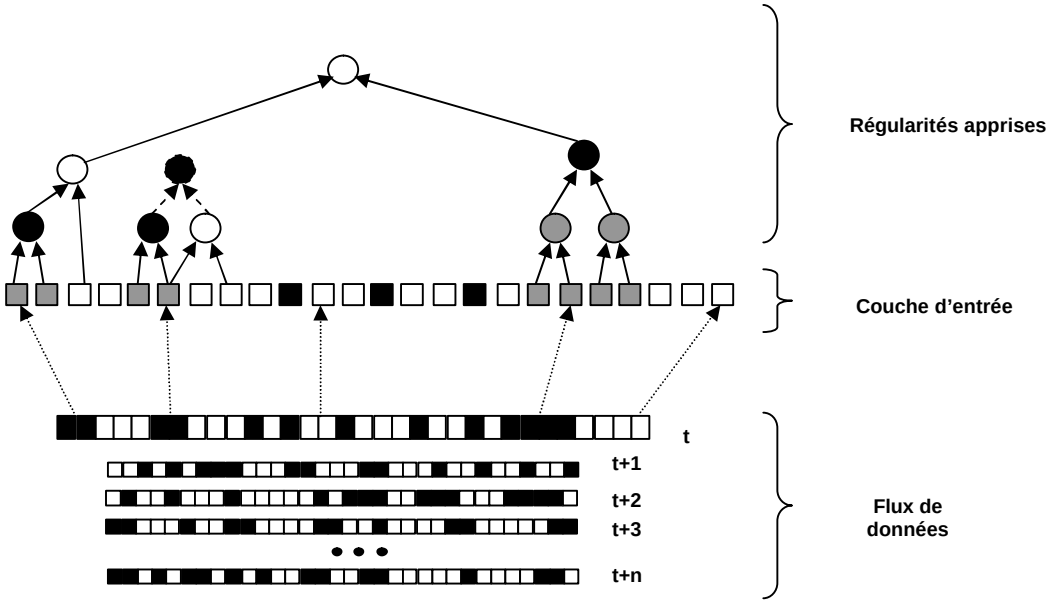


Figure 5-2 : L'apprentissage non supervisé : les régularités sont apprises à partir d'un flux de données.

Les cooccurrences entre chaque paire de nœuds activés non bloqués (les nœuds noirs) sont alors mise à jour et les cooccurrences dépassant un certain seuil donnent lieu à la création d'un nouveau nœud. Les cooccurrences entre nœuds d'un même sous-arbre ne sont pas prises en compte.

5.2.1 Etat initial

L'état initial est un réseau réduit à sa couche d'entrée. Chaque nœud de la couche d'entrée correspond à un trait atomique, la taille de la couche d'entrée est donc égale à celle des vecteurs. Les « compteurs » de cooccurrences entre traits atomiques sont initialisés à zéro :

$$\forall n1, n2 \in N : (n1, n2, 0) \in Coo$$

$$\forall n \in N : Type(n) = Ato$$

Si aucun apprentissage n'avait lieu, chaque exemple serait représenté comme un point dans un espace à n dimensions (où n est le nombre de traits atomiques) : c'est-à-dire de la même manière que dans l'algorithme des k plus proches voisins par exemple.

5.2.2 Apprentissage

L'apprentissage se déroule en deux phases : la propagation du vecteur d'entrée, puis la mise à jour des compteurs de cooccurrence. La première phase a pour but de recoder le vecteur en termes des régularités connues. La deuxième phase met à jour les cooccurrences entre certaines des régularités détectées durant la première phase et crée éventuellement de nouvelles régularités.

5.2.2.1 Phase de propagation

Le vecteur d'entrée V, composé de n réels appartenant à l'intervalle [0,1] est présenté à la couche d'entrée. Si V_i est supérieur à un seuil s, l'entrée correspondante est activée. Le calcul de la valeur du seuil s est pour le moment expérimental.

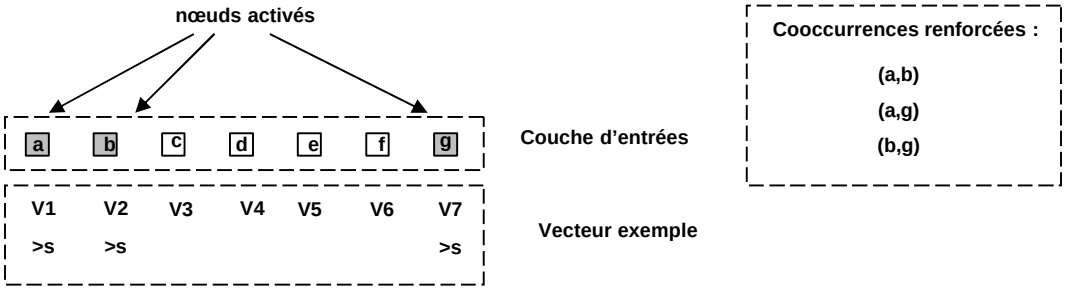


Figure 5-3 : Un vecteur est présenté au réseau initial ne comportant que sa couche d'entrées.

Si, comme à l'état initial, le réseau est réduit à la couche d'entrée (voir Figure 5-3), les valeurs du vecteur supérieures au seuil n'activent que les nœuds de la couche d'entrée :

$$V_i \geq s \Rightarrow Activé(n_i)$$

Dans le cas où le réseau ne se réduit pas à sa couche d'entrée initiale (c'est-à-dire que certain nœuds ont des sorties et sont connectés à d'autres nœuds), les nœuds activés propagent leur activation *via* leurs sorties. Ce signal d'activation est reçu par des nœuds qui s'activeront ou non, selon leur type et selon leurs autres entrées :

- Nœud 'Et' : Il s'active lorsqu'il reçoit un signal de ses deux entrées :

$$(ni, n_j) \in CompD \wedge (n_i, n_k) \in CompD \wedge Activé(n_j) \wedge Activé(n_k) \wedge Type(n_i) = Et \Rightarrow Activé(n_i)$$

Nœud 'Ou' : Il s'active dès la réception d'un signal :

$$((n_i, n_j) \in CompD \wedge ... \wedge (n_i, n_p) \in CompD) \wedge (Activé(n_j) \vee ... \vee Activé(n_p)) \wedge Type(n_i) = Ou \Rightarrow Activé(n_i)$$

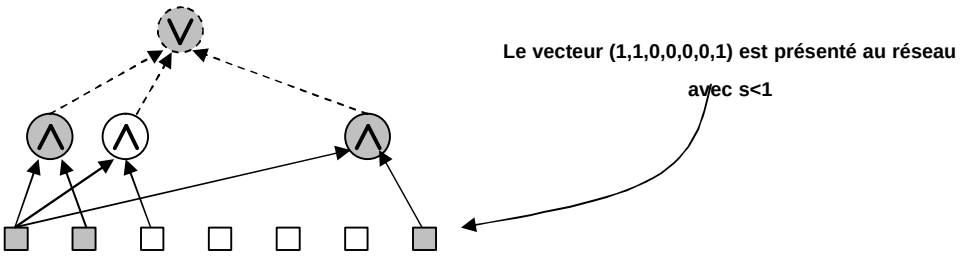


Figure 5-4 : Les nœuds activés de la couche d'entrée propagent leur signal et activent d'autres nœuds. Les nœuds activés grisés représentent des régularités de l'ensemble d'apprentissage présentes dans le vecteur courant.

La Figure 5-4 montre un exemple de propagation. Trois nœuds de la couche d'entrée sont activés par le vecteur présenté au réseau. Ces signaux se propagent *via* les sorties et « stimulent » d'autres nœuds *via* leurs entrées. Dans l'exemple, deux nœuds de type 'Et' sont activés, un autre ne l'est pas car seule une de ses entrée est activée. Le nœud 'Ou' est activé également car il a reçu au moins un signal.

Propager (V):

Pour chaque nœud n_i de la couche d'entrée :
  Si (V[i] > s) Activer(n_i)

Activer (n_i) :

Activé(n_i)=vrai
Pour chaque père n_k de n_i
  Soit NEA(n_k) le nombre d'entrées activées de n_k
  Selon Type(n_k)
    ET :
      NEA(n_k) = NEA(n_k) + 1
      Si (NEA(n_k) = 2) alors Activer(n_k)
    OU :
      Si (NEA(n_k) = 0) alors NEA(n_k) = NEA(n_k) +1 ; Activer(n_k)
Fin

Figure 5-5 : Propagation récursive des valeurs du vecteur dans le réseau.

La Figure 5-5 explicite en pseudo code la première phase de l'apprentissage : une procédure récursive **« Activer »** est exécutée sur tous les nœuds de la couche d'entrée dont la valeur dans le vecteur d'entrée est « vrai ». Puisqu'un nœud représente la présence d'une régularité, il ne doit être activé qu'une fois. C'est la raison pour laquelle, dans le cas d'un nœud 'Ou', le nœud n'est activé que lorsque son nombre d'entrées activées passe de 0 à 1, toute nouvelle entrée activée après n'aura aucun effet. Signalons que préalablement à toute activation, pour chaque vecteur, chaque NEA est initialisé à 0.

A la fin de la phase de propagation, le vecteur d'entrée est donc recodé selon les régularités apprises précédemment. Cependant, comme nous allons le voir maintenant, si une régularité conjonctive (un trait) est activée, on ne tiendra pas

compte de ses composants. C'est là un des points centraux de notre approche : le **sens** d'un trait atomique (sa contribution à la représentation du vecteur d'entrée) dépend de la régularité à laquelle il participe, c'est en cela qu'on peut parler de recodage. Nous détaillons maintenant la seconde phase.

5.2.2.2 Phase de renforcement

Le but de la phase de renforcement est de tenir compte du nouvel exemple présenté en renforçant, dans le réseau, les cooccurrences entre les régularités présentes dans le vecteur (et connues du réseau). Les régularités déjà existantes dans le réseau ne sont donc pas explicitement renforcées, et on pourrait s'en étonner. Il y a là un point clé à comprendre : les régularités existantes (tous les nœuds, hors couche d'entrée) et détectées lors de la phase de propagation sont implicitement « renforcées » dans la mesure où elles servent de brique de base pour la construction de nouvelles régularités. Or, plus une régularité sert de constituant à d'autres régularités plus complexes, plus son influence relative est grande. Dès lors, il n'est pas nécessaire de renforcer explicitement ces régularités.

La phase de renforcement se divise en deux tâches : l'apprentissage des conjonctions, qui renforce les cooccurrences et crée éventuellement des nœuds de type 'Et', et l'apprentissage des disjonctions qui met à jour les nœuds de type 'Ou'.

5.2.2.2.1 Règles de blocage

Nous arrivons au point où il est nécessaire d'expliciter notre biais inductif. Dans le chapitre précédent, nous avions énoncé des contraintes qui, ajoutées à notre choix des régularités représentables, forment ce biais inductif.

Les règles que nous posons ci-dessous découlent de ces contraintes, elles ont pour objectif de restreindre le nombre de nœuds considérés lors de la phase de renforcement de manière à ce que ce nombre soit (à peu près) constant au cours de l'apprentissage, plutôt qu'exponentiel en terme du nombre d'exemples d'apprentissage.

Règle 1 (Le tout remplace les parties) :

Tout nœud appartenant à un sous arbre de nœuds de type 'Et' activés est bloqué, sauf le père.

Règle 2.1 (Règle des multiples niveaux de généralisation) :

Les fils d'un noeud 'Ou' ne sont jamais bloqués.

Règle 2.2 (Règle des multiples niveaux de généralisation) :

Un nœud 'Et', fils d'un nœud 'Et' dont l'autre fils est un nœud 'Ou', n'est pas bloqué.

La règle 2 découle directement de la nécessité de conserver plusieurs niveaux d'abstraction dans la représentation d'un même vecteur, que nous avions évoquée au chapitre précédent. Rappelons que cette nécessité provient du fait que l'on ne puisse pas savoir à l'avance quel sera le niveau d'abstraction des concepts définis par l'utilisateur.

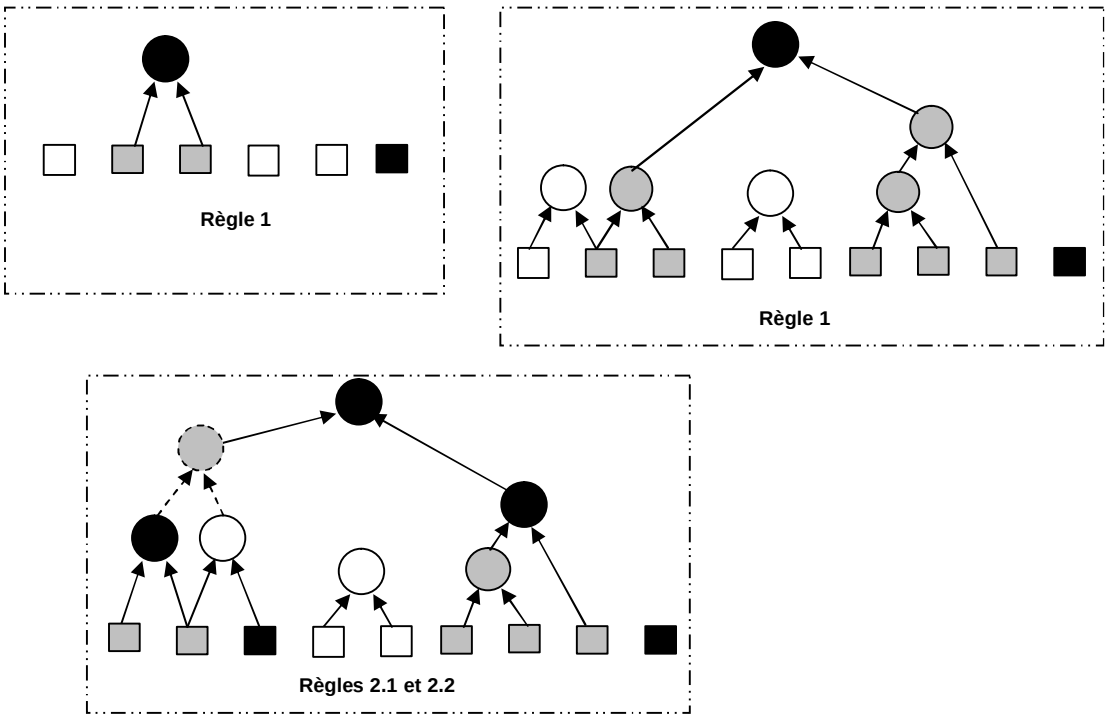


Figure 5-6 : Trois exemples simples d'application des règles.

Graphiquement, nous représentons un nœud bloqué activé par un cercle (ou carré) grisé, les nœuds activés et sélectionnés pour la phase de renforcement étant en noir. La Figure 5-6 illustre les deux règles : les deux schémas du haut montrent des sous arbres conjonctifs dont les nœuds sont bloqués, sauf la racine. Le schéma du bas

illustre la règle deux : les fils du nœud ‘Ou’ ne sont pas bloqués et le ‘Et’ de droite non plus. De cette manière, les régularités sont apprises parallèlement de manière générique et spécifique.

Apprentissage des conjonctions

L’apprentissage porte sur les nœuds activés par le vecteur, mais non bloqués par les règles précédentes. La Figure 5-7 montre un exemple dans lequel une seule cooccurrence est renforcée, les autres nœuds étant désactivés ou bloqués.

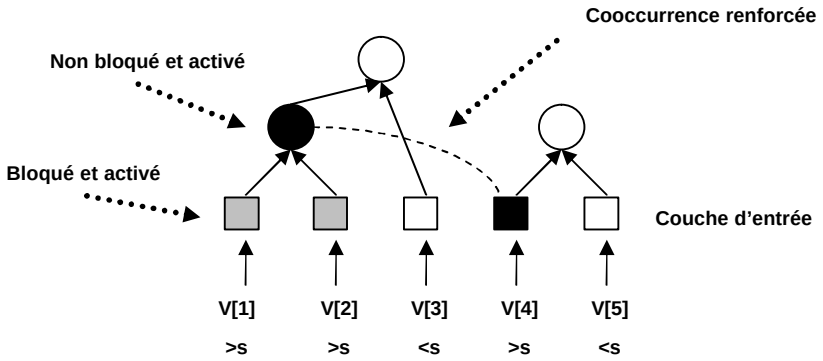


Figure 5-7 : Un exemple dans lequel une seule cooccurrence est renforcée. Parmi les quatre nœuds activés, deux sont en effet bloqués : ce sont les constituants d’un nœud activé.

Lorsque le nombre de cooccurrences entre deux nœuds dépasse un seuil (le seuil de création de nœud), un nouveau nœud de type ‘Et’ est créé, afin de « cristalliser » dans le réseau le fait qu’il s’agisse d’une cooccurrence significative. Dorénavant, et en vertu de la règle 1, lorsque cette cooccurrence se produira, seul le nœud créé sera pris en compte, les nœuds fils seront bloqués. La Figure 5-8 schématise la règle de création de nœud.

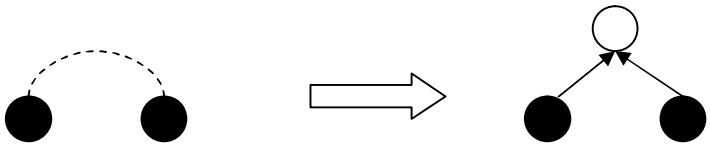


Figure 5-8 : Un nœud de type ‘Et’ est créé lorsque le nombre de cooccurrences entre deux nœuds dépasse un certain seuil.

Lorsqu'un nœud est créé, des liens de cooccurrences le sont également, afin de permettre l'apprentissage des cooccurrences entre ce nouveau nœud et les nœuds existants. Le nouveau nœud est donc relié à tous les nœuds existants, à l'exception de ses descendants.

Le seuil de création d'un nœud conjonctif, qui définit le compromis entre la vitesse et la robustesse de l'apprentissage, sera fixé expérimentalement.

5.2.2.2.2 Apprentissage des disjonctions

La mise à jour ou création des nœuds 'Ou' intervient <u>après</u> le traitement des conjonctions, car si de nouveaux nœuds 'Et' on été créés, il se peut que certain aient un contexte commun. Un nœud disjonctif (de type 'Ou') est créé lorsque deux nœuds conjonctifs (de type 'Et') ont un constituant commun. Si plus de deux nœuds conjonctifs ont un constituant commun, le nœud disjonctif existe déjà et il est simplement mis à jour.

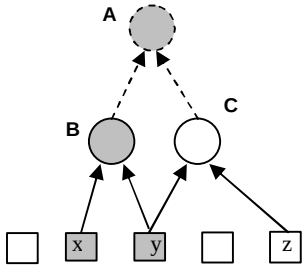


Figure 5-9 : La création du nœud 'Ou' A intervient après la création du nœud 'Et' B.

La Figure 5-9 illustre la création d'un nœud 'Ou' : lors de la phase de renforcement des cooccurrences, une cooccurrence a dépassé le seuil de création de nœud et le nœud de type 'Et' B a été créé. Or, il existait déjà un nœud de type 'Et' (le nœud C) ayant pour fils un constituant commun 'y'. Un nœud de type 'Ou' est donc créé, il représente une abstraction par rapport à ses fils B et C, c'est-à-dire que dans le contexte 'y', 'x' et 'z' sont similaires, ou interchangeables. Etant une généralisation de B et C, A est une régularité plus probable. Par conséquent, il est plus probable que A serve de constituant à de nouvelles régularités que B et C. Remarquons que la généralisation apportée par le nœud A est tout à fait différente du cas où l'on

généraliserait en ne considérant que ‘y’. Certes, le fait que ‘y’ soit activé est indispensable à l’activation de A mais il faut également que ‘x’ ou ‘z’ soit activé.

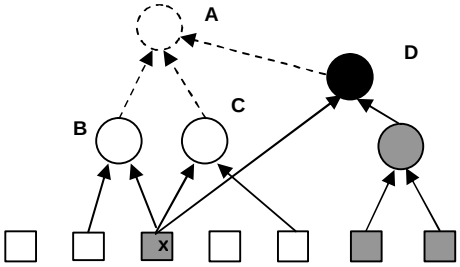


Figure 5-10 : Mise à jour d’un nœud ‘Ou’.

La Figure 5-10 illustre la mise à jour d’un nœud ‘Ou’ : un nœud conjonctif D vient d’être construit, et il a pour constituant un nœud x qui est lui-même constituant de deux autre nœuds conjonctifs (B et C). Il existait déjà un nœud disjonctif A (depuis la création de B ou de C) pour refléter ce constituant commun. La mise à jour consiste à ajouter une entrée à A, en provenance de D.

Il est important de noter qu’un nœud disjonctif peut être mis à jour *même s’il est déjà utilisé* comme constituant de régularités plus complexes. Cela signifie que l’apprentissage à un niveau spécifique, peut avoir des répercussions sur toutes les régularités (et donc tous les concepts) qui utilisent ce niveau. Par exemple, si une régularité disjonctive représente les différentes teintes que peut prendre un ciel clair et que cette régularité est mise à jour (par l’addition d’une nouvelle teinte), toutes les représentations qui utilisent ce nœud « ciel clair » (par exemple « Ciel », « Paysage » ou « Extérieur ») s’en trouvent modifiées et indirectement mises à jour.

5.3 Apprentissage supervisé

La phase d’apprentissage non supervisé a généré, à partir d’un flux de données, un ensemble hiérarchique de régularités qui représentent, par construction, un ensemble de propriétés caractérisant ces données.

Une des hypothèses importantes que nous avons émises au chapitre 4 et qui a motivé l’orientation de ce travail est l’idée que *les concepts, et pas seulement les régularités, sont des propriétés des données*, qui par conséquent, ne dépendent pas (du moins pas complètement) de celui, ou de ce qui observe les données. Cette

indépendance est toutefois nuancée par le fait que l'ordre dans lequel l'observateur perçoit les données est important et peut influer sur les régularités et concepts appris, ainsi que leur importance relative.

Si cette hypothèse est vérifiée, elle entraîne une conséquence très importante : les *mêmes concepts* seront appris *indépendamment* de l'apprenant (pour peu que celui-ci dispose de capacités d'apprentissage suffisantes) mais également indépendamment de la manière dont sont perçues (codées) ces données (pour peu que suffisamment d'information soit conservée).

Si cette hypothèse est vérifiée, on peut alors s'attendre à ce que des régularités apprises de manière supervisée, à partir des données, soient au minimum utiles pour caractériser des concepts. Si c'est le cas, il devrait apparaître des corrélations entre les concepts et certaines régularités. C'est à la mise en évidence de cette corrélation éventuelle que nous nous intéressons maintenant.

5.3.1 Traduction des vecteurs

Comme lors de l'apprentissage non supervisé, la première étape est de traduire les vecteurs en termes des régularités connues par le réseau. Dans la phase d'apprentissage supervisé, ces vecteurs sont étiquetés, c'est-à-dire associés à un concept.

Cette traduction est similaire à celle utilisée lors de l'apprentissage non supervisé et correspond à la phase de propagation décrite en 5.2.2.1. La différence est que lors de l'apprentissage supervisé, toutes les régularités détectées sont conservées : les règles de blocage ne sont pas appliquées.

Il y a deux raisons à cela :

Nous considérons que les exemples étiquetés sont rares et qu'il faut les exploiter au maximum. Pour exploiter un exemple au maximum, il faut considérer toutes les manières possibles de le représenter. C'est pourquoi nous traduisons ici un vecteur comme étant l'ensemble des régularités qu'il contient, quelques soient leur niveau d'abstraction et même si certaines régularités sont des constituants d'autres régularités, ce qui induit une certaine redondance dans l'expression du vecteur.

Les règles de blocage, qui matérialisent les contraintes qui limitent l'exploration de notre espace d'hypothèse, sont nécessaires lors de l'apprentissage non supervisé : sans elles, le nombre de nœuds créés serait exponentiel par rapport au nombre d'exemples d'apprentissage[41]. En apprentissage supervisé, ce problème ne se pose pas puisque le nombre de régularités considérées est sans impact sur les futures représentations.

5.3.2 Corrélations régularités/concepts

L'apprentissage supervisé consiste principalement à comptabiliser les cooccurrences entre concepts et régularités. La Figure 5-11 illustre ce principe : à chaque fois qu'un vecteur, instance d'un concept C_i, est présenté au réseau pour l'apprentissage supervisé et qu'il contient une régularité R_j, le compteur R_jC_i est incrémenté.

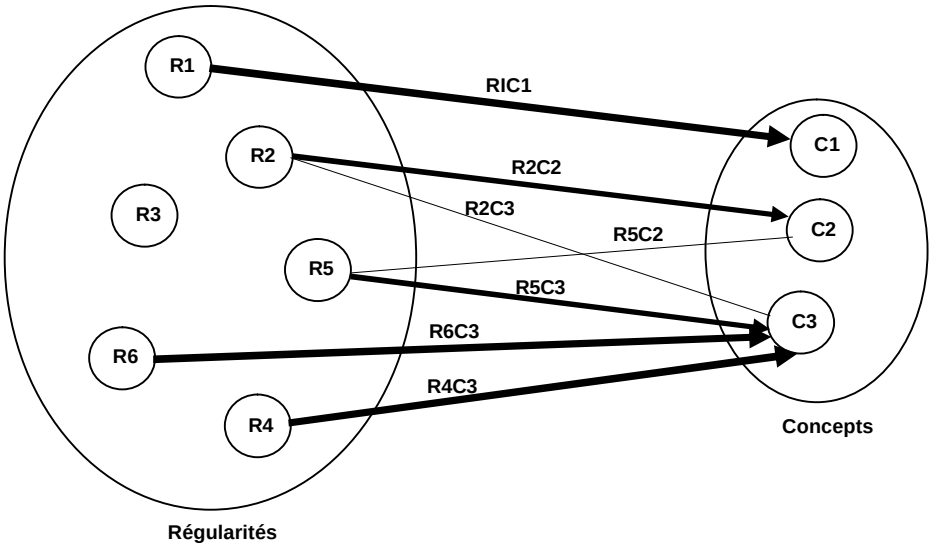


Figure 5-11 : Lors de l'apprentissage supervisé, chaque cooccurrence entre une régularité et un concept est comptabilisée.

Le résultat de l'apprentissage supervisé est simplement l'ensemble de valeurs R_iC_i.

[41] Car le nombre de régularités créées est proportionnel au nombre de régularités détectées.

5.4 Classification

La classification est également très simple : un vecteur inconnu V est tout d'abord traduit en termes de régularités. Pour cela, nous appliquons la phase de propagation ainsi que les règles de blocage.

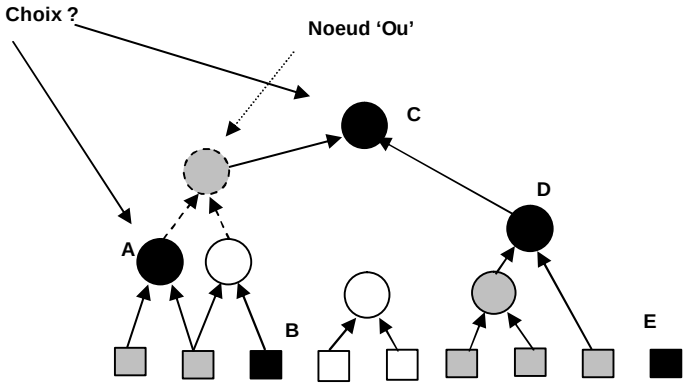


Figure 5-12 : La traduction d'un vecteur requiert le choix entre divers niveaux d'abstraction.

Cependant, contrairement à l'apprentissage non supervisé, nous ne pouvons pas ici conserver plusieurs niveaux d'abstraction dans la représentation du vecteur. Par exemple, la Figure 5-12 montre la traduction d'un vecteur après la phase de propagation et l'application des règles de blocage. Cette traduction présente une redondance car les deux nœuds indiqués par les flèches possèdent des constituants communs. Dans la mesure où la classification d'un vecteur nécessite que celui-ci soit représenté de manière unique et non ambiguë, nous devons choisir un seul de ces nœuds. L'idée est de choisir la représentation qui donne lieu à la classification la moins ambiguë.

La première étape est de calculer l'activation de chaque concept, en représentant le vecteur de la manière la plus abstraite possible. Dans la figure, la représentation la plus abstraite est (C , E), car A, B et D sont à la fois des constituants et des spécifiques de C. L'activation d'un concept j se calcule de la manière suivante :

$$Activation(Cj) = \sum_i \left(R_i C_j \right)$$

Afin de mesurer l'ambiguïté de la classification, nous nous intéressons à la valeur de la différence d'activation entre les deux concepts les plus activés. Plus cette différence est élevée, moins la classification sera considérée comme ambiguë.

L'activation des concepts est une nouvelle fois calculée, en représentant le vecteur de manière plus spécifique. Dans la Figure 5-12, la représentation immédiatement plus spécifique est (A, B, D, E). Si cette représentation est moins ambiguë, l'itération continue et l'activation des concepts est calculée pour une représentation encore plus spécifique. Si, en revanche, cette représentation est plus ambiguë, la représentation précédente est conservée. Cette procédure est appliquée de la représentation la plus générique à la plus spécifique, et s'arrête lorsque la représentation spécifique est plus ambiguë que la représentation générique.

Intuitivement, notre algorithme de classification essaye d'abord de reconnaître une observation comme « un tout » et, en cas d'ambiguïté, décompose ce tout en « parties » puis évalue à nouveau l'observation.

5.5 Conclusion

Nous avons présenté une instanciation de notre modèle d'apprentissage. L'espace initial de représentation des données est concrétisé par une couche d'entrée. Les régularités sont représentées par des nœuds interconnectés formant un arbre dont les feuilles sont les traits atomiques de la couche d'entrée. Les contraintes sur notre biais inductif sont réalisées par des règles de blocage qui restreignent le nombre de nœuds activés. La traduction des vecteurs en termes de régularités est effectuée récursivement par propagation des valeurs du vecteur supérieures à un seuil d'activation. La mise à jour des cooccurrences entre nœuds activés/ non bloqués est itérative et requiert un nombre d'opération proportionnel au carré du nombre de régularités en question. L'apprentissage supervisé et la classification requièrent principalement la propagation de vecteurs dans le réseau auquel des nœuds concepts ont été ajoutés.

Troisième partie

Expérimentation

Chapitre 6

Expérimentations

6.1 Rappel des objectifs

Nous avons proposé une nouvelle approche de l'apprentissage. Cette approche, sous-tendue par la notion de régularités hiérarchiquement combinées, unifie des idées considérées habituellement séparément. Ces idées sont : l'apprentissage non supervisé, l'apprentissage supervisé, la création de traits et la pondération de traits. Cette approche a été imaginée dans le but de pouvoir se confronter à des problèmes de type-2. Parallèlement, le contexte de notre travail impose des exigences supplémentaires : un utilisateur doit pouvoir faire apprendre, rapidement et efficacement, des concepts au système.

Nous commençons par présenter les traits de bas niveau et la segmentation utilisés pour représenter les régions d'images dans nos expérimentations (6.2).

La première expérimentation a pour but d'évaluer l'utilité de la phase d'apprentissage non supervisé (6.3). Si cette étape est utile, elle devrait donner lieu à des régularités (combinaisons de traits atomiques) plus pertinentes que les traits atomiques. Nous mesurerons cette pertinence des régularités par rapport aux traits atomiques en évaluant l'apprentissage supervisé de quelques concepts.

Dans un second temps (6.4), nous allons nous intéresser à l'intérêt des régularités du point de vue de la réactivité du système. Si celles-ci sont pertinentes par rapport à des concepts cibles, la phase d'apprentissage supervisé devrait s'en trouver allégée. Le système devrait être plus réactif qu'un système équivalent ne possédant pas ces régularités, cela devrait permettre un apprentissage rapide des concepts (i.e. l'utilisateur devrait fournir moins d'exemples).

Pour donner une idée des performances globales de notre approche, nous la comparerons à une autre technique, particulièrement adaptée pour les

comparaisons : le 1-NN (i.e. la technique des k plus proches voisins, où le nombre de voisins considérés sera de 1). Nous mesurerons ces performances en utilisant le rappel et la précision.

Finalement, avant de conclure ce chapitre d'expérimentations (6.7), nous observerons quelques topologies de réseaux créés, ainsi que le rapport entre la topologie obtenue et la variabilité des concepts. Nous présenterons également une expérimentation préliminaire sur des données synthétiques.

6.2 Traits de bas niveau & segmentation

La manière dont nous segmentons les images, ainsi que la manière dont nous les représentons est relativement simple et classique et a pour avantage la rapidité de calcul. Etant donné notre contexte, nous recherchons la robustesse des traits par rapport à la diversité des concepts et à la variabilité de leurs apparences.

En ce qui concerne la segmentation, nous divisons l'image en 100 blocs de taille égale. Comme nous l'avons vu dans l'état de l'art, la segmentation intelligente (sémantique) de l'image reste un problème ouvert dont la résolution apporterait également une réponse au problème de l'indexation. Une autre approche plus réaliste, la segmentation visant à produire des régions homogènes du point de vue des couleurs, des textures ou des formes dépend fortement des concepts à apprendre et n'est donc pas adaptée à notre contexte. La segmentation en blocs nous semble être une bonne alternative : elle est instantanée et, mis à part la taille des blocs, ne requiert pas de paramètres.

Nous extrayons de chaque bloc un vecteur de 120 valeurs :

- 80 réels compris entre zéro et un représentant la répartition des teintes (Hue) dans l'espace de couleurs HSV. Ces valeurs sont normalisées de manière à ce que la somme des 80 valeurs soit égale à un.
- Dix réels compris entre 0 et 1 représentant la répartition des saturations (Saturation) dans l'espace de couleurs HSV. Ces valeurs sont normalisées de manière à ce que la somme des dix valeurs soit égale à un.

- Dix réels compris entre zéro et un représentant la répartition des intensités des niveaux de gris (Value) dans l'espace de couleurs HSV. Ces valeurs sont normalisées de manière à ce que la somme des dix valeurs soit égale à un.
- Dix réels compris entre zéro et un représentant une caractéristique de la texture que nous appellerons *rugosité* calculée en faisant glisser une fenêtre sur l'image, cette fenêtre étant sensible à la différence des niveaux de gris avant et après déplacement. Ces valeurs sont normalisées de manière à ce que la somme des dix valeurs soit égale à un.
- Dix réels compris entre zéro et un représentant une caractéristique de la texture que nous appellerons *directionnalité* calculée en faisant glisser cette même fenêtre dans différentes directions (0°, 45°, 90°, 135°, 180°) et en calculant le rapport entre le maximum et le minimum de rugosité. Ces valeurs sont normalisées de manière à ce que la somme des dix valeurs soit égale à un.

Nous caractérisons donc les régions d'image par des traits de couleur et des traits de texture. Les traits de couleur que nous avons choisi sont des plus communs puisqu'il s'agit d'une quantisation des trois canaux de l'espace de couleurs HSV. Toutefois, la manière dont nous comparons les régions ne fait pas appel, comme c'est souvent le cas, à des mesures de similarité entre histogrammes.

La représentation des textures est également classique puisque les traits utilisés (*rugosité* et *directionnalité*) sont deux des traits introduits par Tamura ([Tam78]) et encore largement utilisés aujourd'hui. De plus, Tamura a montré que la rugosité et la directionnalité (*coarseness & directionality)* sont, avec le contraste, les traits correspondant le mieux à la perception humaine des textures (parmi les traits qu'il a étudiés).

6.3 Evaluation de l'apprentissage non supervisé

Cette première expérimentation a pour but d'évaluer l'intérêt de découvrir et utiliser des régularités. Pour cela, nous avons choisi de travailler sur quatre concepts visuellement similaires, donc *a priori* difficilement séparables par un algorithme adapté aux problèmes de type-1. Dans cette expérimentation, nous n'utilisons pas d'images entière mais directement des régions. Ces régions sont des blocs de 64×64 pixels extraits d'images de la base de données Corel, utilisées dans [Pat04]. Les classes choisies sont :

- Champ (132 régions)
- Feuillages (191 régions)
- Fleur (214 régions)
- Herbe (153 régions)

La Figure 6-1 montre des exemples pour chacune des classes.

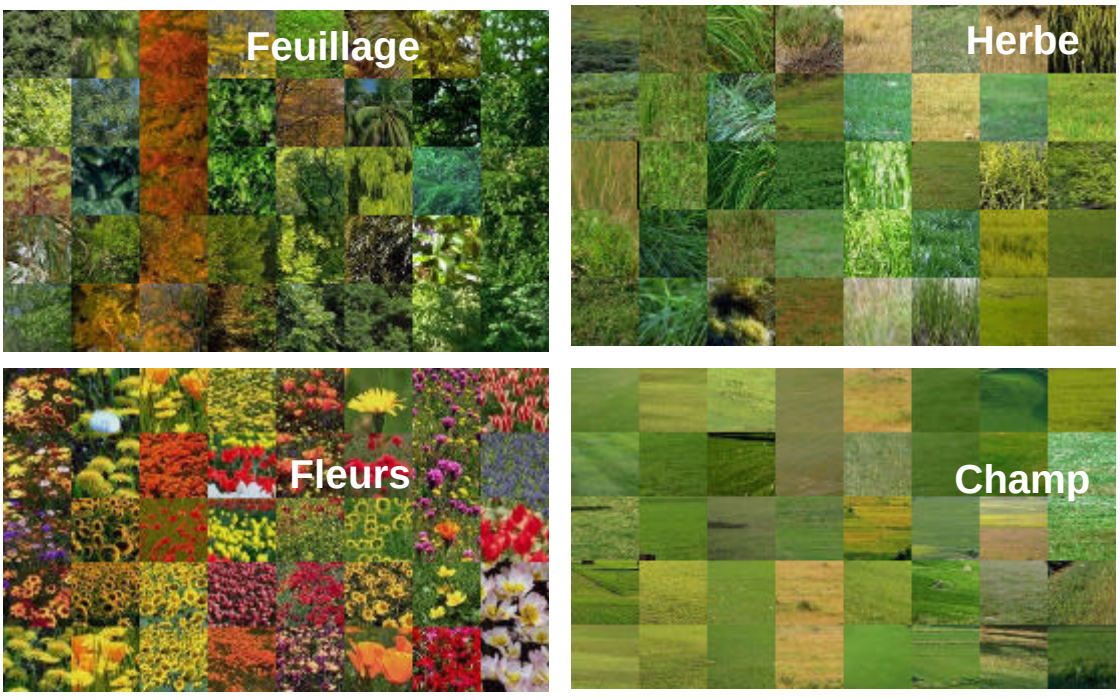


Figure 6-1 : Quelques exemples tirés de la collection utilisée.

Pour chacun des concepts, nous utilisons 120 exemples d'apprentissage, les exemples restants étant utilisés pour la classification. Chaque test est répété dix fois,

les exemples étant répartis aléatoirement entre l'ensemble d'apprentissage et l'ensemble de test.

Afin de placer nos résultats en perspective, nous commençons par présenter les résultats obtenus par l'algorithme 1-NN sur ce problème d'apprentissage particulier.

6.3.1 Résultats de l'algorithme 1-NN

L'algorithme des k plus proches voisins, et en particulier le cas k=1, a la particularité d'être à la fois très simple et très robuste, au point que Jain, Duin et Mao recommandent dans [Jai00] de l'utiliser comme référence pour évaluer un autre algorithme[42]. 1-NN se contente de mémoriser l'ensemble des exemples d'apprentissage et d'affecter à un vecteur test la classe de l'exemple d'apprentissage mémorisé le plus similaire. Dans la mesure où cet algorithme ne cherche en aucune manière à *modéliser* ou *caractériser* les données d'apprentissage, il n'y a nul besoin que ces données soient cohérentes ou qu'elles soient potentiellement caractérisables, c'est-à-dire que les exemples d'apprentissages au sein d'une même classe aient des caractéristiques en commun.

Le Tableau 6-1 montre les résultats obtenus sous la forme d'une matrice de confusion.

La précision moyenne (c'est-à-dire la moyenne des précisions pour les quatre concepts) et le rappel moyen sont :

- Précision : **61,2 %**
- Rappel : **66,2 %**

[42] « The most straightforward 1-NN rule can be conveniently used as a benchmark for all other classifiers since it appears to always provide a reasonable classification performance in most applications. »

		Exemples présentés			
		Fleur	Herbe	Feuillages	Champ
Concepts Reconnus	Fleur	0.69	0	0.07	0
	Herbe	0.05	0.58	0.15	0.31
	Feuillages	0.25	0.18	0.73	0
	Champ	0.01	0.24	0.05	0.69

Tableau 6-1 : Matrice de confusion pour l'algorithme 1-NN.

Une première remarque est que les quatre classes choisies ne semblent pas si difficiles à séparer puisque nous nous situons bien au-delà de l'aléatoire. Le concept « Herbe » semble être le plus problématique, ce qui, étant donné l'algorithme utilisé, peut avoir deux causes principales. La première cause peut être que les exemples d'apprentissages ne ressemblent pas aux exemples de test. La deuxième pourrait être que les exemples d'herbes ressemblent beaucoup aux exemples d'autres classes, comme « Champ » et « Feuillage ». En fait l'explication est un mélange de ces deux causes. Les images d'herbes que nous avons utilisées présentent en effet une très forte variabilité dans l'apparence. La conséquence est que parmi les images utilisées pour le test, bon nombres ne ressemblent à aucunes des images utilisées pour l'apprentissage.

6.3.2 Sans apprentissage non supervisé

Dans cette expérimentation, nous utilisons les exemples d'apprentissage pour évaluer le caractère discriminant des traits atomiques, puis procédons à la classification. Le réseau utilisé est donc réduit à sa couche d'entrée, c'est-à-dire 120 nœuds atomiques. Seule la phase d'apprentissage supervisé est effectuée ici : c'est-à-dire, les cooccurrences entre les nœuds activés (réduit à la couche d'entrée) et les concepts sont comptabilisées. Les nœuds activés sont ceux correspondant à une valeur du vecteur d'entrée supérieure au seuil d'activation, dont nous avons fixé la valeur expérimentalement.

Tout d'abord, nous donnons le rappel et la précision moyenne :

- Précision : **64.4 %**
- Rappel : **65,4 %**

		Exemples présentés			
		Fleur	Herbe	Feuillages	Champ
Concepts Reconnus	Fleur	0.81	0.12	0.12	0.08
	Herbe	0.02	0.27	0.08	0.08
	Feuillages	0.17	0.27	0.79	0.15
	Champ	0	0.33	0	0.69

Tableau 6-2 : Matrice de confusion – 120 nœuds atomiques.

Par rapport au 1-NN, nous assistons à une augmentation de la précision et à une légère baisse de rappel (matrice de confusion, figure 6-2). Cela est principalement dû au concept « herbe » qui est très mal reconnu. Etant donné l'algorithme utilisé, cela signifie que ce concept n'a pas pu être caractérisé par un ou plusieurs traits atomiques. Cette hypothèse est confirmée par l'observation de la pertinence des traits atomiques par rapport aux concepts cibles. La Figure 6-2 montre les traits atomiques discriminants pour chaque concept. On voit que le concept « herbe » n'est quasiment pas associé avec les nœuds de la couche d'entrée, ce qui signifie que ce concept n'est pas caractérisable par l'algorithme et les traits utilisés. A l'inverse, le concept « Fleur » est fortement représenté, ce qui s'explique en particulier par le fait que les instances de ce concepts possèdent des couleurs que ne possèdent pas les instances des autres concepts. On voit qu'il existe une corrélation forte entre les taux de rappel et précision obtenus d'une part, et le nombre de traits atomiques discriminants par concept.

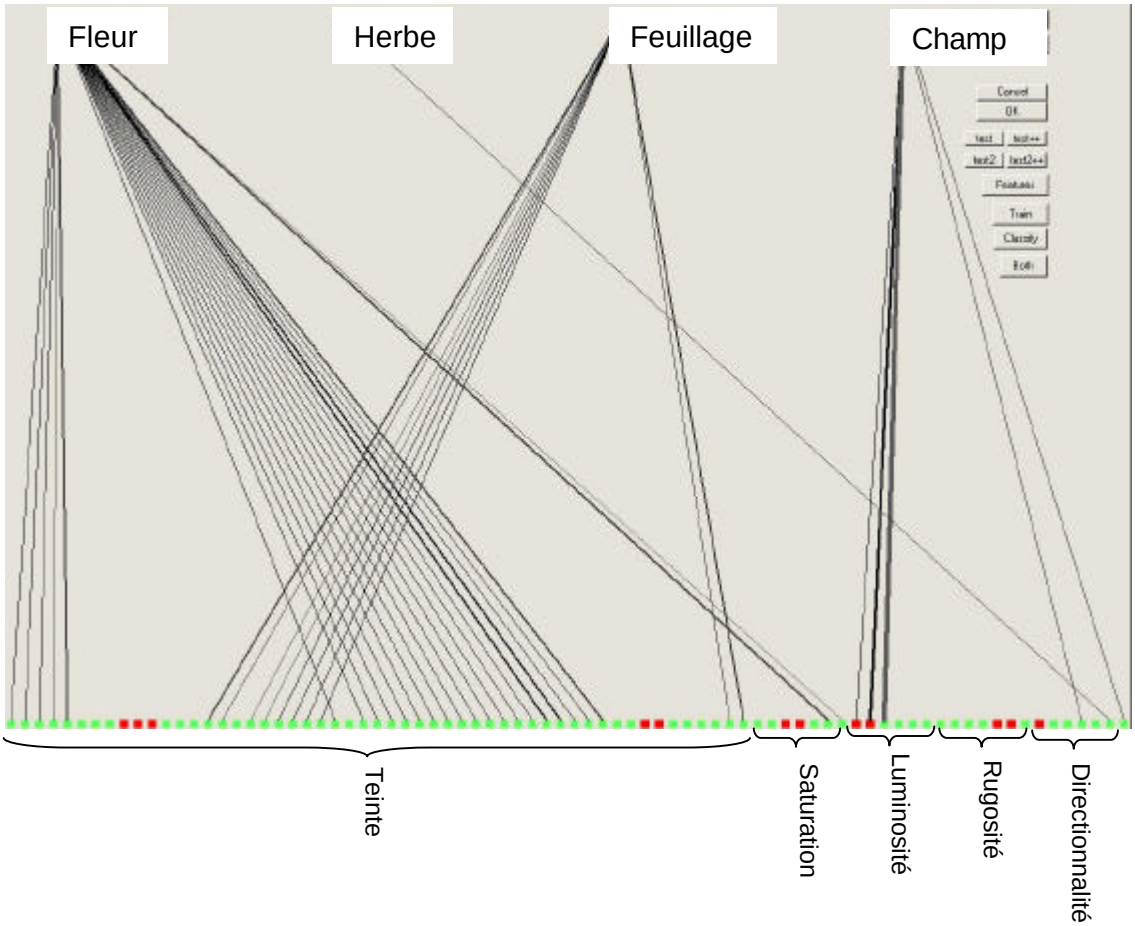


Figure 6-2 : Les nœuds atomiques de la couche d'entrée (en bas), reliés aux concepts (en haut) : de gauche à droite : Fleur, Herbe, Feuillages et Champ. Un lien exprime la pertinence d'un trait atomique par rapport à un concept.

Nous avons vu que les traits atomiques seuls ne sont pas suffisamment discriminants : de nombreux traits ne sont pas spécifiques à une classe particulière d'images. Nous allons maintenant voir si l'apprentissage non supervisé de régularités permet l'apprentissage de traits plus discriminants.

6.3.3 Avec apprentissage supervisé

Nous réitérons l'expérience précédente, à la différence que, cette fois une phase d'apprentissage non supervisé précédera l'apprentissage supervisé et la classification.

Nous présentons plusieurs tests, en faisant varier un seul paramètre : le nombre de nœud du réseau (c'est-à-dire le nombre de nœuds créés par apprentissage non supervisé, sans comptabiliser les 120 nœuds de la couche d'entrée). Ce paramètre représente *la quantité d'apprentissage non supervisé*. Cette quantité n'est pas proportionnelle au nombre d'exemples d'apprentissage présentés au réseau car le nombre d'exemples nécessaires à l'apprentissage d'une régularité dépend de sa fréquence dans l'ensemble d'apprentissage. En faisant varier ce paramètre, nous nous attendons à une augmentation des performances de classification. Nous voulons également observer s'il y a apparition du phénomène de sur-apprentissage.

	Nombre de nœuds	Précision	Rappel
1-NN	~	61.2	66.2
Notre approche	0	64.4	65.4
	160	68.8	68.2
	280	72.2	71.2
	360	70.1	69.4

Tableau 6-3 : Rappel et précision en fonction du nombre de nœuds du réseau. Résultats du 1-NN et de l'apprentissage supervisé uniquement reportés pour information.

Dans le Tableau 6-3, qui présente les résultats de rappel et de précision en fonction du nombre de nœuds créés (i.e. du nombre de régularités apprises), on peut voir que l'apprentissage non supervisé de régularités améliore significativement les résultats. Nous avons effectué un test d'égalité des espérances (où test de Student) sur l'ensemble des valeurs de rappel et de précision obtenus avec 1-NN d'une part et avec notre approche d'autre part (280 nœuds). La probabilité que les valeurs de précisions obtenues avec notre approche (resp. rappel) soient issues d'une distribution ayant la même moyenne que la précision (resp. rappel) obtenue avec 1-NN est égale à : 5.9^{E}-07 (resp. 0.001). Ces probabilités étant bien inférieures au seuil de signifiance couramment utilisé (0.05), ces résultats sont donc significatifs. Dans

notre expérimentation, le réseau est « forcé » d'apprendre, ce qui aboutit à un sur-apprentissage (réseau de 360 nœuds). Les nœuds appris ne correspondent pas nécessairement à de vraies régularités lorsque l'apprentissage est forcé. Or un vecteur traduit en termes de régularités « non représentatives » de l'ensemble d'apprentissage tend à être moins bien reconnu, car un certain bruit est introduit dans le système. Dans un futur proche, nous envisageons d'automatiser le processus d'arrêt de l'apprentissage non supervisé, soit par l'utilisation d'un seuil, soit par validation croisée (ce qui nécessite toutefois des données étiquetées).

Cela signifie que parmi les régularités apprises, certaines sont plus discriminantes que les traits de bas niveau atomiques. Notre approche se révèle utile dans la mesure où ses résultats surpassent ceux du 1-NN ainsi que les résultats obtenus sans apprentissage non supervisé. Ces résultats permettent donc un jugement *relatif*. Cependant, il faut remarquer que nous ne savons pas quel est le résultat *idéal* étant donné ces concepts, et les traits atomiques utilisés. Toutefois, il nous semble quasi certain que les précisions et rappels potentiellement atteignables ne sont pas de 100 %, dans la mesure où même un humain ne saurait pas étiqueter parfaitement les régions utilisées.

En observant ces résultats concept par concept (Tableau 6-1), on s'aperçoit que la détection des concepts n'est pas affectée de manière uniforme par l'utilisation des régularités. En particulier, le concept « Fleur » qui était fortement reconnu lors de l'apprentissage uniquement supervisé n'est pas amélioré de manière significative. Ceci suggère que, pour ce concept, certains traits atomiques étaient déjà suffisamment discriminants. Il faut cependant remarquer que notre approche ne dégrade pas la reconnaissance, ce qui est très appréciable.

160		Exemples présentés			
		Fleur	Herbe	Feuillages	Champ
Concepts reconnus	Fleur	0.84	0.24	0.05	0.08
	Herbe	0	0.21	0.05	0.15

Feuillages	0.16	0.39	0.88	0
Champ	0	0.15	0.01	0.77

280	Fleur	Herbe	Feuillages	Champ
Fleur	0.84	0.09	0.12	0
Herbe	0	0.3	0.04	0
Feuillages	0.16	0.48	0.82	0.08
Champ	0	0.12	0.01	0.92

360	Fleur	Herbe	Feuillages	Champ
Fleur	0.83	0.12	0.16	0
Herbe	0.02	0.3	0.05	0.08
Feuillages	0.15	0.39	0.77	0.08
Champ	0	0.18	0.01	0.85

Tableau 6-4 : Matrices de confusion pour des apprentissages non supervisés de 160, 280 puis 360 régularités.

Le concept « Herbe » en revanche est toujours aussi mal reconnu et les régularités apprises semblent ne rien changer. Pourtant, ces régularités sont discriminantes. La Figure 6-3 montre les régularités apprises pour le concept « Herbe ». Si on compare cette figure à la Figure 6-2 (sans régularités), on voit qu'il existe maintenant de nombreux traits discriminants, ces traits étant ceux reliés par un segment au concept « Herbe » se trouvant en haut de la figure. Un point très intéressant est que ces traits discriminants sont ceux ayant un nombre élevé de constituants (les plus hauts dans la figure) et les nœuds utilisés pour les construire **ne sont pas** discriminants, c'est-à-dire qu'ils n'apportent pas d'information pertinente pour la classification des instances du concept « Herbe ». Cela signifie que les traits discriminants n'auraient pas pu être

découvert par un algorithme d'apprentissage uniquement supervisé car les constituants de ces traits (devant donc nécessairement être créés *avant*) auraient été jugés inutiles. Il faut également noter que les nœuds discriminants créés ne correspondent pas à des instances du concept « herbe » (ce qui pourrait expliquer leur caractère discriminant), il suffit pour s'en convaincre de considérer que la moitié (approximativement) des nœuds sont des régularités disjonctives. Par conséquent ces nœuds discriminants peuvent être activés par de nombreux vecteurs différents : c'est dans les régularités disjonctives que reposent les concepts de similarités et de généralisation. Cette généralisation, étant apprise plutôt qu'arbitraire (comme la généralisation liée à une simple distance dans l'espace), elle permet la reconnaissance de vecteurs inconnus, ne ressemblant pas nécessairement (au sens Euclidien) aux vecteurs d'apprentissage.

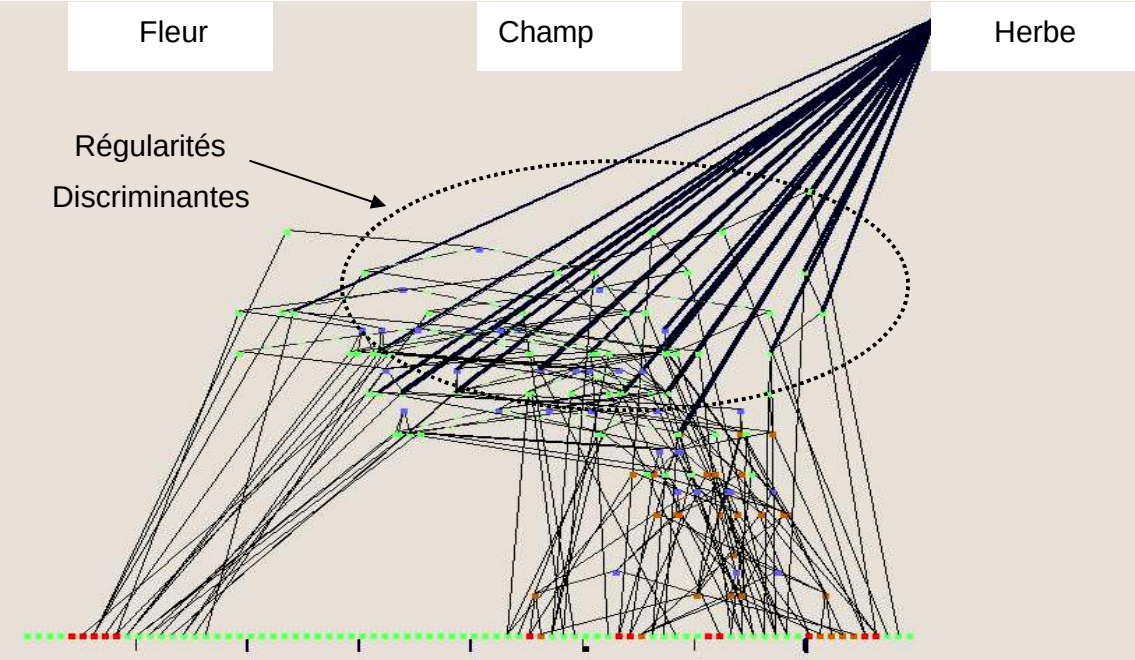


Figure 6-3 : Régularités apprises pour le concept « Herbe »

Notre approche induit une mesure de similarité basée sur des régularités, et non sur un ensemble de traits atomiques. Cette nuance est importante : lors de la classification par la méthode des plus proches voisins, tous les traits sont utilisés et ont le même poids, de plus les vecteurs exemples sont « creux » (comportent beaucoup de 0), il en résulte que la similarité entre deux vecteurs est toujours

supérieure à 88 %. En augmentant le nombre de dimensions, il y aurait convergence des valeurs de similarités entre vecteurs (phénomène connu sous le nom de *curse of dimensionality*) et les concepts seraient encore moins séparables. Notre approche, qui tente de caractériser les concepts plutôt que d'en mémoriser les exemples, ne rencontre pas du tout ce phénomène puisque seuls les traits pertinents sont construits et utilisés pour la comparaison. Outre une indépendance au nombre de dimensions, il existe un autre avantage : les vecteurs étant reconnus avec un certain degré de certitude, plus fiable que celui obtenu avec le 1-NN, il serait possible d'ordonner les résultats d'une requête en montrant d'abord les régions (ou images) dont la valeur de certitude est la plus grande. Par exemple, lors de cette expérimentation, les exemples positifs d'un concept étaient reconnus par 1-NN avec une similarité moyenne normalisées de 0.94. Les exemples négatifs pour ce même concept étaient reconnus avec une valeur moyenne de 0.89. Pour notre approche, ces mêmes valeurs de similarité normalisées étaient respectivement de 0.78 et 0.23. Cette caractéristique appréciable est habituellement obtenue *via* sélection des traits, afin de réduire le nombre de dimensions et donc aussi *la malédiction de la dimensionnalité*. Dans notre approche, cette propriété est obtenue par *construction* de traits.

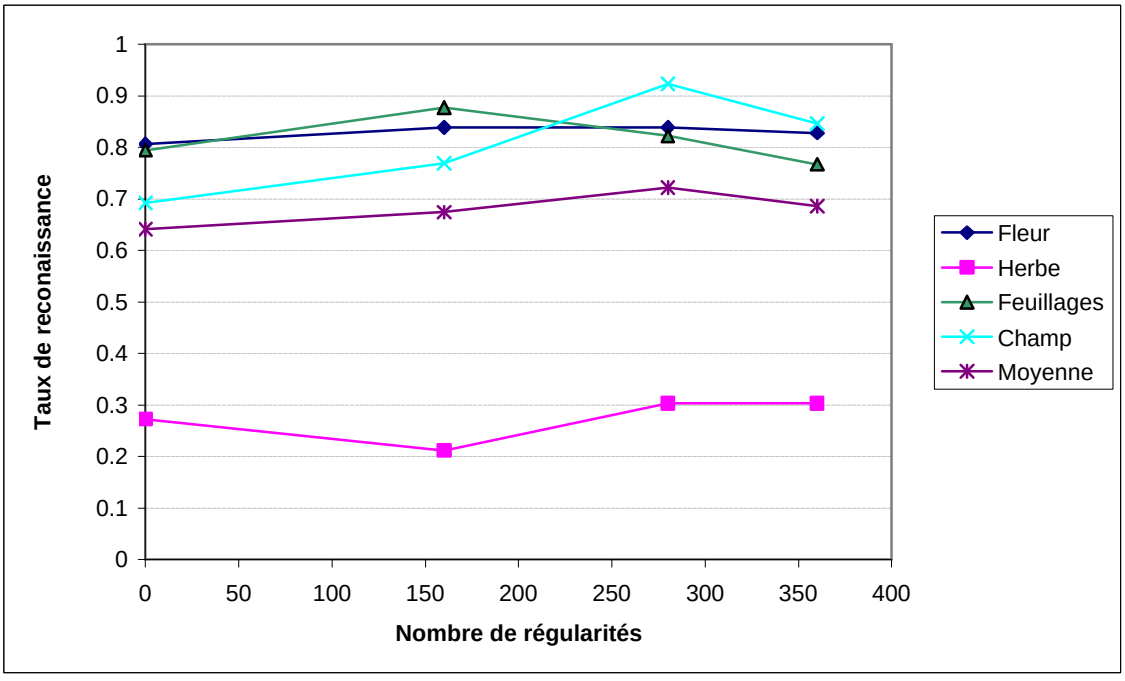


Figure 6-4 : Taux de reconnaissance par concept en fonction du nombre de régularités apprises.

En ce qui concerne les concepts « Feuillages » et « Champ » (Tableau 6-4), la reconnaissance est améliorée par l'apprentissage des régularités. Toutefois, au-delà d'un certain nombre de régularités apprises, on note une diminution du taux de reconnaissance moyen. Nous avons constaté que les régularités apprises pour les trois concepts autres que « Feuillage » tendent à être de « haut niveau », c'est-à-dire qu'elles incluent comme constituants la plupart des traits atomiques activés par le vecteur exemple. En ce sens, ces régularités sont « précises », par opposition à celles apprises pour le concept « Feuillage » qui sont de moins haut niveau. La raison à cela est que le concept « Feuillage » présente une très forte variabilité dans l'apparence de ses instances, et il y a donc plusieurs sous-classes à apprendre. Les régularités apprises ne sont toutefois pas toutes caractéristiques du concept « Feuillage », car ses instances ressemblent assez fréquemment à certaines instances d'herbe ou de champs. Le résultat est donc que le concept « Feuillage » finit par devenir une sorte d'attracteur pour les instances des autres concepts visuellement similaires, ce qui explique cette inversion de tendance.

6.3.4 Conclusion

Dans cette première expérimentation, nous avons voulu mettre en évidence l'intérêt de notre approche sur des concepts visuellement semblables. Le problème est en réalité partiellement de type-1, puisque l'algorithme 1-NN parvient à étiqueter convenablement plus de la moitié des instances. Cependant, nous avons vu que l'apprentissage de régularités permet d'améliorer les résultats en termes de rappel et de précision, même si nous ne connaissons pas les performances théoriques maximales que l'on pourrait atteindre. Nous avons ainsi gagné 5 % de rappel et 11 % de précision par rapport à l'algorithme 1-NN et ce gain a porté nécessairement sur des instances difficilement séparables. Il faut également noter qu'un phénomène de sur-apprentissage apparaît, qu'il nous faudra corriger à court terme.

6.4 Evaluation de la réactivité

Les performances d'un système d'indexation d'images sont certes importantes, mais il existe une contrainte supplémentaire cruciale dans notre contexte de forte interactivité avec l'utilisateur : il s'agit de la réactivité. La réactivité qualifie la qualité

de l'indexation en fonction de la taille de l'ensemble d'apprentissage : une réactivité forte permet une indexation de qualité en dépit de la taille restreinte d'un ensemble d'apprentissage. Un système dont la réactivité est insuffisante est *inutilisable* par un utilisateur.

Le but de l'expérience que nous allons décrire est donc d'évaluer la réactivité de notre système, qui dépend directement de l'algorithme d'apprentissage/classification. Afin de rendre comparables les résultats obtenus, nous utilisons parallèlement l'algorithme 1-NN pour l'apprentissage et la classification. Avant de présenter le protocole et les résultats, nous décrivons brièvement la base d'images utilisée et les paramètres utilisés pour la segmentation.

6.4.1 Segmentation et collection d'images

Segmentation :

Nous divisons chaque image en 100 blocs carrés de taille identique, ce qui produit en tout 74 400 blocs. Chaque pixel de chaque image est étiqueté par un concept unique. Un bloc pouvant contenir des pixels de concepts différents, nous avons défini un seuil de « pureté » au dessus duquel le bloc est indexé par le concept le plus représenté parmi les pixels contenu dans ce bloc. Ce seuil a été expérimentalement fixé à 0.7, c'est-à-dire que nous indexons les blocs dans lesquels 70 % des pixels appartiennent au même concept.

Collection :

Il s'agit d'une collection de photographies personnelles de 744 photographies. Ces photographies ont été prises et indexées manuellement par Yves Chiaramella. Dans la suite, nous ferons référence à cette collection en la désignant par YC744 (exemples Figure 6-5).

Les images sont indexées selon 132 concepts distincts, dont la fréquence d'apparition dans la collection varie fortement. Par exemple, le concept « Porte » n'est représenté que par un unique bloc, alors que le concept « Ciel_clair » est représenté par plus de 5 000 blocs.

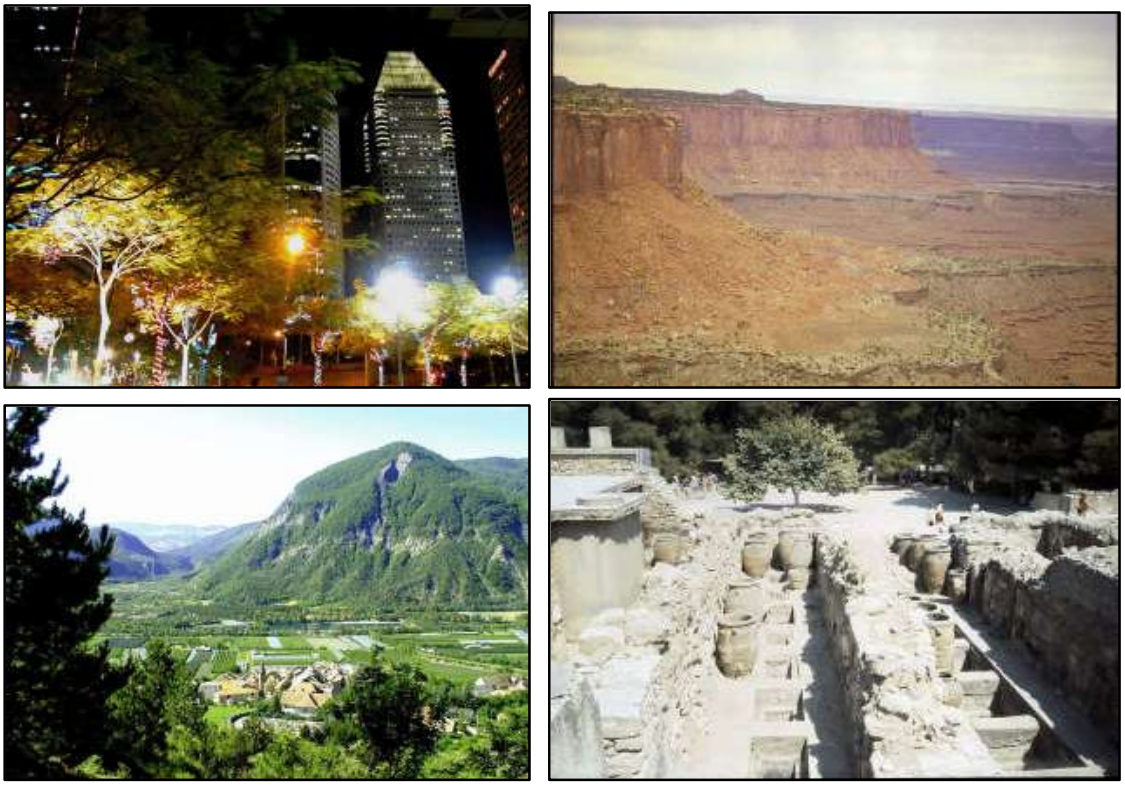


Figure 6-5 : Exemples de photographies tirées de la collection YC744.

La collection YC744 est un peu particulière dans la mesure où les images proviennent de nombreux lieux très différents (montagnes Alpines, villes asiatiques, désert d'Arizona, etc.) mais également d'époques différentes (plusieurs dizaines d'années peuvent séparer la prise de deux photographies). La variabilité géographique induit une variabilité dans l'apparence des concepts, tandis que la variabilité temporelle conduit à une forte variabilité de qualité des images : les plus anciennes sont scannées et leurs couleurs semblent « délavées », les images plus récentes étant numérisées directement à la source (appareil photographique) sont plus nets et leurs couleurs sont plus réalistes. Nous sommes donc bien loin de l'uniformité de la collection Corel utilisée dans l'expérience précédente.

6.4.2 Protocole

Nous utilisons toutes les images de la collection. Dans une première phase, un apprentissage non supervisé est effectué sur 5 000 blocs choisis aléatoirement dans

la collection, parmi ceux dépassant le seuil de pureté. Cette phase a pour but d'apprendre les régularités présentes dans la collection. Le point important est que, cet apprentissage étant non supervisé, il ne requiert aucune interaction avec l'utilisateur. Par conséquent, la phase la plus lourde de l'apprentissage peut être effectuée « en tâche de fond ».

Dans une seconde phase, nous simulons l'interaction utilisateur/système : des images sont sélectionnées au hasard dans la collection (par groupe de 5) et sont considérées comme indexées par l'utilisateur. Les blocs provenant de ces images sont utilisés pour l'apprentissage, d'une part selon notre approche et d'autre par selon le 1-NN. Ces images sont ensuite éliminées et la classification, selon les deux approches, est effectuée sur les images restantes. Dans cette expérimentation, nous limitons le nombre de concepts à quinze : les concepts les plus représentés dans la collection (Tableau 6-5). Ce choix est motivé par le fait que tenir compte de plus de concepts, en particuliers les concepts très peu représentés, conduirait à uniformiser les résultats dans la mesure où les concepts dont les instances sont rares ne peuvent pas être appris correctement.

Concept	Quantité
Ciel_Clair	3266
Arbre_Avec_Feuilles	2618
Rocher	1006
Ciel_Couvert	914
Facade_Immeuble	736
Ciel_Nuit	470
Nuage	435
Herbe	388
Feuillage	323
Gravier-Cailloux	288
Ciel_Crepusculaire	283
Terre	280
Mer	246
Cours_d_eau	178
Ciel_Brumeux	176

Tableau 6-5 : Concepts sélectionnés avec leur nombre d'occurrences.

Nous limitons le nombre d'images indexées à **50** pour deux raisons : la première est que l'indexation manuelle de 50 images par un utilisateur, même patient, est déjà à la limite du réalisme, l'autre raison étant que nous désirons surtout caractériser la réactivité initiale de l'apprentissage.

6.4.3 Résultats

Nous avons effectué cinq cycles apprentissage/classification, et non dix comme dans l'expérience précédente, car le test statistique d'égalité des espérances (test de Student) a révélé que ces données étaient déjà fortement significatives : la probabilité que les valeurs de précision obtenues avec 1-NN et les valeurs de précision obtenues avec notre approches aient été générées par la même source est P = 0.00137893, pour le rappel, cette probabilité est P = 6.9886E-08.

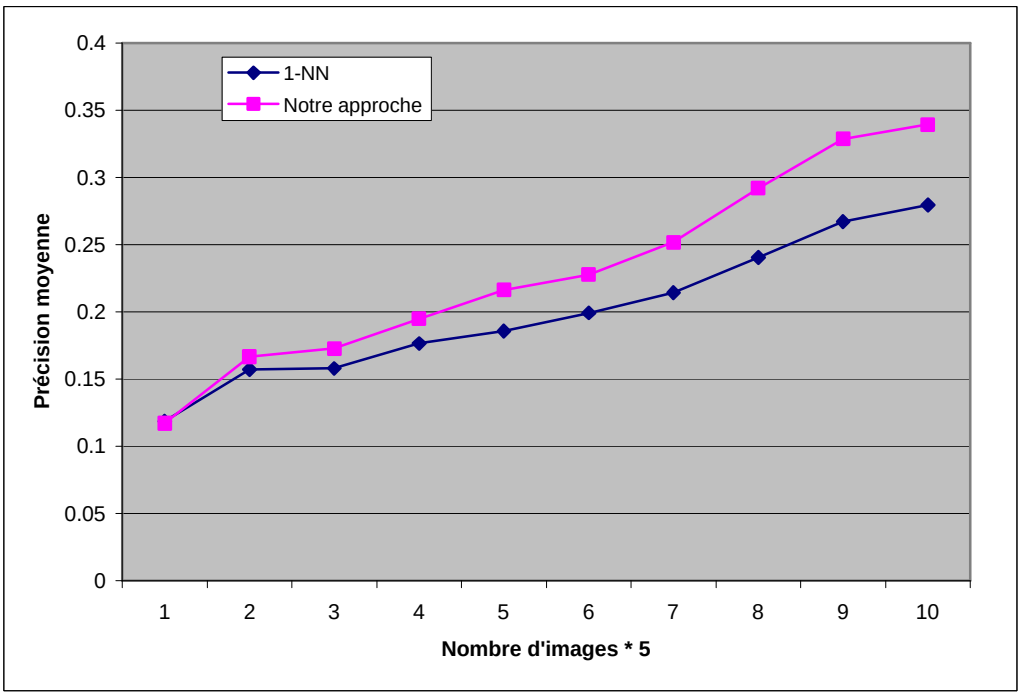


Figure 6-6 : Précision moyenne en fonction du nombre d'images indexées selon quinze concepts.

La Figure 6-6 montre la précision moyenne obtenue pour les deux méthodes en fonction du nombre d'images indexées. Ces courbes montrent que si les précisions initiales (après apprentissage de cinq images), sont quasi identiques (12 %), l'écart se creuse au fur et à mesure de l'augmentation de l'ensemble d'apprentissage, jusqu'à atteindre 6 % d'écart (28 % & 34 %). Même si cet écart est significatif, il reste faible.

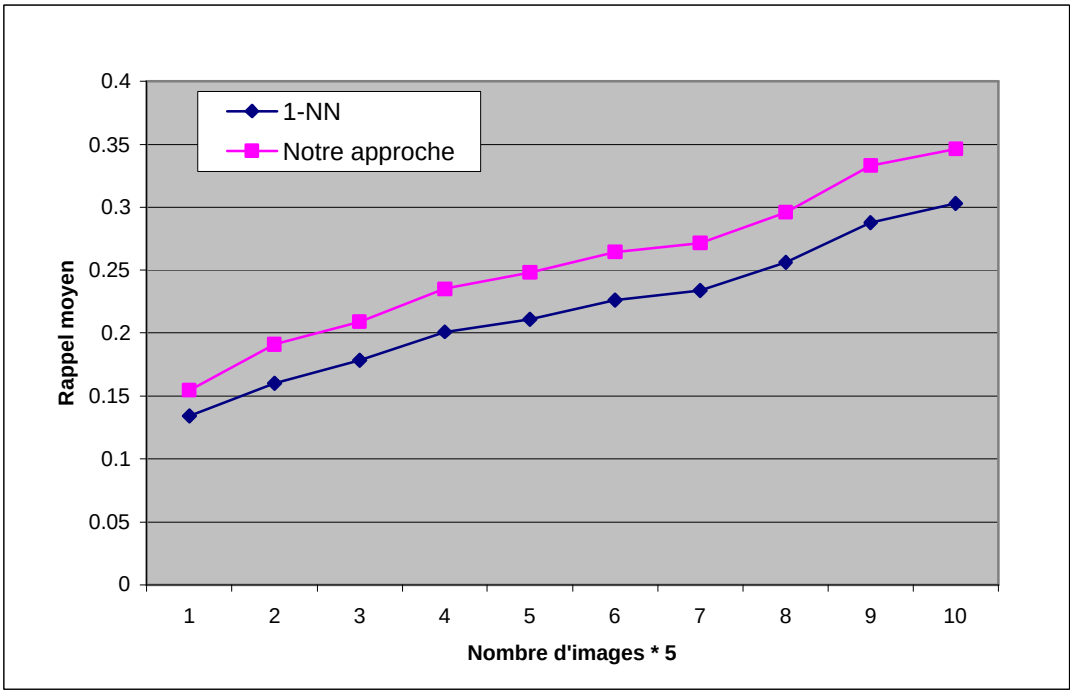


Figure 6-7 : Rappel moyen en fonction du nombre d'images indexées selon qunize concepts.

En ce qui concerne le rappel (Figure 6-7), les deux courbes sont décalées mais évoluent parallèlement. La différence entre ces deux courbes, même si elle est en faveur de notre approche, n'est que de 4 %

Il semble que cela s'explique par le nombre de concepts considérés. La confirmation de cette hypothèse est illustrée par la Figure 6-8 et la Figure 6-9 où le nombre de concepts appris est cette fois de 78. On voit que la différence entre les deux approches est encore plus atténuée (de l'ordre de 2 %). D'un point de vue de l'utilisateur, l'impact de l'algorithme utilisé par le système n'a aucune importance du point de vue qualitatif. D'un point de vue scientifique toutefois, une différence est intéressante si elle est significative (test de Student, résultats similaires à l'expérience avec quinze concepts). Même si les résultats sont similaires, les méthodes employées et les idées sous-jacentes sont totalement différentes et il n'est pas du tout exclu que les résultats puissent être très différents dans d'autres contextes.

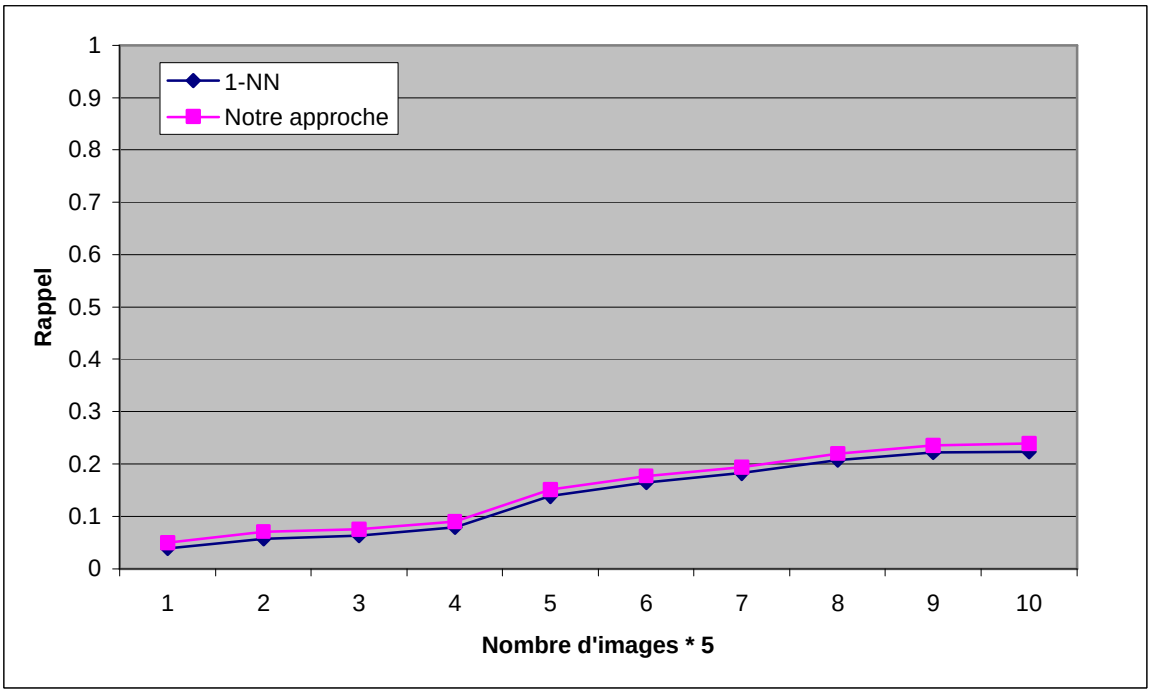


Figure 6-8 : Rappel en fonction de la taille de l'ensemble d'apprentissage, 78 concepts.

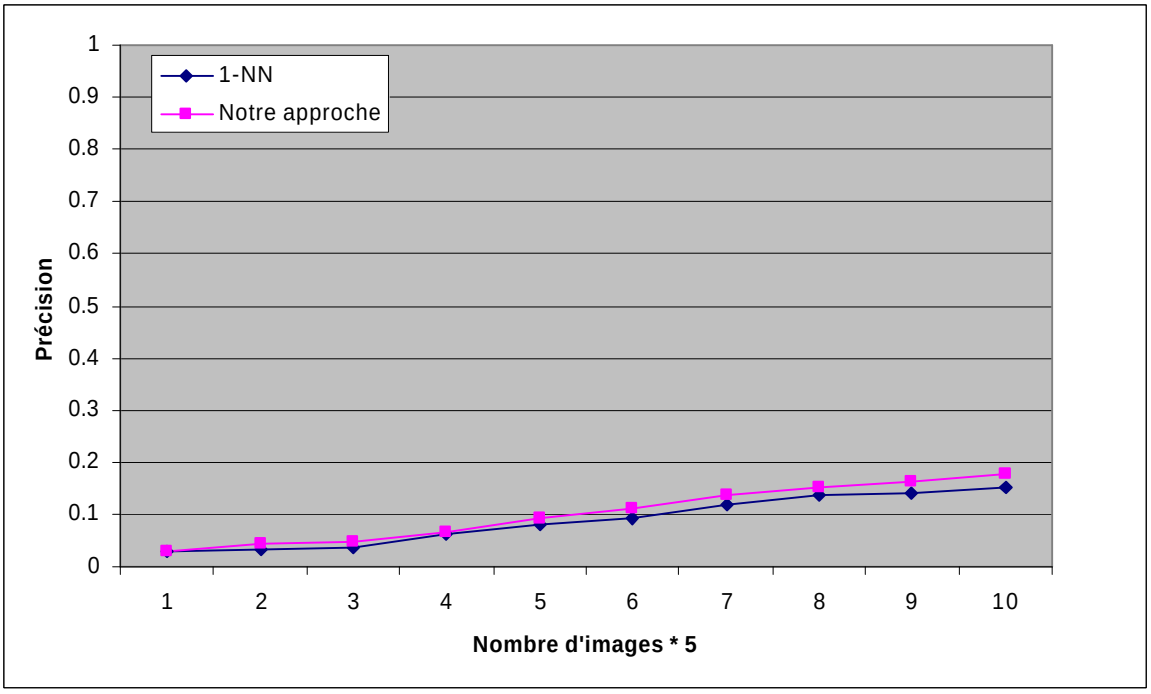


Figure 6-9 : Précision en fonction de la taille de l'ensemble d'apprentissage, 78 concepts.

6.4.4 Conclusion

Cette expérience a montré que notre approche permettait un apprentissage plus rapide que l'algorithme 1-NN, c'est-à-dire une meilleure réactivité. Cependant, cette amélioration dépend du nombre de concepts considérés et s'amoindrit à mesure que celui-ci augmente. La raison à cela est que lorsque le nombre de concepts est important, les confusions augmentent, les résultats en termes de rappel et de précision évoluent dans un intervalle plus petit et les différences relatives entre différentes approches sont donc elles aussi moins importantes. Il n'en reste pas moins que cette différence relative, en notre faveur, se maintient même lorsque le nombre de concepts considérés varie.

Un deuxième point, plus important, est que les calculs les plus lourds ont été effectués de manière non supervisée, c'est-à-dire sans recourir à l'utilisateur. L'apprentissage non supervisé, qui consiste à propager chaque vecteur dans le réseau pour le traduire en termes de régularités, puis à mettre à jour les cooccurrences entre ces régularités, requiert un temps de calcul proportionnel au carré du nombre de régularités activées. L'apprentissage supervisé, et la classification requièrent un temps de calcul qui n'est que linéaire par rapport au nombre de régularités activées, car dans ce cas il n'est pas tenu compte des cooccurrences possibles entre chaque couple de régularités. Or, c'est justement lors de ces deux phases que le temps de calcul est le plus critique, car elles impliquent l'utilisateur.

6.5 Apprentissage et variabilité intra classe

Les concepts appris ne sont pas tous égaux face à la variabilité de leurs instances. Typiquement, le concept « Ciel clair » présente une variabilité bien moindre que « Arbre » : il existe de nombreux types d'arbres, qui n'ont pas le même aspect selon leur âge, ou selon la saison et leur apparence sur une photographie dépend en plus de la distance et de l'angle à l'objectif, des conditions d'illumination, etc. Un « Ciel clair », par contre, ne varie principalement que par sa couleur. La question que nous nous posons ici est de savoir quel effet a cette variabilité sur l'apprentissage.

Nous avons utilisé pour cette expérimentation les régions correspondant aux neuf concepts les plus représentés dans la collection YC744 (voir Tableau 6-5). Pour

chacun des concepts, nous avons effectué un apprentissage non supervisé de régularités en sélectionnant aléatoirement *n* fois une instance parmi toutes les instances du concept. Le nombre n est la seule variable dans cette expérimentation.

6.5.1 Effets de la variabilité

Le Tableau 6-6 indique le nombre de nœuds créés pour chaque concept, en fonction du nombre d'exemples utilisés lors de l'apprentissage. On peut déjà remarquer que le nombre de nœuds augmente en fonction du nombre d'exemples, ce qui était prévisible. On voit également que le nombre de nœuds créés dépend du concept : au bout de 5 000 exemples d'apprentissage, la moyenne des nœuds par concept est 313 mais l'écart type est de 44,4. Le plus grand écart est entre le nombre de nœuds créés pour le concept « Ciel clair » et le nombre de nœuds du concept « Arbre avec feuilles », avec respectivement 245 nœuds contre 378. D'une manière générale, ce résultat montre la corrélation entre la variabilité et le nombre de régularités apprises. Il est intéressant de voir que notre méthode s'adapte dynamiquement (selon les données présentées) à la variabilité des concepts. Ce résultat était prévisible, dans la mesure où lorsque les régularités qui caractérisent un concept de faible variabilité ont été découvertes, les régularités « restantes » sont plus rares et leur découverte nécessite plus de temps. D'un autre côté, lorsqu'un concept présente une forte variabilité, il existe beaucoup de régularités à apprendre.

		Concept								
		Ciel clair	Feuillage	Gravier cailloux	Herbe	Arbre avec feuilles	Ciel couvert	Façade immeuble	Rocher	Terre
Nombre d'exemples	500	132	149	152	152	162	140	151	142	166
	1000	159	187	192	188	203	191	169	197	263
	1500	167	206	215	215	219	176	190	191	235
	2000	200	221	247	227	236	193	205	217	250
	2500	210	257	256	239	284	211	220	229	241
	3000	215	280	292	257	281	219	247	224	274
	3500	228	304	294	250	310	219	258	249	305
	4000	223	313	319	301	319	250	274	267	300
	4500	240	319	328	340	360	255	282	285	329
	5000	245	340	344	350	378	267	295	277	324

Tableau 6-6 : Nombre de nœuds créés par rapport à la taille de l'ensemble d'apprentissage.

Cette observation est confirmée par le Tableau 6-7 qui montre l'accroissement moyen ? du nombre de nœuds par concept (moyenne des différences successives) pour un apprentissage allant de 500 à 5 000 exemples. On voit que les accroissements les plus faibles concernent les concepts « Ciel clair », « Ciel couvert », « Rocher » et « Façade immeuble » qui sont des concepts présentant une faible variabilité (relativement aux autres). Les accroissements les plus forts concernent principalement la végétation (« Arbres avec feuilles », « Herbe », « Feuillage ») dont les instances varient beaucoup selon la situation géographique, la distance de prise de vue, la saison, etc.

	Ciel clair	Feuillage	Gravier cailloux	Herbe	Arbre avec feuilles	Ciel couvert	Façade immeuble	Rocher	Terre
?	12.6	21.2	21.3	22.0	24.0	14.1	16.0	15.0	17.6

Tableau 6-7 : Accroissement moyen du nombre de nœuds après chaque session d'apprentissage (+500 exemples).

Si l'on regarde de plus près le comportement de l'apprentissage pour quelques concepts (Figure 6-9), on voit qu'un concept de faible variabilité comme « Ciel clair » génère de moins en moins de régularités au fur et à mesure de l'apprentissage, contrairement à « Feuillage » ou « Façade immeuble » qui génèrent une quantité de régularités proportionnelle à la taille de l'ensemble d'apprentissage.

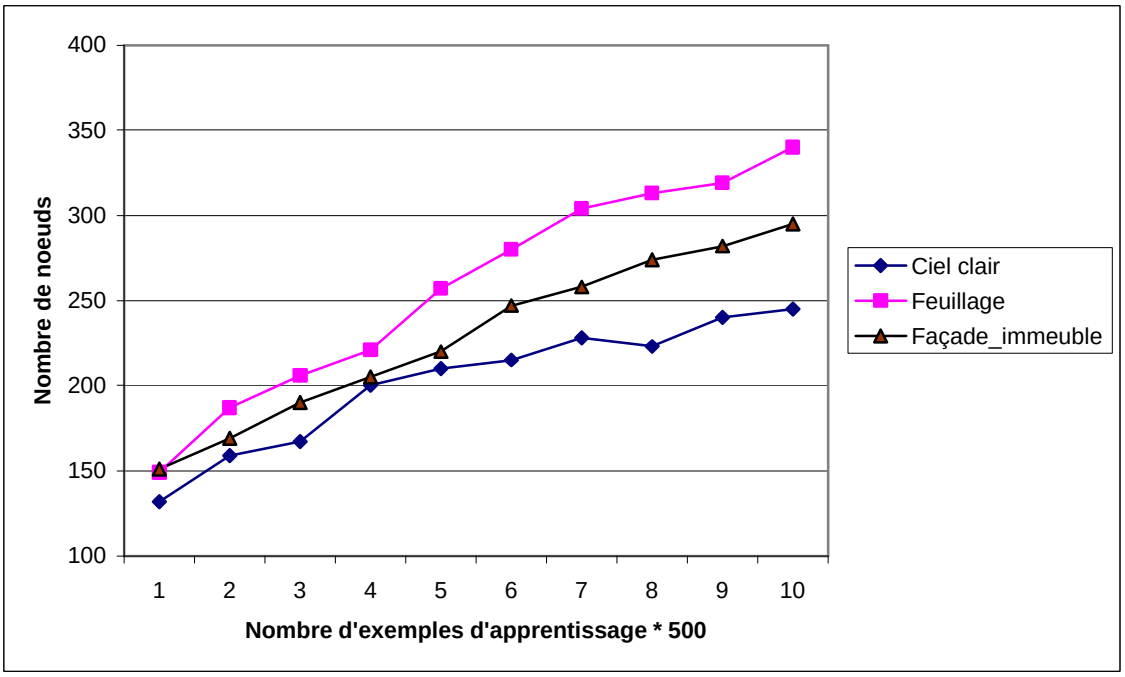


Figure 6-10 : Nombre de nœuds créés en fonction de la taille de l'ensemble d'apprentissage pour trois concepts de variabilités différentes.

6.5.2 Différentes topologies des réseaux de régularités

Nous venons de voir que du point de vue quantitatif, l'apprentissage diffère d'un concept à l'autre par le nombre de régularités apprises. Nous allons examiner maintenant si la nature du concept influe qualitativement sur l'apprentissage, c'est-à-dire sur la topologie des réseaux de régularités.

Les copies d'écran qui suivent montrent les régularités apprises *pour un concept donné*, après un apprentissage basé sur 5 000 exemples sélectionnés aléatoirement parmi toutes les instances de ce concept (un même exemple pouvant apparaître plusieurs fois).

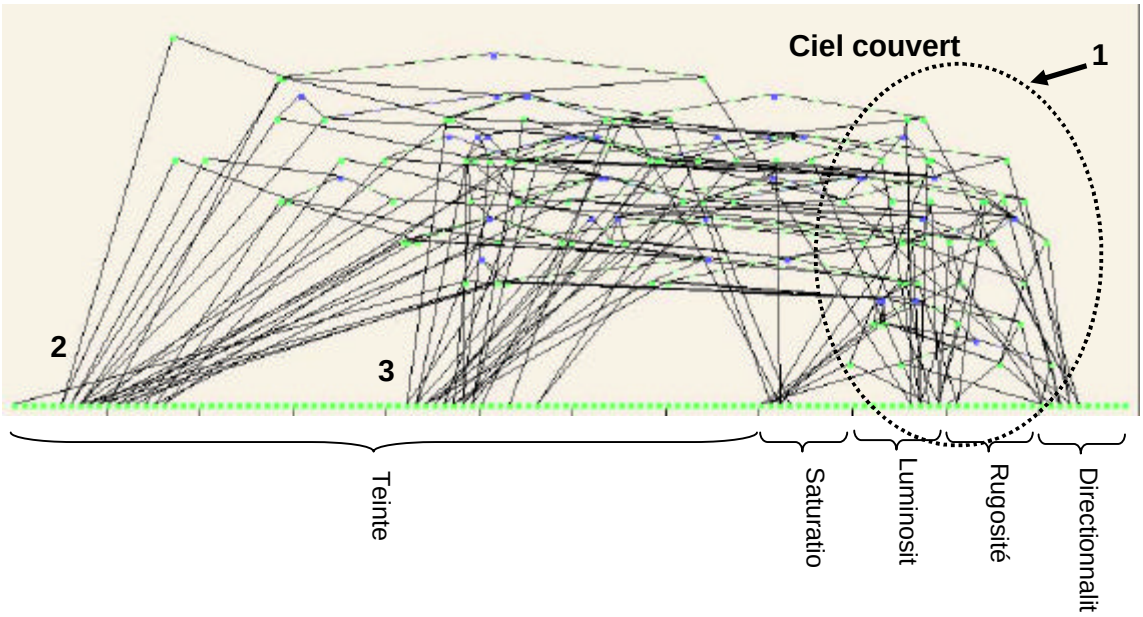


Figure 6-11 : Réseau « Ciel couvert », 252 nœuds.

La Figure 6-11 montre le réseau appris pour le concept « Ciel couvert ». Les premières régularités apprises (la zone 1 sur la figure, les plus proches de la couche d'entrée) concernent la texture (peu rugueux, peu directionnel), la luminosité (forte) et la saturation (faible). Ce n'est que bien après (c'est-à-dire loin de la couche d'entrée) que la teinte devient un constituant des régularités. Ces teintes, divisées en deux zones (2 et 3), correspondant à la couleur du ciel et à la couleur des nuages, varient beaucoup et ont donc moins d'importance. Notons également que la teinte génère beaucoup plus de nœuds 'Ou' que les autres traits, généralement, le nœud « contexte » de ces disjonctions étant une régularité formée de textures, luminosités et saturation, alors que les éléments considérés comme similaires dans ce contexte sont des teintes.

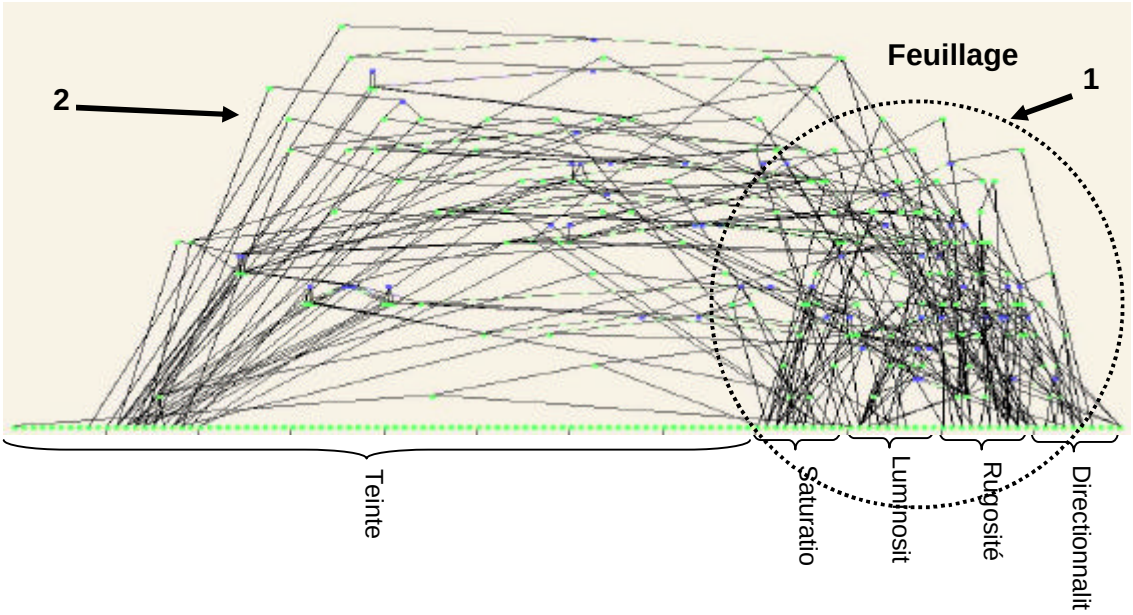


Figure 6-12 : Réseau « Feuillage », 347 nœuds.

Une topologie différente émerge lorsque le concept présente une forte variabilité dans ses instances. C'est le cas par exemple avec le concept « Feuillage » de la Figure 6-12. Cette fois, la variabilité est présente pour tout type de trait : toutes les saturations sont représentées, diverses luminosités, diverses rugosités (avec une tendance pour les fortes rugosités) et toute sorte de directionnalité. Seule la teinte se cantonne à une zone particulière, correspondant aux différentes teintes de vert. Cette variabilité généralisée conduit à une topologie particulière, où les régularités se forment de manière pyramidale (zone 1) : les premières régularités sont plutôt « intra-trait », puis regroupent deux traits, trois traits, etc. La forte densité de nœuds disjonctifs est représentative de cette variabilité et c'est elle qui permet, finalement, la construction de régularités constituées de tous les traits. La teinte semble ne pas être corrélée aux autres traits puisque elle est prise en compte très tard dans les régularités (zone 2). Il se forme donc un « pont » entre la pyramide (zone 1) et la teinte, au final les régularités de cette zone ont la plupart des traits pour constituants.

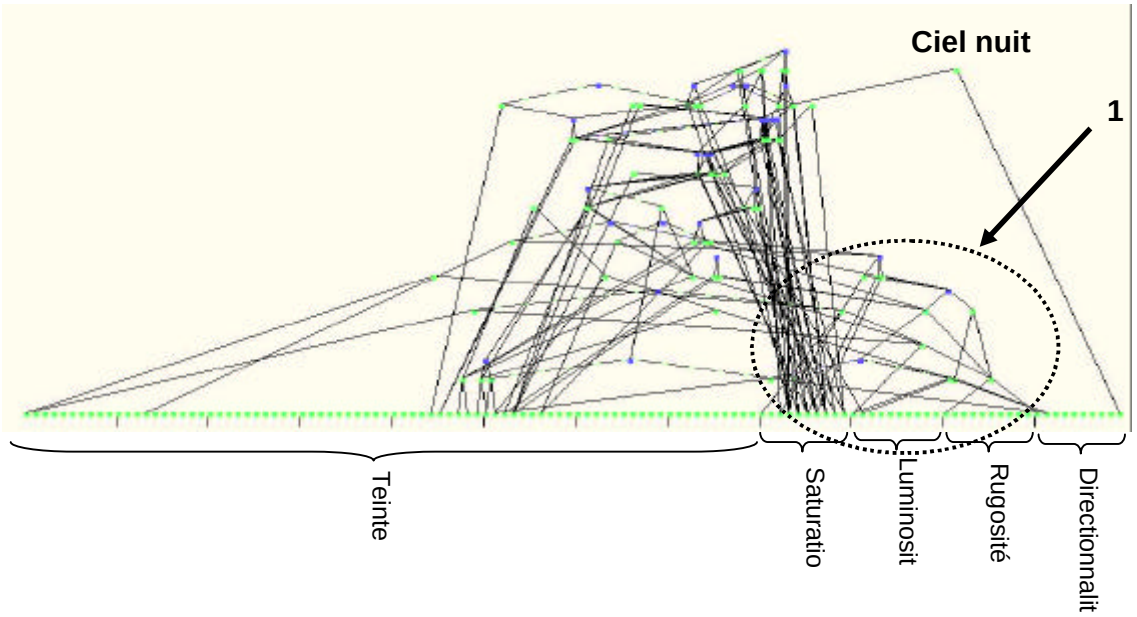


Figure 6-13 : Réseau « Ciel nuit », 219 nœuds

A l'opposé, lorsqu'un concept présente une très faible variabilité, comme pour le concept « Ciel nuit » (Figure 6-13), les régularités correspondantes sont apprises très vite et sont peu nombreuses. Pour le concept « Ciel nuit », on peut voir (zone 1) que des régularités composées de faible luminosité, faible rugosité, faible directionnalité sont apprises. C'est ensuite à partir de ces régularités que sont composées d'autres régularités tenant compte de la teinte, puis de la saturation. Remarquons qu'en pratique, lorsque l'on utilise l'espace de couleurs HSV pour caractériser une zone noire (sans teinte), les valeurs de saturation et de teinte obtenues ne sont pas pertinentes et sont donc réparties de manière quasi aléatoire (la teinte et particulièrement la saturation n'ont pas de sens en l'absence de couleur). On voit donc dans cet exemple, que les régularités formées des traits les plus pertinents sont apprises en priorité, suivies par des régularités tenant compte de traits moins pertinents, suivies finalement de celles ayant pour constituants des traits aléatoires.

6.5.3 Conclusion

Plasticité :

La nature du concept appris influe non seulement quantitativement sur le réseau (nombre de régularités) de régularités appris mais également qualitativement (topologie du réseau). En particulier, la *densité* de nœuds dans le réseau varie selon la quantité d'informations à apprendre. Nous avons vu par exemple pour le concept feuillage que de nombreuses régularités étaient créées pour rendre compte des relations entre textures, saturation et luminosité (i.e. forte densité) alors que les régularités ayant la teinte comme constituant étaient plus rares (densité plus faible). Cela est possible grâce à l'aspect dynamique de la méthode. En effet, si la topologie du réseau était fixée et arbitraire (comme une majorité de réseaux de neurones), cette densité serait uniforme dans tout le réseau, sans qu'il soit tenu compte de la pertinence des traits pour les concepts en question.

Pertinence des régularités :

Nous avons vu que les régularités sont apprises dans un ordre bien particulier : d'abord les régularités les plus fréquentes sont apprises, puis graduellement, les régularités incluent dans leurs constituants des traits atomiques (ou autres régularités) de moins en moins fréquents. Cet ordonnancement est crucial lors de la reconstruction (ou traduction) d'un vecteur : par exemple, pour qu'un vecteur active à 95 % une régularité de haut niveau (correspondant à un grand nombre, voire tous les traits atomiques du vecteur activés), il est indispensable que le vecteur en question active les régularités de la zone 1 (Figure 6-13) car celles-ci, ayant été construites tôt lors de l'apprentissage, sont des constituants de la plupart des autres régularités. A l'opposé, les valeurs du vecteur qui concernent la teinte, mais surtout la saturation, n'auront presque pas d'importance dans le calcul de l'activation, car ces traits ont été inclus tard dans les régularités et n'ont par conséquent que très peu de poids.

6.6 Données artificielles

Dans la deuxième partie de ce document, nous avons critiqué les techniques d'apprentissage traditionnelles, en soulignant que celles-ci n'étaient guère adaptées

aux problèmes d'apprentissage de type-2. Nous avons en outre affirmé que l'apprentissage de concepts à partir d'images était un problème de type-2. Pourtant, nos expérimentations montrent que notre approche et l'algorithme 1-NN présentent des performances comparables.

Nous pensons que cette similarité des performances est due à l'utilisation de traits de bas niveau qui, bien que basiques, induisent des liens statistiques directs entre la représentation des régions et les concepts cibles. La couleur, en particulier, contribue à ce phénomène : la couleur bleue est liée au ciel, la couleur verte aux champs, arbres et herbes, etc.

L'expérimentation que nous présentons maintenant a pour but de montrer plus clairement les capacités de notre approche à traiter un problème d'apprentissage de type-2.

6.6.1 Protocole

Nous avons généré artificiellement des données selon deux classes. La première classe, dite *positive*, correspond à un concept défini de manière principalement relationnelle. La deuxième classe, *négative*, est générée aléatoirement.

Les données d'apprentissage sont constituées de vecteurs de 100 booléens (b_1 à b_{100}), 500 vecteurs positifs et 1 000 vecteurs négatifs. Un mélange de 2 000 vecteurs, 50 % de positifs, 50 % de négatifs, a été généré pour la classification.

Pour générer un exemple positif, nous construisons une instance aléatoire d'un concept C prédéfini. Les booléens ne faisant pas partie de la définition du concept sont choisis aléatoirement. Le nombre de booléens impliqués par la définition des concepts a été fixé arbitrairement à 7.

6.6.2 Résultats

Notre algorithme a été capable d'assigner la bonne classe à 91 % des nouvelles observations présentées. La Figure 6-14 montre un exemple où le concept cible a été parfaitement appris, les nœuds supplémentaires ne représentent pas de « vraies » régularités et résultent d'un sur apprentissage.

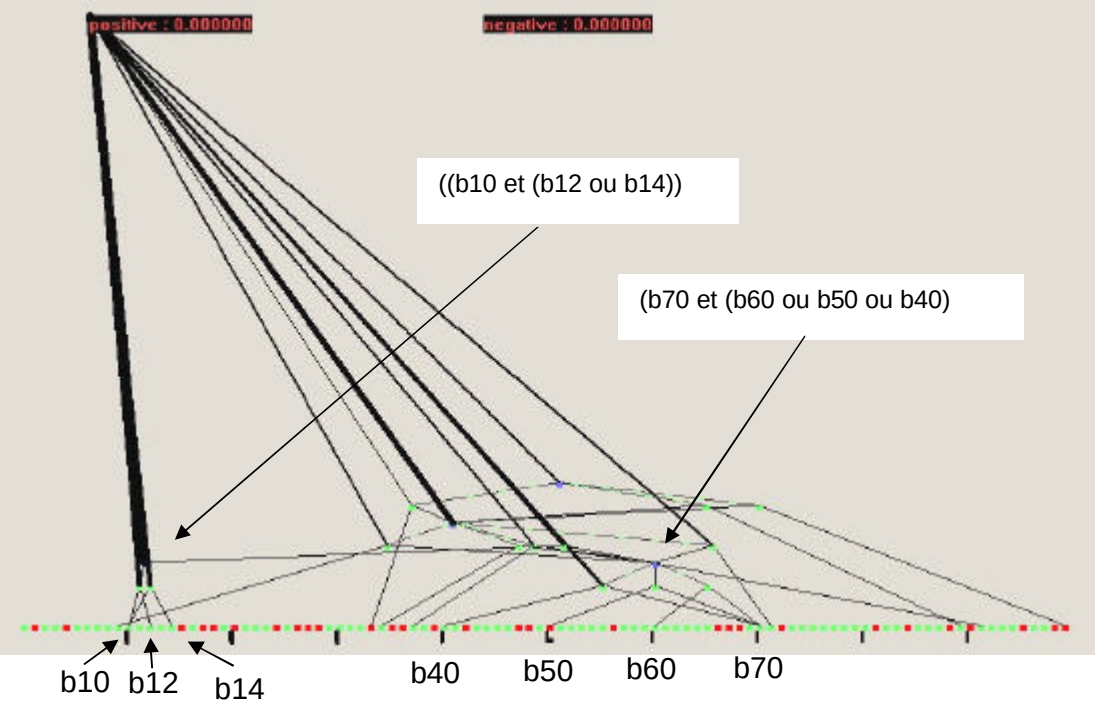


Figure 6-14 : Résultat de l'apprentissage du concept ((b10 et (b12 ou b14)) ou (b70 et (b60 ou b50 ou b40))

Sur les mêmes données, et avec le même protocole, l'algorithme 1-NN obtient un taux de classification correcte de 55 %, c'est-à-dire légèrement plus qu'une classification aléatoire. Ce résultat médiocre était prévisible, dans la mesure ou cet algorithme n'est pas adapté à l'apprentissage de concepts relationnel. De plus le nombre important d'attributs non pertinents dilue fortement l'influence que peuvent avoir les quelques attributs pertinents.

L'algorithme 1-NN n'étant pas un adversaire adapté dans cette expérimentation, nous avons également comparé notre résultat à l'algorithme SVM, capable d'une certaine forme de recodage *via* l'utilisation de fonctions noyaux.

Nous avons utilisé LIBSVM[43], une implémentation particulièrement efficace et qui propose en outre un script permettant l'optimisation automatique de tous les paramètres. Cet algorithme a obtenu un taux de 71 % (toujours sur les mêmes données).

[43] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

6.6.3 Conclusion

Cette expérimentation a montré la capacité de notre approche à apprendre des concepts définis de manière relationnelle. Ce type de concept étant caractéristique des problèmes d'apprentissage de type-2, un recodage des données est nécessaire. L'apprentissage non supervisé mis en œuvre a été capable de découvrir ces recodages grâce aux régularités présentes dans l'ensemble d'apprentissage. Une étude plus poussée sur des données synthétiques nous paraît nécessaire et nous aborderons ce sujet dans la conclusion générale.

6.7 Discussion et conclusion

Au cours de ces expérimentations (principaux résultats présentés Tableau 6-8) nous avons montré que l'approche que nous proposons est compétitive par rapport à l'algorithme 1-NN. Or, cet algorithme a la propriété d'être très robuste par rapport à la diversité des problèmes d'apprentissage et de donner systématiquement de bons résultats (« bon » dans le sens où ces résultats sont toujours proches ou meilleurs que les autres algorithmes testés). Par conséquent, la méthode proposée est performante dans notre contexte.

En termes de performance « absolue », les résultats obtenus, que ce soit avec notre approche ou avec 1-NN, sont clairement insuffisants : un tel système n'est pas suffisamment performant pour intéresser un public large et être utilisé couramment. Toutefois, cela n'était pas notre but *à court terme*, et nous pensons que ce n'est pas un but à court terme dans l'absolu. L'indexation robuste et sémantique des images requiert, au préalable, la résolution de problèmes plus fondamentaux comme la compréhension des mécanismes de la perception **ou** la compréhension/création de mécanismes d'apprentissage généraux aboutissant à cette perception.

Cependant, un point très positif est que dans notre approche, la complexité de l'apprentissage est *déplacée* vers l'apprentissage non supervisé, qui ne nécessite pas la vigilance de l'utilisateur. L'apprentissage supervisé, qui lui met nécessairement l'utilisateur à contribution, s'en trouve allégé, son rôle étant simplement d'établir des liens statistiques entre les concepts et les régularités. De la même manière, la classification est également très simple puisque elle consiste principalement en une propagation des vecteurs dans le réseau.

Type de données	Nombre de concepts	Comparé avec	Résultat relatif
Images "idéales"	4	1-NN	+5 % rappel +11 % précision
Images réelles	15	1-NN	+4 % rappel +6 % précision
	78		+2 % rappel +2 % précision
Données synthétiques	2	1-NN	+36 % classif. Correcte
		SVM	+20 % classif. Correcte

Tableau 6-8 : Principaux résultats des expérimentations

L'utilisation de traits de bas niveau, que nous souhaitons proscrire à moyen terme, rend le problème d'apprentissage partiellement de type-1 et permet à 1-NN d'obtenir des performances similaires à celles de notre approche. Cela rend la comparaison entre les deux approches difficile et, idéalement, il faudrait éliminer toutes les corrélations « évidentes » entre traits et concepts.

Afin de consolider nos résultats, nous avons donc mené une expérimentation sur des données synthétiques, avec des concepts définis de manière relationnelle. Cette fois, la supériorité de notre approche par rapport à 1-NN est beaucoup plus évidente. Celle-ci se montre même meilleure qu'une implémentation performante de SVM (disposant d'une procédure d'optimisation automatique des paramètres, que nous n'avons pas encore mis au point pour notre approche).

Quatrième partie

Conclusion & Perspectives

Chapitre 7

Conclusion

7.1 Synthèse & Contributions

7.1.1 La recherche d'images

Le cadre général de ce travail est celui de la *recherche d'information*, appliquée aux images.

Plus particulièrement, nous nous sommes intéressés au sous-problème de *l'indexation* des images. Nous avons vu, lors de l'état de l'art sur les SRIC (Système de Recherche d'Images par le Contenu) que cette indexation pouvait être faite à bien des niveaux. Ces différents niveaux se différencient principalement par le *niveau d'abstraction* qu'ils permettent. C'est d'ailleurs sur ce critère que nous avons basé notre taxonomie des SRIC. Parmi les systèmes étudiés, certains n'indexent les images que par *des traits de bas niveau*, et ne peuvent proposer que des requêtes par croquis ou image exemple. Ces approches sont dites « Signal ». D'autre système proposent une indexation (des images ou/et de régions de l'image) basée sur des *mots clefs*, ce qui permet bien sûr de formuler des requêtes sous la forme de mots clefs. Ces approches sont qualifiées de « Symboliques ». La limitation des SRIC symboliques est, outre les performances, le fait que les symboles soient prédéfinis arbitrairement et qu'ils ne correspondent pas nécessairement aux besoins des utilisateurs.

7.1.2 L'apprentissage

Nous avons choisi l'orientation *symbolique*, entre autres motivés par les études sur les comportements des utilisateurs de SRIC. Ces études montrent en effet que leurs besoins sont en général abstraits, c'est-à-dire loin des pixels ou même des traits de

bas niveau. De plus, nous avons voulu un système dans lequel l'utilisateur peut lui-même définir ses propres symboles. L'indexation symbolique d'une image implique la connaissance, et la reconnaissance de *concepts*. Or, nous ne sommes pas capables de définir explicitement ces concepts, entre autres car les processus aboutissant à leur reconnaissance sont principalement inconscients. La seule alternative envisageable nous a paru être *l'apprentissage automatique*.

7.1.3 Limitations des approches classiques

Nous avons vu que la plupart des algorithmes d'apprentissage utilisés en SRIC étaient du type *supervisé*. Ces algorithmes affinent donc leur hypothèse en utilisant à tout moment les exemples étiquetés disponibles. Nous avons argumenté sur le fait que cette approche, de type *descente de gradient*, n'était applicable qu'aux problèmes d'apprentissage de *type-1*. Nous avons remarqué que pour contourner ce problème, la recherche progressait parallèlement dans la conception de méthodes toujours plus complexes pour représenter les images. Cette stratégie, bien que permettant d'obtenir des résultats, nous paraît intrinsèquement limitée pour les raisons suivantes :

- On ne sait comment sont représentés nos propres concepts, il est donc difficile de représenter les images de manière adaptée.
- Reconnaître de nouveaux concepts requiert une importante quantité de travail car il faut inventer une représentation adaptée.
- Les efforts dépensés pour créer de nouvelles représentations très spécifiques ne profitent pas ou peu à d'autres domaines[44], même connexes, comme la reconnaissance de la parole.

7.1.4 Un autre point de vue

Nous avons vu que les problèmes d'apprentissage les plus intéressants sont généralement de *type-2*, c'est-à-dire lorsque les concepts sont définis de manière

[44] Certaines avancées suffisamment génériques sont bien sûr profitables à de nombreux domaines, comme les Transformées de Fourier.

relationnelle. La résolution de tels problèmes nécessite un *recodage* des données. Nous avons soutenu que ce recodage ne pouvait être appris de manière purement supervisée. La raison principale est que cet apprentissage nécessite de nombreuses étapes qui n'apportent pas nécessairement de gain vis-à-vis des concepts cibles. Nous avons donc suggéré que l'apprentissage non supervisé est nécessaire initialement[45]. Son rôle est d'apprendre une *représentation* des données telle que l'apprentissage supervisé puisse être facilité, voire rendu possible.

7.1.5 Hypothèses

Ce point de vue requiert cependant que les hypothèses suivantes soient vérifiées :

- Les données contiennent des *régularités* (i.e. ne sont pas générées aléatoirement). Ces régularisées constituent une « prise », un point de départ rendant possible l'apprentissage.

- Tout comme les régularités, les concepts sont une propriété des données. Par conséquent, l'apprentissage des mêmes concepts est possible par des entités différentes. En particulier, une machine en est capable, bien que le support du processus d'apprentissage soit très différent de celui d'un humain.

- Les régularités et concepts peuvent être appris de manière *hiérarchique*, c'est-à-dire par une méthode *constructive* qui représente les nouvelles données en termes des régularités apprises dans le passé.

La première hypothèse est considérée comme vraie. L'argument principal justifiant la validité de la seconde hypothèse est que si la connaissance d'un concept permet de faire des *prévisions justes*, c'est que ce concept est effectivement une propriété des données. Or, nous sommes généralement capables d'effectuer des prévisions, qui s'avèrent vérifiées, sur la base de la perception visuelle[46]. L'argument que nous avons avancé pour la troisième hypothèse est que les entités physiques, qui

[45] L'apprentissage non supervisé n'étant pas réduit au Clustering…

[46] Par exemple, je perçois visuellement un objet, je le reconnais comme étant une instance d'un concept (par exemple "Arbre"), et je peux vérifier, *via* d'autres sens ou expériences, que cette prévision est juste. Si cette expérience se reproduit de nombreuse fois, pour des instances différentes, c'est que le concept visuel « Arbre » que je possède correspond en fait à un concept universel « Arbre ».

correspondent aux instances des concepts, sont elles-mêmes générée de manière constructive et hiérarchique : soumis à des lois physiques, des sous-systèmes stables s'organisent, *via* leurs propriétés macroscopiques plus ou moins invariantes, en système stables plus complexes, etc. Cette troisième hypothèse est celle qui a le plus influencé la définition de notre *biais inductif*.

7.1.6 Biais inductif de notre approche

Un algorithme d'apprentissage peut être vu comme un explorateur qui cherche une certaine hypothèse dans un espace d'hypothèses généralement très grand. Le biais inductif rassemble tous les éléments, exceptées les données, qui le guident dans cette recherche. Notre biais est défini par les contraintes suivantes :

- Les hypothèses sont représentées par des formules propositionnelles limitées aux conjonctions et aux disjonctions (pas de négation, ni de quantificateurs universels).
- Les régularités sont apprises par agglomération de régularités existantes.
- Une régularité remplace, lorsqu'elle est présente, ses composants.
- Lorsqu'il y a ambiguïté entre deux régularités de même niveau hiérarchique (même nombre de feuilles), la plus fréquente dans l'ensemble d'apprentissage est choisie.

7.1.7 Algorithmes

Notre algorithme d'apprentissage *non supervisé* opère de la manière suivante. Chaque vecteur exemple est traduit en termes des régularités connues (initialement, il ne s'agit que de traits atomiques). Les cooccurrences entre chaque paire de régularités présentes sont passées en revue et leur nombre d'occurrences est incrémenté. Lorsqu'un nombre d'occurrence dépasse un seuil, un noeud de type conjonctif est créé. Lorsque'un noeud N est un constituant commun à plusieurs conjonctions, un noeud de type disjonctif est créé et N est appelé noeud contexte.

Notre algorithme d'apprentissage *supervisé* traduit les vecteurs en termes de *toutes* les régularités qu'ils contiennent. Les cooccurrences entre régularités et concepts sont comptabilisées.

L'algorithme de *classification* affecte pour chaque vecteur inconnu une liste de scores à chaque concept. Chacun de ces scores est une combinaison linéaire du degré d'activation de chaque régularité présente dans le vecteur, pondérée par son degré de discrimination vis-à-vis du concept considéré.

7.1.8 Résultats

Nous avons comparé notre approche à l'algorithme des k plus proches voisins (kNN), avec k=1. L'algorithme 1-NN ne requiert l'ajustement d'aucun paramètre[47] et son implémentation est aisée.

Nous obtenons des performances (rappel/précision) supérieures à celle de 1-NN sur deux collections d'images (Corel et collection personnelle). D'un point de vue absolu, les performances obtenues ne sont pas extraordinaires, mais étant donné les traits de bas niveau utilisés et surtout la « jeunesse » de notre approche, nous les considérons très prometteuses, et non inquiétantes.

Une expérience menée sur des données synthétiques montre que notre approche surpasse considérablement 1-NN et largement SVM. Dans cette expérience, le concept à découvrir était défini de manière presque purement relationnelle. C'est pourquoi l'échec de 1-NN étant prévu, nous avons cherché un adversaire plus juste. SVM a été sélectionné pour sa capacité à recoder l'espace initial de représentation, *via* l'exploitation des fonctions noyaux.

Ces expérimentations suggèrent que l'approche proposée ici est compétitive sur des problèmes d'apprentissage classiques. Nous entendons par problème « classique » un problème d'apprentissage initialement de type-2, rendu partiellement de type-1 par l'utilisation de traits de bas niveau pertinents par rapport aux concepts cibles.

[47] Il faut toutefois sélectionner une mesure de similarité. De plus, cet algorithme peut être amélioré de diverses manières : représentation arborescente des données, sélection/pondération des attributs, etc.

Cependant, elles suggèrent également des performances très prometteuses sur des problèmes d'apprentissage relationnels de type-2.

7.2 Perspectives

Les travaux présentés ici posent au moins autant de questions qu'ils apportent des réponses, et cela nous paraît être de bon augure. Même parmi les « réponses », nombreuses sont celles qui devraient être approfondies. Pour la suite de ces travaux, les perspectives suivantes nous semblent prioritaires :

7.2.1 Expérimenter

A cours terme, il nous paraît indispensable de rentrer dans une boucle [*expérimentations→ mise au point→ expérimentation…]* en utilisant initialement des données « simples » et contrôlées, dont nous augmenterons progressivement la complexité. Nous avons certes effectué un certain nombre de ces boucles, mais nous pensons néanmoins que l'application aux collections d'images personnelles, bien qu'attractive, reste un objectif à long terme. Nous pensons en effet que le problème général de la recherche d'image par le contenu est un problème *IA-complet*[48] et qu'il est certainement plus constructif de s'intéresser, comme beaucoup le font, à des problèmes moins ambitieux. Nous pensons en particulier multiplier les expérimentations dans lesquelles les données seront initialement synthétiques, ce qui nous permettra d'une part d'observer en détails le comportement de notre approche, et d'autre part, de la modifier en conséquence.

Une fois l'approche complètement « domptée » dans des domaines limités et contrôlés, il sera envisageable de mener des expérimentations mettant en jeu des utilisateurs et des concepts humains (i.e. généralement très difficiles).

[48] Traduction de **AI-complete**, qui par analogie au concept de NP-complétude en théorie de la complexité, caractérise les problèmes dont la difficulté est équivalente à celle du problème central de l'Intelligence Artificielle, i.e. rendre les machines aussi intelligentes (au sens du test de Turing par exemple) que l'homme.

7.2.2 Utiliser la prévision comme évaluation

Il serait certainement profitable de créer un protocole expérimental ne nécessitant pas de données étiquetées. Il serait par exemple possible d'évaluer l'apprentissage non supervisé en utilisant comme critère la justesse de certaines prévisions. Ces prévisions pourraient par exemple concerner des attributs manquants (mais connus du processus d'évaluation), dont l'algorithme essaierait de prédire les valeurs en fonction des attributs connus. La principale difficulté proviendra probablement de la mesure de similarité utilisée pour comparer la *prédiction* et la *vérité*, à moins que cette comparaison ne soit stricte.

7.2.3 Eliminer la notion de traits arbitraires

De notre point de vue, cette perspective est à la fois la plus ambitieuse et la plus passionnante. Par « éliminer les traits », nous entendons représenter les images (ou autres types d'informations) de la manière la plus simple, tout en conservant un maximum d'information. L'objectif de l'apprentissage non supervisé étant justement de créer, à partir des seules données, les traits ou représentations appropriés. Pour l'image, cette représentation serait probablement sous la forme d'une simple matrice de pixels[49].

Bien entendu, il ne nous paraît pas réaliste d'utiliser *directement* l'approche présentée ici, en passant par exemple d'une couche d'entrée de 120 nœuds à une couche de plusieurs millions de nœuds. Tout comme l'œil humain qui se focalise inconsciemment sur des régions particulièrement saillantes du champ visuel, nous envisageons un apprentissage non supervisé à double objectif : le premier consisterait à apprendre constructivement une représentation hiérarchique des régularités présentes dans les données (ce que nous avons fait jusqu'à présent). Le second objectif serait d'apprendre à choisir le lieu et l'échelle de focalisation, à partir de régularités plus grossières, de manière à ce que la reconstruction de l'image en termes de régularités soit *plus simple* en ce lieu qu'à proximité de celui-ci. Ces deux objectifs devraient être appris *parallèlement :* une possibilité serait d'introduire un nouveau type de nœuds « moteurs » qui, activés, provoquerait un changement de

[49] Certes, un pixel peut être vu comme un "trait". Nous entendons par trait les représentations de l'image qui nécessitent des calculs et qui perdent une partie de l'information initiale.

focus. Une simplification supplémentaire, et probablement nécessaire, serait de limiter spatialement les cooccurrences apprises. Il faudra alors définir une hypothèse supplémentaire sur l'indépendance des informations spatialement distantes, tenant compte toutefois du niveau d'abstraction considéré.

7.2.4 Approfondir la réflexion sur notre biais inductif

L'approche que nous proposons repose sur un certain nombre d'hypothèses. Ces hypothèses permettent de restreindre et de guider l'apprentissage. Ensemble, elles définissent le biais inductif, indispensable à tout « apprenant » homme ou machine.

Cependant, la véracité de ces hypothèses n'est absolument pas évidente. L'hypothèse que nous avons nommée « Le tout remplace les parties », qui est une des clefs de voûte de notre approche, nécessite par exemple que les données aient été générées par agrégation hiérarchique de composants stables[50]. Si ce n'est pas le cas, il ne sera pas possible de *reconstruire* ces données par agrégation de régularités statistiques, de manière également hiérarchique.

L'hypothèse « d'universalité des concepts » que nous avons formulée, considérant que les concepts sont une propriété intrinsèque des données et ne dépendent que peu de l'observateur, qui ne fait que *tendre* vers ces concepts, est une idée que nous allons approfondir. Si cette hypothèse est vérifiée, alors l'apprentissage non supervisé est possible ou plutôt, les données elles-mêmes peuvent en quelque sorte superviser l'apprentissage, sans que le recours à un expert humain soit nécessaire.

7.2.5 Autres types de données

Mis à part les traits de bas niveau et l'aspect « spatial », l'approche décrite n'est pas particulièrement spécifique aux images. Nous avons dans notre liste d'objectifs à moyen terme, le projet d'appliquer notre travail à des données de types différents, en particulier dans le domaine des langues naturelles (texte ou parole). Il nous semble en effet que les représentations utilisées seraient également adaptées à la structure des langues naturelles. Les régularités conjonctives décrivent des structures

[50] Dans le mot "stable" se cache la notion de similarité locale, dépendante du contexte.

imbriquées et pourraient représenter des structures syntaxiques. Les régularités disjonctives représentent la notion de similarité locale (i.e. dans un contexte précis) et pourraient s'avérer tout à fait pertinentes pour assouplir intelligemment les structures apprises, en apprenant comment généraliser à partir des données. Comme pour les images, une partie de la sémantique pourrait finalement émerger de l'apprentissage parallèle d'une structure et d'un ensemble de similarités locales à chaque niveau de cette structure.

Les images, la musique, la voix, les langues naturelles ou encore les séquences d'ADN ont comme point commun la présence de structures, et également le fait de posséder une multitude de similarités locales. De ces structures, articulées par ces similarités contextuelles, émergent finalement les concepts, et donc la sémantique qui nous intéresse en particulier pour la recherche d'information. Nous pensons que l'apprentissage non supervisée de ces structures, concepts et sémantique sera bientôt possible par des machines, ce qui transformera le domaine de la recherche d'informations.

Références bibliographiques

[Aiz64] M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821--837, 1964.

[Ant02] Sameer Antani, Rangachar Kasturi, and Ramesh Jain, "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video", in *Pattern Recognition*, Volume 35, Issue 4, Pages 945-965, April 2002.

[Bez93] J.C. Bezdek, Fuzzy Models - What Are They, and Why? *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 1, February 1993.

[Bis03] Stéphane Bissol, Philippe Mulhem, Yves Chiaramella, Dynamic Learning of Indexing Concepts for Home Image Retrieval, in *Content-Based Multimedia Indexing (CBMI2003)*, Rennes (France), pp87-93, 22-24 septembre, 2003.

[Bis04] Stéphane Bissol and Philippe Mulhem and Yves Chiaramella, Towards Personalized Image Retrieval, in *2nd International Workshop on Adaptive Multimedia Retrieval*, Valencia, Spain, pp89-102, August, 2004.

[Brad00] Ben Bradshaw. Semantic Based Image Retrieval: A Probabilistic Approach. *ACM Multimedia 2000, p.167-176*, Oct. 2000.

[Bur98] Burges, C. «A tutorial on support vector machines for pattern recognition. " *Data Mining and Knowledge Discovery* 2(2): 121-167, 1998.

[Car03] Gail A. Carpenter, Stephen Grossberg: ADAPTIVE RESONANCE THEORY *The Handbook of Brain Theory and Neural Networks*, Second Edition, 2003.

[Car99] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, ,J. Malik, Blobworld: a system for region-based image indexing and retrieval," *Third Int. Conf. on Visual Information Systems*, pp 509-516, D. P. Huijsmans, A. W.M. ,Smeulders (eds.), Springer, Amsterdam, The Netherlands, 1999.

[Cha97] S.-F Chang, J.R. Smith, M. Beigi, et A. Benitez, "Visual Information Retrieval from Large Distributed Online Repositories", *Comm. ACM, vol. 40*, no. 12, pp. 63-71, 1997.

[Che03] Yixin Chen, James Z. Wang, and Robert Krovetz, ``An Unsupervised Learning Approach to Content-Based Image Retrieval," *Proc. IEEE International Symposium on Signal Processing and its Applications*, pp. 197-200, Paris, France, July 2003.

[Che03b] Yixin Chen James Z. Wang Robert Krovetz, Content-based image retrieval by clustering, *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, Pages: 193 - 200, 2003

[Che95] Chellappa, R., Wilson, C.L., and Sirohey, S. , Human and Machine Recognition of Faces: A Survey. *Proceedings of the IEEE*, 83(5):705-740, 1995.

[Cla97] Clark, A & Thornton, C., Trading spaces: Computation, representation, and the limits of uninformed learning. Behavioral and Brain Sciences 20 (1): 57-92, 1997.

[Cla00] Angus A Clark, Tom Troscianko, Neill W Campbell and Barry T Thomas. A comparison between human and machine labelling of image regions. *Perception,volume 29* (9): 1127--1138, September 2000.

[Cuc98] Rita Cucchiara, Genetic algorithms for clustering in machine vision. *Machine Vision and Applications* 11: 1-6 Springer-Verlag, 1998.

[Del99] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, 1999.

[Dem99] A. Demiriz, K.P. Bennett, and M.J. Embrechts. 1999. Semi-supervised clustering using genetic algorithms. *R.P.I. Math Report No.9901*, Rensselaer Polytechnic Institute, Troy, New York, 1999.

[Dor03] A. Dorado and E. Izquierdo, "An approach for supervised semantic annotation", Proc. *Workshop on Image Analysis for Multimedia Interactive Services*, WIAMIS 2003, 9-11 Apr. 2003.

[Dub03] S. Dubnov, G. Assayag, O. Lartillot, G. Bejerano, " Using Machine-Learning Methods for MusicalStyle Modeling ", *IEEE Computer*, Vol. 10, n° 38, p.73-80, October 2003.

[Eak99] John P Eakins and Margaret E Graham Content-based Image Retrieval. *A report to the JISC Technology Applications Programme.* Institute for Image Data Research, University of Northumbria at Newcastle. January 1999.

[Emr04] M. Emre Celebi, Y. Alp Aslandogan, "Content Based Image Retrieval Incorporating Models of Human Perception", *IEEE International Conference on Information Technology, Coding and Computing*, Las Vegas, NV, April 2004.

[Fas99] Daniel Fasulo, *An Analysis of Recent Work on Clustering Algorithm*, Technical Report 01-03-02, University of Washington, 1999

[Fel03] J. C. Felipe, A. J. M. Traina, and C. T. Jr., "Retrieval by Content of Medical Images Using Texture for Tissue Identification", *Proc. 16th IEEE Symposium on Computer-based Medical Systems* (CBMS'2003), New York, June 26-27, pp. 26-27, 2003.

[Fli95] M. Flickher, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D.Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System," *IEEE Computer, vol. 28*, no. 9, pp. 23-32, Sept. 1995.

[Fou02] J. Fournier, M. Cord,S. Philipp-Foliguet, " Stratégie interactive d'exploration pour la recherche d'images par le contenu ", *RFIA 2002, volume 1*, p. 211-220, Angers, France, janvier 2002.

[Gob01] Gobet, F., Lane, P. C. R., Croker, S., Cheng, P. C-H., Jones, G., Oliver, I. & Pine, J. M. Chunking mechanisms in human learning. *TRENDS in Cognitive Sciences*, 5, 236-243. 2001.

[Gos04] Philippe H. Gosselin, A Comparison of Active Classification Methods for Content-Based Image Retrieval. Pages: 51 - 58, *CVBD* 2004

[Gos99] Jaideva C. Goswami et Andrew K. Chan. " Fundamentals of Wavelets: Theory, Algorithms, and Applications ". *Wiley Series in Microwave and Optical Engineering.* Wiley Interscience ISBN 0-471-19748-3, 1999.

[Gri05] Grira, N., Crucianu, Boujemaa, N. Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration, *IEEE International Conference on Fuzzy Systems* (Fuzz'IEEE 2005), pp 867-872, Reno, Etats-Unis, 22-25 mai 2005

[Grü05] P.Grünwald, A Tutorial introduction to the minimum description length principle. In: *Advances in Minimum Description Length: Theory and Applications* (edited by P. Grünwald, I.J. Myung, M. Pitt), MIT Press, 2005.

[Gué03] Guérin-Dugué A., Ayache S., Berrut C., Image retrieval : a first step for a human centered approach, in *Fourth Pacific-Rim Conference on Multimedia* (ICICS-PCM 2003), Singapore, 15-18 December 2003, 2003.

[Hay97] Hayashi, T., Hagiwara, M, "An image retrieval system to estimate impression words from images using a neural network", 1997-IEEE *International Conference on Systems, Man, and Cybernetics-Computational Cybernetics and Simulation*, vol 1, 150-5, IEEE, New York, NY, 1997.

[Hin99] Geoffrey Hinton and Terrence J. Sejnowski , Unsupervised Learning and Map Formation: Foundations of Neural Computation. *MIT Press*, 1999.

[Hua97] Huang, J., Kumar, S., Mitra, M., Zhu, W.-J., and Zabih, R. Image indexing using color correlogram. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition.* , pages 762-768. 1997.

[Ike00] Takashi Ikeda, Masafumi Hagiwara: Content-Based Image Retrieval System Using Neural Networks. *Int. J. Neural Syst.* 10(5): 417-424, 2000.

[Ing00] Ingemar J. Cox, Matthew L. Miller, Thomas P. Minka, Thomas Papathomas, and Peter N. Yianilos. The bayesian image retrieval system, PicHunter: Theory, implementation and psychophysical experiments. *IEEE Transactions on Image Processing*, pages 20-37. 2000.

[Iqb02] Q. Iqbal and J. K. Aggarwal, CIRES: A System for Content-based Retrieval in Digital Image Libraries , *Invited session on Content Based Image Retrieval: Techniques and Applications International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Singapore, pp. 205-210, December 2-5, 2002.

[Jai00] A. Jain, P. Duin, and J. Mao, Statistical pattern recognition: A review," *IEEE Transactions on PAMI* 22(1), pp. 4-37, 2000.

[Jin03] Feng Jing, Mingjing Li, Lei Zhang, Hong-Jiang Zhang, Bo Zhang, Learning in Region-Based Image Retrieval, *CIVR, pages 198—207.* 2003.

[Joh94] Michael P. Johnson, Pattie Maes, and Trevor Darrel, "Evolving Visual Routines," *Artificial Life*, 1(4):373--389, 1994.

[Kem04] C. Kemp, T. L. Griffiths, S. Stromsten, and J. B. Tenenbaum. Semi-supervised learning with trees. *Advances in Neural Information Processing Systems 16, 2004.*

[Kim03] DeokHwan Kim & ChinWan Chung , Qcluster: Relevance Feedback Using Adaptive Clustering for Content-Based Image Retrieval, *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp 536-544. 2003.

[Kuh00] Kuhl, P. K. A new view of language acquisition. *Proceedings of the National Academy of Science*, 97, 11850-11857. 2000.

[Lew96] Michael S. Lew, D. P. Huijsmans, and Dee Denteneer. Content based image retrieval: KLT, projections, or templates. In A. W. M. Smeulders and R. Jain, editors. *Image Databases and Multi-Media Search, proceedings of the First International Workshop* IDB-MMS'96, Amsterdam, The Netherlands. Amsterdam University Press, pages 27-34. , August 1996

[Li03] Jia Li, James Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25*, pages 10, 14. 2003.

[Li04a] Jia Li, James Z. Wang, "Studying digital imagery of ancient paintings by mixtures of stochastic models," *IEEE Transactions on Image Processing*, vol. pages 12, 15., 2004.

[Li04b] T. Li, M. Ogihara ,Music Artist Style Identification by Semi-supervised Learning from both Lyrics and Content (University of Rochester) Page 364. 2004.

[Lim01a] Joo-Hwee Lim: Building Visual Vocabulary for Image Indexation and Query Formulation. *Pattern Anal. Appl*. 4(2-3): 125-139, 2001.

[Lim01b] J. H. Lim. Fuzzy object patterns for visual indexing and segmentation. In *Proc. FUZZ-IEEE*, pages 77-80, 2001.

[Lon02] Fuhui Long, Hongjiang Zhang, David D. Feng: Fundamentals of Content-based Image retrieval, in *Multimedia Information Retrieval and Management - Technological Fundamentals and Applications*, D. Feng, W.C. Siu, and H.J.Zhang. (ed.), Springer, 2002.

[Lon03] Longbin Chen, Bao-Gang Hu, Lei Zhang, Mingjing Li, HongJiang Zhang: Face Annotation for Family Photo Album Management. *International Journal of Image and Graphics*, Volume 3, pp 81-94. 2003.

[Ma99] Wei-Ying Ma and B. S. Manjunath. Netra: A toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184-198, 1999.

[Mar98] Markkula, M and Sormunen, E (1998) "Searching for photos - journalists' practices in pictorial IR", presented at *The Challenge of Image Retrieval research workshop*, Newcastle upon Tyne, 5 February 1998.

[McC95] McCorry, H and Morrison, I O "Report on the Catechism project." National Museums of Scotland. 1995

[Mec95] Mourad Mechkour, Catherine Berrut, and Yves Chiaramella. Using conceptual graph framework for image retrieval. In *International conference on MultiMedia Modeling* (MMM'95), Singapore, pages 127--142, 14-17 November 1995.

[Min96] T. P. Minka An Image Database Browser that Learns from User Interaction , *Master of Engineering Thesis*, 1996.

[Mir98] Mihran Tuceryan, Anil K. Jain, "Texture Analysis", *The Handbook of Pattern Recognition and Computer Vision* (2nd Edition), by C. H. Chen, L. F. Pau, P. S. P. Wang (eds.), pp. 207-248, World Scientific Publishing Co., 1998.

[Mit97] *Machine Learning*, Tom Mitchell, McGraw Hill, 1997.

[Moo95] Mooney, R. J. 1995. Encouraging Experimental Results on Learning CNF. *Mach. Learn. 19*, 1, 79-92, 1995.

[Mül01] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Neural Networks*, 12(2):181-201, May 2001.

[Mul03] P. Mulhem and J-H Lim. Home photo retrieval: time matters. *Lecture Notes in Computer Science*, 2728:321-330. 2003.

[Nad94] J.-P. Nadal, Duality between learning machines: a bridge between supervised and unsupervised learning, *Neural Computation Vol. 6*, Issue 3 (pages 491-508), published by The MIT Press. May 1994.

[Ogl95] Virginia E. Ogle and Michael Stonebraker. Chabot: Retrieval from a relational database of images. *IEEE Computer*, 28(9):40-48, September 1995.

[Orn97] Ornager, S. : Image retrieval: theoretical and empirical user studies on accessing information in images." In ASIS '97: *proceedings of the 60th ASIS Annual Meeting*, vol. 34, 202-211, 1997.

[Ort97] Michael Ortega, Yong Rui, Kaushik Chakrabarti, Sharad Mehrotra, and Thomas S. Huang. Supporting similarity queries in MARS. In *Proceedings of the 5th ACM International Multimedia Conference*, Seattle, Washington, 8-14 Nov. '97, pages 403-413, 1997.

[Pal86] Palmer, D.C. (1986). Chomsky`s nativism: A critical review. In L.J. Hayes & P.N. Chase (Eds.). *Psychological aspects of language.* pp 439-450, Springfield, Il. 1986.

[Pat04] M. C. S. Paterno, F. S. Lim, W. K. Leow. Fuzzy Semantic Labeling for Image Retrieval. In *Proc. Int. Conf. on Multimedia and Exposition*, 2004.

[Pen94] A. Pentland, R. Picard, and S. Sclaroff . Photobook: Tools for Content-Based Manipulation of Image Databases, *SPIE Storage and Retrieval of Image & Video Databases II*, Feb 1994.

[Pol98] Fabio Policarpo, *The Computer Image,* ACM Press. Pages 298-308. 1998.

[Pri95] G. Price, The Nature of Selection, *Journal of theoretical Biology* "0884# 064\ 278_285, 1995.

[Rem01] Remco C. Veltkamp, Mirela Tanase, "Content-Based Image Retrieval Systems: A Survey", Department of Computing Science, Utrecht University, Technical *Report UU-CS-2000-34*, March 8, 2001.

[Ris79] Rissanen, J., & Langdon, G. G. Arithmetic coding. *IBM Journal of Research and Development*, 23, 149-162. 1979.

[Ros05] Bernard Rosell, Lisa Hellerstein, Why Skewing Works: Learning Difficult Boolean Functions with Greedy Tree Learners. *Proceedings of the 22 nd International Conference on Machine Learning*, Bonn, Germany, 2005.

[Rui98] Y. Rui, T.S. Huang, M. Ortega, et S. Mehrota, " Relevance Feedback : A Power Tool for Interactive Content-Based Image Retrieval", *IEEE Trans. Circuits and Systems for Video Technology*, pages 25-36. 1998.

[Rui99] Yong Rui, Thomas S. Huang, and Shih-Fu Chang, Image retrieval: current techniques, promising directions and open issues, *Journal of Visual Communication and Image Representation,* Vol. 10, no. 4, pp. 39-62, April 1999.

[Sal88] G. Salton, M.J. McGill, *Introduction to modern information retrieval*, McGrawHill, New York, 1988.

[Sam92] Samal, A. and Iyengar, P.A. Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey. *Pattern Recognition*, 25(1):65-77. 1992.

[Shi99] Shibata, T., Kato, T, "Kansei image retrieval system for street landscape discrimination and graphical parameters based on correlation of two images", IEEE-SMC'99 *Conference Proceedings-1999 IEEE International Conference on Systems, Man, and Cybernetics*, vol 6, 247-52, IEEE, Piscataway, NJ, 1999.

[Sin01] S. Singh, M. Markou and J.F. Haddon, Nearest Neighbour Classifiers in Natural Scene Analysis, *Pattern Recognition, vol. 34*, issue 8, pp. 1601-1612, 2001.

[Sin02] S. Singh and M. Singh , Evaluation of Segmentation and Texture Algorithms Combinations for Scene Analysis, *IEEE Transactions on SMC*, 2002.

[Sme00] A. Smeulders, M. Worring, S. Santini, and A. Gupta. Content based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349-1380, 2000.

[Smi96] J. R. Smith and S.-F. Chang, Querying by color regions using the VisualSEEk content-based visual query system, *In Intelligent Multimedia InformationRetrieval*. IJCAI, pages 159-173, 1996.

[Tam78] H. Tamura, S. Mori, and T. Yamawaki. Texture features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6):460-473, 1978.

[Tan01] Tan, T. and Mulhem, P. Image Query System Using Object Probes, *ICIP01*, pages 701-704, 2001.

[Tan97] Tanaka, S., Inoue, M., Ishiwaka, M., Inoue, S., "A method for extracting and analyzing kansei factors from pictures", *IEEE Workshop on Multimedia Signal Processing*, 251-6, IEEE, New York, 1997.

[Tho00] Thornton, C. *Truth from trash: How learning makes sense*. Cambridge, Massachusetts: The MIT Press. 2000.

[Tho03] Chris Thorton "Measuring the difficulty of specific learning problems" *Connection Science, 7*, No. 1 (pp. 81-92). 2003.

[Ton01] Simon Tong, Edward Chang, Support vector machine active learning for image retrieval, *Proceedings of the ninth ACM international conference on Multimedia*, Ottawa, Canada, Pages: 107 – 118, 2001.

[Tow00] C.P. Town and D. Sinclair. Content based image retrieval using semantic visual categories. *Technical Report 2000.*14, AT&T Laboratories Cambridge, 2000.

[Tur91] Turk, M. and Pentland, A.P. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71-86. 1991.

[Utg02] Utgoff, P. E., and Stracuzzi, D. J. Many-Layered Learning. *Neural Computation, 14*, 2497-2539. 2002.

[Vai01] Aditya Vailaya, Mário Figueiredo, Anil Jain, HongJiang Zhang. A Bayesian Framework for Semantic Classification of Outdoor Vacation Images, *IEEE Trans. Image Processing*, Vol. 10, No. 1, pp. 157-172, Jan. 2001.

[www-CHart] http://www.chart.ac.uk/index.htm

[www-Wordnet]http://wordnet.princeton.edu/perl/webwn

[www-COD] http://dictionary.cambridge.org/

[www-Encarta] http://encarta.msn.com/encnet/features/dictionary/dictionaryhome.aspx

[Zad65] L.A. Zadeh, "Fuzzy Sets," *Information and Control, Vol. 8,* pp. 338-352, 1965