THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Université Toulouse III - Paul Sabatier*
Discipline ou spécialité : *Informatique*

Présentée et soutenue par *Jérémy Philippeau*
Le *19 juin 2009*

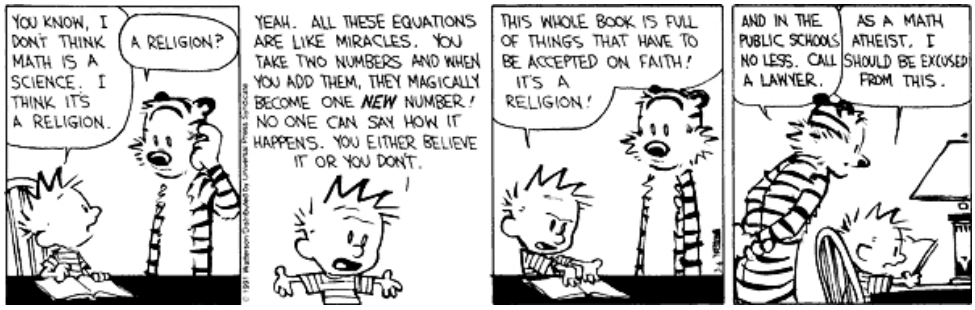
Titre :
Apprentissage de similarités pour l'aide à l'organisation de contenus audiovisuels

---

JURY
Philippe Joly (directeur de thèse)
Julien Pinquier (encadrant)
Jean Carrive (encadrant)
José Martinez (rapporteur)
Georges Quénot (rapporteur)
Florence Sédès (examinatrice)

Ecole doctorale : *Mathématiques Informatique Télécommunications de Toulouse*
Unité de recherche : *UMR 5505*
Directeur(s) de Thèse : *Philippe Joly*
Rapporteurs : *José Martinez et Georges Quénot*

Mis en page avec la classe thloria.



« Calvin & Hobbes », Bill Watterson

Remerciements

Je tiens dans un premier temps à remercier Philippe Joly pour avoir cru en mes capacités et m'avoir soutenu dans le pire comme dans le meilleur tout au long de ces trois années. Il a toujours été disponible pour m'écouter malgré un emploi du temps de ministre, et nos points communs en terme de science fiction et de cinéma en ont fait à mes yeux une figure hautement sympathique et néanmoins extrêmement compétente.

Mes remerciements vont également à Jean Carrive qui m'a été d'un grand secours pour maintenir ma tête hors des eaux tumultueuses du monde chronophage de l'entreprise, me laissant voguer selon mon bon désir. Pour ce respect de ma personne, je l'en remercie vivement. Sa vision critique de la vie m'a nombre de fois permis de prendre du recul sur la mienne, que ce soit au niveau professionnel ou personnel.

Je n'oublie évidemment pas Julien Pinquier, qui fut le pilier le plus solide de mon triptyque d'encadrement. J'attribue cette formidable capacité de soutiens à ses affinités pour les sports collectifs et le remercie d'avoir accompli cette singulière transposition dans son environnement de travail. Toujours un mot encourageant après une réunion, toujours présent pour résoudre un problème d'ordre technique, je fus toujours rassuré après chacune de ses interventions, j'espère qu'il conservera cet état d'esprit pour ses prochains encadrements.

Je tiens à remercier chaudement Régine André-Obrecht, qui ne fut jamais bien loin. Elle fut la première à apposer sa signature sur ce projet et ses compétences en matière de statistiques m'ont été de la plus grande aide. Sans un tel atout en main je n'aurai pu achever mon travail.

Pour conclure ce registre, un grand merci aux autres membres de mon jury de thèse, avec une mention toute particulière pour Florence dont j'apprécie énormément la compagnie.

J'ai eu le privilège de pouvoir évoluer dans deux environnements de travail particulièrement stimulants, et je tiens à remercier mes différents collègues.

Concernant l'Ina tout d'abord, mes compagnons d'armes Quentin, Thomas, Jean-Pascal, Jérôme, Hervé, Félicien et Ludovic ont su être présent et recharger mes batteries lorsqu'elles étaient à plat. Je n'oublie pas JPP, Laurent, Marie-Luce et Olivier pour leur agréable présence à mes côtés et l'émulsion scientifique qu'ils ont su générer tout du long de ce périple. Pour finir, merci à Odile de s'être occupé de moi avec tant de dévotion et de gentillesse.

Pour ce qui est de l'IRIT, je pense dans un premier temps à mes collègues de promotion de DEA : Guillaume, qui m'enseigna si bien que l'habit ne faisait pas toujours le moine ; Vincent, dont la psyché n'influencera je l'espère pas trop celle de sa future progéniture ; et Sylvain, dont l'humour décalé n'a pas d'égal en ce bas monde. Un grand merci à mes co-équipiers Siba, Zein, Hélène (dite la « bêta testeuse »), Hervé, Benjamin, Elie, Lionel, Jérôme (les deux), Christine, Isabelle et José pour leur bonne humeur communicative.

J'ai eu la chance d'avoir de nombreux amis qui m'ont soutenu pendant ce travail et je ne peux tous les citer ici. Un travail bien fait nécessite généralement un ventre bien rempli, je m'attacherai donc à remercier ceux qui m'ont aidé à le remplir d'une façon si plaisante.

À Paris : Ju « mou », Théo, Harald, Damdam, Caro, Sardine, Ced, Aurore, 2 2, Julia et toute la compagnie *Nutella Sunrise*.

Dans le Sud-Ouest : « docteur » Jojo, Manu « capote », Nico « Roux », Vincent, Natcho et ses tortillas, Yannick, Falconnier, Sebalafon, Cirius (*holà amigo mio*), Marion, Irène, Alex, Sophie, Jérôme, Emilie, Gabi, Typhaine et Elya.

Dans le Sud-Est : mes parents, Truong, Nicole, Julie, Jé, K-sk8, Rom1, Simon, Yvon, Véra, Guidu, et tous les autres jongleurs de l'extrême .

Un grand merci à Gabi qui m'a laissé utiliser l'une de ses œuvres merveilleuses pour faire le fond d'écran de mon application. Je ne sais pas si je dois remercier ou blâmer Fred pour m'avoir incité à utiliser la bibliothèque *Clutter*, je le laisse choisir à ma place.

Merci ma Kako pour avoir été là quand rien n'allait plus.

À Tamrill.

Table des matières

Introduction

1 Contexte applicatif

Les travaux présentés dans ce manuscrit ont été effectués dans le cadre d'une convention CIFRE [1] (Convention Industrielle de Formation par la REcherche). Cette thèse cofinancée s'est à la fois déroulée à la direction de la recherche et de l'expérimentation de l'Institut National de l'Audiovisuel [2] (Ina), au sein du thème de recherche DCA (Description des Contenus Audiovisuels) ainsi qu'à l'Institut de Recherches en Informatique de Toulouse [3] (IRIT - UMR CNRS 5505) au sein de l'équipe SAMoVA (Structuration, Analyse et Modélisation de la Vidéo et de l'Audio). Notre étude a été menée de février 2006 à mars 2008 et s'inscrit dans le contexte applicatif suivant : susciter et/ou approfondir l'usage des archives audiovisuelles par le grand public.

Grâce à un plan de sauvegarde numérique de ses archives, l'Ina est aujourd'hui en mesure de mettre à l'usage de tous près de 100.000 émissions de télévision et de radio du XX^e siècle. Il y a un enjeu économique important dans la création de nouveaux outils permettant de faire vivre ces contenus. Les projets de consultation d'archives que sont « Archives pour tous [4] » et « Inamédiapro [5] » témoignent de cette volonté de valorisation des archives numériques.

Ce manuscrit présente nos travaux sur l'organisation interactive d'une base de contenus audiovisuels, à travers une étude de la notion de similarité.

2 Problématique

Cette section est dédiée à la présentation de notre problématique, et se compose de quatre parties :

- Dans un premier temps nous identifierons le propos de notre étude à travers une présentation générale de la tâche à accomplir.
- Nous donnerons également un exemple concret pour aider à une meilleure compréhension de nos ambitions.
- Nous ferons ensuite un état de l'art des travaux qui se rapprochent le plus de nos préoccupations. Nous nous servirons des carences observées dans ceux-ci pour identifier nos besoins réels.
- Pour finir, ces besoins seront formalisés sous forme de contraintes qui constitueront notre base de travail pour le reste du document.

2.1 L'organisation de contenus

Prenons l'exemple de l'offre appelée « Archives pour tous », accessible sur Internet depuis 2006. Plus de 100.000 émissions et extraits provenant des fonds de l'Ina sont « thématisées » et « éditorialisées » à destination du grand public. La page d'accueil du site est partiellement

1. http ://www.anrt.asso.fr/fr/espace_cifre/accueil.jsp
2. http ://www.ina.fr
3. http ://www.irit.fr
4. http ://www.ina.fr/archivespourtous
5. http ://www.inamediapro.fr

illustrée par la figure 1 (seuls les points qui nous intéressaient ont été représentés).

FIGURE 1 – Extrait d'une fenêtre Web « Archive pour tous »

Il s'agit d'une offre commerciale permettant à un utilisateur de consulter des extraits d'archives audiovisuelles, éventuellement en vue de leur achat. On distingue plusieurs façons de procéder à la recherche d'un document :

- par mot(s) clef(s) (figure 1 ①) ;
- par parcours dans un thésaurus (figure 1 ②) ;
- par parcours dans une présélection :
  - de sujets fonction de certaines offres promotionnelles ou du jour de sa naissance (figure 1 ③) ;
  - faite par des tiers (célébrités, public, rédaction) ou par forum thématique (figure 1 ④) ;
  - éditorialisée à des fins éducatives (figure 1 ⑤) ;
  - en rapport avec l'actualité (figure 1 ⑥) ;
- en fonction du jour de consultation du site (figure 1 ⑦) ;
- dans une liste des quelques items les plus consultés par les utilisateurs (figure 1 ⑧).

Une fois qu'un document est consulté, il se retrouve empilé dans un bordereau horizontal (figure 1 ⑨), en haut de l'interface, en vue d'une consultation ultérieure. Ce bordereau permet de visualiser le parcours d'un utilisateur. Il est d'ailleurs possible de se voir proposer des « parcours surprises » préétablis, une autre manière de prendre connaissance du contenu de la base d'archives de l'INA (figure 1 ⑩). Ces données représentent les seules connaissances que la

machine a des goûts de l'utilisateur. Malheureusement, il n'est pas envisagé dans cette interface d'exploiter réellement cette information.

Si nous restons dans une démarche commerciale, interpréter la démarche consultative de l'usager pourrait permettre de proposer d'autres documents susceptibles de l'intéresser, et donc de les lui vendre. Au delà de l'aspect lucratif sous-jacent, un utilisateur peut simplement souhaiter organiser les éléments qu'il a consultés : les ordonner ou les regrouper de telle sorte qu'il puisse s'approprier la base, à travers cette structure qui lui est personnelle.

Notre intérêt se porte sur la conception d'un **système interactif d'aide à l'organisation de contenus audiovisuels**, permettant de mêler les activités de consultation et de documentation en un même processus. Nous avons cherché des outils probabilistes destinés à interpréter une démarche organisationnelle à partir de quelques contenus consultés par l'utilisateur. Pour ce faire nous souhaitons utiliser un opérateur capable d'évaluer de manière précise les ressemblances ou les dissemblances que présentent ces différentes entités : il s'agit d'une fonction de similarité.

La notion de similarité entre contenus audiovisuels est toutefois purement subjective. L'utilisateur peut décider que deux documentaires se ressemblent parce que les voix des narrateurs sont proches, parce que les couleurs du plateau sont sensiblement les mêmes, ou encore parce que les sujets traités sont voisins. Nous souhaitons respecter cette vision subjective de la similarité. Pour ce faire, nous voulons que notre système puisse déterminer quelles caractéristiques auditives et/ou visuelles sont à mettre en relation, et de quelle manière elles doivent l'être pour pouvoir en dégager le sens de la tâche organisationnelle suggérée par l'utilisateur.

2.2 Exemple de scénario

2.2.1 Prise de connaissance avec la base

Illustrons notre démarche par un scénario simple sur lequel nous reviendrons régulièrement (voir figure 2).

Nous souhaitons organiser une base d'images en fonction de leur couleur dominante. Nous positionnons, à l'aide d'une interface dédiée, quelques images de manière à rendre visuellement compte des écarts entre les couleurs (figure 2 ①), et nous demandons au système d'inférer notre démarche sur la totalité de la base.

Nous disposons d'une batterie de caractéristiques extraites de ces images, dont par exemple les valeurs de leur « teinte de couleur dominante ». Le système trouve pertinent d'utiliser cet unique descripteur, et interprète la tâche organisationnelle comme une volonté de répartir sur l'interface, de manière homogène, les entités documentaires selon le cercle chromatique (figure 2 ②). Il établit alors une relation entre la position des images d'une part et les valeurs descriptives correspondant à la teinte d'autre part, pour rapatrier et organiser le reste de la base documentaire (figure 2 ③).

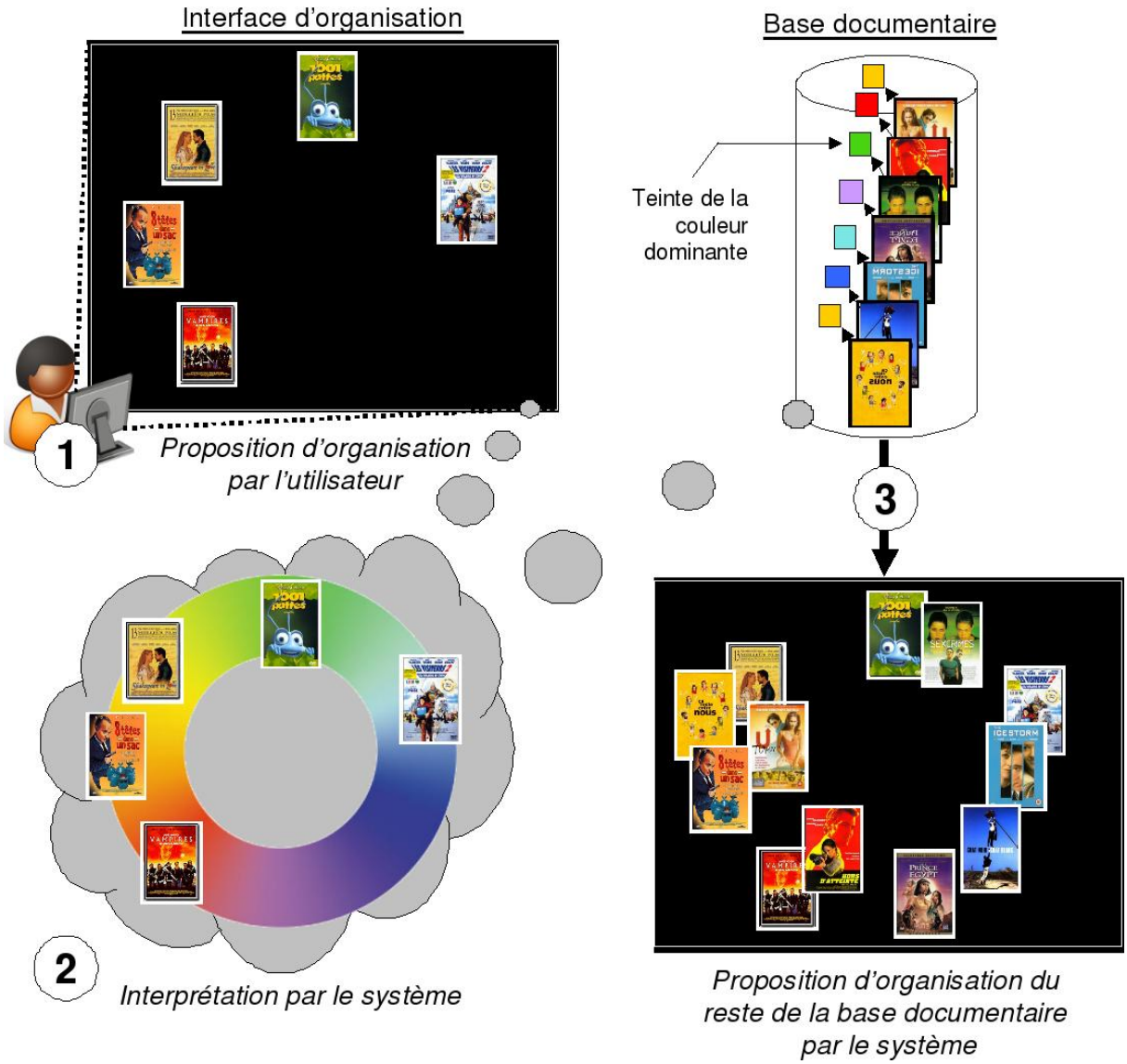


FIGURE 2 – Illustration d'un scénario

2.2.2 Aller plus loin

Supposons maintenant que ces contenus soient des films (l'image représente l'affiche du film). Il serait intéressant pour l'utilisateur de pouvoir organiser ces contenus selon des critères plus personnels (cette liste n'est pas exhaustive) :

- les ranger par degrés d'agressivité ;
- faire des regroupements par préférence ;
- chercher à l'intérieur de ces groupes le film le plus représentatif.

Ce pourrait également être attrayant d'avoir une vue d'ensemble de ses films, ainsi que de ses séries, de ses documentaires, ou de bases moins orthodoxes comme des ensembles de rush, pouvoir les manipuler et naviguer dans chacune d'elles avec aisance.

2.3 Positionnement

Dans cette section, nous présentons notre positionnement à travers un survol de différents travaux entrepris en lien avec notre problématique (les figures 3, 4, 5 et 6 en illustrent certains). Nous dégagerons au fur et à mesure les éléments qui nous occupent en mettant l'accent sur ce qui fait défaut dans la littérature, pour en venir à expliquer notre démarche dans sa globalité.

2.3.1 Regard porté sur l'existant

Les travaux sur *El Niño* [SJ00] (figure 3a), une interface de recherche d'images, nous ont particulièrement intéressés, car l'information de similarité est directement calculée sur l'interface utilisateur à travers la manipulation des documents. Cependant, le contexte ambiant de « Recherche d'Information » (RI) est particulièrement marqué dans le développement de l'outil (tout comme dans [KO95] et [CJHA97] dont il s'inspire, ou dans les travaux de [LSPM+03] sur l'exploration de bases d'images par similarité), et ne correspond pas à nos attentes : **l'organisation est une notion générique** par essence, et n'est pas aussi précise et bien définie que peut l'être une tâche de RI.

Cette notion d'organisation est d'ailleurs très souvent réduite à des tâches de regroupement. Nous pouvons citer à nouveau [SJ00], qui donne la possibilité de créer des *visual concepts* en associant dans une même « boîte » plusieurs images pour les considérer comme un même document. D'autres travaux plus récents utilisent la similarité dans un but de classification d'une base d'images, via un système interactif fondé sur la théorie des croyances [GBV08] (figure 3b). Nous proposons de **pousser plus avant la réflexion sur l'organisation**.

a) Interface de *El Nino*

b) Interface du classifieur basé théorie de croyances

FIGURE 3 – Illustrations de travaux en lien avec l'organisation

Nous projetons également d'utiliser des **techniques de visualisation au service du multi-grains** (entendre par grain « niveau hiérarchique documentaire »). Les travaux portant sur cette problématique existent, mais surtout dans le domaine de l'analyse textuelle. Nous pouvons citer l'outil 3D-XV [JJ02] (figure 4) de navigation en trois dimensions dans des bases de documents XML. Le document texte (le cadre du bas de la figure) est représenté dans la fenêtre principale par une forme en 3 dimensions. La navigation dans cette forme se fait en fonction de la hiérarchie imposée par les balises XML. De tels travaux n'ont pas été entrepris dans le domaine audiovisuel.

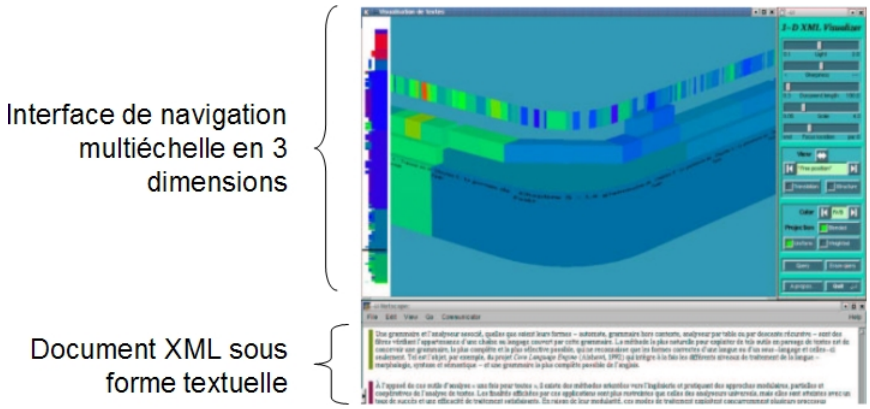


FIGURE 4 – Illustration de l'interface de 3D-XV

Côté interface, nous nous sommes intéressés aux travaux de [MTL+04] portant sur l'élaboration du PDH (*Personal Digital Historian*) (figure 5). Il s'agit d'un outil convivial de manipulation d'images numériques grâce à une sorte de table ronde de salon. Nous regretterons la mise en avant du matériel fabriqué pour favoriser l'aspect social de l'outil, et préférons concevoir un outil logiciel bien plus conventionnel. De plus, il ne traite pas de **données multimédia** et ne répond donc pas à nos attentes.

FIGURE 5 – Illustration de la table PDH

Côté indexation, le mécanisme d'aide à la création de listes de lecture pour disc-jockey de Michel Crampes [CVAR07] (figure 6) propose une approche intéressante de l'organisation. Il met à disposition un procédé dynamique d'assistance à l'indexation à l'aide de descripteurs d'humeurs (*mood descriptors*), sur la base de la création d'un « paysage musical ». Toutefois, l'algorithme de propagation des valeurs descriptives a beau être facile d'usage, il demande une intervention de l'utilisateur, sans quoi l'indexation ne peut pas avoir lieu. Nous préférons œuvrer à l'aide de **descripteurs extraits de manière automatique**, pour délester l'utilisateur de ce poids. De plus, nous envisageons d'**utiliser des caractéristiques portant un œil plus objectif**, plus « neutre », sur le contenu manipulé.

Pour finir, les descripteurs utilisés sont calculés par des intervenants provenant d'un corps de métier très particulier (DJ). Cela cible, à travers la spécificité de la tâche à accomplir, le type

d'utilisateur susceptible d'interagir avec le système. Nous préférons élargir ce champ et **créer un outil destiné au grand public**.

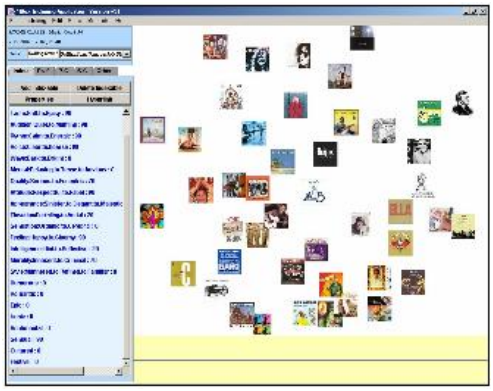


FIGURE 6 – Illustration du paysage musical indexé par les *mood descriptors*

2.3.2 L'organisation, une notion généraliste

La notion d'organisation est particulièrement difficile à cerner car très généraliste. La tâche d'organisation n'est pas une tâche décisionnelle, mais structurelle. Un système d'aide à l'organisation n'a pas pour vocation directe de répondre à un besoin spécifique, comme pourraient l'être :

- « trouver le nombre de regroupements d'objets » pour une tâche de clustering ;
- « déterminer la meilleure frontière existant entre deux classes » pour de la classification ;
- « retrouver un document précis » pour de la recherche d'information.

Il s'agit de répondre à une tâche potentiellement imprécise, qui englobe tous ces besoins, eux-mêmes plus ou moins bien définis.

Il n'existe pas à notre connaissance de système informatique qui se soit intéressé à ce genre de problématique. Nous aurions pu envisager, sous la forme d'un travail d'ingénierie, une éventuelle combinaison de méthodes efficaces, propres à chaque tâche identifiée. L'utilisateur pourrait à loisir évoluer dans un univers déterministe, et choisir de manière explicite la tâche qu'il souhaite accomplir.

Nous pensons plutôt qu'un outil d'aide à l'organisation se doit de prendre en compte l'étendue de l'hétérogénéité de ces tâches sans pour autant privilégier un choix plutôt qu'un autre. Porter avec lui la subjectivité de l'utilisateur sans étouffer sa créativité. C'est pour cela, par exemple, que nous avons choisi d'utiliser, à travers une modélisation de descripteurs audio et vidéo, un vocabulaire autre que ce que l'on pourrait trouver dans une notice documentaire de l'Ina, afin de répondre à des envies qui ne sont pas facilement définissables avec des mots.

2.3.3 Schéma technique

Nous souhaitons aider un utilisateur à **organiser**[6] des **contenus audiovisuels**, c'est-à-dire les **classifier**, les **caractériser**, les **identifier** ou les **ordonnancer**. Nous pensons que la notion de **similarité** peut expliquer ces tâches. La similarité numérique nous semble être un bon outil au regard des éléments que nous manipulons, à savoir des **objets informationnels** présentés sur un écran d'ordinateur et des **valeurs descriptives de « bas niveau »** audio et vidéo extraites de manière automatique. Nous pensons également que la similarité entre ces éléments peut être **prédite** grâce à un modèle statistique. Parmi les nombreux modèles existants, la prédiction statistique basée sur la **régression univariée** nous semble adaptée. Le schéma suivant (figure 7) résume les grandes lignes de cette réflexion.

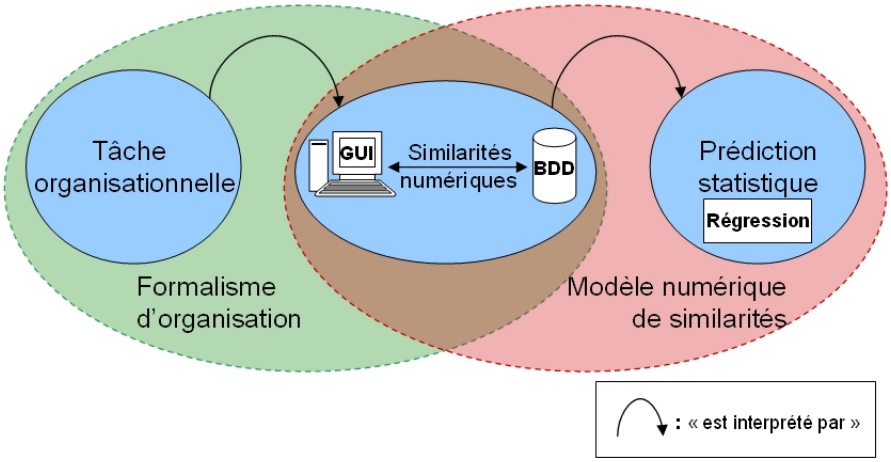


FIGURE 7 – Vision globale de nos travaux

2.4 Formalisation des contraintes

Voici un résumé des spécificités de l'outil que nous avons implanté :

1. une interface graphique qui permet à un utilisateur issu du grand public d'organiser des contenus audiovisuels, comme s'ils étaient des objets posés sur une plateforme en deux dimensions (l'utilisateur peut ne pas avoir connaissance des contenus manipulés) ;
2. un système d'aide à l'organisation qui s'appuie sur un mécanisme d'apprentissage basé sur la disposition de ces objets, ainsi que sur le traitement automatique du signal des contenus correspondants ;
3. un mode de restitution des propositions faites par la machine qui se rapproche au mieux du principe d'interaction énoncé dans le point 1 ; le système doit s'en inspirer à la fois sur le fond, par une bonne compréhension de la tâche organisationnelle suggérée par l'utilisateur, et sur la forme, en considérant les contenus comme des objets physiques ;

6. Les mots en gras utilisés ici sont des termes techniques définis dans le contexte spécifique de cette étude. Ils seront expliqués en temps utile tout au long du document.

4. un système qui permet l'exploration multi-grains des contenus.

Cette sous-section est consacrée à la définition formelle des contraintes que nous nous posons et qui s'imposent à nous. Nous y ferons régulièrement référence tout au long du document. Afin de rendre plus lisible la suite du manuscrit, nous définissons ici quelques notations :

- une **tâche organisationnelle** identifiera toute manipulation des contenus effectuée par l'utilisateur en vue de les organiser. Ces manipulations peuvent être de type « classification », « identification », « caractérisation » ou « ordonnancement » (ces termes sont définis dans la section 2.2.1 du chapitre 2) ;
- les **valeurs descriptives** sont les données numériques issues de différents détecteurs (sonores ou visuels) présentés dans le chapitre suivant ;
- les **distances utilisateur** sont les distances calculées entre les entités manipulées sur l'interface d'organisation visuelle lors de la phase d'apprentissage de la tâche organisationnelle. L'ensemble de ces distances sera interprété par le système comme l'expression de la similarité entre contenus audiovisuels proposée par l'utilisateur.

2.4.1 Contraintes liées au mode d'interaction

$\mathscr{C}_0$ **: La tâche organisationnelle est modélisable**.

Il s'agit ici plus d'une hypothèse que d'une contrainte. Nous supposons qu'il existe une relation entre les valeurs descriptives et les distances utilisateur. Nous cherchons une formulation mathématique de cette relation, valable dans le domaine où les mesures ont été effectuées.

$\mathscr{C}_1$ **: Cette modélisation s'appuie sur l'analyse des similarités**.

Nous voulons étudier les similarités entre les contenus audiovisuels. Pour ce faire, nous nous appuyons sur une interface graphique interactive permettant d'interpréter, en termes de similarités, la tâche organisationnelle suggérée par un utilisateur.

$\mathscr{C}_2$ **: Le temps de calcul doit être humainement acceptable**.

L'interface doit permettre une organisation rapide de la base documentaire. De ce fait il nous faut construire un modèle avec une durée d'apprentissage relativement courte. Le temps d'attente maximum pendant lequel il est possible de conserver l'attention d'un utilisateur est estimé à environ 10 secondes [Nie93]. Notre système doit être en mesure de respecter ce délai.

$\mathscr{C}_3$ **: Le modèle doit gérer les changements de granularité**.

Nous souhaitons utiliser des mécanismes de navigation hiérarchique qui vont nous amener à considérer les entités documentaires sous diverses granularités : images, plans, séquences, émissions, collections de documents... La cohérence du modèle de descripteurs doit perdurer ou s'adapter lors d'un changement de grain.

$\mathscr{C}_4$ **: L'utilisateur peut remettre en question la tâche organisationnelle à tout instant**.

Nous supposons que l'utilisateur peut ne pas avoir connaissance des contenus qu'il manipule, découvrant peu à peu la base documentaire. Nous voulons que notre système s'adapte

aux éventuels changements de tâche, pouvant intervenir suite à une prise de connaissance de nouveaux contenus.

2.4.2 Contraintes liées au nombre d'interactions

$\mathscr{C}_5$ **: Le nombre de distances utilisateur est faible**.

Nous voulons permettre à un utilisateur d'organiser une grand nombre de contenus en n'en manipulant que quelques uns. Cela implique que toute l'information nécessaire à la conception d'un bon modèle ne sera pas forcément explicitée ; le moteur d'apprentissage devra extrapoler de manière efficace pour combler ce manque.

2.4.3 Contraintes liées à l'objectivité des valeurs descriptives

$\mathscr{C}_6$ **: Le système doit pallier l'imprécision des valeurs descriptives**.

Cette imprécision peut provenir de différentes sources :

- $\mathscr{C}_{6.1}$ **: Le système doit accepter tout type de valeur numérique extraite de manière automatique**. Nous souhaitons que notre modèle puisse intégrer toute valeur descriptive numérique, sans se soucier du fait que celles-ci soient extraites d'un opérateur linéaire ou non, continu ou non, borné ou non...
- $\mathscr{C}_{6.2}$ **: Les valeurs descriptives utilisées peuvent être perturbatrices**. Certaines valeurs peuvent être très fortement corrélées, la sémantique exprimée par certaines d'entre elles peut être inadaptée à la tâche organisationnelle envisagée par l'utilisateur, etc. Il faut donc se pencher sur un processus permettant d'élaguer ces éléments perturbateurs.

2.4.4 Contraintes liées à la subjectivité des distances utilisateur

$\mathscr{C}_7$ **: Le système doit pallier l'imprécision des distances utilisateur**.

Cette imprécision peut provenir de différentes sources :

- $\mathscr{C}_{7.1}$ **: L'ensemble des contenus explorés peut être inconnu**. L'outil que nous créons sert à s'approprier une base documentaire. Celle-ci peut autant appartenir à l'utilisateur que lui être présentée pour la première fois (comme pour le site « Archive pour tous » par exemple). Cette contrainte ne nous permet donc pas d'anticiper la position exacte des contenus présentés.
- $\mathscr{C}_{7.2}$ **: La tâche organisationnelle est approximative**. Un humain qui positionne des objets dans l'espace, de la manière dont nous le proposons, définit entre eux des relations qui ne seront jamais parfaitement interprétées par un outil mathématique. Nous devons créer un modèle suffisamment souple pour prendre en compte l'approximation engendrée par un positionnement imprécis des contenus.
- $\mathscr{C}_{7.3}$ **: L'utilisateur est issu du grand public**. Nous ne possédons aucune connaissance *a priori* sur le bagage socioprofessionnel de l'utilisateur. Nous ne pouvons nous appuyer sur un profil préexistant pouvant définir ses besoins.

3 Présentation du manuscrit

Les chapitres un, deux et trois, bien que différemment structurés, sont tous constitués :

- d'une présentation des besoins spécifiques au problème abordé ;
- d'un état de l'art correspondant à ces besoins ;
- de notre prise de position sur le sujet ainsi que de la contribution que nous y apportons, que cet apport soit de nature informelle (chapitre un) ou technique (chapitres deux et trois).

Le premier chapitre présente la matière première qui sera disséquée tout au long du reste du manuscrit : le contenu audiovisuel. Nous y exposons différents points de vues portés sur ces objets, autant du côté humain que du côté machine. Cette exploration nous permet de nous positionner vis-à-vis de la communauté sur ce que nous entendons faire avec ces éléments.

Le deuxième chapitre est une analyse du propos de notre étude du point de vue de l'interaction entre l'Homme et la Machine. Il met l'accent sur les aspects propres à la visualisation de similarités numériques. Nous y présentons le formalisme qui nous permet de visualiser des contenus audiovisuels dans un espace dynamique en deux dimensions.

Le chapitre trois est dédié au moteur d'apprentissage de notre système, partie immergée de notre iceberg technologique. Nous y parlons de la modélisation mathématique globale du problème grâce au modèle numérique de similarité. Nous nous appuierons sur une régression univariée pour prédire ces similarités. Les discussions porteront sur l'existant ainsi que sur les méthodes que nous avons choisi d'exploiter et qui nous permettront d'adapter ce modèle pour qu'il puisse répondre à nos attentes.

Le chapitre quatre est une présentation du prototype implanté. Nous argumentons sur la manière dont nous nous y prenons pour relier les aspects « visualisation » et « apprentissage » du problème. Il y sera question de la pertinence des différents choix technologiques compte tenu des contraintes qui nous sont posées et que nous nous sommes imposées. Nous conclurons par plusieurs séries d'expériences permettant d'évaluer ces choix.

Notions de Mathématiques

Quelques notions élémentaires de Mathématiques vont être décrites afin de faciliter la lecture de ce manuscrit. Les domaines concernés sont l'Algèbre linéaire, l'Analyse et les Probabilités.

Algèbre linéaire

Soit un ensemble E non vide.

Groupe

Le couple $(E, +)$ est un groupe si :
- $+$ est une loi interne sur E ;
- $+$ est associative : $\forall x, y$ et $z \in E,\ (x + y) + z = x + (y + z)$;
- $+$ possède un élément neutre : $\exists\ e$ tel que $e + x = x + e = x$;
- tout élément de E possède un symétrique dans E :
  $\exists\ x, y \in E$ tels que $x + y = y + x = e$, où e est l'élément neutre de E.

Si la loi interne est commutative, c'est-à-dire que pour tout x et y de E, $x+y = y+x$, le groupe est dit commutatif (ou abélien).

Corps

Un corps est un ensemble $\mathbb{K}$ muni de deux lois internes $+$ et $*$ vérifiant
- $(\mathbb{K}, +)$ forme un groupe commutatif dont l'élément neutre est noté 0 ;
- $(\mathbb{K} \setminus \{0\}, *)$ forme un groupe multiplicatif ;
- la multiplication est distributive à gauche et à droite pour l'addition :
  $\forall (x, y, z) \in \mathbb{K}^3, \quad x * (y + z) = x * y + x * z \quad$ et $\quad (y + z) * x = y * x + z * x$

Espace vectoriel

On appelle $\mathbb{K}$-espace vectoriel tout ensemble E muni d'une loi interne $+$ et d'une loi externe $*$ tel que :
- $(E, +)$ est un groupe commutatif ;
- l'élément neutre du groupe $(\mathbb{K}, *)$ est neutre à gauche pour $*$:
  $\forall\, x \in E, \quad 1 * x = x$;
- $*$ est distributive à gauche par rapport à l'addition de E :
  $\forall\, k \in \mathbb{K},\ \forall\, x \in E,\ \forall\, y \in E, \quad k * (x + y) = (k * x) + (k * y)$;
- $*$ est distributive à droite par rapport à l'addition de $\mathbb{R}$:
  $\forall\, k \in \mathbb{K},\ \forall\, k' \in \mathbb{K},\ \forall\, x \in E, \quad (k + k') * x = (k * x) + (k' * x)$;
- $*$ est associative par rapport à la multiplication de $\mathbb{R}$:
  $\forall\, k \in \mathbb{K},\ \forall\, k' \in \mathbb{K},\ \forall\, x \in E, \quad (k.k') * x = k * (k' * x)$.

Par abus de langage, nous appellerons **espace vectoriel** tout $\mathbb{R}$-espace vectoriel.

Produit scalaire dans E

On appelle **produit scalaire** sur E un $\mathbb{R}$-espace vectoriel toute application $\pi : E \times E \to \mathbb{R}^+$ telle que :

- π est symétrique : $\forall x, y \in E, \pi(x, y) = \pi(y, x)$;
- π est linéaire par rapport à la seconde variable :
  $\forall x, y, y' \in E,\ \forall k \in \mathbb{R},\ \pi(x, ky + y') = k * \pi(x, y) + \pi(x, y')$;
- π est définie positive : $\pi(x, y) = 0 \Leftrightarrow x = y$.

On dit qu'un produit scalaire sur un $\mathbb{R}$-espace vectoriel est une forme bilinéaire, symétrique et définie positive. On le note $\pi(x, y) = \langle x, y \rangle$.

Norme

Soit $\mathbb{K}$ un corps muni d'une valeur absolue et E un $\mathbb{K}$-espace vectoriel.

Une norme sur E est une application $\mathcal{N}$ sur E à valeurs réelles positives et satisfaisant les hypothèses suivantes :

- séparation : $\forall x \in E,\ \mathcal{N}(x) = 0 \Rightarrow x = 0_E$;
- homogénéité : $\forall (\lambda, x) \in \mathbb{K} \times E,\ \mathcal{N}(\lambda * x) = |\lambda| * \mathcal{N}(x)$;
- inégalité triangulaire : $\forall (x, y) \in E^2,\ \mathcal{N}(x + y) \leq \mathcal{N}(x) + \mathcal{N}(y)$.

La norme d'un vecteur x se note $\|x\|$.

Distance

On appelle distance sur un ensemble E une application $f : E \times E \to \mathbb{R}^+$ telle que :

- f soit symétrique : $\forall x, y \in E, f(x, y) = f(y, x)$;
- f soit définie positive : $f(x, y) = 0 \Leftrightarrow x = y$;
- f respecte l'inégalité triangulaire : $f(x, z) \leq f(x, y) + f(y, z)$.

Espace euclidien

Un espace euclidien est un espace vectoriel de dimension finie muni d'un produit scalaire.

Espace préhilbertien

Un espace préhilbertien est un espace vectoriel muni d'un produit scalaire. Il peut être vu comme une généralisation de l'espace euclidien, l'hypothèse de la dimension finie étant omise.

Analyse

Base canonique

Soit $\mathbb{K}$ un corps et $n \in \mathbb{N}^+$. La base canonique de $\mathbb{K}^n$ se compose des vecteurs $\{e_i\}_{i\in\{1,\dots,n\}}$ tels que $e_i = (\delta_{1,i}, \delta_{2,i}, \cdots, \delta_{n,i})$, avec $\delta_{i,j}$ le symbole de Kronecker :

$$\delta_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$$

Dérivées partielles

La dérivée partielle d'une fonction f est la dérivée par rapport à l'une de ses variables, les autres sont considérées comme constantes. La dérivée partielle du premier ordre par rapport à la variable x est notée $\frac{\partial f}{\partial x}$ ou $\partial_x f$.

Différentielle

Soient E et F deux espaces vectoriels normés, et f une application de E dans F. Soit a un point de E. On dit que f est différentiable en a si et seulement s'il existe une application linéaire continue L de E dans F telle que :

$$\forall h \in E, \quad f(a+h) = f(a) + L(h) + o\left(\|h\|\right)$$

L est appelée différentielle de f en a et se note $L = df(a)$.

Gradient

Soit U un ouvert (ensemble qui ne contient pas sa frontière) de $\mathbb{R}^n$. Soit $f : U \mapsto \mathbb{R}$ une fonction différentiable. Soit $df(a)$ la différentielle de f en a, avec $a \in U$. On note $(df(a), u)$ l'image par cette différentielle d'un vecteur $u \in \mathbb{R}^n$.

Il existe un vecteur A de $\mathbb{R}^n$, $(df(a), u) = \langle A, u\rangle$, avec $\langle ., .\rangle$ le produit scalaire de $\mathbb{R}^n$. Le vecteur A est appelé gradient de f en a, et se note $\nabla_a f$. Il vérifie donc :

$$\forall u \in \mathbb{R}^n, \langle \nabla_a f, u\rangle = (df(a), u)$$

Exprimé dans la base canonique $\{e_i\}_{i\in\{1,\dots,n\}}$ de l'espace vectoriel $\mathbb{R}^n$, le gradient se définit grâce aux dérivées partielles sous la forme suivante :

$$\nabla_a f = \sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i}(a)\, \mathbf{e}_i \right)$$

Matrice jacobienne

La matrice jacobienne d'une fonction vectorielle est la matrice de ses dérivées partielles du premier ordre.

Soit F une fonction d'un ouvert de $\mathbb{R}^n$ à valeurs dans $\mathbb{R}^m$. Une telle fonction est définie par ses m fonctions composantes à valeurs réelles :

$$F : \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \longmapsto \begin{pmatrix} f_1(x_1, \ldots, x_n) \\ \vdots \\ f_m(x_1, \ldots, x_n) \end{pmatrix}$$

Les dérivées partielles de ces fonctions en un point M, si elles existent, peuvent être rangées dans une matrice J à m lignes et n colonnes, appelée matrice jacobienne de F :

$$J = \begin{pmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{pmatrix} \tag{1}$$

Probabilité

Univers, probabilité et espace de probabilité

Soit Ω un **univers**, c'est-à-dire l'ensemble de tous les résultats possibles qui peuvent être obtenus au cours d'une expérience aléatoire. On appelle **événement** un sous-ensemble de Ω.

Une **probabilité** P sur l'ensemble Ω est une application de l'ensemble des parties de Ω dans l'intervalle $[0, 1]$ qui vérifie les propriétés suivantes :

- $P(\Omega) = 1$;
- P est additive, c'est-à-dire $P(A \cup B) = P(A) + P(B)$ pour tout A et B disjoints.

Un **espace de probabilité** est un couple (Ω, P) où Ω est un ensemble et P une probabilité sur cet ensemble.

Une **variable aléatoire**, notée X, est un nombre dépendant du résultat d'une expérience aléatoire.

Variable aléatoire discrète

On dit qu'une variable aléatoire est **discrète** si elle ne prend qu'un nombre fini ou dénombrable de valeurs. Soit (Ω, P) un espace probabiliste.

On note $X(\Omega)$ l'ensemble des valeurs possibles de la variable aléatoire discrète X. On peut définir un autre espace de probabilité qui ne tient compte que des résultats X, donné par le couple $(X(\Omega), P_X)$ où P_X est la probabilité définie pour tout événement B de $X(\Omega)$ par :

$$P_X(B) = P(\{\omega, X(\omega) \in B\}) = P(X^{-1}(B)) = P(X \in B)$$

Cette probabilité P_X s'appelle la **loi** de X. Dans le cas où Ω est fini, cela revient à donner $P(X = k)$ pour tout $k \in X(\Omega)$.

L'**espérance** d'une variable aléatoire discrète X, notée $E(X)$, est une moyenne pondérée des valeurs de cette variable :

$$E(X) = \sum_{k \in X(\Omega)} kP(X = k)$$

La **variance** d'une variable aléatoire discrète X, notée $V(X)$, mesure la dispersion des valeurs de cette variable par rapport à sa valeur moyenne :

$$V(X) = E((X - E(X))^2) = E(X^2) - E(X)^2$$

Variable aléatoire continue

On dit qu'une variable aléatoire est **continue** si l'ensemble des valeurs de X est un intervalle de $\mathbb{R}$.

On appelle **fonction de répartition** de la variable aléatoire X l'application F définie sur $\mathbb{R}$ à valeur dans $[0;1]$ par $F(x) = P(X \leq x)$. Voici quelques unes de ses propriétés :

- $P(X > x) = 1 - F(x)$ pour tout réel x ;
- $P(a < b) = F(b) - F(a)$ pour tous réel a et b tel que $a < b$;
- F est croissante et continue sur $\mathbb{R}$.

On appelle **densité de probabilité** toute fonction f définie continue et positive sur $\mathbb{R}$ telle que :

$$\int_{-\infty}^{+\infty} f(x)dx = 1$$

Soit X une variable aléatoire continue de fonction de répartition F alors :

- pour tout réel x , la fonction f définie sur $\mathbb{R}$ par $f(x) = F'(x)$ est une densité de probabilité appelée densité de probabilité de X ;
- pour tout réel x :

$$F(x) = \int_{-\infty}^{x} f(t)dt$$

L'**espérance** d'une variable aléatoire continue X est le nombre réel, noté $E(X)$, défini par :

$$E(x) = \int_{-\infty}^{+\infty} xf(x)dx$$

La **variance** d'une variable aléatoire continue X est le nombre réel, noté $V(X)$, défini par :

$$V(x) = E(X^2) - E(X)^2 = \int_{-\infty}^{+\infty} x^2 f(x)dx - \left(\int_{-\infty}^{+\infty} xf(x)dx\right)^2$$

L'**écart-type** de la variable aléatoire (discrète ou continue) X se définit ainsi :

$$\sigma(X) = \sqrt{V(X)}$$

Chapitre 1

Description d'un contenu audiovisuel

Sommaire

1.1 Introduction

Le contenu audiovisuel est l'élément central de notre problématique. Ce chapitre nous permet de faire sa connaissance, tout d'abord d'un point de vue linguistique et sémiotique, que nous qualifierons d'humain, par opposition à un point de vue orienté signal, que nous lierons à la machine. Contenu, humain et machine, un triptyque avec lequel nous évoluerons tout au long de ce manuscrit et qui nous permettra de nous pencher sur la manière dont chaque élément constitue une clef pour interpréter les deux autres.

1.2 Le contenu audiovisuel vu par l'humain

Nous n'avons en aucun cas la prétention de nous improviser spécialiste en sémiotique dans cette section, nous proposerons seulement notre point de vue plutôt généraliste sur le sujet en s'aidant de celui de personnes particulièrement qualifiées que nous citerons au fil de la plume.

1.2.1 Qu'est-ce qu'une description ?

Décrire un objet est un acte interprétatif. À ce propos, la métaphore visuelle est souvent utilisée pour parler d'une description personnelle : une description est une « vue », une « vision » sur un objet.

Comme le propose Peter Stockinger [Sto02], une description (au sens d'acte interprétatif, d'une interprétation) présuppose toujours :

– d'un point de vue ;
– d'un besoin, c'est-à-dire d'une intention qui oriente la description.

Prenons par exemple cette photographie (figure 1.1).

FIGURE 1.1 – Exemple de document

Il existe autant de façon de la décrire que de point de vue de l'aborder :

- d'un point de vue **historique**, cette photo fut prise le 28 juin 2008 lors d'une soirée spectacle chez Marie-Luce et Jean-Luc...
- d'un point de vue **anecdotique**, les enfants se sont déguisés et ont dansé dans le salon toute la fin de l'après-midi en imitant...
- d'un point de vue **poétique**, il s'agit d'une scène de joie infantile dans laquelle sont mises en avant diverses évocations d'expressions artistiques...
- d'un point de vue **esthétique**, nous voyons sur cette photo cinq personnes, mis en scène dans un intérieur urbain, la principale source de lumière venant essentiellement du flash de l'appareil. Au premier plan...
- d'un point de vue **technique**, il s'agit d'une photographie de résolution 3328x1872 réduite au 10^e, prise par un appareil photo numérique...
- etc.

Le point de vue fait partie d'une culture et implique un degré de connaissance partagé par une communauté.

Si l'on souhaite modéliser l'utilisateur d'un outil informatique semi-automatique, il est courant de le considérer comme représentatif d'une catégorie socioprofessionnelle particulière. C'est cette dernière qui constituera la communauté dont nous parlions plus tôt et qui dictera quel point de vue adopter :

- une documentaliste aura plutôt une vision historique du document ;
- un technicien de l'image aura une vision technique ;
- un vidéaste aura une vision à la fois poétique et technique ;
- etc.

Toutefois, dans le cadre de nos travaux, l'utilisateur vient du grand public. Il n'entre dans aucune catégorie socioprofessionnelle précise, et dans toutes à la fois.

1.2.2 Qu'est-ce qu'un document audiovisuel ?

Considérons tout d'abord le terme qui se rapproche le plus de ce que nous entendons par contenu audiovisuel, à savoir le document.

En sémiotique, un document audiovisuel est considéré comme un « signe » audiovisuel. Les signes établissent la relation entre un signifiant et un signifié, c'est-à-dire le liant entre image mentale et image acoustique.

On peut considérer comme signe audiovisuel tout signal pouvant être perçu par la vue et l'ouïe, se soumettant aux contraintes de :

- **linéarité temporelle :** l'information présente dans le contenu prend forme à mesure que le temps s'écoule, de manière conforme à notre perception de celui-ci ;
- **forme d'expression syncrétique :** l'information peut se présenter via différents médias (son, images, paroles, écrits...) qui sont coordonnés entre eux pour une bonne interprétation globale.

Cette définition peut prêter à confusion, car elle est suffisamment large pour englober les impressions audiovisuelles que nous laissent notre environnement, et qui nous permettent de se construire une image mentale de celui-ci.

Afin donc de ne pas diverger de notre sujet d'analyse, et contraint par notre contexte applicatif et l'environnement industriel dans lequel nous avons évolué, nous appellerons document audiovisuel les signes nommés **« film »** et **« vidéo »**. Par extension, tout signe relatif à un média qui pourrait intervenir dans ces deux précédent stéréotypes sera également considéré comme un document audiovisuel, c'est-à-dire les signes **« son »**, **« texte »** et **« image »**.

Toutefois, nous verrons par la suite que nous ne pourrons nous contraindre à respecter la règle de linéarité temporelle sur les objets que nous manipulerons. Pour lever toute ambiguïté, nous préciserons ce qu'est pour nous un contenu audiovisuel à travers la notion de grain documentaire, définie dans la section suivante.

1.2.3 Comment décrire un contenu audiovisuel ?

Un document audiovisuel est un ensemble de signes perceptibles soumis à des contraintes de structure et de cohérence entre les médias utilisés pour véhiculer une information.

Cette définition est généraliste : un document donné est souvent lui-même considéré comme un ensemble de documents, et peut également faire partie d'un document englobant. Par exemple, une chanson est constituée de différentes parties (les refrains, les ponts, les chorus, etc.), et cette chanson fait partie d'un album, lui-même issu d'une discographie. Toutefois, il n'est pas possible d'appeler « documents » ces ensembles, car la linéarité temporelle n'est pas respectée. C'est pourquoi nous souhaitons définir une autre entité comme extension du document audiovisuel : le contenu.

Avant de décrire un document, il est nécessaire de s'intéresser à son grain. C'est l'homogénéité de l'information que l'on souhaite analyser qui va nous servir à structurer et donc à définir le grain. En parallèle, se pose la question de la description à adopter afin de témoigner du sens porté par ces grains : les termes utilisés pour comparer deux discographies ne sont pas les même que ceux utilisés pour comparer deux refrains.

Voici un exemple (illustré par la figure 1.2) de ce que peuvent être différents grains de documents, présentés à travers une segmentation hiérarchique d'une collection d'archives.

1.2.3.1 Exemples de grains documentaires

Posons le contexte du paysage télévisuel français (les termes qui sont spécifiques à ce domaine, utilisés dans cet exemple, sont définis dans la thèse de Jean Philippe Poli [Pol07]).

7. Le grain le plus grossier que l'on puisse considérer, et qui est commun à tout type de format audiovisuel, est **la collection**. Dans ce contexte, une collection est un ensemble de programmes télévisuels. Prenons comme collection l'ensemble des programmes répondant à un **genre télévisuel** (figure 1.2, point 7) : les journaux télévisés, les magazines, les retransmissions,

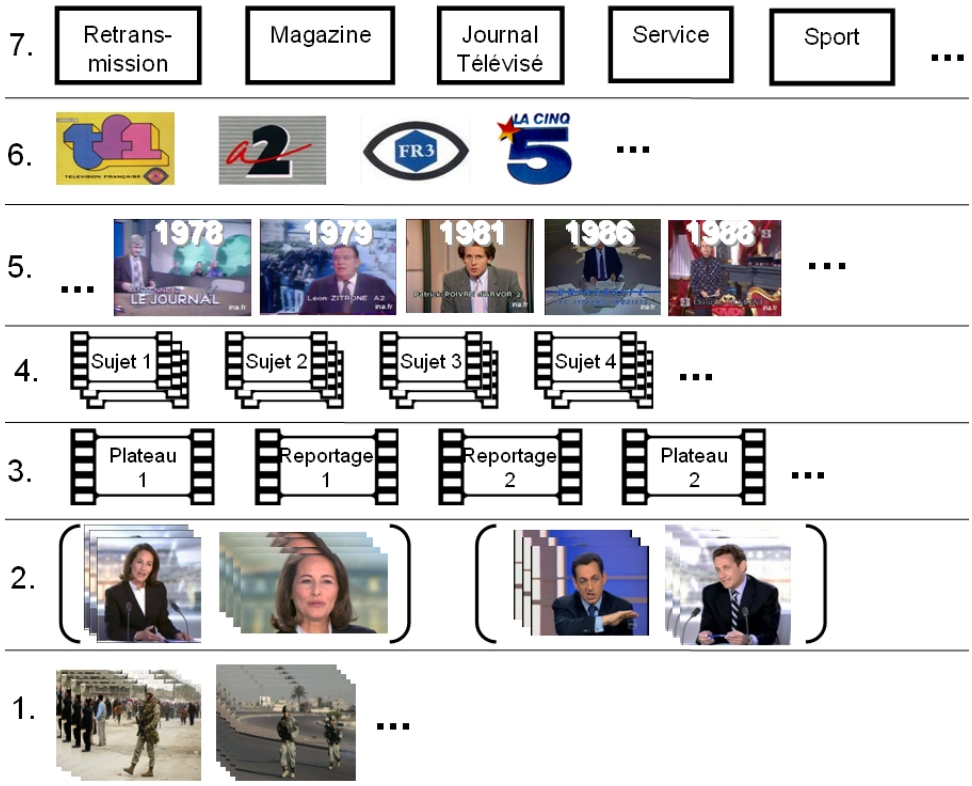


FIGURE 1.2 – Exemple de segmentation hiérarchique d'une collection d'archives audiovisuelles

les émissions de service et le sport sont des genres télévisuels. On peut, dans le cadre d'une étude sociologique ou commerciale, vouloir tirer des statistiques en prenant les tranches d'âges des spectateurs susceptibles de regarder tel genre de programme, et les confronter aux plages horaires dédiées à ces programmes.

6. Un grain plus fin peut être l'**ensemble des journaux d'une chaîne de télévision** (figure 1.2, point 6), et témoigne par exemple de l'image propre à une chaîne. D'une analyse globale de l'atmosphère des journaux d'une chaîne peut émerger une forme de signature de celle-ci.

5. Nous segmentons cette même collection selon différentes **saisons** (figure 1.2, point 5), représentatives d'une certaine forme d'habillage de la chaîne. Une telle démarche isole donc un ensemble de documents ayant des caractéristiques globales homogènes : même réalisateur, même présentateur, même environnement de diffusion... L'information porte ici sur l'ambiance générale conditionnée sur un temps donné. La signature obtenue sur le grain précédent s'affine, et témoigne d'une époque : la présentation des journaux de TF1 des années 70 sur fond psychédélique n'a rien à voir avec l'austérité des journaux des années 90.

4. Un journal peut être vue comme une succession de **sujets** (figure 1.2, point 4). Sont souvent employés les termes d'« information spectacle » lorsqu'on parle des journaux de TF1 : fidéliser son auditoire en choisissant des sujets en fonction de leur sensationnalisme, afin de

jouer sur les émotions. Il peut être intéressant d'analyser quels sujets sont abordés et comment ils sont structurés par rapport à d'autres journaux, comme ceux de Arte ou de TMC par exemple, pour identifier en termes audiovisuel ce qu'est, ou ce que n'est pas, de l'information spectacle.

3. Un sujet peut être vu également comme une alternance de **séquences**, entre plateaux de lancement et reportages (figure 1.2, point 3). Les plateaux peuvent servir d'introduction (respectivement de conclusion) au reportage qui lui succède (respectivement qu'il précède). Nous pouvons alors comparer la manière dont on témoignait des violences urbaines dans les reportages d'Antenne 2 de 1979 et dans les reportages de la Cinq de 1992.

2. Il peut être intéressant de s'attarder sur chaque **locuteur** (figure 1.2, point 2) qui intervient dans les plateaux ou les reportages des journaux télévisés. Cela permet par exemple de comparer le temps de parole de Ségolène Royale à celui de Nicolas Sarkozy lors du débat du second tour de l'élection présidentielle de 2007, où de noter les idées de chacun afin de les confronter à celles débattues en 1995 dans le duel opposant Lionel Jospin à Jacques Chirac.

1. Du point de vue de l'indexation vidéo, une segmentation intéressante et amplement usitée, qui peut intervenir à ce niveau de la hiérarchie, est le **plan** (figure 1.2, point 1). Ce grain est l'unité de base lors d'une activité de dérushage par exemple. Il peut être intéressant pour un monteur vidéo de retrouver tous les plans montrant des militaires américains lors de la guerre du golfe, afin de créer un documentaire personnel.

1.2.3.2 Quel grain choisir ?

Notons qu'il n'existe pas de consensus sur la valeur du sens porté par un type d'information. Cet indéterminisme donne donc libre cours à l'interprétation de ce que peut être le grain d'un document.

Prenons le point 2 de l'exemple précédent. Ce grain s'inscrit entre les niveaux 1 et 3, à savoir le Plan et la Séquence. Cela n'est pas sans rappeler le **Macrosegment** [Jol96], grain provenant de la communauté vidéo, dont la définition est la suivante : *segment temporel s'insérant entre le plan et le document*, le document étant ici la Séquence.

Cependant, le point 2 est le Locuteur, qui est un grain purement audio. Nous l'avons considéré d'un niveau hiérarchique supérieur au plan, mais il ne s'inscrit pas dans le format du macrosegment :

- sa structure peut être morcelée, ce n'est donc pas un segment temporel à proprement parler mais un regroupement de segments ;
- un Locuteur peut exister sans qu'il y ait eu de changement de plan.

Nous aurions également pu choisir, pour ce même niveau hiérarchique, un autre grain d'analyse qui est l'**Intervenant** [PPJ06]. Il s'agit d'un locuteur qui se caractérise conjointement à sa présence ou son absence sur la vidéo pendant son élocution. Le point de vue est désormais multimédia et les descripteurs employés le sont également :

- le locuteur est visible, il est « IN » ;

– le locuteur n'est pas visible, mais a déjà été filmé ou le sera durant son élocution, il est « OUT » ;
– le locuteur n'est jamais visible pendant toute son intervention, il est « OFF ».

Locuteur, Macrosegment et Intervenant, chacun de ces grains porte sa propre information et ses propres descripteurs issus de leurs domaines respectifs : l'audio, la vidéo et le multimédia. Il y a donc une interdépendance forte entre la taille du grain, l'information qu'il porte et les outils utilisés pour la description (figure 1.3).

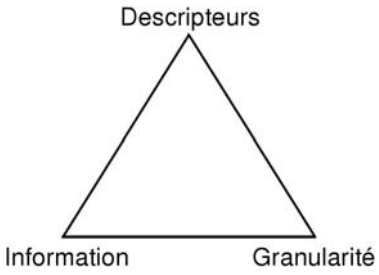


FIGURE 1.3 – Interdépendances nécessaires à une bonne description d'un document

1.2.3.3 Grain et contenu

Nous voyons bien dans l'exemple 1.2 qu'un grain documentaire est défini par le type d'information que les documents qu'il porte souhaitent véhiculer.

Il est de ce fait naturel de souhaiter considérer comme un document audiovisuel l'ensemble des segments temporels portant sur un même locuteur : il serait intéressant de rassembler un ensemble d'extraits portant sur un candidat à la présidentielle en une même entité documentaire, afin de la comparer à d'autres entités représentatives. Cette considération brise la règle de linéarité temporelle énoncée plus haut (section 1.2.2).

Pour éviter toute confusion de langage, nous définissons un contenu audiovisuel comme un **ensemble de documents audiovisuels, porteurs d'une information provenant d'une même granularité** ; Nous considérons que les différents grains s'organisent selon une hiérarchie dont la profondeur est arbitraire, allant du grain le plus fin (de niveau 1) qui est « le document », jusqu'au grain le plus grossier (de niveau le plus élevé) qui est « la collection ».

Il devient ainsi possible, et surtout cohérent, de comparer n'importe quel contenu d'une même granularité grâce à des descripteurs adaptés. Définir un contenu, c'est d'abord définir son grain documentaire : pour tout contenu A_j d'une granularité n donnée,

$$\begin{cases} A(1)_j := (nom\_du\_grain, A_j), \text{tel que } A_j \text{ soit un document} \\ A(n)_j := (nom\_du\_grain, \{A(n-1)_1, A(n-1)_2, A(n-1)_3, \ldots\}) \end{cases} \tag{1.1}$$

Nous pouvons redéfinir l'exemple de la section 1.2.3.1 à l'aide de ce formalisme (avec $A(i)_j$ un contenu A_j quelconque d'une granularité i).

Formalisme	Exemples
$A(7)_j$:= (Collection, $\{A(6)_1, A(6)_2, ...\}$)	(Collection, {Retr.}), (Collection, {Mag.}), (Collection, {J.T.}), ...
$A(6)_j$:= (J.T., $\{A(5)_1, A(5)_2, ...\}$)	(J.T., {TF1}), (J.T., {a2}), (J.T., {FR3}), (J.T., {La Cinq}), ...
$A(5)_j$:= (Saison, $\{A(4)_1, A(4)_2, ...\}$)	(Saison, {TF1_1978}), (Saison, {a2_1978}), (Saison, {FR3_1983}), (Saison, {a2_1984}), ...
$A(4)_j$:= (Sujet, $\{A(3)_1, A(3)_2, ...\}$)	(Sujet, {TF1_1978_25/11/79_20h}), (Sujet, {Arte_25/11/2008_20h}), ...
$A(3)_j$:= (Séquence, $\{A(2)_1, A(2)_2, ...\}$)	(Séquence, {TF1_1978_25/11/79_20h_sujet3}), (Séquence, {Arte_25/11/2008_20h_sujet7}), ...
$A(2)_j$:= (Locuteur, $\{A(1)_1, A(1)_2, ...\}$)	(Locuteur, {Royale1, Royale2}), (Locuteur, {Sarkozy1, Sarkozy2})
$A(1)_j$:= (Plan, A_j)	(Plan, Rush1), (Plan, Rush2), (Plan, Rush3), ...

TABLE 1.1 – Réinterprétation de l'exemple de la section 1.2.3.1

1.3 Le contenu audiovisuel vu par la Machine

Cette section est dédiée à la description des caractéristiques sur lesquelles reposera le cœur du mécanisme d'apprentissage. Pour contrebalancer la section précédente, ces caractéristiques symbolisent le point de vue porté par la Machine sur les contenus audiovisuels. Après avoir expliqué brièvement leur nature, nous décrirons, en fonction du média concerné, les méthodes utilisées pour les obtenir.

1.3.1 Nature des caractéristiques utilisées

Nous avons choisi d'utiliser des caractéristiques :
- **globales** afin de cadrer au mieux avec la liberté laissée à l'utilisateur sur la tâche d'organisation à accomplir ;
- **de « bas niveaux »** car nous souhaitons conserver une certaine forme d'objectivité machine sur les données observées. Le grain documentaire que nous avons privilégiés dans nos expériences étant le macrosegment, il était pour nous préférable d'utiliser une stratégie de description ascendante ;
- **extraites automatiquement** pour pouvoir facilement analyser un grand nombre de contenus de nature différentes.

L'ensemble des caractéristiques que nous avons utilisées ont été extraites par des outils développés au sein de l'équipe SAMoVA. Nous avons choisi de concevoir un modèle qui prend en charge des valeurs statistiques calculées sur des séries temporelles comme celles que nous

allons énumérer dans les sections suivantes. Toute autre caractéristique de nature numérique pourrait convenir pour notre étude.

Nous n'avons pas implémenté de caractéristiques textuelles faute de temps. Toutefois, comme dit précédemment, l'intégration de valeurs numériques provenant de descripteurs textuels, comme par exemple les distances entre les mots calculées sur un thésaurus spécialisé, serait tout à fait envisageable.

1.3.2 Caractéristiques Audio

Un son est une *sensation auditive engendrée par une onde acoustique* [7]. Toute variation de pression se propageant dans l'air et pouvant impressionner l'oreille entre dans cette catégorie. Nos analyses se portant essentiellement sur des contenus issus des archives audiovisuelles de l'INA, nous nous intéresserons aux variations de pression dites périodiques (par opposition aux autres variations arbitrairement qualifiés de « bruits ») que sont la parole et la musique. Nous avons donc naturellement choisi des caractéristiques audio qui sont utilisées pour discriminer la parole de la musique.

La musique est difficile à définir d'un point de vue acoustique, car sa perception est étroitement liée aux caractéristiques physiologiques du canal auditif humain. Nous nous contenterons d'étudier la musique à travers les divergences qui existent entre son signal et celui de la parole.

La parole est le résultat d'une « phonation » (source) et d'une « articulation » (filtre) selon un modèle simple de la théorie acoustique de la production de la parole ([Mar02], [Cal89]).

La source est un signal composé d'une partie périodique (vibration des cordes vocales) et d'une partie bruitée. Le conduit vocal sert à transformer le signal de source par des phénomènes de résonance et d'anti-résonance. La parole est donc une alternance de sons voisés (quasi-périodiques) et de sons non-voisés (bruit).

Pour analyser le signal de parole, l'unité classiquement utilisée est la trame acoustique. Il s'agit d'un signal d'environ 10 à 40 ms durant laquelle le signal de parole est supposé quasi stationnaire. Les différents paramètres statistiques que nous allons présenter dans la suite sont calculés sur une trame.

Pour plus de détails sur cette section, se référer à la thèse de Julien Pinquier [Pin04].

1.3.2.1 Le taux de passage à zéro (*Zero Crossing Rate* ou ZCR)

Il s'agit d'un des paramètres les plus utilisés pour la classification parole/musique [Sau96, SS97, ZC99].

Le ZCR d'une trame *i* est défini par :

$$ZCR(i) = \frac{1}{2N}\left(\sum_{n=1}^{N} |sign(x_n(i)) - sign(x_{n-1}(i))|\right) \tag{1.2}$$

7. http ://www.larousse.fr

où $x_n(i)$ est le énième échantillon de la trame *i* et *N* le nombre d'échantillons dans la trame *i*.

De brusques variations du ZCR caractérisent les alternances voisé/non-voisé du signal de parole, alors qu'une absence de variations caractérise l'absence de parole. Pour la musique, ces variations sont très faibles.

1.3.2.2 L'énergie

L'énergie à court terme est une mesure du volume sonore. L'énergie E d'un signal échantillonné $(x_n(i))_{n=1,\ldots,N}$ à support fini est défini par :

$$E(i) = \sum_{n=1}^{N} x_n(i)^2 \tag{1.3}$$

Une faible énergie détectée sur une durée conséquente est synonyme de silence [SS97].

Pour un signal échantillonné de longueur infinie, l'énergie à court terme est calculée sur une fenêtre glissante, d'une taille de l'ordre de 10 ms, ce qui correspond en général à une trame acoustique.

Ce paramètre présente de plus grandes variations pour la parole que pour la musique, et sert donc à les différencier. De plus, couplée avec le ZCR, l'énergie devient un bon discriminant des sons voisés et non-voisés : un ZCR faible et une énergie forte caractérisent un son voisé, alors qu'un ZCR élevé et une énergie moyenne signifient la présence d'un son non-voisé.

1.3.2.3 Le centroïde spectral

Le centroïde spectral est le « centre de gravité » du spectre pour une trame donnée [SS97]. Il est défini par :

$$C(i) = \frac{\sum_{n=1}^{N} \omega_n \cdot S_i(\omega_n)}{\sum_{n=1}^{N} S_i(\omega_n)} \tag{1.4}$$

où $S_i(\omega_n)$ est la composante spectrale de la trame i de fréquence ω_n.

Un centroïde spectral élevé caractérise la présence de musique (et non de parole), car la répartition fréquentielle des hauteurs de sons se fait sur une zone plus importante en musique qu'en parole. De plus, une variation importante de celui-ci est significatif d'une alternance voisé/non-voisé.

1.3.2.4 Le flux spectral

Le flux spectral est défini comme la variation du spectre entre deux trames consécutives :

$$FS = \sum_{n=1}^{N} \left(\frac{S_i(\omega_n)}{\|S_i\|} - \frac{S_{i-1}(\omega_n)}{\|S_{i-1}\|} \right)^2 \tag{1.5}$$

La parole est caractérisée par une variation importante et une valeur faible du flux spectral. La musique, quant à elle, se démarque par une variation faible et une valeur haute de ce même flux.

1.3.2.5 Le *spectral rolloff point*

Le **spectral rolloff point** (exemple figure 1.4) est le point de roulement du spectre : c'est la fréquence sous laquelle on trouve la majorité de l'énergie spectrale (typiquement 95%). Il est plus élevé pour un son non-voisé (riche en hautes fréquences) que pour un son voisé (l'énergie est concentrée dans des fréquences plus faibles). Il permet de caractériser les alternances voisé/non-voisé de la parole.

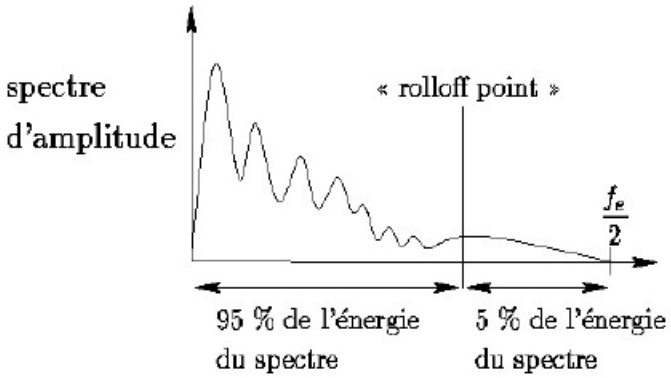


FIGURE 1.4 – Présentation du « Spectral Rolloff Point »

1.3.2.6 La modulation de l'énergie à 4 Hertz

Il s'agit d'un paramètre à la fois issu de l'analyse fréquentielle et temporelle [SS97].

Si nous supposons qu'une syllabe est la combinaison d'une zone de faible énergie (consonne) et d'une zone de forte énergie (voyelle), l'analyse de ce paramètre permet de bien différencier la parole de la musique. En effet, la musique présente une variation de l'énergie autour de 4 Hertz beaucoup plus faible que la parole (voir figure 1.5). Voici la procédure qui nous a permis d'extraire ce paramètre :

1. Le signal est d'abord segmenté en trames de 16 ms sans recouvrement ;
2. Les artefacts indésirables, qui peuvent apparaître lors du passage du domaine temporel au domaine fréquentiel par effets de bords, sont minimisés grâce à un fenêtrage de Hamming ;
3. Pour tenir compte de la perception humaine qui est non-linéaire, on se place sur l'échelle Mel, puis on extrait 40 coefficients spectraux. Nous obtenons ainsi l'énergie des 40 bandes de fréquence, appelés canaux, en accord avec les propriétés de l'oreille humaine [HS85] ;
4. L'énergie dans chacun des canaux fait apparaître des syllabes. On y applique un filtre à Réponse Impulsionnelle Finie (RIF) passe-bande de fréquence centrale 4 Hertz ;

5. On somme sur l'ensemble des canaux l'énergie filtrée, qu'on normalise par l'énergie moyenne ;
6. La modulation est obtenue en calculant, sur une seconde de signal, la variance de l'énergie filtrée en décibels.

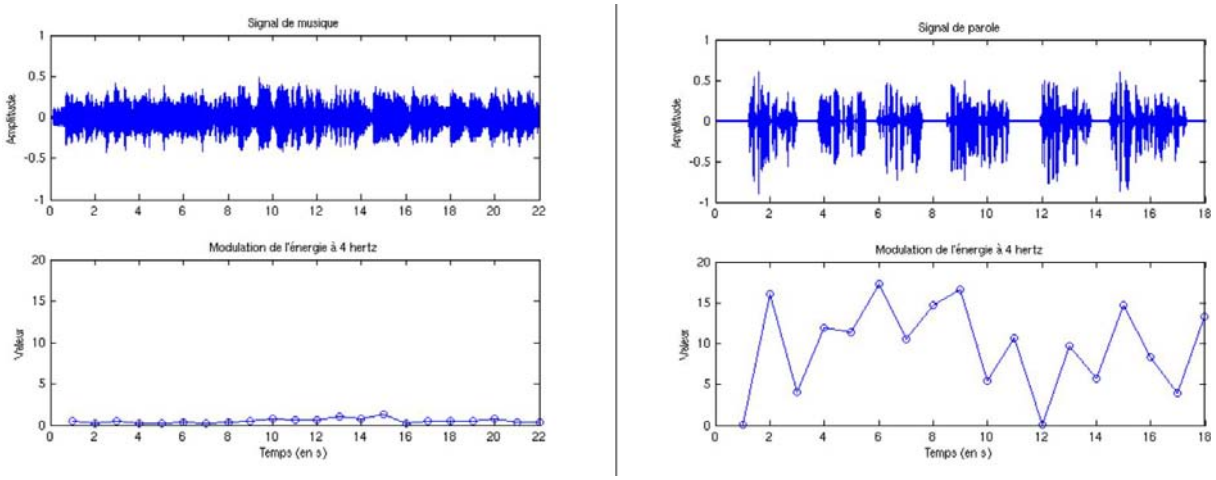


FIGURE 1.5 – Modulation de l'énergie filtrée autour de 4Hz sur un extrait de signal de musique (Mozart) et de parole (6 phrases de parole lue)

1.3.2.7 Modulation de l'entropie

Des études menées sur le signal ont montrées que la structure du signal de musique est plus « ordonné » que le signal de parole [Pin04]. La mesure du désordre de ce signal pourrait donc être intéressant comme outil de discrimination. Nous utilisons un paramètre fondé sur la mesure de l'entropie du signal [Mod89] :

$$H = \sum_{i=1}^{k} -p_i log_2 p_i \tag{1.6}$$

tel que p_i soit la probabilité de l'événement i et k le nombre d'évènements.

L'extraction de ce paramètre se fait selon une procédure semblable à celle utilisée pour la modulation de l'énergie à 4Hz :

1. Le signal est coupé en trames de 16ms sans recouvrement ;
2. On utilise un estimateur non biaisé pour estimer l'entropie. On calcule au préalable un histogramme pour préciser la notion d'événement :

   – Calcul de l'histogramme :
   Soit N le nombre d'échantillons contenus dans la fenêtre considérée. Les bornes minimum et maximum de l'histogramme, respectivement min_h et max_h, sont calculés

ainsi :

$$min_h = min(x) - \frac{\Delta}{2},\ et\ max_h = max(x) + \frac{\Delta}{2}$$

avec $\Delta = \frac{max(x)-min(x)}{N-1}$.

Le pas de quantification de l'histogramme est défini de telle sorte que le nombre N_k de classes soit l'arrondi supérieur de la racine carrée du nombre d'échantillons :

$$N_k \approx \sqrt{N}$$

– Estimation de l'entropie :
L'histogramme précédent nous fournit les probabilités d'apparition des différentes valeurs de l'amplitude. Soit h_i l'effectif de la classe i, pour $i = 1, \ldots, N_h$. On considère que les échantillons sont indépendants, en d'autres termes $\widehat{H} = \sum_{n=1}^{N} \widehat{H_n}$. L'estimateur biaisé s'exprime alors sous la forme suivante [Mod89] :

$$\widehat{H_{biased}} = \frac{\sum_i(-h_i log(h_i))}{N} + log(N) + log(\frac{max_h - main_h}{N_h})$$

Le biais vaut $nbias = -\frac{N_h-1}{2N}$

3. La modulation de l'entropie est obtenue en calculant sa variance sur une seconde de signal.

1.3.2.8 La fréquence fondamentale

La fréquence fondamentale ou *pitch* (notée F_0), caractérise la hauteur d'une note pour la musique ou encore la fréquence de vibration des cordes vocales. Elle permet une discrimination parole/musique par comparaison à une référence acoustique.

1.3.2.9 Quelques autres paramètres

De nombreux autres paramètres existent dans la littérature et sont décrits dans [CP00].
En voici quelques uns que nous n'avons pu implémenter dans notre étude, faute de temps :
– les **coefficients cepstraux** pondérés selon les propriétés perceptives des bandes de fréquences (*Mel Frequency Cepstral Coefficient* ou MFCC) sont utilisés, particulièrement en reconnaissance automatique de la parole et en identification du locuteur [Foo97], pour déconvoluer le signal du conduit vocal et l'excitation des cordes vocales ;
– la détection de pulsation ou **pulse métric** [SS97] qui caractérise le rythme et donc la musique ;
– l'**harmonicité** [WBKW99] permet de savoir si l'on peut se fier à la fréquence fondamentale ;
– la **largeur de bande** [WBKW99] caractérise l'étalement en fréquence d'un spectre ;
– le **timbre** [ZK98, WBKW99] permet de comparer deux sons de mêmes hauteur, puissance et durée ;

– le **nombre de segments de basses énergie** [SS97], paramètre à la fois fréquentiel et temporel.

1.3.3 Caractéristiques vidéo

Nous nous sommes servis des outils développés par Siba Haidar [Hai05] dans le cadre du projet KLIMT [CFJV03] pour extraire la plupart des caractéristiques vidéo décrites dans cette section.

Pour plus de détails sur ces caractéristiques, se référer à la documentation du standard de description de contenus MPEG7 [8].

1.3.3.1 La luminance moyenne

Le calcul se fait sur les coefficients DC (Direct Component) des blocs 8×8 des images : ce coefficient est obtenu en effectuant une TCD (Transformation en Cosinus Discrète) [Bas89] sur le bloc 8×8, puis en prenant la somme des 64 pixels du bloc, divisée par 8. Il s'agit donc de la moyenne des luminances des pixels appartenant à ces blocs :

$$L_p = (R_p + G_p + B_p)/3$$

$$Luminance\_Moyenne = \Sigma\ L_p/nb\_coef\_DC$$

La luminance moyenne est comprise entre 0 et 255.

1.3.3.2 Les deux couleurs dominantes

Tout d'abord, nous utilisons une succession de filtres pour obtenir la première couleur dominante (illustration figure 1.6).

– Grâce à une analyse des histogrammes de couleurs en HLS (Hue, Luminance, Saturation), nous déterminons la teinte H1 la plus représentée, puis nous éliminons tous les pixels qui ne possèdent pas cette teinte (filtre 1 de la figure 1.6) ;
– Par le même procédé, nous localisons sur les pixels qui restent la saturation dominante S1 et nous éliminons les pixels n'ayant pas cette saturation (filtre 2 de la figure 1.6) ;
– Nous ne conservons des pixels restants que ceux qui ont la luminance L1 la plus représentée (filtre 3 de la figure 1.6).

La deuxième couleur dominante est obtenue selon le même procédé, mais en portant les filtres 2 et 3 sur les pixels dont la teinte H2 est la plus éloignée sur le cercle chromatique de H1 (filtre 1 bis de la figure 1.6) : la teinte de la deuxième couleur dominante maximise l'expression suivante :

$$\mathbf{distance\_circulaire(H2, H1)} \times (proportion\ de\ pixels\ ayant\ la\ teinte\ H2)$$

8. http ://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm

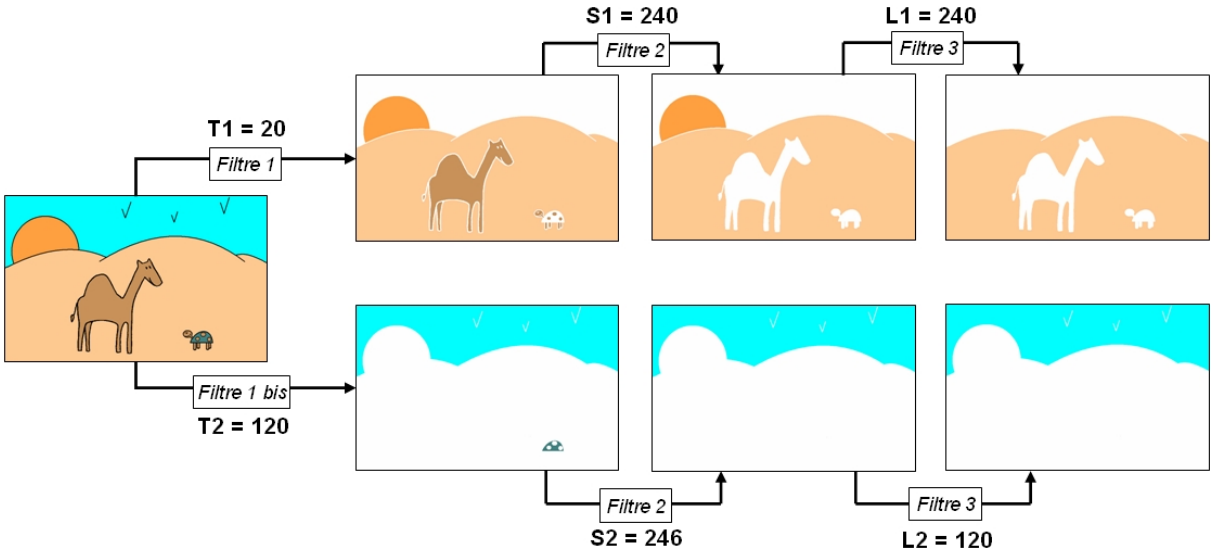


FIGURE 1.6 – Exemple d'extraction des deux couleurs dominantes

L'ordre dans lequel s'appliquent ces filtres a été étudié dans un précédent projet nommé « ViewTime » [9] auquel l'équipe SAMoVA a participé.

1.3.3.3 Le contraste

Nous prendrons comme définition du contraste l'éloignement entre les deux couleurs dominantes.

$$contrast = |L_1 - L_2| + distance\_circulaire(H1, H2) \times log\left(\frac{S1 + S2}{2}\right)$$

1.3.3.4 Le taux d'activité

Cette caractéristique permet de rendre compte de l'évolution globale de l'activité d'une vidéo d'une image sur l'autre. Elle est très utilisée pour aider à la détection de changements de plans ou de mouvements de caméra.

À l'échelle du pixel, nous considérons qu'il y a activité si son intensité lumineuse a « suffisamment » varié. Une phase d'égalisation de l'histogramme des niveaux de gris est faite comme prélude à ce calcul.

$$Taux = \sum_{p=1}^{nb_pixels} \left(1 \times \begin{cases} 1 \text{ si } lum_{image}(p) - lum_{image\_precedente}(p) > 128 \\ 0 \text{ sinon} \end{cases}\right)$$

9. ViewTime : Projet industriel mené avec la compagnie PeakTime, de 1997 à 1998.

1.3.3.5 D'autres caractéristiques visuelles

Voici quelques autres caractéristiques, utilisées dans la communauté de l'indexation de l'image et de la vidéo, qui pourraient être utilisées dans le cadre de cette étude :

- les **orientations et granularités de texture**. Pour analyser de telles caractéristiques, on applique une DFT (transformée de Fourier discrète) sur l'image puis on calcule l'énergie du spectre fréquentiel par l'application de bancs de filtres de Gabor ;
- le **descripteur SCD** (*Scalable Color Descriptor*). Il s'agit d'une quantification uniforme de l'espace HSV [10]. Les valeurs des « bins » subissent une quantification non-linéaire pour réaliser un encodage efficace. La transformée de Haar [Sol98] est ensuite appliquée sur ces valeurs pour rendre le descripteur compact et avoir une représentation multiéchelle de l'histogramme.

1.4 Fossé sémantique : mythe ou réalité ?

Cette section est un point de réflexion porté sur la place de notre travail, en regard de la communauté scientifique au sein de laquelle nous avons évolué durant ces trois années d'études.

Nos travaux portent sur la réalisation d'un outil industriel permettant d'aider un humain à organiser une base de données audiovisuelles, qui utilise des technologies issues de la communauté de l'indexation automatique. Selon un point de vue communément défendu dans ce même domaine, notre outil tenterait de résorber le « fossé sémantique » (appelé *semantic gap* en anglais).

Une interprétation répandue est que cette image symbolise la distance qui existe entre un concept audiovisuel, pouvant être exprimé et compris par un humain, et sa représentation numérique, directement interprétable par la machine.

Si combler ce fossé sémantique constitue un réel problème pour la communauté, peut-être est-ce parce que les objets qu'il sépare sont mal identifiés. À première vue, ces objets pourraient être (cette liste n'est pas exhaustive) :

- les données extraites du signal et les concepts audiovisuels, comme dit précédemment ;
- les méthodes numériques et les méthodes symboliques ;
- les tâches manuelles et les outils automatiques. Le gouffre se définirait ici comme une forme d'intelligence artificielle ;
- ce qu'un ordinateur est capable de faire et ce qu'il ne peut pas réaliser ;
- ce qu'un utilisateur attend de la part d'un ordinateur et ce qu'il ne fera jamais.

C'est une notion particulièrement à la mode aujourd'hui, qui sert de support à l'élaboration de nombreux projets scientifiques, et qui exprime de manière très didactique une réalité scientifique forte : les domaines de compétence des chercheurs issus du monde de l'indexation automatique et les connaissances mises en jeu pour analyser les documents sont extrêmement

10. Espace HSV : espace colorimétrique défini sur la teinte, la saturation et la valeur des couleurs.

variés.

Si nous prenons l'exemple du réseau d'excellence *K-Space* [11] (avec lequel nous avons collaboré), qui est un réseau européen créé dans le but de résorber ce fossé, les intervenants mis en relation sont issus des domaines de l'Indexation Conceptuelle, du Traitement du Signal monomédia (audio, vidéo, textuel) ou multimédia, de la Recherche d'Information, du Web Sémantique... Il y a finalement autant de points de vue pour affronter la manière de combler ce gouffre que de visions du gouffre lui-même. La question est abordée d'une manière très générale, alors que les solutions apportées par ces mêmes scientifiques le sont pour des problèmes extrêmement spécifiques. Entretenir cette vision trop globale de l'activité d'automatisation des tâches d'indexation, maintient la réflexion dans le flou duquel elle a émergé.

Nous proposons de traiter la question d'un point de vue plus pragmatique [Car07], qui tient compte des spécificités des acteurs réellement impliqués dans la mise en œuvre d'un outil comme le nôtre. Ces acteurs mis en relation sont **les technologies** utilisées pour élaborer le système, **l'utilisateur** du système et **le corpus** étudié. La figure 1.7 schématise cette proposition.

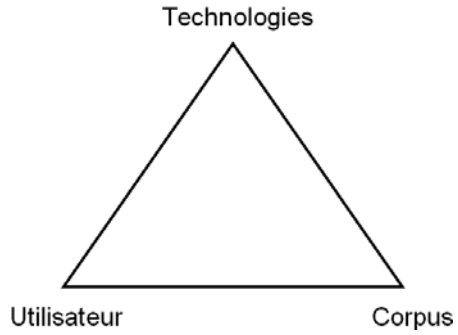


FIGURE 1.7 – L'analyse automatique multimédia vue comme un problème triangulaire

Nous pensons que les problèmes liés au « fossé sémantique » se posent car il se fondent sur une vision bilatérale de l'activité de recherche. Nous ouvrons la discussion en énumérant ci-dessous quelques uns de nos questionnements. Nous ne portons aucun jugement sur les intervenants mis en cause, ces points sont posés pour insister sur le fait que prendre les trois acteurs uniquement par paires pose certes les bases de réflexions nécessaires, mais qui ne peuvent que demeurer incomplètes :

1) **Réflexion sur les usages réels** [utilisateur, corpus, ~~technologies~~]
Il est important que les idées liées aux usages soient réalistes pour pouvoir prendre forme dans une étude comme la nôtre.

Il est souvent dit des chercheurs qu'ils créent de toute pièce des problèmes qui n'ont pas lieu d'être pour pouvoir y apporter des solutions. On peut aussi voir cela comme un processus utile car il nourrit l'imagination et pousse à la créativité. Nous pouvons citer à leur crédit Roei

11. Knowledge Space of Semantic Inference for Automatic Annotation and Retrieval of Multimedia Content

Amit [12], responsable des éditions de l'Ina : « *Il n'y a pas un usage ou un besoin immédiat, spontané des archives. C'est à nous de les créer* ».

Dans le cadre d'une thèse CIFRE, il est important que l'apport théorique, particulièrement soutenu par le laboratoire scientifique impliqué, soit soldé par la création d'un outil (ici un logiciel) qui puisse profiter à l'entreprise. Avoir en tête les limites des technologies dont on dispose (ou que l'on souhaite créer) ; éprouver leurs limites dans le contexte posé par le couple {utilisateur, objets étudiés} ; en d'autres termes, poser un cadre applicatif concret, est nécessaire pour mener à bien un tel projet scientifique.

2) **Réflexion sur les possibilités du système en regard des données mises à disposition** [~~utilisateur~~, corpus, technologies]
Prenons comme exemple les tests d'évaluation des technologies, qui sont directement issus de ce type de questionnement. Ces tests peuvent être portés à grande échelle dans des campagnes d'évaluation, comme par exemple TRECVID [SOK06], ou à l'échelle de son propre prototype. Ici l'utilisateur est effectivement identifié : il n'y en a pas ! Nous sommes dans un cadre scientifique clairement défini, qui répond à un problème totalement déterminé, structuré autour d'un corpus précis, mais qui ne répond à aucun usage effectif. Le risque est ici de se laisser aveuglément guider par la technologie en créant un utilisateur factice idéal, dont les attentes correspondent à des besoins biaisés. On peut alors se demander quel est l'intérêt d'intégrer de telles technologies dans un cadre aussi sensible à l'interaction avec l'utilisateur qu'est le nôtre.

3) **Réflexion sur la place de l'utilisateur vis-à-vis du système** [utilisateur, ~~corpus~~, technologies]
Les tests d'évaluation utilisateur sont à mettre en relation avec ces préoccupations. De la même manière que dans le point précédent, se contenter de créer des outils sur des jeux de tests spécifiques aux technologies utilisées n'est pas suffisant. Si les contenus réellement utilisés par l'utilisateur final ne sont pas pris en compte, les usages ne sont alors pas clairement définis et l'utilisabilité du système pourra en pâtir.

1.5 Conclusion

Ce chapitre a présenté les différents interlocuteurs qui seront amenés à cohabiter tout au long de notre étude, à savoir la machine, l'humain et le contenu. Nous avons exposé différents points de vue que les uns peuvent avoir sur les autres.

Nous avons tout d'abord porté notre attention sur la description d'un contenu audiovisuel par un humain, de la vision que ce dernier pouvait avoir sur le média audiovisuel et les différents grains qu'un contenu pouvait revêtir.

Nous avons ensuite énuméré les différentes caractéristiques, audio et vidéo, utilisées par notre moteur d'apprentissage.

L'accent a été mis, au travers des différents sujets abordés, sur l'aspect générique inhérent à un outil d'aide à l'organisation. En ce sens notre système devra assurer :

12. Propos recueillis par Isabelle Didier dans le carde du dossier « *L'avenir de l'audiovisuel passe-t-il par le web ?* » mis en ligne en décembre 2008 sur le site de l'INA.

– la gestion de contenus hétérogènes, de par leur nature multimédia et la possible navigation hiérarchique entre les différentes granularités ;
– la généricité des usages, car les définitions mêmes des termes « organisation » et « utilisateur grand public » se prêtent à ce type de considération.

Ces points feront l'objet d'une étude dans le chapitre suivant.

Chapitre 2

Visualisation de similarités pour l'organisation de données

Sommaire

2.1 Introduction

Maintenant que nous avons défini le contenu audiovisuel, il nous faut étudier comment le manipuler.

Notre objectif est d'aider à organiser des contenus, mais qu'est-ce qu'organiser signifie vraiment ? N'y a-t-il pas un outil mathématique qui puisse se rattacher à cette notion ? Cela nous permettrait de construire un modèle statistique dédié à l'interprétation d'une organisation et qui serait directement exploitable par la machine.

En étudiant la notion de similarité, nous avons trouvé un candidat potentiellement très intéressant pour élaborer notre système semi-automatisé. Toutefois, il a été nécessaire de créer un formalisme capable de mettre en relation organisation et similarité, afin de cadrer avec les contraintes qui définissent notre étude.

2.2 Organisation et similarité

Cette section présente ce que nous entendons quand nous parlons d'organisation. Nous examinons ce terme de près et présentons comment la similarité est en mesure de satisfaire nos besoins.

2.2.1 Que veut dire « organiser des contenus » ?

Les éléments que nous souhaitons manipuler, c'est-à-dire des objets qui symbolisent des contenus, sont un sous-ensemble des artefacts cognitifs définis par Norman comme des « objets informationnels » [Nor94]. Ce sont des médiateurs qui permettent d'améliorer la perception dans un contexte de travail avec des ordinateurs. Les éléments peuvent être des textes, des images, des images animées, ou encore des constructions dynamiques, des réponses de systèmes, etc. Nous nous restreindrons aux images et images animées que nous nommerons icônes ou vignettes.

Reprenons la définition du terme organiser [13] :

> **Organiser :**
> S'occuper de chacun des éléments d'un ensemble de façon à constituer un tout cohérent et adapté à sa destination.

Lorsqu'on organise une base de vignettes représentant des contenus audiovisuels, parler de cohérence revient à donner à cet ensemble de contenus de nature hétérogène une structure qui facilite leur consultation, la rendant plus intuitive, plus lisible.

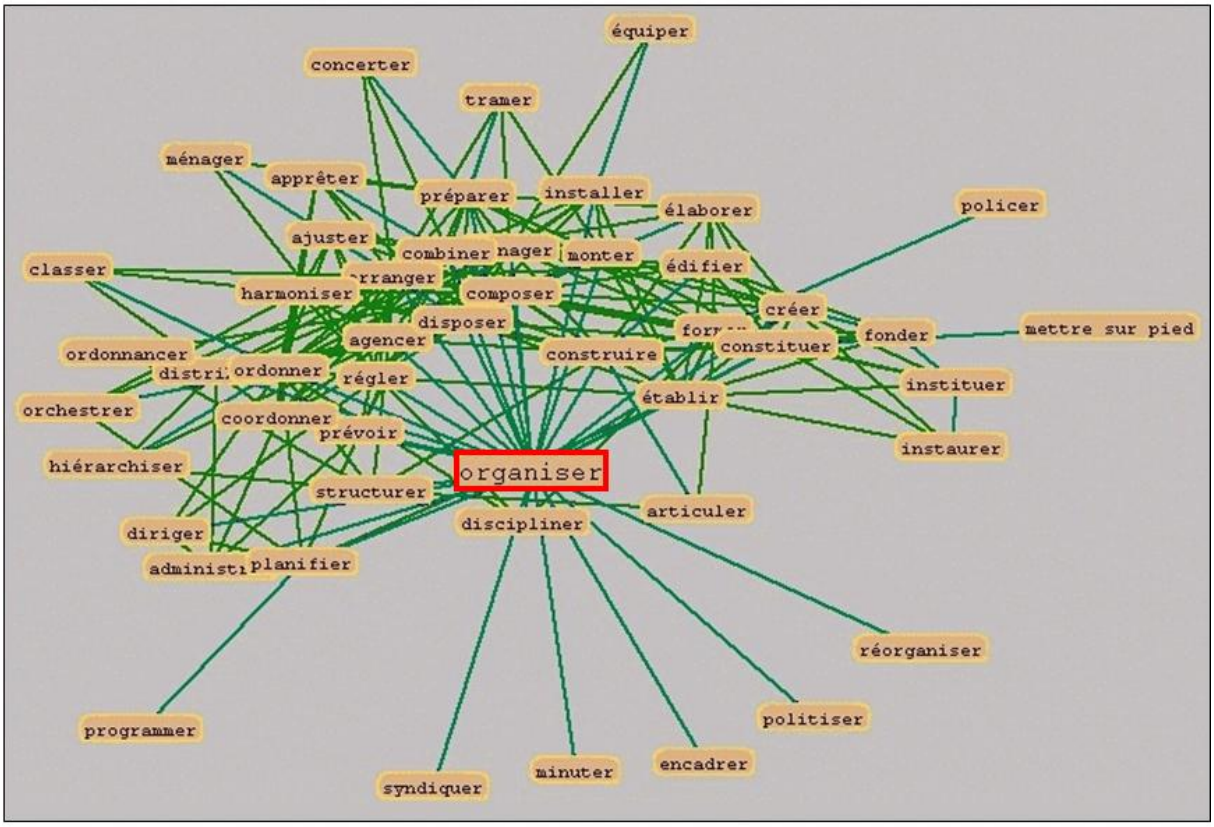
L'organisation touche à des idées qui sont d'apparence très proches, mais de nature très distinctes. Il n'y a qu'à s'attarder sur les 49 synonymes de ce terme (figure 2.1a, ces synonymes sont extraits de DycoSyn [14] pour percevoir la complexité de la chose.

13. http://www.larousse.fr

14. DicoSyn est un dictionnaire de synonymes constitué de sept dictionnaires classiques (Bailly, Benac, Du Chazaud, Guizot, Lafaye, Larousse et Robert) dont ont été extraites les relations synonymiques à l'ATILF (http://www.atilf.fr/) puis homogénéisés au CRISCO (http://elsap1.unicaen.fr/).

administrer ; agencer ; ajuster, disposer ; aménager, équiper ; apprêter, arranger, concerter, harmoniser, ménager ; articuler, construire, édifier ; classer, hiérarchiser, orchestrer, ordonnancer, ordonner ; combiner, tramer ; composer, élaborer ; constituer, fonder, instaurer, instituer, mettre sur pied ; coordonner ; créer ; diriger ; discipliner ; distribuer ; encadrer ; établir ; former, policer ; installer ; minuter ; monter ; planifier, programmer ; politiser ; préparer ; prévoir ; régler ; réorganiser ; structurer ; syndiquer ;

a. Liste des 49 synonymes.



Légende: Il existe une arête (A,B) si les verbes représentés par les sommets A et B sont synonymes.

b. Graphe des synonymes.

FIGURE 2.1 – Synonymes du verbe « organiser » extraits de DicoSyn

Toutefois, ces synonymes présentent des caractéristiques communes très fortes qui peuvent nous être profitables. Le graphe des synonymes (figure 2.1b) contient trois cliques [15], soit trois unités de sens dominantes dans la structure du graphe :

Clique1 = *{organiser, classer, ordonner, arranger, distribuer}*

Clique2 = *{organiser, classer, ordonner, arranger, harmoniser}*

Clique3 = *{organiser, classer, ordonner, distribuer, hiérarchiser}*

15. Clique : terme employé en *théorie des graphes*, ensemble de sommets deux-à-deux adjacents (notion de graphe complet).

L'intersection de ces trois cliques est l'ensemble de termes suivant :

$$\textbf{Clique1} \cap \textbf{Clique2} \cap \textbf{Clique3} = \{organiser, classer, ordonner\}$$

Nous voyons que, même sans avoir à contextualiser le précédent graphe des synonymes autour de l'organisation de contenus (il va de soit que nous ne sommes pas concernés par des termes comme « policer » ou « politiser »), il émane deux idées maîtresses qui vont guider la nature de la structure sous-jacente à une organisation : d'une part la notion de **regroupement** et d'autre part la notion d'**ordre**.

Il existe de nombreux amalgames dans les termes utilisés couramment pour parler de regroupement et c'est en étudiant leur utilisation dans différents domaines que nous en sommes venus à mettre en avant quatre sous-tâches présentes derrière les termes « organisation d'objets informationnels » (les exemples en italique dans la liste ci-dessous font référence à la figure 2.2 qui servira à illustrer le propos) :

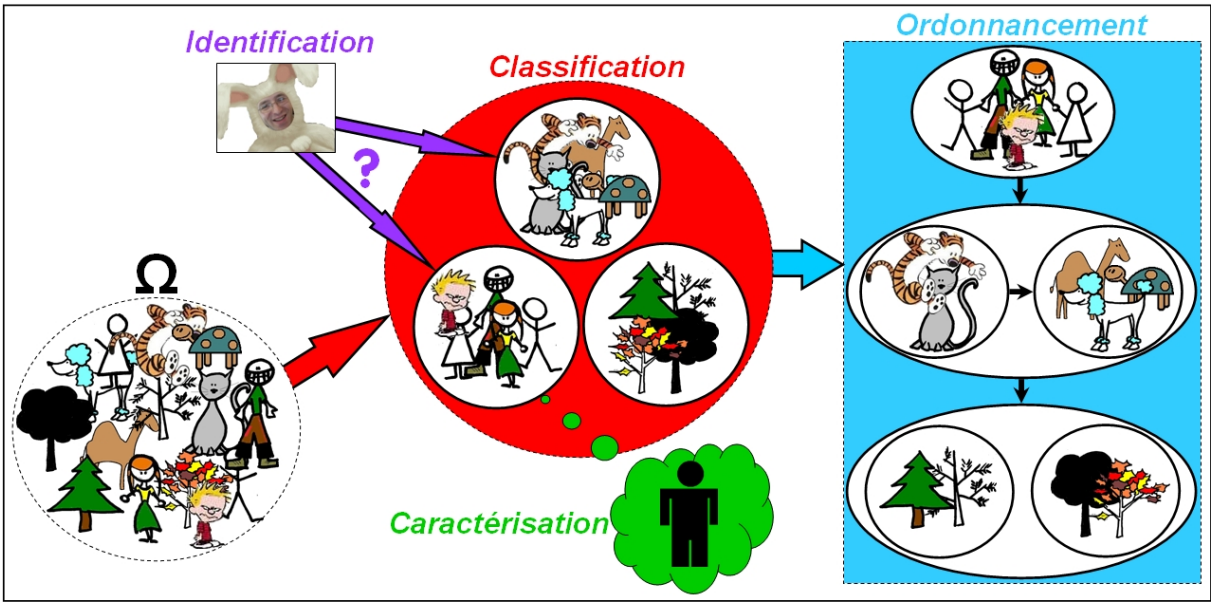


FIGURE 2.2 – Exemple d'organisation de contenus

La classification a pour but de structurer les données de manière à composer des groupes ou « classes », à la fois contrastés et homogènes. Il s'agit donc de mettre en avant les ressemblances et les dissemblances des éléments pour constituer ces classes.

*Exemple : les images de l'univers Ω sont agencées en trois classes : **Humains**, **Animaux** et **Végétaux**.*

L'identification consiste à déterminer la classe à laquelle un élément inconnu appartient (on parle alors de classement), ou à trouver à quel élément de Ω il ressemble le plus.

*Exemple : utiliser les caractéristiques des classes **Humains** et **Animaux** pour savoir où doit être rangée la photo inconnue.*

La caractérisation (ou généralisation) consiste à rechercher des informations communes à un ensemble d'éléments, permettant ainsi d'en construire une représentation explicite.

*Exemple : extraire les invariants des éléments de la classe **Humains** pour définir un ensemble de caractéristiques définissant le genre humain.*

L'ordonnancement d'objets a pour but de structurer les éléments ou groupes d'éléments de manière à les ordonner entre eux.

Exemple :

- *Les trois classes sont ordonnées de manière à considérer d'abord les **Humains**, puis les **Animaux** et enfin les **Végétaux**.*
- *Les éléments de la classe **Animaux** sont séparés en deux classes : les **Félins** et les **Autres**, puis ordonnés de telle sorte que les **Félins** soit avant les **Autres**.*
- *Les éléments de la classe **Végétaux** sont scindés en deux classes : les **Conifères** et les **Feuillus**. Ces classes ne sont pas ordonnées.*

2.2.2 Quel lien avec la similarité ?

Revenons-en à la similarité. Lorsque nous cherchons la définition de la similarité, qui est le fait d'être similaire, dans un dictionnaire [16], nous obtenons ceci :

> **Similaire :**
> Qui est plus ou moins de même nature qu'une/que d'autre(s) entité(s) ;
> Qui peut, sur certains points, être assimilé à une/à d'autre(s) entité(s).

Chercher à représenter la similarité peut relever du quantitatif (« plus ou moins ») et/ou du qualitatif (« sur certains points »).

L'utilisation de la similarité dans les tâches de classification, d'identification et de caractérisation a été explorée par Giles Bisson [Bis00]. Son étude porte sur différents domaines : l'analyse de données, la reconnaissance des formes, l'apprentissage symbolique et les sciences cognitives. Nous pouvons également citer le travail de Christine Michel [Mic02], qui porte sur la définition de mesures de similarités dédiées à l'ordonnancement d'objets.

Bien que nos travaux ne s'inscrivent pas dans la totalité des domaines précédemment cités, ils s'en inspirent fortement. Nous proposons dans les sections suivantes d'explorer quelque peu l'incidence de l'étude de la similarité sur ces différents domaines de l'informatique.

2.2.2.1 Classification et similarité

Le critère fondamental de la constitution d'une classe consiste à rapprocher les éléments qui se ressemblent le plus, tout en éloignant ceux qui présentent des dissemblances. En d'autres termes, il s'agit de maximiser la similarité intra-classe tout en minimisant la similarité inter-classe.

16. Dictionnaire « le Trésor de la Langue Francaise informatisé » (http://atilf.atilf.fr/)

L'étude de la classification dans le domaine de la reconnaissance des formes a conduit à la création de deux familles de méthodes :

- les méthodes non hiérarchiques ou à partitionnement, qui consistent, à partir d'une partition initiale, à chercher à améliorer itérativement la partition en minimisant un certain critère. Un très bon représentant de ces méthodes est l'algorithme des nuées dynamiques [Did91], duquel découle l'algorithme du K-means [DHS73].
  Les algorithmes de clustering topographique, tels que les cartes auto-organisatrices de Kohonen [Koh82] et ses dérivées ([Lut94], [BSW98], [GO98]) entrent également dans cette catégorie.
- les méthodes hiérarchiques ([LW67], [CBT04]), qui procèdent par fusions successives d'ensembles de points : à un niveau de précision donné, deux individus peuvent être confondus dans un même groupe, alors qu'à un niveau de précision plus élevé, ils seront distingués et appartiendront à deux sous-groupes différents.

Nous pouvons également évoquer ici les nombreux travaux sur les algorithmes de création de distances pour la classification semi-supervisée et non-supervisée ([XNJR03], [KT03], [XNZ08], [SJ04], [GR06], [BHHSW06], [HLLM06]), dont nous reparlerons mieux dans le chapitre suivant (section 3.3.1.2).

En apprentissage symbolique, l'étude de la classification dans le cadre non supervisé a donné naissance à des méthodes de classification conceptuelle ([Han90], [MS84]) et de formation de concepts ([GLF89], [Fis89]).

Dans le domaine des sciences cognitives, le lien entre classification et similarité a été largement abordé, dans le sens où la « catégorisation » est un processus central du système cognitif humain [Gol94], [Tve77].

2.2.2.2 Identification et similarité

Du point de vue de l'analyse de données, pour déterminer la classe ou l'élément le plus ressemblant à un objet α inconnu, il est souvent suffisant de trouver l'objet de Ω qui maximise la mesure de similarité avec α. La méthode des k plus proches voisins (Kppv) et ses dérivées sont des applications directes de ce principe.

On utilise particulièrement cette approche dans le domaine de l'Intelligence Artificielle, à travers des disciplines comme l'apprentissage à partir d'instances (*Instance Based Learning* [AKA91], [TW06]), ou le raisonnement à partir de cas (*Case-Based Reasoning* [Voß94], [ZS04], [PAO05]).

On peut également trouver dans le domaine des sciences cognitives, des études faites sur la similarité à travers les liens qui l'unissent au raisonnement par analogies (*Analogical Reasoning* [GM98], [LSG07]).

Notons que les principes de classement et de classification peuvent être étroitement liés, puisque certains concepts de classification reposent sur des mécanismes d'identification incrémental.

Pour conclure, nous remarquerons que lorsqu'on emploie le terme de similarité dans le sens plus général de la « ressemblance », une mesure de similarité est symétrique en classification, alors qu'elle peut ne pas l'être en identification. C'est particulièrement le cas lorsque l'objet de référence est une description de concept et non une instance.

2.2.2.3 Caractérisation et similarité

La construction d'une représentation explicite d'un ensemble de données a été étudiée en apprentissage symbolique ([Mit82], [KG86], [Nib88]). L'idée dominante est ici de se servir du résultat d'une classification et d'affecter une *fonction de reconnaissance symbolique* aux classes créées.

Toutefois peu de travaux présentent la notion de similarité comme fondement de la généralisation. Ce lien direct a été étudié en sciences cognitives, et montrent que le stimulus de généralisation peut être expliqué en termes de similarités, dans des tâches de conditionnement et d'identification par Shepard [She57], puis dans des tâches de catégorisation [MS78] et de raisonnement par induction [Osh90]. Ces études présument que les différences multidimensionnelles entre les stimuli peuvent être résumées par une valeur unique, résultat d'une fonction de similarité, avant d'entrer dans le processus décisionnel humain (présomptions remises en cause dans de récents travaux [JML06]).

2.2.2.4 Ordre et similarité

Dans le domaine de la recherche d'informations, les systèmes travaillant avec des objets informationnels présentent leurs réponses sous forme d'éléments structurés de manière ordonnée, afin de témoigner d'une hiérarchie, d'un système de classement (*ranking*), ou autre selon le besoin.

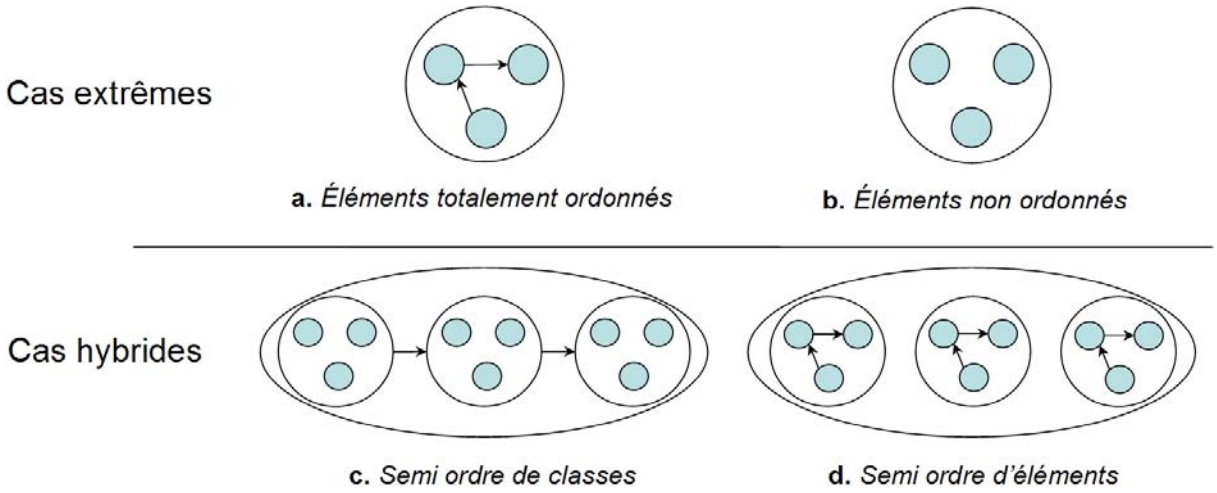


FIGURE 2.3 – Les différentes structures de représentation visuelle d'objets informationnels

Ces réponses peuvent se présenter (voir figure 2.3) :
– sous la forme d'éléments totalement ordonnés (figure 2.3a) ;

- sous la forme d'éléments non ordonnés (figure 2.3b, notons ici que l'absence de relation d'ordre s'inscrit comme un processus d'ordonnancement) ;
- sous des formes hybrides de semi ordre [Mic02], qui prennent en considération d'éventuelles classes d'éléments :
  - le semi-ordre de classe (figure 2.3c) : les classes sont ordonnées mais les éléments à l'intérieur ne le sont pas ;
  - le semi-ordre d'éléments (figure 2.3d) : les classes ne sont pas ordonnées mais les éléments à l'intérieur le sont.

2.2.2.5 Conclusion

Nous trouvons pertinent, à la vue de cet état de l'art, d'utiliser l'outil « similarité » pour témoigner de tout type de structure propre à une tâche de regroupement ou d'ordonnancement de contenus audiovisuels.

De plus, nous pensons qu'une fois mêlé à la notion d'identification, il nous sera possible de pleinement exploiter cet outil pour témoigner du reste des tâches organisationnelles, si toutefois un formalisme fonctionnel nous permet d'œuvrer dans ce sens.

2.3 Formalisme d'organisation fondé sur la similarité

Il nous est nécessaire de nous pencher dans un premier temps sur la visualisation des données et sur les principes d'interactions offerts à l'utilisateur. Nous avons besoin d'un formalisme dédié à l'organisation assistée de contenus audiovisuels :

- qui permette à l'utilisateur de consulter ces contenus ;
- qui s'adapte aux quatre sous-tâches identifiés dans la section précédente : la classification, la caractérisation, l'ordonnancement et l'identification (section 2.2) ;
- qui permette l'exploration multi-grains de notre base documentaire ;
- avec lequel nous puissions constituer simplement le corpus nécessaire au moteur d'apprentissage du système (entrées du moteur d'apprentissage) ;
- qui permette de visualiser clairement la proposition du système (sorties du moteur d'apprentissage).

Nous décrivons notre proposition dans cette section. Nous commençons par présenter les choix que nous avons faits en matière de visualisation, pour ensuite décrire le formalisme que nous avons adopté.

2.3.1 Nos choix concernant la visualisation des données

L'interdépendance entre le moteur d'apprentissage et l'interface utilisateur de notre système est assez forte, et nous a menés à faire des choix technologiques en matière de visualisation de données que nous allons préciser.

2.3.1.1 Postulats

Nous prenons le parti de représenter les contenus sur un écran d'ordinateur, sous formes d'objets informationnels tels que des images fixes ou animées.

Nous considérons que tous les contenus présents à l'écran, à un instant donné, sont d'une même granularité documentaire.

2.3.1.2 Structure des données : le graphe

Par définition, un graphe $G = (V, E)$ est constitué d'un ensemble V de sommets (ou nœuds) et d'un ensemble E d'arêtes, $E \subseteq V \times V$.

Une manière simple de déterminer si l'utilisation d'un graphe est légitime pour représenter la structure de ces données, est de répondre à la question suivante : « *existe-t-il une relation inhérente aux données manipulées qui puisse être visualisée ?* ».

Si la réponse est « non », alors les éléments n'ont pas de structure et le but du système de visualisation est d'aider à découvrir des relations entre les données à l'aide de moyens visuels. Si la réponse est « oui », alors la structure de données peut être représentée par un graphe dit « général ».

Partant de l'hypothèse $\mathscr{C}_0$ présupposant l'existence d'une relation entre la position des objets informationnels et leurs valeurs descriptives, et forts de l'utilisation de la similarité pour témoigner d'une relation entre les contenus (contrainte $\mathscr{C}_1$), nous trouvons pertinent d'utiliser un graphe :

- un noeud représente un contenu ;
- une arête relie deux contenus. Elle représente la relation de similarité qui existe entre ceux-ci ; sa longueur quantifie leur dissimilarité : plus la dissemblance entre deux contenus est faible, plus la longueur de l'arête est petite.

2.3.1.3 Placements du graphe par modèle d'énergie

Nous allons être amenés à visualiser des données dans un espace à deux dimensions (l'écran de l'ordinateur). Le mécanisme d'aide que nous mettons en place va calculer des similarités entre certains contenus, pour en inférer sur d'autres. Nous voulons que des sommets du graphe puissent se déplacer de manière automatique, pour traduire visuellement la réponse du système. Afin de bouger ces sommets, les contraintes de similarité doivent être traduites sur les arêtes, pour leur imposer une certaine longueur. Pour ce faire, nous avons décidé d'utiliser un modèle d'énergie.

Les algorithmes de placement par modèle de force ou d'énergie sont les plus couramment utilisés pour la visualisation des graphes généraux. Lorsque les graphes sont modérément complexes, ils fournissent de bons résultats [Thi06].

Ces techniques partent toutes du même postulat : il existe une analogie entre un graphe et un modèle physique « masse-ressort ». Un sommet est une particule qui possède une masse. Une arête est considérée comme un ressort, caractérisé par un coefficient de raideur et une longueur au repos. Les particules bougent, s'attirent et se repoussent les unes les autres. Elles sont

contraintes par les arêtes qui leur imposent de rester à une distance la plus proche possible de la longueur de leur ressort au repos. Le système bouge jusqu'à trouver une position d'équilibre. Cette stabilité n'est atteinte que pour une configuration d'énergie minimale.

Les différents algorithmes ([Ead84], [KK89], [DH96], [FR91]) procèdent de façon similaire : une fois la position initiale des sommets déterminée, leur nouvelle position est recalculée à chaque itération. L'algorithme s'arête lorsqu'un état de stabilité est trouvé. Les différences entre ces algorithmes dépendent du choix des forces d'attraction et de répulsion, et de la façon dont l'énergie est minimisée. Nous rentrerons dans le détail de ce modèle dans la section 4.3.2, portant sur les choix technologiques propres à notre moteur de visualisation.

2.3.1.4 Espace dynamique

La figure 2.4 présente une capture d'écran annotée de notre Interface Graphique dédiée à l'Utilisateur (communément appelée GUI pour *Graphic User Interface*).

Elle se compose de quatre zones principales :
- la **zone I** est une fenêtre dans laquelle il est possible d'organiser des contenus audiovisuels ;
- la **zone II** fournit des informations sur les contenus de la zone I ;
- la **zone III** présente une vue horizontale de la base de données. Un curseur de défilement horizontal permet de s'y déplacer, d'y sélectionner des contenus qui peuvent être **rapatriés** vers la zone I afin d'y être organisés ;
- la **zone IV** propose des outils d'interaction pour agir sur l'ensemble de la GUI : sauvegarde et chargement des données, quitter, lancement de l'apprentissage via le rapatriement d'un ou plusieurs contenus, etc...

Nous nous attardons ici sur la zone I, que nous avons appelé **espace dynamique**. Il s'agit d'une interface en deux dimensions de type WIMP (*Windows, Icons, Menus, Pointing device*) [VD97] : une fenêtre de visualisation (figure 2.4 ①) permet d'appréhender des contenus représentés sous forme d'icônes (figure 2.4 ③), que l'on peut manipuler (sélectionner, déplacer, consulter) à l'aide d'un pointeur, ou « souris » (figure 2.4 ②).

Une vignette représentative du contenu audiovisuel constitue le corps de l'icône. Des fonctionnalités permettant d'entrer en interaction avec le système sont présentées dans un menu (figure 2.4 ④).

Dans la suite nous parlerons indifféremment de contenu ou d'icône, ces deux termes désignant des objets informationnels manipulables via la GUI.

Notre interprétation du terme **dynamique** est la suivante : toute opération effectuée par le système entraînant des répercussions dans la fenêtre de visualisation de la GUI (comme une réorganisation automatique des entités présentes après un rapatriement de contenus depuis la base) doit engendrer un mouvement des entités dans cet espace. Nous voulons donner la sensation que l'humain et la machine modifient l'environnement de la même manière, c'est-à-dire en manipulant des entités physiques régies par des lois mécaniques (simulées par le modèle

FIGURE 2.4 – Capture d'écran annotée de notre prototype

d'énergie) perceptibles pour l'utilisateur.

Notons pour conclure que cette GUI permet de porter deux regards différents sur ces objets en mêlant deux espaces :

- un **espace de représentation**, qui correspond au point de vue porté par l'humain sur le contenu, et dans lequel un objet est une icône ;
- un **espace de description**, qui fait référence au jugement porté par la machine sur le contenu, et qui considère ce même objet comme un vecteur.

Un complément d'information sur notre interface est donné dans l'annexe F.

2.3.2 Description du formalisme

Cette section explique le travail de recherche que nous avons mené sur un formalisme suffisamment souple et lisible pour permettre à un utilisateur de travailler convenablement sur l'organisation de contenus.

2.3.2.1 Suggérer sans imposer grâce à l'identification

Nous allons dans un premier temps porter un regard global sur la suite de cette section, afin de mieux comprendre certains choix que nous avons faits.

Comme nous l'avons expliqué dans l'introduction (section 2.3.2), la nature généraliste du terme « organiser » nous impose de ne pas choisir la sous-tâche organisationnelle à accomplir à la place de l'utilisateur. Nous intervenons en amont pour faciliter sa prise de décision.

Corollaire à cela, l'**attribution d'une classe** (section 2.3.2.2) à un contenu ne se fera que par l'intervention explicite de l'utilisateur. C'est un parti pris fort que nous avons fait ici : cela implique que le système d'aide ne sera pas en mesure d'attribuer explicitement une classe à des contenus, et nous nous interdisons donc toute possibilité de faire du clustering ou de la classification semi-supervisée. Ces tâches seront suggérées à travers le processus d'**identification** visuelle, en tenant compte du rapprochement (resp. l'éloignement) des contenus en fonction de leur ressemblance (resp. dissemblance).

C'est cette même notion qui sera également chargée d'évoquer une **structure d'ordonnancement**, ou de rendre compte de la **représentativité d'un contenu** (sections 2.3.2.3 et 2.3.2.6).

2.3.2.2 Organisation des données et organisation de l'espace de travail

Dans l'espace dynamique, l'utilisateur peut **classer** des contenus : les contenus mis dans une même classe sont encadrés d'une même couleur. Les icônes qui ne sont pas encore classées ne possèdent pas de cadre coloré.

La limite d'une classe n'est pas physiquement définie. Par exemple, il n'y a pas de cercle englobant ou de polyèdre constitué par l'enveloppe convexe des éléments de la classe. Seule la couleur qui enveloppe les contenus permet d'identifier une classe.

Il est possible d'afficher jusqu'à huit classes à la fois à l'écran. Les couleurs des différentes classes proviennent d'une palette adaptée à la visualisation cartographique, issue des travaux de [Bre94] (voir figure 2.5).

FIGURE 2.5 – Palette de couleurs utilisée pour illustrer les différentes classes

Nous avons interprété un contenu classé comme un contenu qui interviendra dans l'organisation automatique des données : *si je classe un contenu, c'est que je reconnais qu'il a des caractéristiques qui sont communes aux autres contenus de la même classe, et que je souhaite que ces caractéristiques soient exploitées*.

Toutefois, l'utilisateur doit également avoir la possibilité de positionner un contenu dans l'espace que nous lui offrons sans que les coordonnées de celui-ci soient constamment perturbées par des réorganisations automatiques. Son but peut être de simplement écarter un élément pour y revenir plus tard, sans pour autant affecter une sémantique particulière à la place où il l'a déposé.

Nous avons décidé de symboliser ceci en « verrouillant » ces contenus sur l'environnement graphique. Nous avons appelé ce principe l'**ancrage** : toute intervention de la machine pouvant modifier les coordonnées d'un contenu ancré est proscrite. L'icône tourne de 45 degrés horaires pour représenter l'ancrage du contenu.

Par opposition, la position d'un contenu non ancré pourra être modifiée par la machine durant une phase dynamique.

Pour résumer, nous concevons notre espace dynamique comme un outil ayant deux fonctionnalités, non antagonistes au demeurant (voir figure 2.6) :

- une fonction d'*organisation de l'espace de travail*. La notion de contenu ancré, noté A (qui s'oppose au contenu libre ou non ancré, noté $\overline{A}$) y est directement attachée. Un contenu est ancré si l'on ne veut pas qu'il bouge ;
- une fonction d'*organisation des données*. La notion de contenu classé, noté C (qui s'oppose au contenu non classé, noté $\overline{C}$) y est directement attachée. Un contenu est classé si l'on veut qu'il intervienne dans l'organisation automatique des contenus de sa classe.

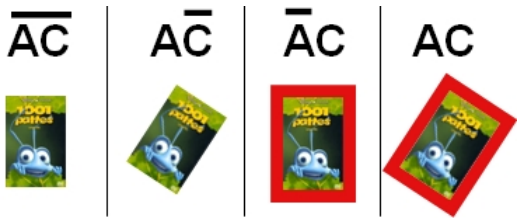


FIGURE 2.6 – Les différentes représentations d'un contenu dans l'interface

Nous prenons le parti de considérer que l'ensemble des contenus d'une même classe a un comportement qui lui est propre, sans rapport avec tout autre élément de l'espace dynamique. Appartenir à une classe a une symbolique d'autant plus forte que celle-ci constitue un espace autonome, dont la cohérence est assurée par un graphe dédié.

2.3.2.3 Le représentant, une entité polysémique

Nous avons décidé de créer une entité abstraite nommée le **représentant**, qui va nous servir à illustrer différents concepts.

D'un point de vue « classification », le représentant d'une classe est à la fois :

- le centroïde géométrique de la classe (du point de vue de l'espace de représentation) : il s'agit de la moyenne des coordonnées des entités ancrées constituant la classe.
- le centroïde descriptif de la classe (du point de vue de l'espace de description) : le représentant doit témoigner de l'ensemble des valeurs descriptives provenant des contenus

ancrés qui composent sa classe. Les détails sur cet aspect du représentant seront développés dans le chapitre suivant, section 4.2.5.1.

Cette double assimilation nous permet de faire l'amalgame entre le représentant d'une classe et l'ensemble des contenus de cette même classe : cette vision du problème se rapproche du point de vue « caractérisation » (ou « généralisation ») d'une tâche d'organisation (chapitre 2 section 2.2.2.3), car elle permet de considérer le représentant comme l'abstraction qui généralise les caractéristiques communes à une classe.

2.3.2.4 Interlude : éléments de réflexion

Nous avons dans un premier temps pensé proposer à l'utilisateur d'agir, dans une certaine mesure, sur les liens qui unissaient les contenus au sein d'une même classe. Nous pensions le laisser choisir à tout instant entre quatre types de configurations qui nous semblaient fonctionnelles et intéressantes (modèle T.A.R.N., illustré sur la figure 2.7) :

- la relation du type **Total**, qui relie tous les éléments d'une classe, y compris le représentant de celle-ci. Cette configuration avait été adoptée pour la mise en relation des contenus sans classe avec les représentants des éventuelles classes définies par l'utilisateur ;

- la relation du type **Absolu**, qui relie tous les éléments d'une classe, sans tenir compte du représentant. Cette configuration avait été adoptée pour la mise en relation des contenus au sein d'une même classe ;

- la relation du type **Relatif**, qui relie chaque élément de la classe au représentant de sa classe et uniquement à celui-ci. Nous pensions utiliser cette configuration pour visualiser directement la représentativité de chacun des éléments (plus il aurait été proche du représentant, plus il aurait généralisé les attributs de sa classe) ;

- la relation de type **Néant**, qui laisse les contenus sans lien ; configuration rattachée aux contenus ancrés, qui ne bougent pas dans l'espace dynamique.

Nous ne nous attarderons que peu sur ce formalisme, car nous ne l'avons pas retenu, mais la réflexion que nous avons eue à son sujet a fait émerger différents problèmes liés :

1. à la portée de l'interaction utilisateur vis-à-vis du graphe : doit-on lui laisser l'opportunité d'interagir sur la nature du graphe entre les contenus ?
   - si oui, lui propose-t-on d'agir directement sur les liens entre les contenus (ajout, suppression) ou lui impose-t-on les configurations pré-établies du formalisme T.A.R.N. ?
   - sinon, quelles relations privilégier et quelles mécanismes adopter pour couvrir au mieux l'éventail des possibilités offertes par une interface organisationnelle telle que nous la concevons ?
2. à l'intérêt de la visualisation du représentant : faire cohabiter des contenus réels et des entités virtuelles peut porter à confusion. Comment désambiguïser le problème ?

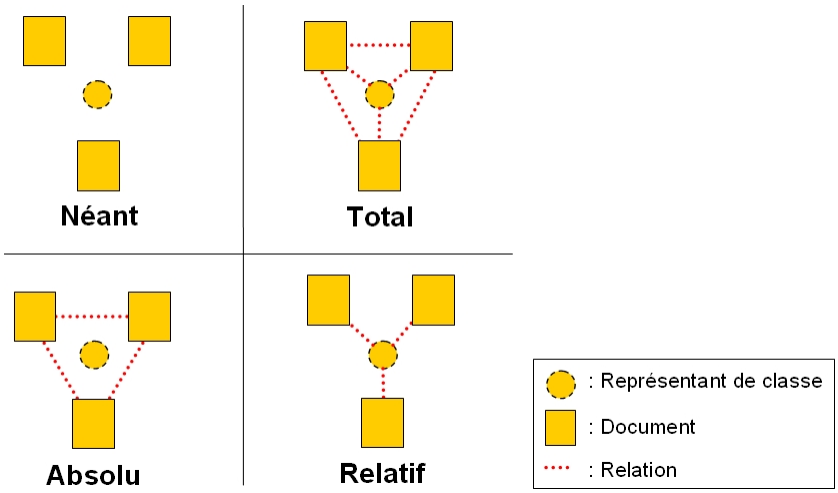


FIGURE 2.7 – Formalisme T.A.R.N.

3. à la perception de différents modèles comportementaux des contenus : comment présenter, sur un même espace visuel, deux groupes d'objets de même nature qui n'obéissent pas aux mêmes lois physiques ?
4. aux liens existant entre des contenus classés et non classés : que signifie « classer un contenu » ? Comment se servir de cette symbolique ?
5. à la cohabitation humain-machine sur une même interface dynamique : quelles sont les répercussions de la manipulation des mêmes contenus à la fois par la machine et par l'utilisateur ?

La suite de cette section présente les solutions que nous avons adoptées en fonction des différents problèmes précédemment mis en lumière.

2.3.2.5 Interaction avec l'utilisateur

L'utilisateur n'a pas la possibilité d'agir directement sur les liens qui unissent les contenus.

Nous avons préféré imposer les configurations suivantes, qui nous semblaient les plus pertinentes au vu du mode de visualisation et des propriétés fonctionnelles des contenus (ancrage et/ou classement). La figure 2.8 illustre ces propriétés.

1) Les entités ***ancrées et non classées*** *($A\overline{C}$) ne seront en interaction avec personne.*

Nous proposons également de faire apparaître un contenu non classé comme une icône légèrement transparente et sans cadre coloré. Son intervention dans une classe sera symbolisée par une prise de consistance illustrée à la fois par la levée de la transparence et par la matérialisation du cadre de la couleur représentant la classe choisie.

2) Chaque entité ***non ancrée et non classée*** *($\overline{AC}$) sera uniquement en relation avec tous les représentants virtuels de classes.*

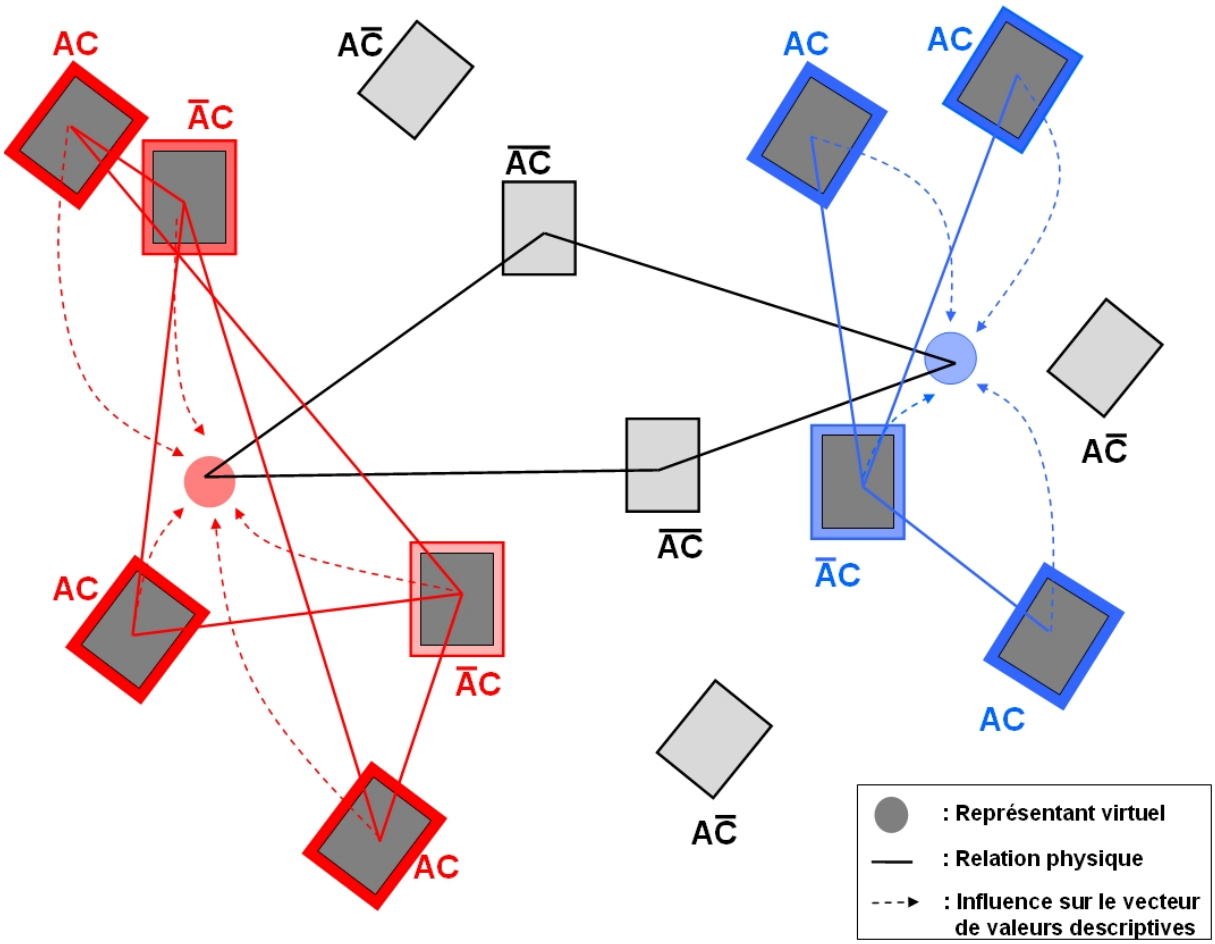


FIGURE 2.8 – Illustration des choix retenus pour la visualisation des données

En partant de la constatation générale que les contenus non classés ($\overline{C}$) sont probablement ceux qui auront le plus de chance de ne pas encore avoir été consultés [17], nous assumons le fait de ne pas les mettre en relation entre eux pour ne pas surcharger les calculs et ainsi respecter les contraintes du temps interactif pour l'utilisateur de l'interface.

3) Le vecteur des valeurs descriptives et la position du représentant virtuel d'une classe seront calculés en considérant tous les éléments de la classe, faisant de lui un représentant statistique de celle-ci.

4) Les entités ***non ancrées et classées*** *($\overline{A}C$) seront uniquement en relation par les contenus ancrés et classés (AC) de la même classe.*

Il y a donc deux types de représentations de similarités évoquées en même temps sur l'espace dynamique : une **inter-classe globale** définie par la position des représentants virtuels de

17. En effet, tout contenu présenté pour la première fois à l'utilisateur est non classé. Parmi eux, se trouvent les contenus rapatriés par le système à la demande de l'utilisateur, et leur nombre peut être réellement conséquent suivant la tâche.

classe uniquement et agissant sur les contenus $\overline{AC}$ (sur la figure 2.8, ces similarités sont les arêtes noires), et une **intra-classe locale** propre à chaque classe, définie par ses contenus ancrés et classés (AC) et n'affectant que ses éléments libres et classés ($\overline{A}C$) (sur la figure 2.8, les arêtes ont la couleur de la classe).

	Ancré *A*	**Libre** *A*
Classé *C*	- Modifie tout sauf la position des *AC*	- Modifie la position des *A* de la même classe - Est modifié par la mise à jour des *AC* de la même classe
Non Classé *C*		- Est modifié par la mise à jour de n'importe quel *AC*

← Organisation de l'espace de travail

↑ Organisation des données

FIGURE 2.9 – Influence de la manipulation d'un contenu

Le tableau 2.9 récapitule l'influence de la manipulation d'un contenu par l'utilisateur en regard de sa propriété (A, $\overline{A}$, C ou $\overline{C}$).

Voici quelques remarques sur les conséquences de tels choix :

- *Concernant l'attribution d'une classe à un contenu :*

tout contenu $\overline{AC}$ auquel on affecte une classe ou tout contenu $\overline{A}C$ dont on change la classe perd automatiquement son ancrage.

- *Concernant les contenus non ancrés :*

la notion de contenu non ancré telle que nous la proposons met l'accent sur le point de vue « identification » de la tâche organisationnelle (voir figure 2.10).

Si l'utilisateur veut intégrer un contenu $\overline{AC}$ dans une classe préexistante, la réponse du système peut être naturellement interprétée comme une aide portée sur le choix de la classe : le contenu est placé automatiquement plus ou moins loin des centroïde des classes, le système propose donc de l'intégrer dans la classe qui attire le plus ce contenu (voir la figure 2.10 b.1).

De la même manière, si un contenu $\overline{A}C$ (resp. $\overline{AC}$) doit être identifié à un autre élément, le système propose l'élément de la même classe (resp. l'élément non classé) qui attire le plus ce contenu (voir la figure 2.10 b.2 pour un contenu classé).

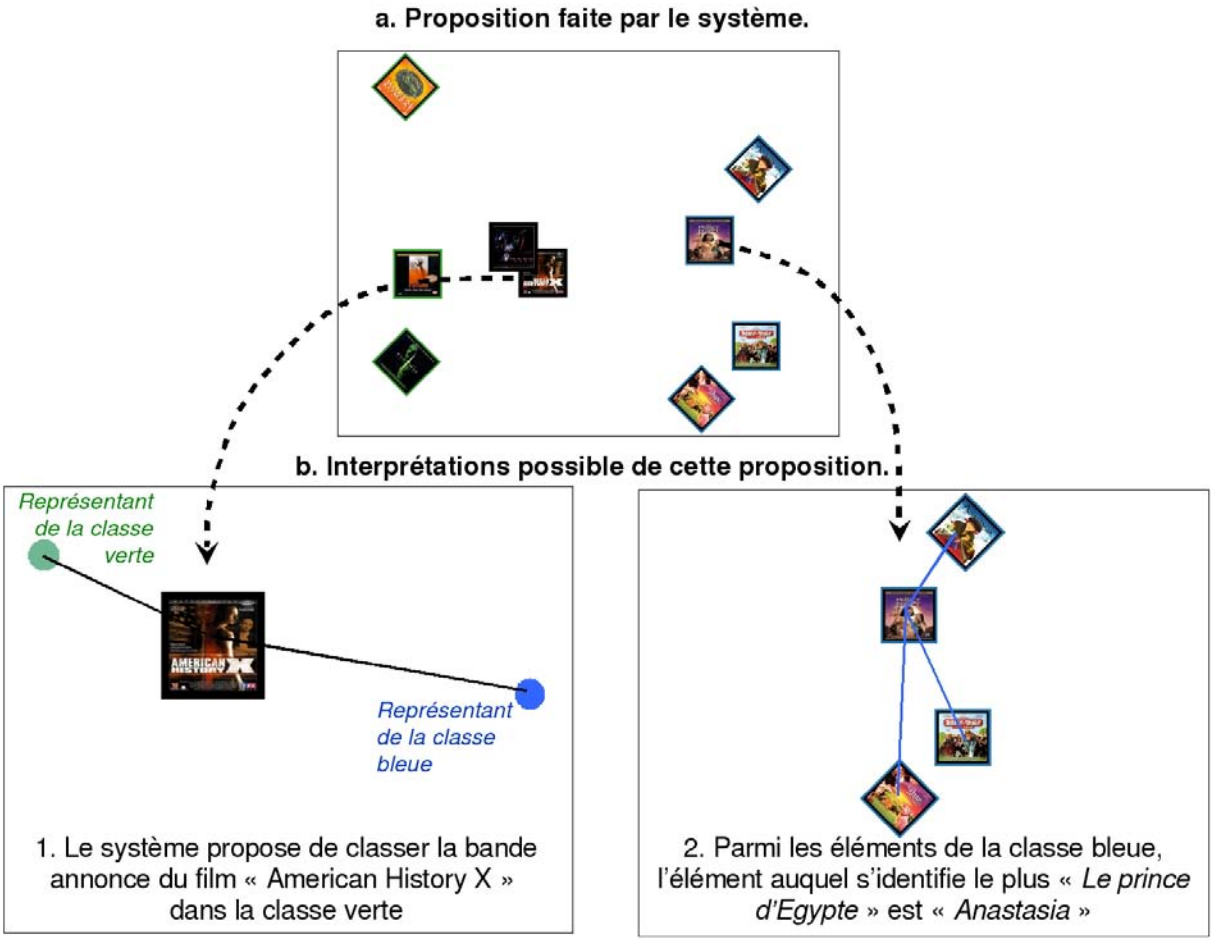


FIGURE 2.10 – Interprétation de la réponse système d'un point de vue « identification »

2.3.2.6 Symbolique visuelle du représentant

Devons-nous matérialiser le représentant d'une classe dans la GUI ? C'est un choix critique qui s'est posé à nous, car laisser à l'utilisateur l'opportunité d'interagir directement avec une entité virtuelle peut amener à renforcer le caractère intangible des données manipulées, alors que nous souhaitons le contraire.

Suite à cette réflexion, nous avons choisi de faire coexister contenus et représentants, mais de jouer sur leur visibilité pour éviter toute confusion :

1) Le représentant est invisible et les contenus de sa classe sont visibles.

Pour illustrer le fait qu'un élément classé est plus ou moins représentatif de sa classe (en d'autres termes, dont le vecteur de description est plus ou moins similaire à celui du représentant virtuel de sa classe), nous jouons sur l'intensité plus ou moins forte de la couleur assimilée à la classe (voir figure 2.11).

Cet artifice visuel nous permet de mettre en avant le point de vue « caractérisation » (cf. section 2.2.2.3) d'une tâche organisationnelle, sans surcharger ni le modèle structurel du graphe

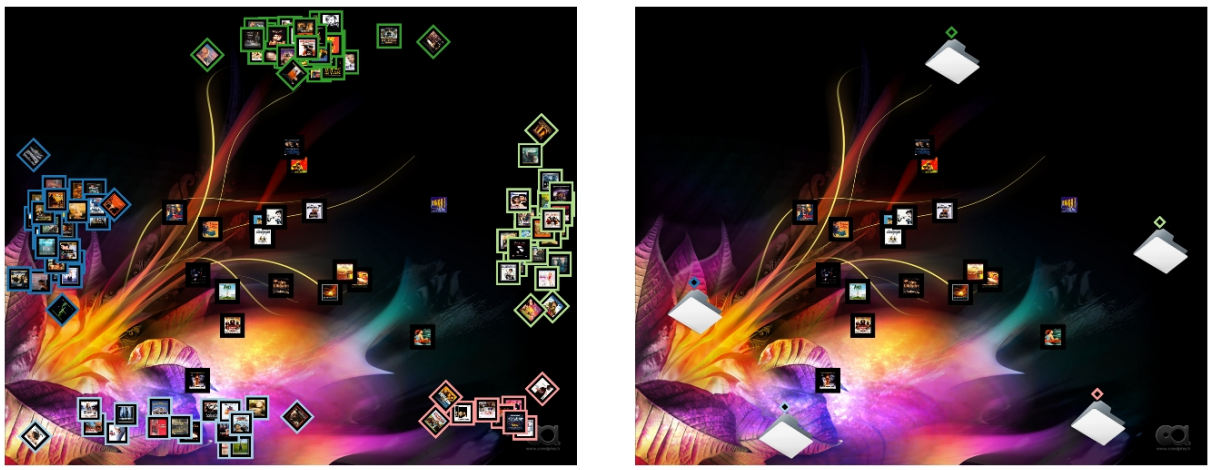
FIGURE 2.11 – Visualisation de la proximité avec le représentant de classe

sous-jacent, ni l'interface.

2) Le représentant est visible et les contenus de sa classe sont invisibles.

Cela peut être vu comme le fait de ranger tous les contenus d'une même classe dans un dossier étiqueté. La symbolique du dossier reste cohérente avec la définition du représentant sur laquelle nous fondons notre formalisme.

Il est alors possible de déployer ou de ranger les contenus, si l'on souhaite ou non s'attarder sur ceux-ci. La figure 2.12 est une illustration de l'utilisation du représentant dans ce contexte.



a. Mode « déployé » b. Mode « rangé »

FIGURE 2.12 – Utilisation du représentant comme dossier de classement

2.3.2.7 La multi-granularité

Dans notre formalisme, la navigation d'une granularité à une autre se fait par la création de classes : une classe est un regroupement de contenus, ce qui définit de fait un contenu d'une granularité supérieure.

La hiérarchie entre ces grains peut être pré-établie (pour l'exploration d'une base documentaire comme celle présentée sur la figure 1.2 par exemple) ou déterminée « à la volée » par l'utilisateur en constituant différentes classes d'un même niveau.

En créant des représentants de classes, nous permettons la mise en relation entre éléments de niveaux différents. Il serait même tout à fait possible avec notre formalisme de monter dans la hiérarchie en créant des représentants de représentants de représentants... La seule limite étant le nombre de contenus présents dans la base.

Nous traiterons, dans cette étude, de l'exploration multi-grains sur seulement deux niveaux hiérarchiques consécutifs : pour une base documentaire donnée, nous supposons que l'ensemble $\{A(n)_j\}_{j=1,\dots,M}$ de ses contenus sont d'une même granularité de niveau n, et proposons d'assister un utilisateur dans l'organisation de contenus de niveaux n (les contenus) et $n+1$ (les représentants de classes).

Nous considérons notre choix du représentant comme une base de réflexion sur l'exploration multi-grains et testerons son comportement dans l'expérience 4.5.5.

2.4 Conclusion

Dans ce chapitre, nous avons présenté ce que nous entendions par organisation de contenus et nous avons étudié de quelle manière il était possible d'utiliser un système semi-automatisé pour aider un humain dans cette tâche.

Nous avons mis en avant les liens qui existent entre cette notion et celle de similarité. Nous pensons qu'il s'agit d'un bon outil pour construire notre système, si toutefois l'utilisateur est orienté dans une direction favorable à son utilisation.

C'est dans cet esprit que nous avons établi un formalisme favorisant le lien entre ces deux univers. Il nous fournit, en exploitant la notion d'identification, la matière nécessaire à une potentielle interprétation de la tâche organisationnelle. Le chapitre suivant sera consacré à la mécanique de cette interprétation.

Chapitre 3

Apprentissage d'un modèle numérique de similarité par régression univariée

Sommaire

3.1 Introduction

Nous avons défini dans les chapitres précédents les données que nous souhaitons traiter et la manière dont nous permettons à l'utilisateur d'interagir avec elles. Il s'agit maintenant d'interpréter ces interactions et de construire la mécanique du système automatisé sous-jacent. En d'autres termes, il nous faut modéliser le problème posé par l'utilisateur afin que la machine puisse le résoudre.

Huet, Jolivet et Messéan [HJM92] définissent un modèle, dans le contexte des statistiques appliquées, comme la représentation d'une réalité de nature aléatoire. Cette représentation se fonde sur des lois (ou des éléments de loi) qui gouvernent cette réalité et qui sont déductibles grâce à la confrontation de l'observation et du modèle considéré.

Dans la vision que nous avons de l'organisation d'une base documentaire, une réalité correspond à une tâche organisationnelle suggérée par l'utilisateur. Notre démarche, inscrite dans la lignée des systèmes d'apprentissage semi-supervisé, est d'expliquer autant que possible cette réalité grâce à des outils statistiques.

Ce chapitre est une description du modèle mathématique que nous avons choisi d'utiliser et la manière selon laquelle nous avons procédé pour qu'il puisse exprimer une notion de similarité.

3.2 Définition d'un modèle numérique de similarité

Pour comparer des entités, il est possible d'évaluer leurs ressemblances ou leurs dissemblances à l'aide d'un modèle de similarités, formellement défini de la sorte :

Soit Ω un univers. Un **modèle de similarité** est un quadruplet (L_D, L_S, χ, f) défini sur Ω, tel que :

- L_D est le langage de description des données ;
- L_S est le langage de description des similarités ;
- χ est l'ensemble de connaissances que nous possédons sur l'univers Ω ;
- f est la fonction de similarité telle que :

$$f : L_D \times L_D \rightarrow L_S$$

Dans notre étude, Ω est l'ensemble des contenus que nous souhaitons organiser.

Les langages L_D et L_S peuvent être de différentes natures. Concernant L_D, nous pouvons manipuler des vecteurs, des graphes, des relations d'ordres, etc. Les langages de similarité, quant à eux, se scindent en deux catégories [RI92] : les similarités symboliques qui « qualifient » les ressemblances et les similarités numériques qui les « quantifient » sous la forme d'une valeur dans $\mathbb{R}$.

Nous pouvons relever, comme différents modèles de similarité, les modèles conceptuels, comportementaux, sémantiques, graphiques, événementiels, numériques (cette liste n'est pas

exhaustive).

Nous nous focalisons sur l'étude de similarités numériques, aux vues des caractéristiques que nous utilisons (chapitre 1, section 1.3) et des informations sur la similarité avec lesquelles nous allons travailler (notion de distances entre les contenus). Pour ce faire, nous définissons ce que nous entendons par modèle numérique de similarité :

Soit Ω un univers. Un **modèle numérique de similarité** est un quadruplet (L_D, L_S, χ, f) défini sur Ω, tel que :
- L_D est restreint à l'ensemble des données descriptibles par des valeurs numériques (scalaires, vecteurs, matrices) ;
- L_S est une valeur réelle comprise entre 0 et 1 ;
- χ est un ensemble de données statistiques sur Ω ;
- $f : L_D \times L_D \rightarrow [0 \dots 1]$;

Le terme générique pour désigner f est **mesure de similarité**. C'est ce terme que nous employons lorsque nous parlons de similarité dans le sens général de la « ressemblance » (comme dans le propos de la section 2.2.2.2, sur la différence entre identification et classification).

L'intérêt d'une mesure de similarité se trouve soit dans la quantification de la ressemblance, soit dans la quantification de la dissemblance. Lorsque nous aurons besoin d'être plus rigoureux à ce sujet, nous emploierons le termes de **fonction** de similarité ou de dissimilarité, dont voici les définitions :

Soit f une mesure de similarité provenant d'un modèle numérique de similarité. Pour tout $(x_i, x_j) \in L_D \times L_D$:

- f est une **fonction de similarité** si :
  1) $f(x_i, x_j) \geq 0$ (positive) 2) $f(x_i, x_j) = f(x_j, x_i)$ (symétrique) 3) $f(x_i, x_i) \geq f(x_j, x_i)$

- f est une **fonction de dissimilarité** si :
  1) $f(x_i, x_j) \geq 0$ (positive) 2) $f(x_i, x_j) = f(x_j, x_i)$ (symétrique) 3) $f(x_i, x_i) = 0$

- f est une **fonction de dissimilarité propre** si :
  1) f est une fonction de dissimilarité 2) $(f(x_i, x_j) = 0) \Rightarrow (x_i = x_j)$

À titre d'exemple, nous voyons qu'une distance est une fonction de dissimilarité propre qui vérifie l'inégalité triangulaire, dont nous rappelons la formule :

$$\forall x_i, x_j, x_k \in L_D,\ f(x_i, x_k) \leq f(x_i, x_j) + f(x_j, x_k) \tag{3.1}$$

3.3 Réflexions sur la construction d'un modèle pour l'organisation

Nous avons choisi d'interpréter une tâche organisationnelle par un modèle numérique de similarité. Il nous faut maintenant définir le cœur de ce modèle, à savoir sa mesure de similarité. Nous expliquons dans cette section quel chemin nous avons parcouru, à travers l'exploration du domaine de la création de distances, pour en arriver à nous orienter vers les méthodes de prédiction statistique, et plus particulièrement vers la régression univariée.

3.3.1 La piste de l'apprentissage de distances

Nous nous sommes dans un premier temps orientés vers l'apprentissage de distances. C'est à travers l'exploration de l'existant que nous avons pris position sur la manière dont nous souhaitions exploiter l'espace de description.

3.3.1.1 Introduction

Il existe de nombreuses méthodes pour concevoir des mesures de similarité. Toutes ne sont pas présentées comme telles. La littérature (3.3.1.2) a coutume de citer des méthodes de projection telles que l'Analyse en Composantes Principales (PCA) ou le *Multidimensional Scaling* (MDS) (toutes deux présentées dans la section 3.4.6.2) comme références de tâches connexes à la création de mesures, ce que nous trouvons peu pertinent :

- Les algorithmes tels que le PCA modifient l'espace de description pour concentrer l'information sur un nombre restreint d'axes. Nous avons besoin de focaliser sur des corrélations dans cet espace, certes, mais relatives aux similarités exprimées par un tiers, ce que ne nous permet pas cette classe d'algorithmes.
- Les algorithmes tels que le MDS utilisent directement la notion de similarité pour positionner des objets dans un espace souvent en deux ou trois dimensions (plus facilement visualisables). L'information de similarité entre les éléments est déjà établie, alors que nous souhaitons créer la relation qui permet de fournir cette information. En d'autres termes, il n'y a aucun apprentissage dans cette classe d'algorithmes.

Nous présentons dans cette section un survol de différents outils que nous avons recensés pour apprendre une distance numérique. Ces techniques ne sont pas toutes incompatibles, et peuvent également être utilisées pour l'apprentissage d'autres modèles de similarité, comme en similarité symbolique. Cette section nous permettra également de porter un premier regard sur les méthodes à noyau, que nous serons amenés à détailler par la suite.

3.3.1.2 Présentation

Beaucoup de travaux relativement récents ont été menés sur l'apprentissage de distances, notamment pour aider aux tâches de recherche d'information ou de classification non-supervisées et semi-supervisées.

Le principe consiste à créer une distance euclidienne paramétrique qui se définit de la sorte : pour $(x_i, x_j) \in \mathbb{R}^n \times \mathbb{R}^n$ et Λ une matrice semi-définie positive de dimension $n \times n$,

$$\delta_\Lambda(x_i, x_j) = \|(x_i - x_j)\|_\Lambda = \sqrt{(x_i - x_j)^T \Lambda (x_i - x_j)} \tag{3.2}$$

Quelques remarques :

- si Λ est la matrice identité de $\mathbb{R}^n$, nous retombons sur une distance euclidienne ;
- si Λ est diagonale, alors ses coefficients sont appelés des poids, et δ_Λ est appelée distance euclidienne pondérée ;
- si Λ est l'inverse de la matrice de covariance de (x_i, x_j), alors la distance est dite de Mahalanobis.

De manière générale, Λ caractérise la famille des distances pondérées dites de Mahalanobis, paramétrées sur $\mathbb{R}^n$. Cette matrice porte avec elle les informations sur corrélations et poids des variables.

De nombreux travaux portent sur la transformation de l'équation 3.2 en un problème d'optimisation convexe. En introduisant Φ, une fonction non linéaire de projection, 3.2 devient :

$$\delta_\Lambda(x_i, x_j) = \sqrt{(\Phi(x_i) - \Phi(x_j))^T \Lambda (\Phi(x_i) - \Phi(x_j))} \tag{3.3}$$

La résolution de 3.3 se fait alors par projection dans un espace appelé **espace de redescription** (ou *feature space*) à l'aide d'une fonction noyau (ou *Kernel*). Cette technique, appelée le *Kernel Trick*, sera expliquée en détails dans la suite du manuscrit (section 3.4.5.2).

Le but de l'apprentissage de distances paramétriques est de trouver une estimation $\tilde{\Lambda}$ de Λ qui réponde au mieux aux besoins de la tâche à accomplir. Notons que nombre de ces travaux utilisent la notion de similarité sous forme d'appartenance ou non à une classe.

Un travail de référence sur l'aspect linéaire du problème est celui de [XNJR03] qui utilise une approche qualitative de la similarité, en considérant les ensembles Sim des éléments similaires et $Dissim$ d'éléments dissimilaires ($Dissim = \Omega \setminus Sim$), pour en venir à résoudre le problème d'optimisation suivant :

$$\tilde{\Lambda} = \arg\min_{\Lambda} \sum_{(x_i,x_j)\in Sim} \|x_i - x_j\|_\Lambda^2 \text{ , tel que } \sum_{(x_i,x_j)\in Dissim} \|x_i - x_j\|_\Lambda \geq 1 \tag{3.4}$$

avec Λ semi-définie positive.

Ces travaux ont été étendus aux cas non linéaires par [KT03], qui optent pour la création d'un noyau « idéal » dédié à l'expression de telles similarités.

Une autre approche intéressante est celle de [XNZ08], qui reprend les travaux de [XNJR03]. La matrice Λ est décomposée en un produit de matrices de pondérations WW^T. La matrice optimale $\Lambda = (W^*)(W^*)^T$ est obtenue en trouvant W^* par la résolution de la fonction objectif

suivante :

$$W^* = \underset{WW^T=I}{\arg\max} \frac{tr(W^T Cov_{\{Sim\}} W)}{tr(W^T Cov_{\{Dissim\}} W)} \tag{3.5}$$

avec $Cov_{\{E\}}$ la matrice de covariance des paires de points de l'ensemble E, et $WW^T = I$ une contrainte d'orthogonalité qui permet d'éviter les solutions dégénérées.

[SJ04] portent leur intérêt sur la similarité relative entre les échantillons d'apprentissage en exploitant l'assertion « A est plus proche de B que de C » pour édifier leur distance.

L'approche dite *Maximally Collapsing Metric Learning* [GR06] repose sur l'intuition géométrique que tous les éléments appartenant à une même classe (et donc très fortement similaires) pourraient être projetés sur un même élément (une sorte de représentant de classe) dans l'espace de redescription.

Les travaux de [BHHSW06] ont conduit à la création de l'algorithme *Relevant Component Analysis* (RCA), qui consiste à identifier puis à diminuer la variabilité globale indésirable entre les données. Cette méthode modifie l'espace de description par une transformation linéaire globale qui assigne des poids forts (resp. faibles) aux dimensions « pertinentes » (resp. sans importance). Ces dimensions pertinentes sont estimées à partir d'un ensemble réduit de points (appelés *chunklets*) qui sont supposés appartenir à une même classe.

Cette méthode a été améliorée par [HLLM06] avec la création de l'algorithme Discriminative Component Analysis (DCA) et son extension aux cas non linéaires (Kernel DCA), qui optimise RCA en maximisant la variance totale des données entre les *chunklets* discriminants tout en minimisant la variance totale des données sur les mêmes *chunklets*.

3.3.1.3 Une fausse piste

Toutes ces techniques ont retenu notre attention car elles pourraient naturellement être intégrées comme mesure dans un modèle numérique de similarité. Cependant, il s'avère qu'elles ne sont pas en accord avec notre philosophie : ces distances modifient la nature quantitative de la relation de similarité dans l'espace de description, pour mieux la qualifier dans l'espace de représentation. Au lieu de chercher à concilier la similarité entre les deux espaces, elle la force.

3.3.2 Une solution : la régression univariée

Il faut que notre mesure de similarité s'adapte aux contraintes imposées par l'utilisateur. Son jugement ne doit pas être remis en question par la procédure d'apprentissage, comme pouvaient le faire les techniques vues précédemment. C'est dans les méthodes de prédiction statistique que nous avons trouvé une solution.

Nous interprétons une relation de similarité numérique entre les contenus comme un phénomène, au sens probabiliste du terme, dont il est possible d'estimer le comportement à partir d'un jeu d'observations. Cette section présente la technique statistique que nous avons retenue, à savoir la régression univariée.

3.3.2.1 Introduction

Revenons sur l'exemple de scénario d'usage présenté dans le chapitre d'introduction générale (figure 2, section 2.2.1).

Soient deux contenus A et B quelconques, $x = (x_A, x_B)$ l'observation correspondant aux valeurs d'angles de teinte extraites de leur couleur dominante, et y la distance euclidienne calculée sur l'interface entre les objets informationnels correspondants. La relation qui illustre cette tâche est la fonction suivante (voir figure 3.1) :

$$f(x) = sin|x_A - x_B| \tag{3.6}$$

En effet, le sinus de l'angle α pris entre les deux valeurs d'angles correspondant aux teintes des contenus A et B *explique* l'écart résidant entre les deux contenus.

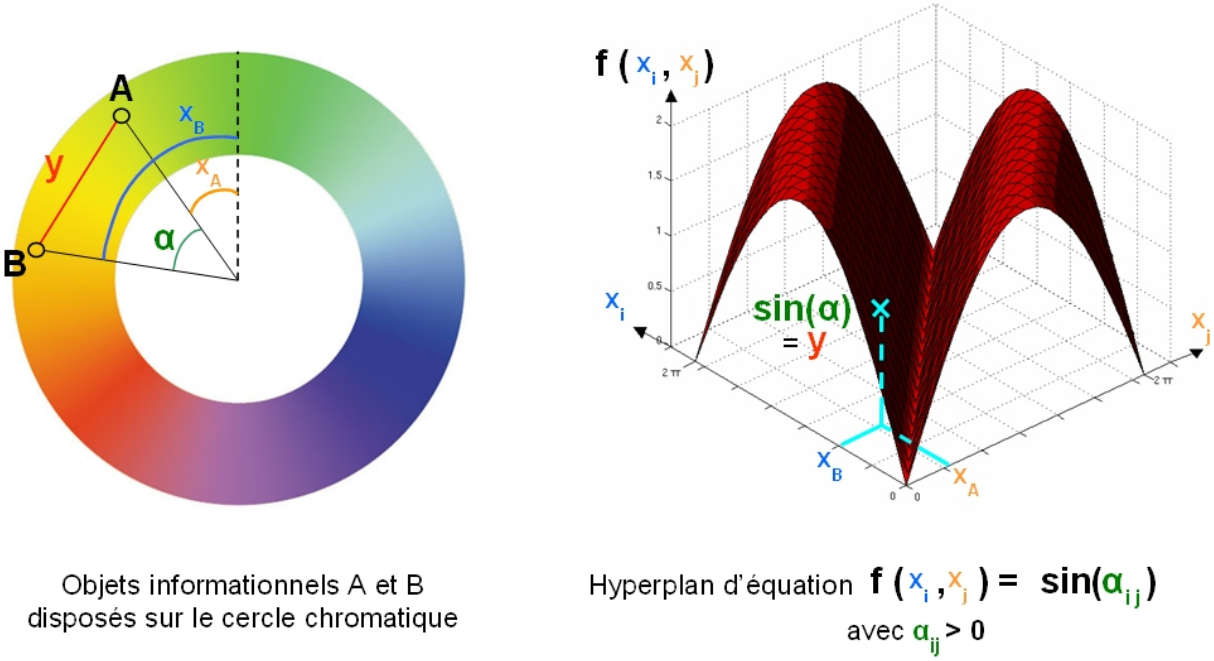


FIGURE 3.1 – Illustration du scénario (2)

Afin de modéliser une telle organisation, nous avons choisi comme stratégie d'approcher la fonction non linéaire f de manière efficace par un modèle mathématique qui mettrait en relation un vecteur x avec une unique variable y. Nous avons ici une définition simpliste d'une régression.

3.3.2.2 Notations et définitions spécifiques

Définir une régression revient à créer une relation qui s'appuie sur un ensemble de valeurs observées à partir d'un ensemble de **variables prédictives** (également appelées variables **exogènes**) pour estimer des **variables à prédire** (aussi connues sous le nom de variables **endogènes**).

En nous plaçant dans le contexte de la modélisation par régression, nous pouvons reprendre les notations de l'introduction générale (section 2.4) : les « valeurs descriptives » deviennent des variables exogènes, et les « distances utilisateur » deviennent des variables endogènes.

Dans la suite nous considérerons le couple (x, y) tel que :
- $y \in \mathbb{R}$ soit la réalisation de la variable aléatoire Y ;
- $x \in \mathbb{R}^D$ soit la réalisation de la variable aléatoire X.

Nous définissons le vecteur Y de la façon suivante :

$$Y = f_\theta(X) + E \tag{3.7}$$

avec $\theta = \{\theta_1, \ldots, \theta_p\} \in \mathbb{R}^p$ le vecteur de paramètres de f et E une variable aléatoire.

Nous supposons E d'espérance nulle et de variance σ^2. En d'autres termes, E est une variable aléatoire de loi $\mathcal{N}(0, \sigma^2)$. Les réalisations e_i de la variable E sont appelées les aléas.

Nous définissons un modèle de régression par la réunion de la fonction f_θ et de la loi probabiliste du vecteur E. f_θ est appelée l'équation de régression et E représente l'écart résiduel à la tendance générale exprimée par le modèle de régression.

3.3.2.3 Pourquoi univariée ?

Utilisée comme fonction dans un modèle numérique de similarité, une régression peut être univariée ou multivariée.

Une fonction de régression est **univariée** si une variable exogène ne peut expliquer qu'une valeur de variable endogène :

$$y = f_\theta(x) + e \tag{3.8}$$

Contrairement aux fonctions univariées, les fonctions **multivariées** considèrent que pour une même variable exogène, le lien entre deux objets est exprimé de J manières différentes, par J composantes d'une variables endogènes :

$$\begin{cases} y^1 = f_\theta^1(x) + e^1 \\ \vdots \\ y^J = f_\theta^J(x) + e^J \end{cases} \tag{3.9}$$

Dans le contexte de notre étude, l'emploi d'une fonction multivariée est envisageable dans deux cas qui ne soutiennent pas nos hypothèses de travail :

1) J réalisations y^j de Y^j, $j \in \{1, \ldots, J\}$, sont comparées à un même instant t. Chaque y^j provient donc d'un utilisateur différent. L'expression de la similarité à un instant donné par J variables endogènes s'interprète comme la mise en correspondance de l'opinion de J utilisateurs sur l'organisation des mêmes données. Toutefois, dans notre étude une variable endogène est obtenue par une demande explicite d'apprentissage faite par un unique utilisateur.

2) chaque réalisation y^j de Y^j est prise à un instant t_j, $j \in \{1, \ldots, J\}$. Dans ce cas, les y^j représentent les J propositions d'organisations faites par un même utilisateur.

Exprimer la notion de similarité par J variables endogènes à différents instants s'interprète comme la mise en correspondance de différentes opinions d'un utilisateur sur l'organisation d'un même jeu de données. C'est en soi ce que nous souhaitons faire grâce à notre système : que d'un instant t_j à un instant t_{j+1}, l'utilisateur remette en cause la disposition des contenus pour améliorer la mesure de similarité apprise.

Ce simple processus est toutefois trop restrictif, dans la mesure où la cardinalité du jeu de données, elle, n'est pas remise en cause : nous avons pris le parti qu'une reconsidération de la tâche organisationnelle à chaque nouvel instant puisse se faire en ajoutant ou en enlevant des contenus du corpus d'apprentissage, ce que ne permet pas de faire ce cas précis.

Pour ces raisons, nous avons choisi de travailler sur l'élaboration d'un modèle de similarités reposant sur une fonction de régression univariée.

3.3.2.4 Remarque : prédiction ou explication ?

Il convient de prendre un peu de recul afin de noter quelques considérations de langage, abordées par Rouanet et Lebaron [RH02] dans une étude portant sur la régression en mathématiques et en sciences humaines.

La régression est de nature prédictive. En ce sens, on peut se servir d'une régression uniquement pour un usage **prédictif**. Dans ce cas, le nombre de variables prédictives n'est pas limitatif et les liens entre les variables sont peu gênants. D'un point de vue explicatif, les variables prédictives sont appelées « variables explicatives », et les variables à prédire sont des « variables à expliquer ».

La prédiction est une problématique bijective alors que l'explication ne l'est pas. Prenons un exemple :

- la longueur d'une barre de métal peut être prédite à partir d'une température, on peut de ce fait régresser la longueur sur la température. Il est également possible de partir de la longueur de la barre de prédire la température (c'est le principe du thermomètre) et de régresser la température sur la longueur de la barre ;
- du point de vue de la théorie physique de la dilatation, la température explique la longueur de la barre, mais la réciproque est fausse.

Il devient intéressant pour nous d'utiliser la régression dans un contexte **explicatif**. C'est cet objectif que nous souhaitons atteindre à travers notre projet : expliquer le « pourquoi » d'une organisation est bien plus gratifiant que considérer celle-ci uniquement comme la description d'une réalisation probabiliste d'agencement de données. C'est également parce que nous avions cela en tête que nous avons choisi la régression comme modèle d'apprentissage.

Mais de telles considérations nous écartent du contexte expérimental qui nous contraint à évoluer dans un environnement statistique. Nous ferons toutefois sciemment l'amalgame entre ces deux univers, mêlant réalités et idéaux, et citerons J.P. Fénelon [Fen82] pour clore le sujet : *« dire descriptif, c'est péjoratif ; dire explicatif, c'est abusif »*.

3.4 Méthode pour l'utilisation de la régression comme modèle numérique de similarité

Nous présentons dans cette section une méthode, inspirée des différents domaines que sont l'apprentissage supervisé et l'estimation fonctionnelle, permettant d'utiliser la régression comme un modèle numérique de similarité.

Nous avons orienté cette analyse d'un point de vue pragmatique, en centrant le débat autour de la notion de similarité, et en minimisant les points théoriques liés à la régression qui porteraient la réflexion sur un terrain trop général.

3.4.1 Présentation de la méthode

Cette méthode se décline en plusieurs points :

---

1. **La choix des variables numériques** endogènes et exogènes. C'est une première étape qui pose la réflexion sur :
   - **a. Le type de variables numériques** qui va contraindre le mode d'expression de la similarité ;
   - **b. Le prétraitement des données** pour poser une base d'appréciation commune entre les différentes observations.
2. **Le choix du modèle de régression.** Quel cœur donner au moteur d'apprentissage ? Les questions à se poser sont les suivantes :
   - **a. La fonction de régression choisie est-elle linéaire ou non linéaire ?** La réponse dépend du phénomène à expliquer ;
   - **b. Quelle est la nature de la fonction ?** La régression est une discipline extrêmement étudiée et de nombreuses fonctions existent. Le choix de celle-ci détermine le contexte du modèle, constitué de l'**espace des hypothèses**, de la **fonction de perte** et des **termes de régulation**.
3. **Le choix de l'algorithme de réduction de la dimensionnalité,** qui est une étape d'optimisation importante afin de rendre le modèle à la fois robuste et parcimonieux.

4. La mise en œuvre, qui comprend :

a. Le type de fusion qui détermine comment mettre en relation des observations provenant de phénomènes hétérogènes, comme peuvent l'être les différentes modalités d'un contenu audiovisuel ;

b. Le choix des méthodes d'évaluation pour valider le modèle et ses différentes briques.

Le schéma suivant (figure 3.2) illustre les différentes étapes de la méthode.

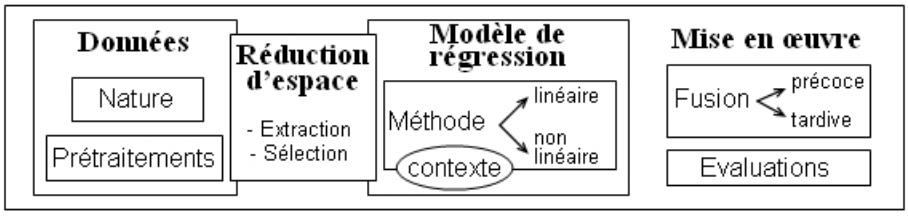


FIGURE 3.2 – Schéma général de la méthode

La suite de cette section présente en détail les différents points de cette méthode.

3.4.2 Choix des variables

Cette section porte la réflexion sur la nature de la matière première : les variables endogènes et exogènes (cf. figure 3.3). Avant d'être mises en relation, au travers de leur intégration au modèle de régression, elles doivent subir des transformations, en fonction de leur provenance.

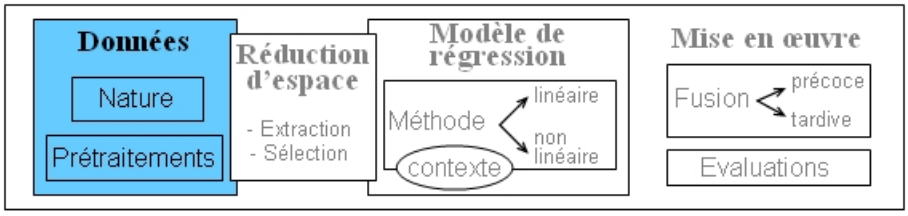


FIGURE 3.3 – Positionnement du choix des variables dans le schéma général

3.4.2.1 Type de variables

Nous posons :
- N variables endogènes $(y_1, \ldots, y_N)$ comme les N réalisations de la variable aléatoire Y ;
- N variables exogènes $(x_1, \ldots, x_N)$ correspondantes comme les N réalisations de la variable aléatoire X.
- $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ l'ensemble d'apprentissage.

Elles sont supposées indépendantes et identiquement distribuées.

Ces variables, qu'elles soient endogènes ou exogènes, posent un double problème. Il faut d'abord choisir quel type d'information fournir au système, pour qu'ensuite il puisse faire le tri dans celles-ci afin de ne conserver que les plus significatives.

Avant de parler du tri (qui sera discuté dans l'étape du choix de l'algorithme de réduction de la dimensionnalité dans la section 3.4.6), attardons nous sur la nature de nos valeurs descriptives.

Le langage L_D d'expression des variables exogènes se restreint à l'ensemble des données descriptibles par des valeurs numériques, que sont les scalaires, les vecteurs et les matrices.

Les variables endogènes, elles, sont le reflet de la notion de similarité. Leur énonciation est relative au choix fait entre une fonction de similarité et un fonction de dissimilarité (définies dans la section 3.2). Quel que soit ce choix, ces variables s'expriment au travers d'un réel compris entre 0 et 1.

3.4.2.2 Prétraitement

Une variable exogène provient de l'expression de multiples réalisations, rien ne garanti que ces différentes observations soient toutes homogènes. Dans notre étude, la question de la mise en relation des paramètres audio et vidéo entraîne nécessairement une réflexion sur la création d'une référence commune, pour ne pas biaiser leurs comparaisons. Cette question est considérée dans l'étape de prétraitement des données. Nous nous servirons de l'étude de ces deux médias pour illustrer notre propos.

- L'échantillonnage

Comme nous l'avons vu dans le premier chapitre (section 1.3), l'interprétation d'un contenu audiovisuel par la machine peut se fonder sur l'analyse de son signal. C'est cette approche, dite « bas niveau », que nous avons décidé d'explorer dans cette étude.

L'analyse des descriptions issues des différents médias s'effectue sur un signal numérique discret. Le signal analogique d'origine supposé continu doit donc subir une transformation pour pouvoir être traité. Cette transformation s'appelle l'échantillonnage, ou *sampling*.

En vidéo, le cinéma a fait le choix de restituer des images à une cadence de 24 Hz, soit 24 images par secondes (30 Hz pour le format NTSC), de manière à dépasser la limite perceptive humaine d'une image fixe qui est de l'ordre du dixième de seconde. La majeur partie des descriptions voulant témoigner d'un événement propre à ce média se fondent sur cette **fréquence d'échantillonnage**.

Pour ce qui est de l'audio, la représentation numérique fidèle d'un son ne peut être obtenue qu'en échantillonnant celui-ci au moins au double de sa fréquence. L'oreille humaine peut percevoir les fréquences acoustiques allant en moyenne de 20 Hz jusqu'à 20 kHz [Boi00],

tout échantillonnage numérique doit théoriquement s'effectuer à 40 kHz [18] (un CD audio, par exemple, est échantillonné à 44,1 kHz).

Le respect de cette contrainte n'est toutefois pas nécessaire à l'obtention de paramètres audio représentatifs du signal acoustique, preuve en est des communautés du traitement de la parole et de la musique qui utilisent des descriptions échantillonnées à une fréquence de 16 kHz.

Un autre facteur qui intervient dans la qualité de l'échantillonnage acoustique est la **résolution**. Elle détermine le nombre de valeurs du vecteur qui servira à caractériser l'échantillon. La résolution s'exprime en bits : un codage 8 bits permet d'obtenir 256 valeurs par échantillon, un codage 16 bits en autorise 65.536, etc. Chaque bit supplémentaire améliore le système de 6 dB. En théorie, le signal audio doit se coder sur une résolution de 20 bits pour exploiter l'ensemble des possibilités de l'oreille humaine. En pratique, il est suffisant de coder le signal sur 16 bits (ce qui correspond à la résolution d'un CD audio).

Une fois la fréquence d'échantillonnage et la résolution déterminées, reste éventuellement à établir la taille de la **trame** (chapitre 1, section 1.3.2), unité de traitement spécifique à la description du signal de parole. Cette taille est comprise entre 10 et 40 ms, mais rien ne garantie qu'elle soit la même pour toute description, exemple en est du calcul de l'énergie (section 1.3.2.2) qui se fait sur des trames de 10 ms, alors que celui de la modulation de l'entropie (section 1.3.2.7) s'effectue sur des trames de 16 ms.

• La *normalisation*

La normalisation d'un vecteur de caractéristiques est relative à une valeur de description (une coordonnée) et à un corpus de référence ([TLN+03], [SWS05]). Cette étape permet principalement deux choses :

1. apprécier chacune des descriptions d'un vecteur en les comparant à celles calculées sur l'ensemble du corpus ;
2. mettre les différentes descriptions sur un même pied d'égalité.

Prenons un vecteur $v = (v_1, \ldots, v_n)$ possédant n caractéristiques, et notons $\overline{v} = (\overline{v_1}, \ldots, \overline{v_n})$ son vecteur normalisé. Nous citons ici quelques exemples de fonctions de normalisation couramment utilisés en analyse du signal :

– la normalisation par le maximum (resp. minimum) :

$$\overline{v_i} = \frac{v_i}{max_i} \tag{3.10}$$

avec max_i (respectivement min_i) la valeur maximum (resp. minimum) de la i^e description du corpus de référence. Cette fonction est une mise à l'échelle des vecteurs, de manière relative aux valeurs maximales de chaque description, qui, de fait, sont projetées

18. Suivant le *théorème de Shannon*, il est nécessaire d'échantillonner un signal analogique à une fréquence double de la plus haute fréquence qu'il contient pour pouvoir le reconstruire parfaitement. Cette fréquence d'échantillonnage minimale est appelée *fréquence de Nyquist*

sur 1.

– la fonction que nous nommons $MinMax$:

$$MinMax(v_i) = \overline{v_i} = \frac{v_i - min_i}{max_i - min_i} \tag{3.11}$$

Cette fonction a pour effet de contraindre chacune des coordonnées du vecteur sur l'intervalle [0,1], en projetant la valeur minimum de chaque description sur 0 et la valeur maximum sur 1. Une variante de cette fonction, également très utilisée pour l'élaboration de machines à vecteurs de support, permet de projeter les extrema sur l'intervalle [-1,1].

– la fonction suivante :

$$\overline{v_i} = \frac{v_i - min_i}{\sqrt{Card(v_i)} \times (max_i - min_i)} \tag{3.12}$$

avec $Card(v_i)$ la dimension du vecteur v_i. Cette variante de $MinMax$, utilisée dans le cas où le nombre de descriptions ne seraient pas le même pour tous les vecteurs, permet d'obtenir une même norme moyenne pour tous les éléments du corpus [AQG07].

3.4.3 Le modèle de régression

Maintenant que nous avons défini la nature de notre échantillon d'apprentissage, il nous faut nous attaquer au modèle de régression à proprement parler, modèle que nous présenterons dans le contexte d'expression de la similarité (cf. figure 3.4).

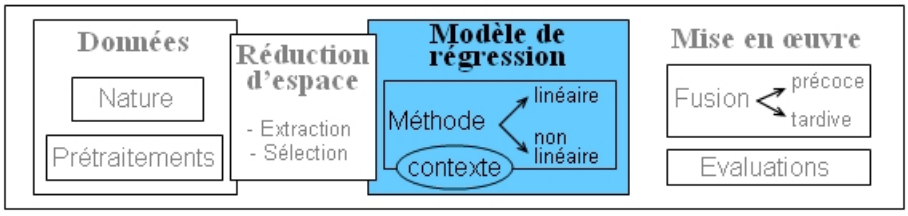


FIGURE 3.4 – Positionnement du modèle de régression dans le schéma général

Ce modèle peut être vu comme un ensemble constitué d'une fonction de régression f et d'un contexte χ. Ce dernier se compose d'un espace d'hypothèses, d'une fonction de perte et d'un terme de régulation. Toutes ces notions seront définies dans cette section.

3.4.3.1 Définitions

La fonction de régression f qui nous servira de mesure de similarité se trouve dans un espace fonctionnel particulier, appelé **espace des hypothèses**, et noté $\mathcal{H}$.

Le modèle de régression est chargée de trouver la fonction f dans l'espace $\mathcal{H}$. Ces fonctions sont dites **linéaires** si leur dépendance aux paramètres θ de f est linéaire. Dans le cas contraire, elles sont **non linéaires**. Quelques unes de ces fonctions seront décrites respectivement dans les

sections 3.4.4 et 3.4.5. L'espace $\mathcal{H}$ dépend directement du choix de la fonction utilisée.

Un modèle de régression qui utilise une fonction linéaire (respectivement non linéaire) est appelé **modèle de régression linéaire** (respectivement **modèle de régression non linéaire**).

La fonction f doit minimiser l'écart entre la valeur de similarité réelle (le vecteur y) et la valeur estimée (le vecteur $f(x)$). Cet écart est appelé le risque réel associé à la fonction f et s'écrit :

$$R[f] = \int_{\mathbb{R}^D \times \mathbb{R}} L(y, f(x)) dp(x, y) \tag{3.13}$$

La fonction L, appelée **fonction de perte** (section 3.4.3.3), définit ce risque.

Bien souvent, il n'est pas possible de calculer le risque réel $R[f]$, la densité $p(x, y)$ étant inconnue. La solution est de suivre l'idée naturelle qui est de sélectionner une fonction f_0 de $\mathcal{H}$ qui décrit du mieux possible les données S. Pour ce faire, le **risque empirique** $R_{emp}[f]$ de f, qui est une estimation de $R[f]$, est utilisé. Il est défini par la moyenne de la fonction de perte sur l'échantillon S :

$$R_{emp}[f] = \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i)) \tag{3.14}$$

Nous cherchons une fonction $f_0 = \arg\min_{f \in \mathcal{H}} R_{emp}[f]$ qui minimise $R_{emp}[f]$. Il est possible que $\mathcal{H}$ la contienne, mais il se peut qu'elle présente de mauvaises capacités de généralisation. Pour éviter cela et maîtriser l'expressivité de l'espace d'hypothèses, un terme de contrôle est ajouté au risque empirique. On obtient ainsi le **risque empirique régularisé** :

$$R_{reg}[f] = R_{emp}[f] + \upsilon Reg(f) \tag{3.15}$$

avec υ une constante dite de régulation, qui permet de contrôler l'apport du risque empirique et du terme de régulation $Reg(f)$ [GBD92].

Nous reviendrons sur cette dernière équation dans la section 3.4.7.2 qui aborde la question du sur-apprentissage.

3.4.3.2 Espace des hypothèses

La nature d'un espace d'hypothèse varie selon les fonctions de régression choisies. Voici les différentes expressions possibles de $\mathcal{H}$ correspondant aux fonctions auxquelles nous ferons allusion dans ce manuscrit :

Si $\mathcal{H}$ est engendré par une famille génératrice $\{g_i(x)\}_{i=1,\dots,m}$, alors il peut s'écrire sous la forme suivante :

$$\mathcal{H} = \left\{ f : f(x) = \sum_{i=1}^{m} v_i g_i(x), \text{tel que } v = (v_1, \dots, v_m) \text{ et } \forall\ i,\ v_i \in \mathbb{R} \right\} \tag{3.16}$$

Dans le cas où $\mathcal{H}$ est un espace préhilbertien et qu'il existe une fonction dite noyau qui satisfait les conditions de Mercer (voir section 3.4.5.2), alors f peut être décrite en fonction de ce noyau. Pour ces espaces, il est possible de définir une fonction g symétrique définie positive, telle que :

$$\forall f \in \mathcal{H}, f(x) = \langle f, g \rangle_{\mathcal{H}} \tag{3.17}$$

L'espace d'hypothèse s'écrit dans ce cas :

$$\mathcal{H} = \left\{ f : f(x) = \sum_{i=1}^{n} v_i g(x, x_i), \text{tel que } v = (v_1, \ldots, v_n) \text{ et } \forall\ i,\ v_i \in \mathbb{R} \right\} \tag{3.18}$$

Pour finir, si $\mathcal{H}$ est engendré par une famille génératrice $\{g_i(x, w_i)\}_{i=1,\ldots m}$ et que ces fonctions sont dépendantes d'un paramètre $w = \{w_1, \ldots, w_m\}$ issu des données, alors $\mathcal{H}$ peut s'écrire :

$$\mathcal{H} = \left\{ f : f(x) = \sum_{i=1}^{n} v_i g_i(x, w_i), \text{tel que } v = (v_1, \ldots, v_n) \text{et } \forall\ i,\ v_i \in \mathbb{R} \right\} \tag{3.19}$$

Cette variante de 3.16 permet d'introduire de la flexibilité dans la modélisation, incarnée par le paramètre w. Cet artefact permet, dans le cas où seul une partie de la famille génératrice de base est suffisante à la construction de $\mathcal{H}$, de faire preuve de parcimonie en n'utilisant que ce sous-ensemble.

3.4.3.3 Fonction de perte

Deux types de fonction de perte sont communément utilisés pour la construction de modèles de régression [RGCV05] :

- la fonction L_2 de coût quadratique :

$$L_2(y, f(x)) = (y - f(x))^2 \tag{3.20}$$

- la fonction L_ε dite ε-insensible :

$$L_\varepsilon(y, f(x)) = \|y - f(x)\|_\varepsilon = max(0, |y - f(x)| - \varepsilon) \tag{3.21}$$

3.4.3.4 Terme de régulation

Trois types de termes de régulation sont communément utilisés pour la construction de modèles de régression [RGCV05] :

- le premier repose sur une pénalisation quadratique :

$$Reg(f) = \sum_i v_i^2 \quad \text{ou} \quad Reg(f) = \sum_{i,j} v_i v_j g(x_i, x_j) \tag{3.22}$$

la forme dépendant du choix de l'expression de $\mathcal{H}$. Ces termes pénalisent généralement de fortes valeurs pour les v_i. Ils correspondent à la norme quadratique de $f(x)$ dans $\mathcal{H}$ (soit $\|f\|^2$) ;

- le second est également utilisé pour favoriser la parcimonie de la solution $f(x)$:

$$Reg(f) = \sum_i |v_i| \tag{3.23}$$

- le troisième, appelé norme l_0 par abus de langage, représente le nombre de coefficients v_i non nuls :

$$Reg(f) = card\{v_i \neq 0\} = \|v_i\|_0 \tag{3.24}$$

3.4.4 Les fonctions linéaires

3.4.4.1 Introduction

Nous sommes dans l'optique de construire une relation entre x et y afin d'arriver à prédire une nouvelle valeur de y en fonction de nouvelles données x.

En statistiques (ainsi qu'en probabilités), lorsque l'intensité du lien entre ces deux variables est étudié, leur **corrélation** est calculée. En quantifiant cette dépendance, la corrélation peut alors jouer le rôle d'une mesure de similarité. Lorsque la relation recherchée est affine, la corrélation est dite linéaire, et se détermine à partir d'une régression linéaire entre les variables.

La régression linéaire est un outil particulièrement utilisé lorsque le type de dépendance entre x et y est connu. Même si le lien entre ces variables n'est pas linéaire, il est souvent possible de linéariser la relation afin d'en calculer la corrélation. Par exemple :

- pour une dépendance de type exponentielle, une corrélation linéaire entre $ln(y)$ et x est cherchée. Citons comme exemple la courbe de croissance d'une population qui, si elle n'est pas contrainte par la pression du milieu (famine, guerre, etc.) prend une forme exponentielle (loi de Malthus). Une telle courbe peut se noter $y = \theta_0 e^{\theta_1 x}$ et peut facilement se linéariser en $ln(y) = \theta_1 x + ln(\theta_0)$.
- pour une dépendance de type puissance, une corrélation linéaire entre $ln(y)$ et $ln(x)$ est cherchée. Pendant la croissance d'un organisme par exemple, toutes les parties ne se développent pas à la même vitesse, ce qui altère ses proportions générales (phénomène d'allométrie). Un tel cas est décrit par la relation de rapport de puissance constant entre les variables, et se note $y = \theta_0 x^{\theta_1}$, qui peut se linéariser sous la forme $ln(y) = \theta_1 ln(x) + ln(\theta_0)$.

En des termes plus proches de notre problématique, lorsqu'une relation de similarité est connue, si cette relation est linéaire, ou qu'elle est non-linéaire mais linéarisable, elle peut être modélisée par une régression linéaire.

3.4.4.2 Définition

Une régression linéaire est une équation de régression qui suppose la relation suivante entre les variables endogènes et exogènes :

$$y = f(x) = \langle w, x \rangle + b \tag{3.25}$$

Elle est appelée la *régression linéaire simple*. Cette définition se pose comme un cas particulier de ce que nous entendons par fonction de régression linéaire (section 3.4.3.1). Nous concevons une fonction de régression linéaire comme une fonction dont le vecteur de paramètres θ de f_θ peut engendrer n'importe quelle forme de relation linéaire entre les variables, pas seulement une relation affine. C'est le propre d'une *régression linéaire multiple*. Nous la définissons de la sorte :

$$y = f_\theta(x) = \sum_{i=1}^{p} \theta_i \varphi_i(x) \tag{3.26}$$

avec φ_i une fonction quelconque de x.

Nous voyons, par exemple, que :
- pour $\varphi = \{\varphi_1 : x \mapsto x, \varphi_2 : x \mapsto b\}$ et $\theta = \{\theta_1, \theta_2\} = \{w, 1\}$, nous obtenons l'équation 3.25 ;
- pour $\varphi = \{\varphi_1 : x \mapsto x^0, \ldots, \varphi_p : x \mapsto x^{p-1}\}$, nous obtenons n'importe quel équation polynomiale de degrés $p - 1$.

Plus généralement, toute fonction fondée sur des bases fonctionnelles classiques (splines, base de Fourier, base d'ondelettes, etc.) est une fonction de régression linéaire au sens de l'équation 3.26, les fonctions de bases étant les φ_i.

Nous cherchons une fonction capable d'estimer correctement la variable y de l'équation précédente (équation 3.26), pour toute fonction φ de x. De nombreuses fonctions existent dans la littérature, nous pouvons citer parmi les plus utilisées la méthode des moindres carrés, du maximum de vraisemblance, etc.

3.4.4.3 Exemple : les moindres carrés

Supposons que nous disposons de notre ensemble d'apprentissage $S = \{(x_i, y_i)\}_{i=\{1,\ldots,N\}}$. Cette méthode consiste à minimiser la somme des carrés des résidus :

$$R_f = \sum_{i=1}^{N} r_i^2 = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left(y_i - \sum_{j=1}^{p} \theta_j \varphi_j(x_i) \right)^2 \tag{3.27}$$

Remarque : l'équation précédente peut être interprétée de manière géométrique. Minimiser l'équation 3.27 revient à minimiser la distance euclidienne entre y et $f_\theta(x)$. Le risque peut ainsi s'écrire $R_f = \|y - f_\theta(x)\|^2$.

Si le minimum de R_f existe, alors le gradient s'annule. Il va nous falloir différentier la relation ci-dessus par rapport à chaque inconnue θ_j. Nous obtenons ainsi un système linéaire de p équations *normales* à p inconnues, qui s'écrit sous la forme suivante :

$$\frac{\partial R_f}{\partial \theta_j} = 2\sum_{i=1}^{N} r_i \frac{\partial r_i}{\partial \theta_j} = 2\sum_{i=1}^{N} \frac{1}{\sigma_i^2}\left(y_i - \sum_{k=1}^{p} \theta_k \varphi_k(x_i)\right)\varphi_j(x_i) = 0, j = 1, \ldots, p \tag{3.28}$$

Nous introduisons la matrice Φ, appelée *matrice modèle*, de dimension $N \times p$ qui va nous aider à mieux formaliser le système. Un élément de Φ se note $\Phi_{ik} = \varphi_k(x_i)/\sigma_i$, pour $i = 1, \ldots, N$ et $k = 1, \ldots, p$.

Notons qu'en réorganisant les termes des équations normales, nous pouvons faire apparaître le terme suivant :

$$\sum_{i=1}^{N} \frac{\varphi_k(x_i)\varphi_j(x_i)}{\sigma_i^2} \tag{3.29}$$

qui est, de fait, la composante d'indice jk de la matrice $\Phi^T\Phi$ de dimension $p \times p$. En définissant le vecteur $b = (b_1, \ldots, b_N)$ tel que $b_i = y_i/\sigma_i$, le système d'équations 3.28 devient :

$$(\Phi^T\Phi)\theta = \Phi^T b \tag{3.30}$$

La résolution de l'équation 3.30 peut se faire de plusieurs manières (grâce à la décomposition de Cholesky ou d'une décomposition LU par exemple).

3.4.5 Les fonctions non linéaires

3.4.5.1 Introduction

Il se peut toutefois que le type de dépendance entre les données, qui caractérise l'expression de la similarité, ne soit pas linéarisable, ou ne soit pas connu *a priori*. Dans ce cas, une fonction non-linéaire est utilisée afin de prédire la similarité entre les données.

De nombreuses méthodes d'estimation de paramètres non-linéaires sont utilisées en prédiction statistique (les moindres carrés partiels par noyau [RLGW94], les algorithmes génétiques [VA98], les réseaux de neurones [HSW89], les moindres carrés non linéaires [Kel99], la régression par vecteur de support [CV95], etc.). La plupart des méthodes d'estimations linéaires ont été portées dans le domaine du non-linéaire, et ce en grande partie grâce à l'astuce du noyau, dont nous avions succinctement parlé dans la section 3.3.1.2.

Dans cette section, nous expliquons plus en détail comment fonctionne le *kernel trick*, et présentons deux algorithmes très utilisés pour créer des prédicteurs non linéaires : les moindres carrés non linéaires, qui n'utilisent pas l'astuce du noyau, et la régression par vecteur de support qui, elle, l'utilise.

3.4.5.2 L'astuce du noyau

L'astuce du noyau, ou *kernel trick* [ABR64], est une méthode qui consiste à obtenir un algorithme non linéaire en se servant d'un algorithme linéaire. Il suffit pour cela que le problème

initial puisse se formuler par une expression qui ne dépend que du produit scalaire entre les variables d'observation.

Soient x_1 et x_2 deux variables de $\mathbb{R}^N$. Soit une fonction K dont les arguments sont à valeurs dans un espace mesurable ψ et qui satisfait les conditions de Mercer [19]. Alors il existe un espace préhilbertien E et une fonction $\Phi : \mathbb{R}^N \rightarrow E$ continue sur $\mathbb{R}^N$ telle que :

$$K(x_1, x_2) = \langle \Phi(x_1), \Phi(x_2) \rangle \tag{3.31}$$

Le *kernel trick* consiste à remplacer un produit scalaire dans un espace de grande dimension, par une fonction noyau très simple à calculer. Voici quelques exemples de noyaux utilisés couramment dans la littérature :

- le noyau linéaire : $K(x_1, x_2) = \langle x_1, x_2 \rangle$;
- le noyau polynomial de degré n : $K(x_1, x_2) = \langle x_1, x_2 \rangle^n$, $avec\ n \in \mathbb{N}$;
- le noyau polynomial non homogène de degré n : $K(x_1, x_2) = (\langle x_1, x_2 \rangle + a)^n$ avec $n \in \mathbb{N}\ et\ a \geq 0$;
- le noyau gaussien : $K(x_1, x_2) = e^{-\frac{\|x_1 - x_2\|^2}{2\sigma^2}}\ avec\ \sigma > 0$;
- le noyau sigmoïde : $K(x_1, x_2) = tanh(a \langle x_1, x_2 \rangle - b)$.

3.4.5.3 Exemple sans *kernel trick* : les moindres carrés non linéaires (NLLS)

Il s'agit d'une des techniques les plus répandues pour estimer les paramètres de régressions non linéaires. On la note NLLS pour *Non-Linear Least Square* [Kel99].

Supposons que nous disposons de N observations (x_i, y_i). Comme son pendant linéaire, cette méthode consiste à minimiser R_f, la somme des carrés des résidus, qui, dans le cadre de l'estimation non linaire, s'écrit sous la forme suivante :

$$R_f = \sum_{i=1}^{N} r_i^2 = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} (y_i - f_\theta(x_i))^2 \tag{3.32}$$

Un des techniques de minimisation les plus utilisées dans ce cas s'appelle la *méthode du gradient*. Si le minimum de R_f existe, il est atteint au point d'annulation du gradient. Une fonction de régression à p paramètres génère ainsi p équations normales :

$$\frac{\partial R_f}{\partial \theta_j} = 2 \sum_{i=1}^{N} r_i \frac{\partial r_i}{\partial \theta_j} = 0, \text{avec } j = 1, \ldots, p \tag{3.33}$$

Les dérivées partielles dépendent à la fois des variables explicatives et des paramètres. Les équations normales sont résolues à l'aide du procédé itératif suivant : $\theta^{k+1} = \theta^k + \Delta\theta$. Le point fort de ce processus est qu'il fournit des approximations successives θ^k, de plus en plus proches

19. Théorème de Mercer : une fonction noyau K satisfait les conditions de Mercer si elle est continue, symétrique et semi-définie positive. Elle peut alors s'exprimer comme produit scalaire dans un espace de plus grande dimension [SBS99].

de θ^*, valeur idéale et inconnue du vecteur de paramètres. Il est toutefois dépendant du choix de $\Delta\theta$ qui, s'il est mal choisi, peut entraîner la divergence de l'algorithme.

3.4.5.4 Exemple avec *kernel trick* : la ε *Support Vector Regression* (ε-SVR)

La technique de la ε-SVR [CV95] consiste à trouver une fonction $f(x)$ qui dévie de la cible y d'au plus ε. Autrement dit, les erreurs d'approximation de y_i par $f(x_i)$ sont acceptées tant qu'elles restent inférieures à ε. Cette fonction doit également être suffisamment lisse pour éviter les phénomènes de surajustement (dont nous parlerons dans la section 3.4.7.2).

Pour ce faire, il convient de construire une fonction f qui régresse au mieux les éléments réellement représentatifs de l'ensemble d'apprentissage : les points trop éloignés de la fonction idéale ne doivent pas intervenir dans la modélisation, au risque d'introduire un biais trop important.

Nous expliquons dans cette section comment procéder pour obtenir une équation de régression non-linéaire basée sur de tels éléments.

Considérons dans un premier temps une fonction f linéaire (la forme affine de la régression linéaire) d'équation :

$$f(x) = \langle w, x \rangle + b \tag{3.34}$$

La distance $dist(x)$ d'un point x à la fonction f est donnée par sa projection orthogonale sur f :

$$dist(x) = \frac{|w \cdot x + b|}{\|w\|} \tag{3.35}$$

La plus petite distance entre les vecteurs d'apprentissage et la fonction f est appelée la **marge**. Une grande marge permet d'obtenir une régression de biais faible [SS03], propriété nécessaire à la construction de notre fonction. Notre objectif est d'obtenir une marge maximale sur les vecteurs qui serviront de support à l'édification de la régression.

Au regard de l'équation 3.35, maximiser cette marge revient à minimiser $\|w\|$ sous les contraintes suivantes :

$$\min_w \frac{1}{2} \|w\|^2 , \text{ tel que } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon \end{cases} \tag{3.36}$$

En écrivant l'équation 3.36 nous supposons de manière implicite qu'il existe bien une fonction f qui approxime les couples (x_i, y_i) avec une précision de ε, c'est-à-dire que ce problème d'optimisation convexe est faisable.

Il se peut toutefois que ce ne soit pas le cas, c'est pourquoi il est coutume d'apporter un peu de souplesse dans la modélisation en introduisant une certaine tolérance à l'erreur. Cela est traduit par l'ajout de variables de relâchement ξ et ξ^*, appelées *slack variables* (méthode analogue au calcul de la fonction de perte dite *soft margin* de [BM92], utilisée pour les machines à vecteur support par [CV95]).

Le problème se reformule alors de la sorte :

$$\min_{w} \frac{1}{2} \|w\|^2 + \mathbb{C} \sum_{i=1}^{N} (\xi_i + \xi_i^*) \quad \text{tel que} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, i = 1, \ldots, N \end{cases} \tag{3.37}$$

Il est appelé problème primal, les variables (w, b, ξ, ξ^*) se nomment les contraintes primales du problème d'optimisation.

La constante $\mathbb{C} > 0$ de contrôle de l'erreur, appelée le coût, est chargée d'établir un compromis entre le caractère lisse de f et la marge de déviation tolérée au-delà de ε. Elle est directement en lien avec la fonction de coût ε-insensible L_ε (c.f. équation 3.21). La figure 3.5 illustre son utilisation : seuls les points hors de la région grisée sont linéairement pénalisés par la fonction de coût, proportionnellement au paramètre de contrôle $\mathbb{C}$.

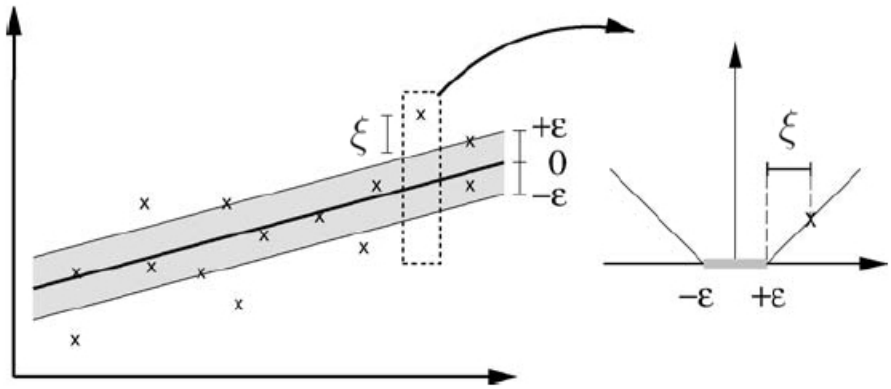


FIGURE 3.5 – Paramétrisation de la marge souple d'un SVM linéaire [SS02]

La technique d'optimisation courante est basée sur la méthode du Lagrangien, et passe par la résolution du dual de 3.37. La construction du Lagrangien $\mathcal{L}$ de la fonction objectif du problème primal s'opère en introduisant les variables duales α_i, α_i^*, η_i et η_i^*, appelées multiplicateurs de Lagrange :

$$\begin{aligned} \mathcal{L} = \tfrac{1}{2} \|w\|^2 + \mathbb{C} \sum_{i=1}^{N} (\xi + \xi_i^*) - \sum_{i=1}^{N} (\eta_i \xi + \eta_i^* \xi_i^*) - \sum_{i=1}^{N} \alpha_i (\varepsilon + \xi_i - y_i + \langle w, x_i \rangle + b) \\ - \sum_{i=1}^{N} \alpha_i^* (\varepsilon + \xi_i^* + y_i - \langle w, x_i \rangle - b), \quad \text{tel que } \alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0 \end{aligned} \tag{3.38}$$

La solution de l'équation 3.37 est atteinte au point critique du Lagrangien 3.38 [McC83] (c'est-à-dire au minimum, au maximum ou au point de selle [20] de celui-ci). En d'autres termes, pour qu'il y ait optimalité, il faut que les dérivées partielles de $\mathcal{L}$ s'annulent sur les contraintes primales :

$$\partial_w \mathcal{L} = \partial_b \mathcal{L} = \partial_\xi \mathcal{L} = \partial_{\xi^*} \mathcal{L} = 0. \tag{3.39}$$

20. Un point selle, ou *saddle point*, pour une fonction f à valeur dans $\mathbb{R}$ est un point pour lequel les dérivées de f s'annulent sans que soit un extremum local.

La résolution du problème aux dérivées partielles nous permet de réécrire w comme une combinaison linéaire des éléments d'apprentissage x_i :

$$w = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) x_i \tag{3.40}$$

Nous pouvons ainsi redéfinir f en écrivant l'équation 3.34 sous la forme suivante :

$$f(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b \tag{3.41}$$

Les conditions de Karush-Kuhn-Tucker (KKT) [KT82] nous garantissent qu'au point de solution de notre problème d'optimisation, le produit entre les variables duales et les contraintes primales s'annule, soit :

$$\begin{cases} \alpha_i(\varepsilon + \xi_i - y_i + \langle w, x_i \rangle + b) = 0 \\ \alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle w, x_i \rangle - b) = 0 \end{cases} \tag{3.42}$$

Il résulte de cette équation que les multiplicateurs de Lagrange seront non nuls uniquement pour $\|f(x_i) - y_i\| \geq \varepsilon$: pour tous les exemples à l'intérieur du tube (la région grisée de la figure 3.5) ε, c'est-à-dire pour tout x_i tel que $\|f(x_i) - y_i\| < \varepsilon$, le deuxième terme de l'équation 3.42 est non nul, ce qui implique que α_i et α_i^* doivent être nuls pour que les conditions de KKT soient satisfaites.

L'ensemble des exemples pour lesquels ces coefficients ne sont pas nuls serviront à construire f. Ces éléments sont appelés les **vecteurs supports**.

Il est maintenant trivial, grâce au *kernel trick*, de transposer ce problème dans le domaine du non-linéaire :

Nous supposons la réalisation de notre problème de régression faisable dans un espace E de plus grande dimension (même infinie), grâce à une fonction non-linéaire Φ inconnue.

Supposons $\Phi : \mathbb{R}^N \rightarrow E$ cette fonction non-linéaire, telle que E soit un espace préhilbertien et que Φ soit continue sur $\mathbb{R}^N$. Selon la même procédure que pour la forme linéaire, nous construisons une variante non-linéaire de l'équation 3.41 de f :

$$f(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \langle \Phi(x_i), \Phi(x) \rangle + b \tag{3.43}$$

et souhaitons trouver une solution sans avoir à expliciter Φ.

Considérons K une fonction à valeur dans un espace mesurable et qui satisfait les conditions de Mercer. Alors elle peut être substituée au produit scalaire $\langle \Phi(x_i), \Phi(x) \rangle$ (sous la contrainte des hypothèses de définition de Φ et E) :

$$f(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) K(x_i, x) + b \tag{3.44}$$

Les performances de f sont directement liées aux choix de la fonction K (et donc de ses paramètres), de la largeur du tube ε et du coût $\mathbb{C}$.

3.4.6 Réduction de la dimensionnalité

Au delà du choix de la fonction utilisée, la performance d'un modèle de régression est très souvent fonction de la qualité des données d'apprentissage. Pour avoir la prétention d'expliquer un phénomène, il faut que les valeurs descriptives (les témoins de ce phénomène) soient le moins bruitées possibles.

Ce lien entre les données et le modèle se fait via des algorithmes de réduction de la dimensionnalité (cf. figure 3.6).

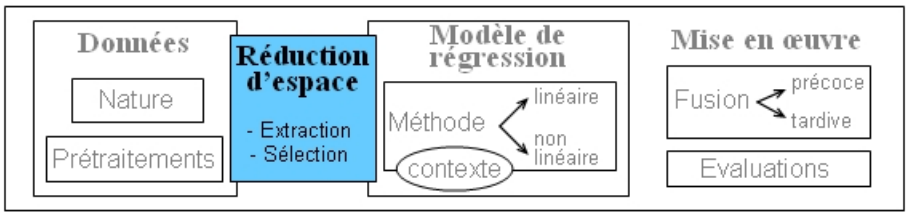


FIGURE 3.6 – Positionnement de la réduction de la dimensionnalité dans le schéma général

3.4.6.1 Modèle parcimonieux

Il est plus que souhaitable d'arriver à dissocier les effets entre les composantes des variables exogènes.

Le problème se porte sur l'espace de description : est-il nécessaire de prendre en compte toutes les valeurs descriptives afin de construire le modèle, ou quelques valeurs seulement ne seraient-elles pas suffisantes ?

Il se peut également que l'utilisation de certaines de ces valeurs porte nettement préjudice à la pertinence du modèle. Un problème peut se poser lorsque, par exemple, celles-ci sont fortement corrélées. La *quasi-colinéarité* des variables, voire leur colinéarité complète, entraîne une instabilité des valeurs des paramètres du modèle (forte variance) et une perte de leur interprétabilité.

Il nous faut alors choisir entre :

– un modèle *exhaustif*, qui va utiliser un ensemble de variables pertinentes aussi complet que possible, mais qui risque de conduire à ces phénomènes indésirables ;
– et un modèle *parcimonieux*, qui va minimiser le nombre de données nécessaires à l'apprentissage, mais qui risque d'en oublier certaines essentielles.

Nous pensons qu'un modèle numérique de similarité se doit d'être parcimonieux sur le choix de ses valeurs exogènes, quitte à perdre en exhaustivité.

En plus de potentiellement accroître les performances du système, un modèle parcimonieux a l'avantage de réduire le coût en temps de calcul de l'apprentissage piloté par la fonction de régression, d'améliorer la compréhension des hypothèses et de soulager du problème du « fléau de la dimensionnalité » [21] [Bel61].

La question est la suivante : comment peut-on réduire la dimensionnalité de l'espace de description en limitant la perte d'information ? La réponse théorique est simple : éliminer les valeurs redondantes, qui ne sont pas porteuses d'information, et trouver une structure pertinente de l'espace de dimension moindre.

En pratique, des algorithmes de réduction de la dimensionnalité sont utilisés. La littérature les regroupe en deux catégories :

- les algorithmes de sélection d'attributs (*Feature Selection Algorithms*), qui choisissent un sous-ensemble de variables sans agir sur la topologie de cet espace (figure 3.7 a.) ;
- les algorithmes d'extraction d'attributs (*Feature Extraction Algorithms*), qui transforment l'espace de description pour le réduire (figure 3.7 b.).

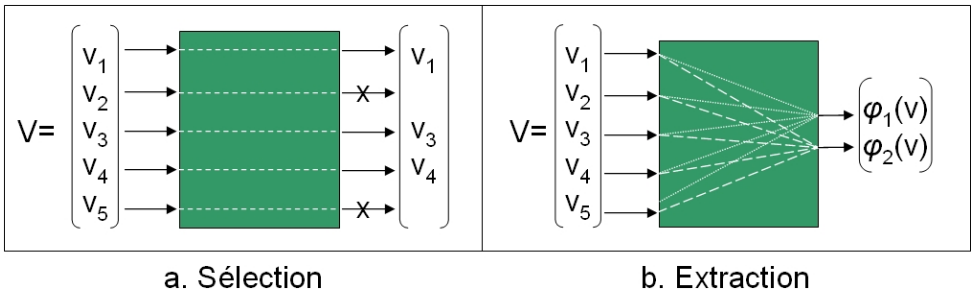


FIGURE 3.7 – Méthodes de réduction de dimensionalité

Nous présentons, dans la section suivante, quelques unes des principales méthodes d'extraction d'attributs, avant de nous focaliser un peu plus en profondeur sur les principes de la sélection d'attributs.

3.4.6.2 Principaux algorithmes d'extraction d'attributs

Les algorithmes d'extraction de variables consistent à transformer un espace de grande dimension pour en obtenir une représentation concise (compression de l'information) et fidèle (préservation de l'information). Le tableau 3.1 fait une synthèse des méthodes que nous présentons dans cette section (voir la thèse de José Arias [Ari08] pour un état de l'art complet). Nous nous attardons sur les méthodes issues du domaine de l'apprentissage non-supervisé, les

21. Appelé *curse of dimensionnality* en anglais : dans des espaces de grande dimension, la représentativité d'un ensemble d'apprentissage est toujours compromise car le nombre d'échantillons est en relation exponentielle avec la dimensionnalité des variables.

méthodes supervisées étant pour la plupart des extensions de celles-ci.

L'algorithme d'**Analyse en Composantes Principales** (PCA pour *Principal Component Analysis* [Jol86]) consiste en une recherche des directions de l'espace étudié qui représentent le mieux les corrélations entre les vecteurs. Pour ce faire, on effectue une diagonalisation de la matrice de covariance des vecteurs. Leurs nouvelles coordonnées dans la base des vecteurs propres (i.e. leur projection sur les vecteurs propres) sont appelées « composantes principales ». Les axes où la variance des données est faible peuvent être éliminés, l'espace est ainsi réduit avec une perte minimale d'information.

Beaucoup de techniques d'extraction linéaires se basent sur les mêmes principes que cet algorithme, qu'elles soient issues du domaine de l'apprentissage non supervisé (la méthode *Independant Component Analysis* (ICA) [Com94] par exemple) ou étendues à de l'apprentissage supervisé (comme l'Analyse Factorielle Discriminante (AFD) [Sap06], et l'algorithme *Linear Discriminant Analysis* (LDA) [HUN92]).

La transformation PCA est par nature linéaire, mais de nombreux travaux ont portés sur des adaptations non-linéaires de l'algorithme (*Kernel* PCA (KPCA) [SSM97], *Non Linear* PCA (NLPCA) [SKG+05], *Iterative Kernel* PCA (IKPCA) [KFS05], *Approximate Kernel* PCA (AKPCA) [CS06]).

Une méthode très rencontrée dans la littérature est le **Multidimensional Scaling** [CC01] (MDS), appelée aussi Analyse en Coordonnées Principales. C'est une méthode qui s'inspire de la similarité pour réduire la dimension de l'espace. Si le système possède l'information de similarité existant entre les points du corpus d'apprentissage sous la forme d'une matrice de distances, il est alors possible d'obtenir une représentation de plus faible dimension (typiquement deux ou trois, pour de la visualisation d'information) qui préserve les distances entre les points, en se basant sur les valeurs propres de cette matrice.

L'algorithme **Isomap** [TSL00] (voir les méthodes S-Isomap [GZZ05] et Multiclass Isompap [YC04] pour des versions supervisées) est une extension du MDS aux cas non-linéaires. L'idée est de construire un graphe de similarité des données localement connectées, dépendant du k-voisinage de chaque point. Les distances entre couples de points sont calculées par l'exploration du graphe, en prenant en compte la longueur du chemin le plus court entre les points. Cette distance dite « géodésique » est une approximation de la distance réelle entre les couples. On utilise pour finir la technique des MDS pour trouver des ensembles de points similaires à dimensions réduites en fonction de la distance géodésique calculée précédemment.

Isomap est souvent confronté à la méthode de réduction appelée ***Locally Linear Embedding*** (LLE) [RS00] (voir l'algorithme SLLE [dRKO+03] pour une version supervisée), qui exploite également la géométrie des points de l'espace d'origine pour la reproduire dans un espace de plus faible dimension. L'idée est de considérer un vecteur comme une combinaison linéaire de ses voisins et de conserver cette relation dans l'espace de dimension réduite.

TABLE 3.1 – Récapitulatif des méthodes d'extraction de caractéristiques.

Type	**Méthodes**
Linéaire	PCA, ICA, MDS
Linéaire supervisée	LDA, AFD
non-linéaire	*Isomap*, LLE, NLPCA KPCA, IKPCA, AKPCA
non-linéaire supervisée	S-*Isomap*, SLLE, *Multiclass Isomap*

3.4.6.3 Principes fondamentaux des procédures de sélection d'attributs

De nombreux domaines de recherche, tels que l'apprentissage automatique (*machine learning*) ou la fouille de données (*data mining*), ont recours à la sélection d'attributs.

Rechercher un sous ensemble optimal de p variables parmi P revient à explorer les $\mathbf{C}_p^P$ combinaisons de sous-ensembles constructibles. La complexité d'un tel algorithme est exponentielle ($O(p^P)$), il s'agit d'un problème NP complet. L'exploration exhaustive de tous ces sous-ensemble devient rapidement irréaliste, même pour des valeurs faibles de p et de P. Les algorithmes de sélection ont pour tâche de rendre réalisable la recherche d'un sous-ensemble (optimal ou non) pour améliorer les performances d'un système.

Le sujet ayant été déjà largement abordé dans la littérature (voir [DBM07] pour un état de l'art complet), nous ne nous attardons que sur les points qui nous ont semblé utiles pour une bonne compréhension du sujet et/ou incontournables de par leur très fréquente utilisation.

De manière générale, nous pouvons définir une procédure de sélection selon les étapes suivantes (figure 3.8) [DL97] :

1. une **procédure de génération** qui crée les différents sous-ensembles candidats (nous en parlons en détail plus bas dans cette section) ;
2. une **fonction d'évaluation** qui mesure la pertinence du sous-ensemble en cours d'analyse. Elles se répartissent en quatre familles, suivant comment le concept de pertinence est défini :
   - les **mesures statistiques de corrélation** et les **mesures de divergence** entre distributions de probabilité. Ces mesures permettent de distinguer parmi les attributs pertinents ceux qui le sont fortement (et qui doivent impérativement être intégré au modèle) de ceux qui le sont faiblement (et dont l'apport en information dépend du contexte) [KJ97] ;
   - les **mesures de cohérences**, qui proviennent d'une réflexion sur les travaux précédents [AD91]. La notion de pertinence s'appuie également sur la distinction forte / faible, mais est directement liée aux échantillons étudiés [BL97] ;
   - les **mesures d'information**. Elles se basent sur la définition de la pertinence entropique faible et forte [BW00], qui tente de quantifier la pertinence d'un attribut en mesurant l'incertitude du gain en information grâce à l'entropie conditionnelle ;

3. un **critère d'arrêt** qui détermine quand mettre fin au processus ;
4. un **processus de validation** qui vérifie si le sous-ensemble trouvé est valide. Ce processus n'appartient pas, à proprement parler, à la procédure de sélection elle-même, mais cette étape est à envisager en pratique. Il s'agit de vérifier la validité du sous-ensemble en faisant différents tests, et de comparer les résultats soit par validation croisée, soit en comparant les méthodes de sélection avec des jeux de données artificielles et/ou réelles.

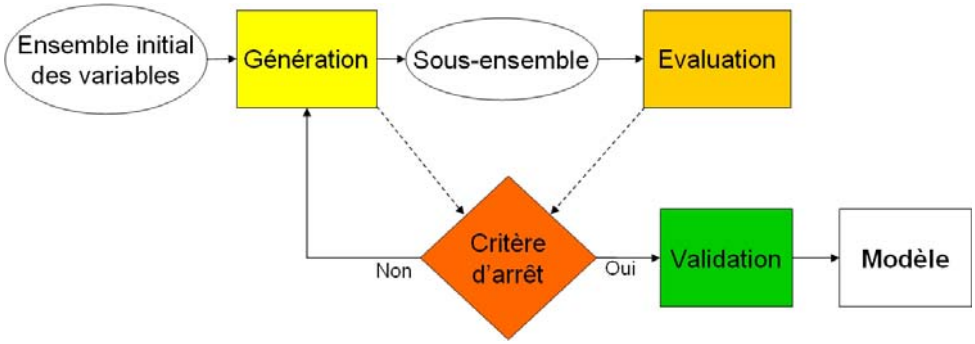


FIGURE 3.8 – Procédure générale de sélection avec validation

C'est sur la procédure de génération que nous portons notre attention. Il s'agit de la méthode utilisée pour explorer l'espace des sous-ensembles possibles.

La procédure de sélection idéale serait de générer tous les tuples possibles, à savoir 2^P sous-ensembles, afin de conserver le meilleur en regard de l'objectif choisi (construction d'un classifieur, d'une régression...). Une telle opération s'avère beaucoup trop coûteuse en temps de calcul pour être appliquée, même pour un P faible. De nombreuses alternatives existent, choisir parmi celles-ci revient à établir un compromis entre précision et temps de calcul.

Blum et Langley [BL97] proposent de différencier ces méthodes selon qu'elles relèvent de l'un des trois types d'approches suivants (schématisées sur la figure 3.9) :

- **approches symbioses** (ou *wrapper methods*) : la pertinence des sous-ensembles générés est évaluée en fonction des performances du modèle en construction (figure 3.9 a.),
- **approches filtres** (ou *filter methods*) : le choix du sous-ensemble est indépendant de l'apprentissage du modèle. Les variables sont sélectionnées dans une phase de prétraitement (figure 3.9 b.),
- **approches intégrées** (ou *embedded methods*) : la mesure de pertinence est directement incluse dans la fonction de coût à optimiser par le modèle. En d'autres termes, ces méthodes cherchent le sous-ensemble et entraînent le modèle simultanément (figure 3.9 c.).

Nous suivons Isabelle Guyon [GE03] sur le constat que la nomenclature n'est toutefois pas très bien établie sur le sujet. La distinction *filter / wrapper / embedded* ne fait pas une partition des méthodes : par exemple, le très utilisé *Branch and Bound* [NF77] (détaillé plus bas dans la partie des méthodes optimales) est un procédé de recherche qui peut être utilisé comme filtre pour prétraiter les données ou être implémenté par un *wrapper* pour parcourir l'espace des variables. En effet, ce n'est pas la méthode employée qui fait un filtre, mais uniquement le

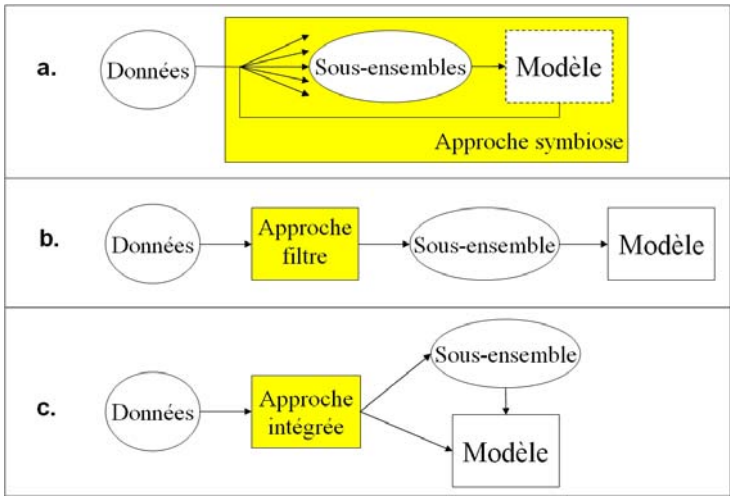


FIGURE 3.9 – Différentes approches pour la procédure de génération

découplage entre la procédure de génération et la procédure d'apprentissage du modèle.

Pour ce qui est de la méthode, ce qui fait la distinction entre une méthode *wrapper* et une méthode *embedded*, c'est que la *wrapper* utilise le modèle d'apprentissage comme une boîte noire, en s'en servant pour sélectionner les attributs. Pour chaque sous-ensemble d'attributs, un modèle est entraîné et évalué (généralement à l'aide de l'estimateur de validation croisée). Par exemple le *Sequential Forward Selection* [DK82] (détaillé plus bas dans la partie des méthodes sous-optimales) version symbiose fait $P(P+1)/2$ entraînements (à chaque itérations, tous les attributs candidats qui restent sont testés) tandis que la version intégrée ne fait que P entraînements (on ne considère à chaque itération que l'attribut qui est le plus prometteur du point de vue de l'objectif).

Nous préférons exposer ces méthodes en fonction de leur caractère optimal et non optimal, et présentons dans cette section les algorithmes les plus utilisés. Les tableaux 3.2 et 3.3 sont un récapitulatif de ces algorithmes.

• Principales méthodes de sélection optimales

La méthode ***Branch and Bound*** (B&B) [NF77] dont nous parlions un peu plus tôt est un algorithme de parcours de graphe sans cycle avec possibilité de retour en arrière. L'idée générale est de découper l'ensemble des variables initial (considéré comme un arbre) en sous problèmes plus petits (sous-arbres), et de remettre en question le coût de la meilleure solution obtenue jusqu'alors. On applique la fonction d'évaluation sur chaque sous arbre pour prédire si un sous problème donnera ou non une solution optimale, ce qui détermine si la branche correspondante mérite ou non d'être explorée. Dans la négative, on élague cette branche de l'arbre. Si, au contraire, l'évaluation est satisfaisante, on calcule la borne inférieure du sous problème sur le coût optimal (généralement par relaxation linéaire) : si cette borne est supérieure au coût optimal, on supprime le sous-problème, sinon on subdivise le sous problème et on applique

l'algorithme à nouveau jusqu'à obtenir le sous-ensemble de caractéristiques optimal.

L'algorithme **FOCUS** ([AD91], [AD94]), défini à l'origine pour des problèmes de classification binaires à données non continues et non bruitées, examine tous les sous-ensembles de manière exhaustive et sélectionne celui qui suffit à étiqueter tout l'ensemble d'apprentissage.

L'algorithme ***Las Vegas*** utilisé comme filtre (LVF) [BB96] (une version *wrapper* existe sous le nom de LVW [LS96]) est une méthode probabiliste qui s'appuie sur un processus incrémental aléatoire et sur un critère d'inconsistance monotone. Á chaque itération, un sous-ensemble d'attributs E est généré. Si les performances de A sont meilleures que A^* le meilleur sous-ensemble au regard du critère d'inconsistance, alors $A^* = A$. On continue jusqu'à ce obtenir une solution optimale.

TABLE 3.2 – Récapitulatif des méthodes optimales de sélection de caractéristiques.

	Déterministes	**Probabilistes**
Solution unique	B&B, FOCUS	
Solution multiple		LVF, LVW

• Principales méthodes de sélection sous-optimales

Les algorithmes les plus couramment utilisés, connus sous le nom d'**algorithmes séquentiels**, sont la *Sequential Forward Selection* (SFS, illustré figure 3.10 a.) et la *Sequential Backward Selection* (SBS, illustré figure 3.10 b.) [DK82]. Le principe est simple : pour la SFS (resp. SBS), nous partons d'un ensemble de variables vide (resp. de l'ensemble de toutes les variables), puis nous ajoutons (resp. éliminons) une variable à chaque étape tant que les performances du modèle s'améliorent. Ces deux méthodes sont rapides, car de complexité polynomiale ($O(N^2)$), et très simples à mettre en œuvre. Toutefois, ces méthodes sont sous-optimales car les variables ajoutées (resp. enlevées) ne sont jamais remises en question.

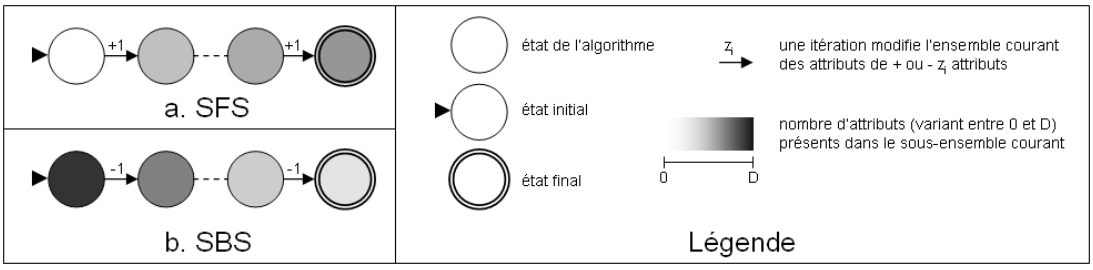


FIGURE 3.10 – Illustration du fonctionnement des algorithmes SFS et SBS

Leurs **versions flottantes** (*Sequential Floating Search methods*) d'acronymes SFFS (illustré figure 3.11 a.) et SBFS correspondant respectivement à la SFS et à la SBS [PNK94], sont considérées comme les plus efficaces parmi les algorithmes sous-optimaux [CE94]. La SFFS consiste

à appliquer, après chaque étape *forward*, x étapes *backward*, x étant déterminé dynamiquement : parmi l'ensemble des variables constituant le sous-ensemble courant, on enlève une à une des variables tant que cela améliore les performances du modèle.

Aucune paramétrisation de x n'est nécessaire, contrairement à l'algorithme ***Plus l Take Away r*** (PTA(l,r)) [Ste76] (illustré figure 3.11 b.) dont il s'inspire, qui nécessite quant à lui une paramétrisation fine des nombres l et r, correspondant respectivement aux nombre d'étapes $forward$ et $backward$. Il suffit d'intervertir les étapes *forward* et *backward* à SFFS pour obtenir la SBFS. Les performances sont améliorées par rapport aux versions non flottantes et la complexité de ces algorithmes reste polynomiale ($O(N^2)$).

Les **versions adaptatives** de ces méthodes flottantes, notées *Adaptive* SFFS (ASFFS, illustré figure 3.11 c.) et ASBFS [SPNP99], sont des générations des précédentes approches séquentielles flottantes. Elles permettent de se rapprocher du sous-ensemble optimal en considérant l'ajout et le rejet des attributs par tuples et non par individus. La taille de ces tuples est déterminée de manière adaptative. Toutefois, la complexité de ces algorithmes est accrue car dépendante du nombre d'étapes *forward* et *backward* implémentées.

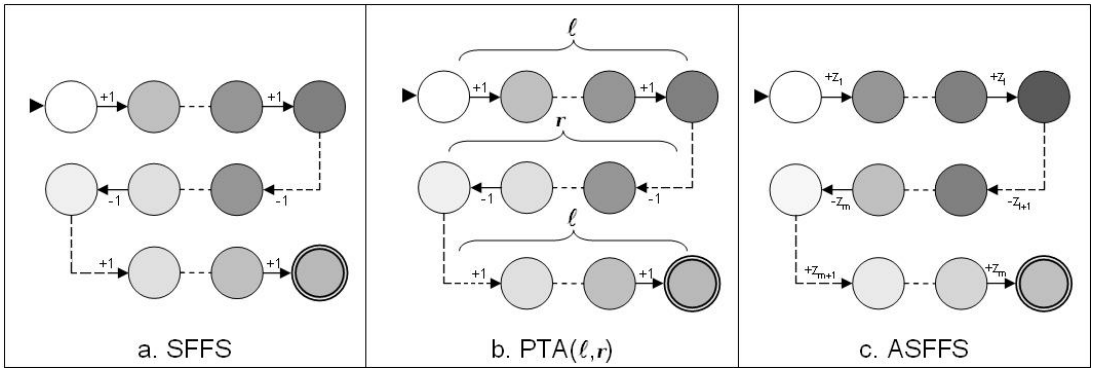


FIGURE 3.11 – Illustration du fonctionnement des algorithmes SFFS, PTA(l,r) et ASFFS. La légende est celle de la figure 3.10

Une autre forme de généralisation de la recherche séquentielle appelée ***Beam search*** [AB95] est très utilisée en sélection d'attributs. Au lieu de garder une unique solution à chaque itération, comme les algorithmes séquentiels précédents, *Beam search* conserve un ensemble de solutions. Á chaque itération, les n meilleurs états sont stockés dans une liste. Pour chaque état, les m états suivants sont générés et évalués. On conserve les n meilleurs parmi les $(n \times m)$ propositions, puis on itère. Cette méthode permet de parcourir plusieurs chemins en même temps, en évinçant les plus mauvais au fur et à mesure de l'algorithme.

L'algorithme sous-optimal **Relief** ([KR92a], [KR92b]), également utilisé en classification binaire, affecte à chaque variable un poids qui dénote la pertinence de celle-ci en regard du concept considéré. Relief sélectionne aléatoirement des sous-ensembles de données depuis l'ensemble d'apprentissage. Il réestime ensuite les poids en se servant de la différence entre les instances les plus proches, instances prises dans la même classe (« *near-hit* ») et dans les classes opposées (« *near-miss* »). Une version multi-classes appelée Relief-F a été implémentée

par [Kon94].

Les **algorithmes génétiques** (AG) [SS93] ont des performances proches de SFFS lorsqu'ils sont utilisés pour de la sélection d'attributs. Ce sont des méthodes probabilistes intéressantes du point de vue de leur analogie avec la biologie. Le principe est de considérer un sous-ensemble d'attributs comme un chromosome. Chaque chromosome est caractérisé par une valeur d'*adaptation*, qui correspond à une valeur du critère d'évaluation. Au départ, la population (l'ensemble des chromosomes) est initialisée aléatoirement, ce qui ne donne pas forcément des individus avec de bonnes valeurs d'adaptations. L'algorithme va faire évoluer cette population, en utilisant des opérateurs de croisements et de mutations pour recomposer les individus, jusqu'à obtenir un solution acceptable répondant le critère d'arrêt, ou atteindre un nombre de générations prédéterminé.

L'algorithme itératif ***Inductive Decision Tree*** (ID3) [Qui86] (dont l'évolution a donné la méthode **C4.5** [Qui92]) est une méthode de construction et d'exploration d'arbres de décisions. Il choisit au hasard un sous-ensemble d'attributs appelé une « fenêtre » et construit un arbre de décision à partir de celle-ci de telle manière qu'il classifie correctement tous les objets dans cette fenêtre. Il applique ensuite cet arbre sur l'ensemble complet des attributs, et teste s'il donne une solution optimale. Sinon, il ajoute une sélection des attributs mal classifiés à la fenêtre pour construire un nouvel arbre et on réitère le processus.

Dans la famille de la sélection par arbre décisionnel, nous pouvons également citer l'algorithme **CART** [LB84], qui est une méthode à partitionnement récursif.

TABLE 3.3 – Récapitulatif des méthodes sous-optimales de sélection de caractéristiques.

	Déterministes	**Probabilistes**
Solution unique	SFS, SBS, SFFS, SBFS, ASFFS, ASBFS, PTA(l,r), ID3, C4.5, CART	Relief
Solution multiple	Beam Search	Relief-F AG

3.4.7 Mise en œuvre

Il nous reste maintenant à considérer le dernier aspect de ce modèle, qui détermine la manière dont les différentes pièces sont assemblée (point 4 du processus général, voir figure 3.12). Nous avons relevés dans cette section deux points cruciaux qui touchent à la gestion de la cohérence de cet assemblage, le premier porte sur le regard porté aux données et le second sur l'ensemble des éléments mis en œuvre.

3.4.7.1 Schéma de fusion

Il nous faut construire un modèle de traitement de données qui puisse prendre une décision en s'appuyant sur des observations. Bien qu'elles soient de natures différentes, ces observations

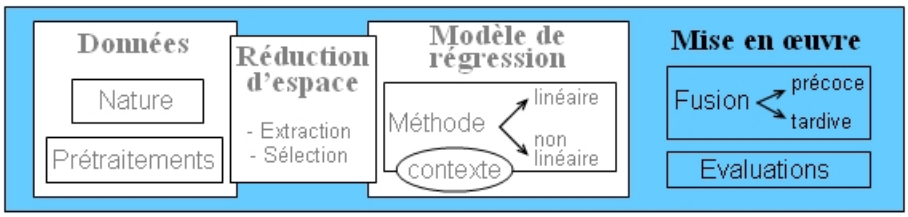


FIGURE 3.12 – Positionnement de la mise en œuvre dans le schéma général

peuvent être regroupées en différentes familles en fonction de leur provenance. Il est nécessaire de prendre conscience du rôle que chacune de ces familles va jouer dans la modélisation. Ces préoccupations sont en grande partie le fait du schéma de fusion des données.

De nombreux travaux ont été portés sur le problème de la fusion, plus ou moins complexes en fonction du type de concepts à mettre en relation ([SWS05], [AQ07], [IN03], [WLCS04], [AQG07]). Les concepts que nous traitons ici sont des valeurs numériques, les schémas de fusion les plus en adéquation avec un modèle numérique de similarité (et qui sont également les plus rencontrés) sont la fusion précoce et la fusion tardive.

Voici une présentation sommaire de ces méthodes :

- la fusion précoce, ou *early fusion*, intègre les caractéristiques comme un tout dans le moteur d'apprentissage final, sans faire de distinction préalable entre les familles d'observations (voir figure 3.13).

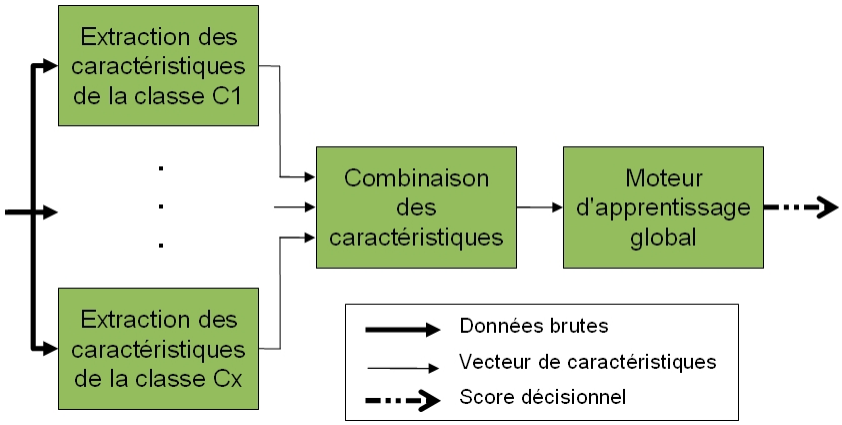


FIGURE 3.13 – Schéma de fusion précoce

- la fusion tardive, ou *late fusion*, traite d'abord chaque famille d'observations avec un moteur d'apprentissage dédié, fournissant ainsi un score par famille, pour que chacun d'eux soit intégré dans un moteur d'apprentissage final (voir figure 3.14).

Ces deux modèles peuvent se combiner entre eux si plus de deux familles d'observations entrent en compte dans la modélisation. Nous pouvons par exemple citer le schéma de fusion de [SYET07], dédié à la synchronisation de documents audiovisuels, et fondé sur l'analyse canonique de la corrélation entre l'observation des lèvres (descripteurs de mouvement et de

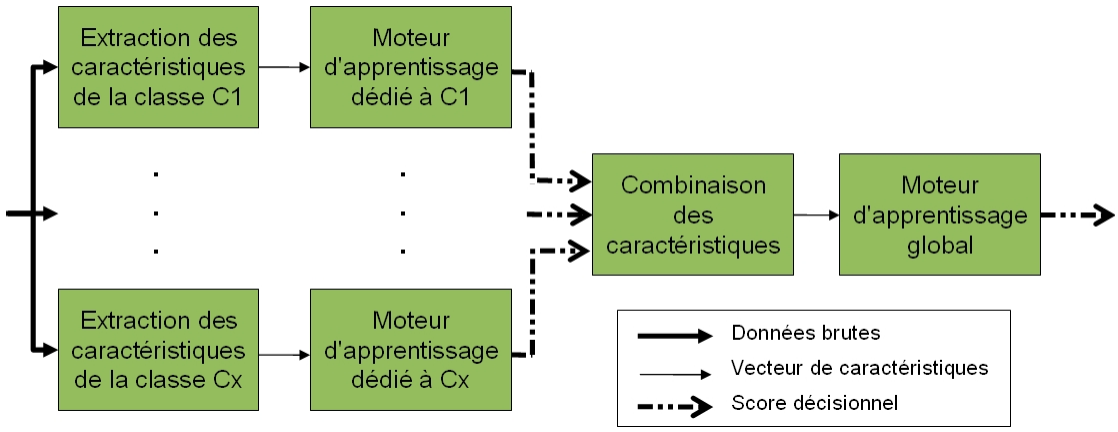


FIGURE 3.14 – Schéma de fusion tardive

texture) et l'activité de parole.

Dans le cadre où nous nous trouvons, la méthode utilisée dans les moteurs d'apprentissages globaux des figures 3.13 et 3.14 est une régression. Le score décisionnel final n'est autre que son résultat, sous forme de valeur réelle.

Le principal atout du schéma de fusion précoce est sa simplicité de mise en œuvre, puisque la combinaison des caractéristiques se fait par une simple concaténation des vecteurs de descriptions. Ce schéma propose un avantage certain pour les questions de corrélation entre descripteurs, les valeurs étant considérées conjointement à un même niveau d'analyse. Toutefois, cette dernière propriété implique d'aborder la question de la normalisation avec soin.

La fusion tardive est un schéma intéressant si l'étude de la similarité repose sur une confrontation des différentes familles de phénomènes, et que nous disposons d'une modélisation performante pour chacune de ces familles (les moteurs d'apprentissages dédiés de la figure 3.14).

Dans une étude portant sur des données audiovisuelles, une famille peut, par exemple, correspondre à une modalité (vidéo, audio ou textuelle), ou bien à un regroupement de « sous-modalités » : pour la vidéo, il est courant de faire des familles de descriptions, en fonction de la provenance des valeurs descriptives, qu'elles soient issues de l'analyse des textures, des formes ou des couleurs de la vidéo (à la manière des systèmes CBIR de recherche d'images basées sur le contenu). Ce type de considérations induit de faire un travail d'abstraction sur les données, en créant un concept numérique par famille d'observation.

Notons que l'étape d'intégration des différentes familles dans le moteur d'apprentissage global, implique une nouvelle phase d'entraînement qui peut conduire à des phénomènes de sur-apprentissage.

3.4.7.2 Méthodes d'évaluation du modèle

Nous nous intéressons ici aux techniques permettant de valider les différentes briques de notre modèle.

Une méthode statistique pertinente ne doit pas modéliser trop finement l'ensemble d'apprentissage, mais plutôt établir un bon compromis entre les performances de généralisation et les performances de d'apprentissage. Sa validité dépend de l'évaluation de cet ajustement, appelé « compromis biais / variance » [GBD92].

Cette section nous permet d'appréhender cette notion statistique, étroitement liée à celle de la performance. Nous mettons l'accent sur les méthodes existantes pour contrôler ce biais de manières *a priori* et *a posteriori*.

• Le compromis biais / variance

Nous illustrons ce principe à l'aide de la figure 3.15. Dans celle-ci, la fonction g, qui représente la régression idéale, est supposée inconnue ; la fonction f est une approximation de g (la sortie du modèle de régression) ; les croix représentent les éléments de l'ensemble d'apprentissage.

Un modèle ayant un biais trop faible génère une variance trop forte. Dans l'exemple de la figure 3.15 (partie a.), la fonction f s'ajuste de manière presque parfaite aux données d'apprentissage : son biais est quasiment nul. Toutefois sa forme varie beaucoup trop, car elle est totalement fonction des données. Ce risque est appelé le surajustement (ou *overfitting*) ;

Un modèle ayant un biais trop fort possède une faible variance. Dans l'exemple de la figure 3.15 (partie b.), la fonction affine f s'écarte énormément du modèle idéal, son biais est très important. Toutefois cet écart ne dépend que très peu de l'ensemble d'apprentissage, sa variance est donc faible. Ce risque s'appelle la surgénéralisation ;

Un bon modèle génère une fonction ayant un bon équilibre entre ces paramètres. La fonction f de la figure 3.15 (partie c.), a un comportement proche de l'idéal. Son biais et sa variance sont faibles, la régression donnera une réponse moyenne satisfaisante (bonne capacité de généralisation) tout en dépendant le moins possible de l'échantillon d'apprentissage (bonne capacité de généralisation).

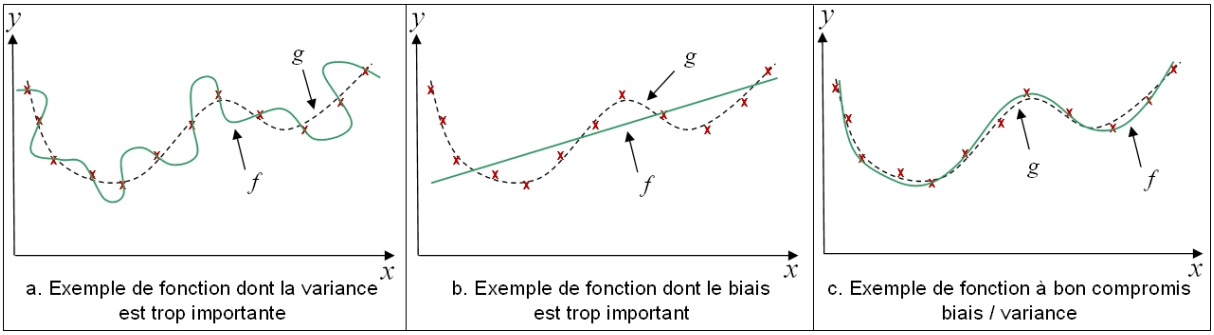


FIGURE 3.15 – Illustration portant sur le compromis biais / variance

Il existe des techniques permettant d'éviter ces problèmes, ou du moins de les contrôler. Elles se classent en deux familles qui dépendent de la manière de résoudre le problème (pendant

ou après la procédure d'apprentissage). Nous les présentons dans les deux sections suivantes.

• Gestion *a priori*

Nous pouvons exprimer le problème du dilemme biais / variance de la façon suivante : l'espace des hypothèses doit être contrôlé pour que les fonctions soient recherchées de manière à minimiser les risques de surgénéralisation et de surajustement. Lorsque ce problème est abordé *a priori*, c'est-à-dire pendant (voire avant) l'apprentissage, nous parlons de régularisation.

Lorsque nous avons défini un modèle de régression dans la section 3.4.3.1, nous avons abordé la question de la performance de la fonction qui lui est liée au travers de la minimisation du risque régularisé, dont nous rappelons ici l'équation 3.15 :

$$R_{reg}[f] = R_{emp}(f) + \upsilon Reg(f)$$

Nous avons fait le choix d'utiliser cette technique, dite du *weight decay*, qui est la plus connue des méthodes de pénalisation de la fonction de coût *a priori*. La constante de régulation υ gère le compromis entre biais et variance. Si υ est trop grand, le biais sera élevé. S'il est trop petit, la variance sera trop forte. Il faut donc estimer υ, au même titre que les autres paramètres de la fonction de régression.

Des alternatives de pénalisation existent, mais ne nous ont pas séduites. Nous pouvons par exemple citer [Bis92] qui utilise une méthode pénalisant les fonctions à forte courbure, en ajoutant à la fonction de coût la norme du vecteur dérivé seconde de la sortie de la fonction (le score décisionnel de la régression). Cette technique a été spécifiquement développée pour être intégrée à des réseaux de neurones, pour lesquelles l'interopérabilité entre le *weight decay* et l'algorithme standard de construction du réseau (rétropropagation du gradient) pose des problèmes.

Une autre famille d'algorithmes connue sous le nom de l'*early stopping* [MB89] permet de s'affranchir d'un terme de pénalisation. Les algorithmes de cette classe de méthodes arrêtent prématurément l'apprentissage, même si la fonction de coût n'a pas convergé vers un minimum. Nous ne retiendrons pas cette technique, car même si elle permet parfois d'obtenir de bons résultats, elle se fonde sur une partition arbitraire de la base des exemples en une base d'apprentissage et une base de validation ; la répartition des données dans ces deux partitions n'est pas traitée et peut nuire à la reproductibilité des résultats.

• Gestion *a posteriori*

Une autre manière de contrôler le surajustement est de s'intéresser aux performances de généralisation. L'évaluation se fait après la construction de celui, elle est dite *a posteriori*.

Ces techniques possèdent toutes la même philosophie : il s'agit de comparer des modèles entre eux et d'en conserver le meilleur. Il en existe de nombreuses, telles que les tests d'hypothèses [Guj04] ou le critère d'information [AK99]. Nous détaillons dans cette sous-section la

méthode de validation croisée [PC84].

Cette méthode (illustrée dans la figure 3.16) consiste, comme pour l'*early stopping*, à partager la base d'apprentissage, afin d'estimer les performances en utilisant des exemples n'ayant pas servis à l'élaboration du modèle. La base est divisée en Q parties de tailles égales. Q apprentissages sont ensuite réalisés, chacun d'eux laissant de côté une des parties qui servira à valider le modèle courant. Un critère $\mathcal{E}$ d'évaluation est utilisé pour récupérer le score de validation de chaque apprentissage. La performance de généralisation du modèle est calculée en réalisant la moyenne $\mathcal{M}_{\mathcal{E}}$ (modifiée selon le critère) des Q scores de validation précédents. Quelques exemples de critères :

- l'erreur $\mathcal{E} = (f(x) - y)$ qui génère une moyenne $\mathcal{M}_{\mathcal{E}} = \frac{1}{Q}\sum \mathcal{E}$
- l'erreur absolue $|\mathcal{E}| = |f(x) - y|$ qui génère une moyenne $\mathcal{M}_{|\mathcal{E}|} = \frac{1}{Q}\sum |\mathcal{E}|$
- l'erreur quadratique $\mathcal{E}^2 = (f(x) - y)^2$ qui génère une moyenne $\mathcal{M}_{\mathcal{E}^2} = \frac{1}{Q}\sum \mathcal{E}^2$

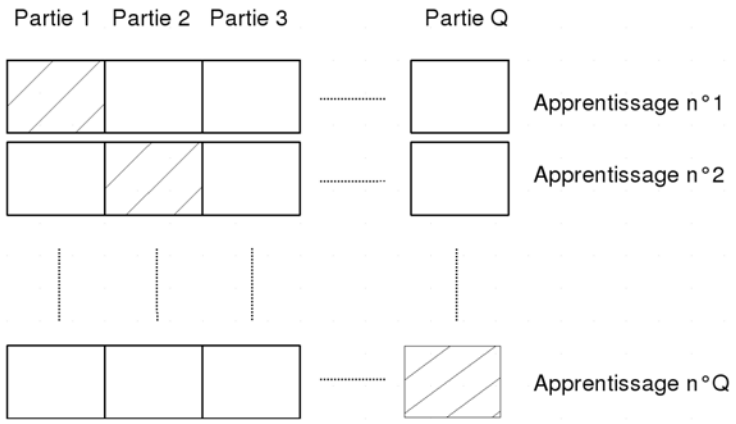


FIGURE 3.16 – Illustration du principe de validation croisée. Les parties blanches sont utilisées pour l'apprentissage, les hachurées pour la validation

Un cas particulier de la validation croisée s'appelle le *leave-one-out*. Cette technique consiste à valider chaque phase d'apprentissage par une partie contenant un exemple unique. Dans ce cas extrême, le nombre d'exemples est égal à Q.

• Autre technique

Lorsqu'une estimation de la valeur d'un paramètre statistique est recherchée, il est de coutume de définir un intervalle appelé **intervalle de confiance** qui contient, avec un certain degré de confiance, cette estimation. Ces intervalles sont de fait étroitement liés à la notion de performance de généralisation du modèle de régression.

Ils peuvent être calculés de différentes façons :
- de manière analytique [SW05],
- suivant des techniques fondées sur l'approche bayésienne et le *weight decay* [DVSSU98],

– par des méthodes, type *boostrappng* [ET93].

3.5 Conclusion

La vision que nous avons de l'organisation de contenus aidée par la similarité nous a poussés à définir un modèle numérique de similarité. Il servira de cadre à l'élaboration d'un moteur d'apprentissage dédié à la problématique qui nous incombe.

Nous avons pensé le cœur de ce modèle comme un organe de nature statistique : à l'aide d'une régression univariée, nous souhaitons construire une mesure de similarité qui nous permet de rester en accord avec nos contraintes structurelles. Cette proposition détermine la notion de similarité comme un phénomène de nature prédictible.

Nous avons de ce fait suggéré une méthode qui nous autorise à intégrer la régression grâce à des techniques provenant des domaines des statistiques et de l'apprentissage (semi-)supervisé. Celle-ci, aidée du formalisme défini dans le chapitre précédent, va nous permettre par la suite de construire un prototype et d'évaluer la pertinence des choix que nous avons faits.

Chapitre 4

Prototype et expérimentations

Sommaire

4.1 Introduction

Le formalisme visuel d'expression des similarités, que nous avons caractérisé dans le deuxième chapitre (section 2.3), nous permet de récupérer l'information de similarité sous forme de distances entre les différents contenus, une distance étant une fonction de dissimilarité propre.

Nous cherchons à construire, grâce à la procédure explicitée dans le chapitre précédent (section 3.4.1), un modèle de régression dont la méthode associée est une fonction de dissimilarité qui s'inspire de ces valeurs de distance.

Nous décrivons dans ce chapitre la structure de notre prototype dédié à l'organisation de contenus. Sa réalisation a été orientée afin de répondre au mieux aux contraintes formalisées dans l'introduction (section 2.4). Nous étudierons son comportement sur plusieurs séries d'expériences, mettant ainsi en pratique les principaux points théoriques abordés dans ce manuscrit.

4.2 Modèle numérique de similarité expérimental

Sous l'hypothèse $\mathscr{C}_0$ selon laquelle la tâche organisationnelle suggérée par l'utilisateur est effectivement modélisable, nous choisissons d'utiliser un modèle numérique de similarité pour ce faire.

Cette section décrit en détail les différentes briques qui composent l'architecture de notre modèle. Nous avons suivi le plan fourni par la procédure de création du chapitre précédent. Nous y expliquons pourquoi nous avons choisi tous ces éléments en fonction des différentes contraintes auxquelles nous devons nous soumettre (voir le tableau 4.2 pour un récapitulatif), ainsi que la manière dont nous les avons assemblés.

4.2.1 Notations

Nous appelons **corpus documentaire** l'ensemble des M contenus présents sur l'espace dynamique lorsque la procédure d'apprentissage est initiée par l'utilisateur à l'instant t. À cet instant, le nombre de classes formées est c' (c est l'indice de classe tel que $C_c \in \{C_0, \ldots, C_{c'}\}$).

Compte tenu du fait que seuls deux niveaux de grains consécutifs sont présents sur l'interface à un instant donné (voir section 2.3.2.7), nous allégeons les notations liées au formalisme de la granularité (définition 1.1) : un contenu $A(n)_j$ sera noté A_j et le représentant $A(n+1)_c$ d'une classe C_c sera noté R_c.

De manière à qualifier les contenus non classés, nous leur affectons une classe arbitraire C_0, constituée des contenus libres non classés et des représentants des c' classes. Nous pouvons alors harmoniser les notations en faisnat l'amalgame, illustré par la figure 4.1, de C_0 avec une classe conventionnelle, en considérant que :

- les représentants (R1, R2 et R3 sur la figure) des autres classes deviennent des ancrés classés (AC) de la classe C_0 ;
- les contenus libres et non classés ($\overline{AC}$) (de A1 à A6 sur la figure) deviennent les libres classés ($\overline{A}C$) de la classe C_0 ;

– les contenus ancrés non classés ($A\overline{C}$) (A7, A8 et A9 sur la figure) n'interviennent pas dans C_0.

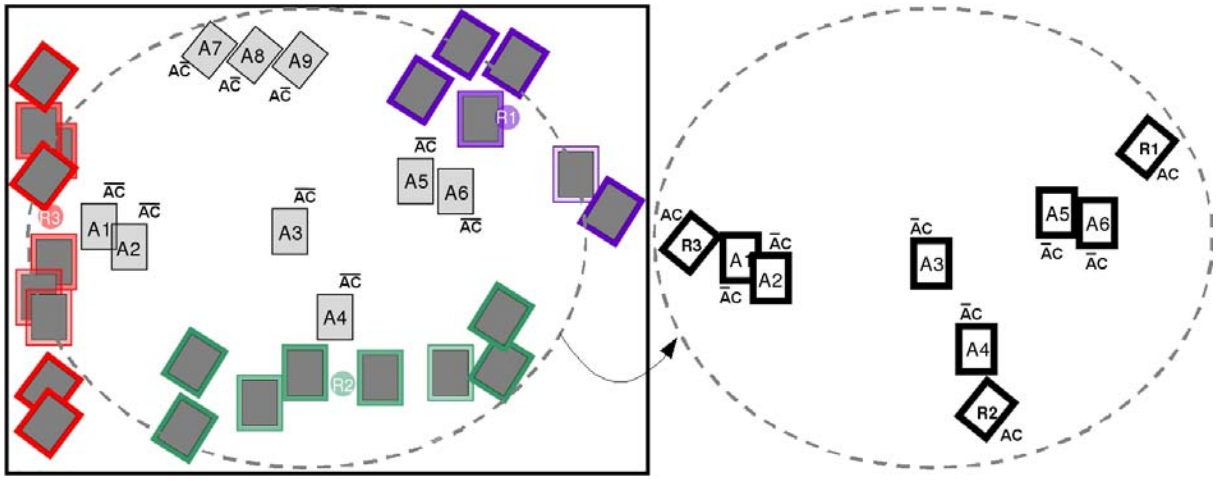


FIGURE 4.1 – Amalgame entre des contenus non classés et une Classe C_0 (de couleur noire)

Un **corpus d'apprentissage** est relatif à une classe C_c, $0 \leq c \leq c'$, et se compose des M_c contenus ancrés de C_c.

Chaque classe C_c porte avec elle le bagage nécessaire pour apprendre une mesure de similarités, symbolisé par le couple (x_c, y_c). Cela signifie qu'à l'instant t il nous faut créer c' couples de variables endogène et exogène.

Lorsqu'il sera spécifié que nous parlons d'un seul corpus d'apprentissage, nous traitons une classe unique de manière à généraliser le propos. Les notations s'en trouveront simplifiées et l'indice c relatif à la classe sera momentanément oublié.

4.2.2 Type de fusion

L'intérêt de travailler avec un modèle numérique de similarité réside dans l'aptitude naturelle de ce modèle à s'abstraire des informations qualitatives portant sur le contenu que nous souhaitons traiter, et de travailler à même les données brutes. Nous ne souhaitons pas y introduire une forme quelconque de qualification en structurant nos descriptions à la manière de la fusion tardive avant de les introduire dans le moteur d'apprentissage.

Nous préférons considérer sur un pied d'égalité l'ensemble de nos descriptions et préserver ainsi cet aspect « objectif » auquel nous sommes attachés (c.f. contrainte $\mathscr{C}_{6.1}$: le système doit accepter tout type de valeur numérique extraite de manière automatique). Le schéma de fusion adopté est le schéma précoce.

4.2.3 Fonction de normalisation

Pour toutes nos variables, nous avons utilisé la fonction de normalisation $MinMax$ (équation 3.11).

4.2.4 Variables endogènes

Les variables endogènes interprètent la tâche organisationnelle sous forme de similarités numériques, répondant ainsi à la contrainte $\mathscr{C}_1$ (utiliser la similarité pour interpréter une tâche). Nous précisons dans cette section le procédé de récupération et les différentes transformations subies par ces variables, pour conclure par une définition formelle de celles-ci.

4.2.4.1 Nature

Considérons un corpus d'apprentissage particulier. L'utilisateur positionne les M contenus de ce corpus sur l'espace dynamique. Ce faisant il fournit un jeu d'apprentissage, sous forme d'une matrice carrée Λ, appelée matrice des distances :

Soit $E = \{A_1, \ldots, A_M\}$ l'ensemble des M contenus du corpus d'apprentissage. Pour tout $(A_i, A_j) \in E^2$, le terme Λ_{ij} situé à la ligne i et à la colonne j de Λ est défini par la distance euclidienne $\delta(A_i, A_j)$ calculée entre les deux contenus A_i et A_j sur l'espace dynamique :

$$\Lambda = \begin{pmatrix} \Lambda_{11} & \cdots & \Lambda_{1M} \\ \vdots & \ddots & \vdots \\ \Lambda_{M1} & \cdots & \Lambda_{MM} \end{pmatrix} = \begin{pmatrix} \delta(A_1, A_1) & \delta(A_1, A_2) & \cdots & \delta(A_1, A_M) \\ \delta(A_2, A_1) & \delta(A_2, A_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \delta(A_{M-1}, A_M) \\ \delta(A_M, A_1) & \cdots & \delta(A_M, A_{M-1}) & \delta(A_M, A_M) \end{pmatrix} \tag{4.1}$$

De par les propriétés de la distance euclidienne, cette matrice carrée est symétrique, à diagonale nulle, et ses valeurs sont toutes positives. Nous ne nous intéresserons qu'à sa partie triangulaire supérieure, c'est à dire aux Λ_{ij} avec i et j entiers positifs, tels que $1 \leq i < j \leq M$. Nous ne considérerons donc que $N = \frac{M(M-1)}{2}$ valeurs de distances.

Illustrons cela grâce à la figure 4.2, qui reprend l'espace dynamique de l'exemple de la figure 4.1 en mettant l'accent sur le corpus d'apprentissage de deux classes différentes.

- Concernant les contenus la classe C_0, l'apprentissage se fait sur les représentant R1, R2 et R3 des classes respectives C_1 (la bleue), C_2 (la verte) et C_3 (la rouge).

Dans ce cas M_0=3, $N_0 = 3$ et $\Lambda_0 = \begin{pmatrix} 0 & \delta(R1, R2) & \delta(R1, R3) \\ & 0 & \delta(R2, R3) \\ & & 0 \end{pmatrix}$

Une fois la mesure apprise, elle servira à repositionner les contenus A1, A2, A3, A4, A5 et A6 (les libres de C_0).

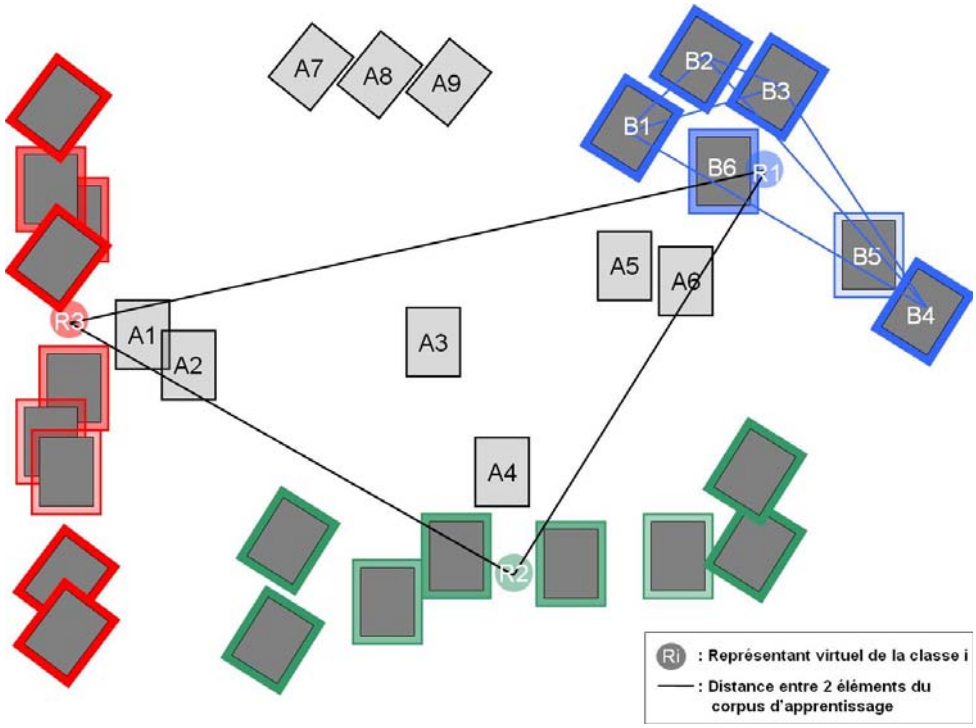


FIGURE 4.2 – Illustration de la construction des matrices de distances relatives aux classes C_0 et C_1

- Concernant la classe C_1 (de couleur bleue), l'apprentissage se fait sur les ancrés classés de la classe (B1, B2, B3 et B4).

Dans ce cas M_1=4, $N_1 = 6$ et $\Lambda_1 = \begin{pmatrix} 0 & \delta(B1,B2) & \delta(B1,B3) & \delta(B1,B4) \\ & 0 & \delta(B2,B3) & \delta(B2,B4) \\ & & 0 & \delta(B3,B4) \\ & & & 0 \end{pmatrix}$

Une fois la mesure apprise, elle servira à repositionner les contenus B5 et B6 (les libres de C_1).

Nous allons construire des variables endogènes qui s'inspireront directement de cette matrice de distances.

4.2.4.2 Normalisation

Pour une classe donnée, à laquelle correspond une variable endogène, la partie triangulaire supérieure de la matrice des distances Λ est projetée en un vecteur λ de dimension N :

$$\Lambda = \begin{pmatrix} 0 & \Lambda_{12} & \cdots & \Lambda_{1j} & \cdots & \Lambda_{1M} \\ & \ddots & \ddots & \vdots & & \vdots \\ & & \ddots & \Lambda_{ij} & \cdots & \Lambda_{iM} \\ & & & \ddots & \ddots & \vdots \\ & & & & \ddots & \Lambda_{(M-1)M} \\ & & & & & 0 \end{pmatrix} \Longrightarrow \lambda = \begin{pmatrix} \Lambda_{12} \\ \Lambda_{13} \\ \vdots \\ \Lambda_{ij} \\ \vdots \\ \Lambda_{(M-2)M} \\ \Lambda_{(M-1)M} \end{pmatrix} \tag{4.2}$$

Les composantes du vecteur λ sont ensuite normalisées par MinMax :

$$\overline{\lambda} = (\overline{\Lambda_{12}}, \overline{\Lambda_{13}} \ldots, \overline{\Lambda_{ij}}, \ldots, \overline{\Lambda_{(M-2)M}}, \overline{\Lambda_{(M-1)M}}) \tag{4.3}$$

4.2.4.3 Définition

Pour une classe donnée, les variables endogènes y_i, $i \in \{1, \ldots, N\}$, associées au corpus d'apprentissage sont égales aux valeurs des distances normalisées $\overline{\lambda_i}$ issues de la matrice des distances Λ :

$$y_1 = \overline{\Lambda_{12}}, \ldots, y_N = \overline{\Lambda_{(M-1)M}} \tag{4.4}$$

4.2.5 Variables exogènes

Pour construire les variables exogènes, nous utilisons les valeurs descriptives « bas niveau ». Elles proviennent des D détecteurs audio ou vidéo présentés dans le chapitre 2, et sont extraites de manière automatique sur les M contenus du corpus documentaire. Nous décrivons dans cette section le traitement que nous leur faisons subir afin de les intégrer dans le moteur d'apprentissage, pour finir par leur définition formelle.

4.2.5.1 Nature

Nous avons focalisé notre étude expérimentale sur des séries temporelles : une série des valeurs extraites ponctuellement tout au long du contenu. Cependant, la dimension temporelle de ces séries nous pose un réel problème.

Le principal intérêt de travailler sur la similarité existant entre ces séries porte sur la propriété de linéarité du temps (voir [Hai05] pour un état de l'art complet du sujet). Nous sommes toutefois soumis à deux contraintes qui ne permettent pas d'exploiter directement cette propriété :

1. le système doit accepter tout type de valeur numérique extraite de manière automatique (contrainte $\mathscr{C}_{6.1}$) : nous souhaitons pouvoir intégrer d'autres valeurs descriptives, comme

pourraient l'être des dates, des *tags*[22] numériques extraits de sites communautaires, ou encore l'âge minimum conseillé pour visionner l'ouvrage ;

2. le modèle doit gérer les changements de granularité (contrainte $\mathscr{C}_3$) : des grains documentaires tels que le locuteur ou l'intervenant sont des regroupements de segments temporels audiovisuels, et ne sont donc pas linéaires dans le temps.

Pour répondre à ces deux contraintes, nous avons préféré ne pas exploiter directement cette dimension temporelle, et avons ainsi pris en considération des valeurs représentatives. Nous avons choisi d'utiliser la **valeur du minimum**, la **valeur du maximum**, la **valeur moyenne** et la **variance** de ces séries.

Pour un corpus d'apprentissage donné, nous appelons une **description** l'ensemble des valeurs issues d'une de nos caractéristiques « bas niveau », restreintes par l'une des quatre fonctions précitées. Citons comme exemple de description le minimum de la luminosité moyenne, la variance de la modulation de l'entropie, ou encore le maximum du taux d'activité d'une vidéo. Pour D caractéristiques étudiées sur un contenu, nous obtenons $D' = 4D$ descriptions de ce contenu.

Se pose alors la question des c' représentants de classes, qui constituent le corpus d'apprentissage de la classe C_0. Comme dit dans la section 2.3.2.3 du chapitre consacré sur la visualisation, « le représentant doit témoigner de l'ensemble des valeurs descriptives provenant des contenus ancrés qui composent sa classe ». Pour ce faire, nous avons choisi d'exploiter la moyenne des valeurs descriptives spécifiques aux entités ancrées constituant la classe : chaque description du représentant de la classe C_c est égale à la moyenne des D' descriptions de même nature provenant des M_c contenus du corpus.

En conclusion, une variable exogène se présentera sous la forme d'un vecteur de descriptions.

4.2.5.2 Normalisation

Soient D_i une description ($i \in \{1, \ldots, D'\}$) et A_j un contenu du corpus d'apprentissage ($j \in \{1, \ldots, M\}$). La figure 4.3 illustre l'ambivalence de notre système :

1. d'un point de vue documentaire, chaque A_j est représenté dans l'espace de description par un vecteur $a_j = \{d_{1,j}, \ldots, d_{D',j}\}$ constitué de ses D' valeurs descriptives, chacune issue d'une description D_i différente. Nous appelons a_j « vecteur du contenu A_j » (les lignes de la matrice Mat de la figure 4.3) ;
2. d'un point de vue descriptif, chaque D_i existe à travers son vecteur $d_i = \{d_{i,1}, \ldots, d_{i,M}\}$, constitué des valeurs descriptives des M contenus du corpus d'apprentissage. Nous appelons d_i « vecteur de la description D_i » (les colonnes de la matrice Mat de la figure 4.3).

Nous avons également opté pour une normalisation $MinMax$. Nous appelons **vecteur normalisé** du contenu A_j, que nous écrivons par abus de notation $\overline{a_j}$, le vecteur $\{\overline{d_{1,j}}, \ldots, \overline{d_{D',j}}\}$,

22. *tag* : marqueur sémantique ou lexical utilisé sur les sites dits « réseaux sociaux ».

avec chaque $\overline{d_{i,j}}$ provenant de la normalisation $MinMax$ de la description D_i. Sur la figure 4.3, cela s'illustre sur la matrice normalisée $\overline{Mat}$: pour un vecteur ligne donné (un a_j), chacune de ses coordonnées provient d'une colonne qui a été normalisée indépendamment des autres.

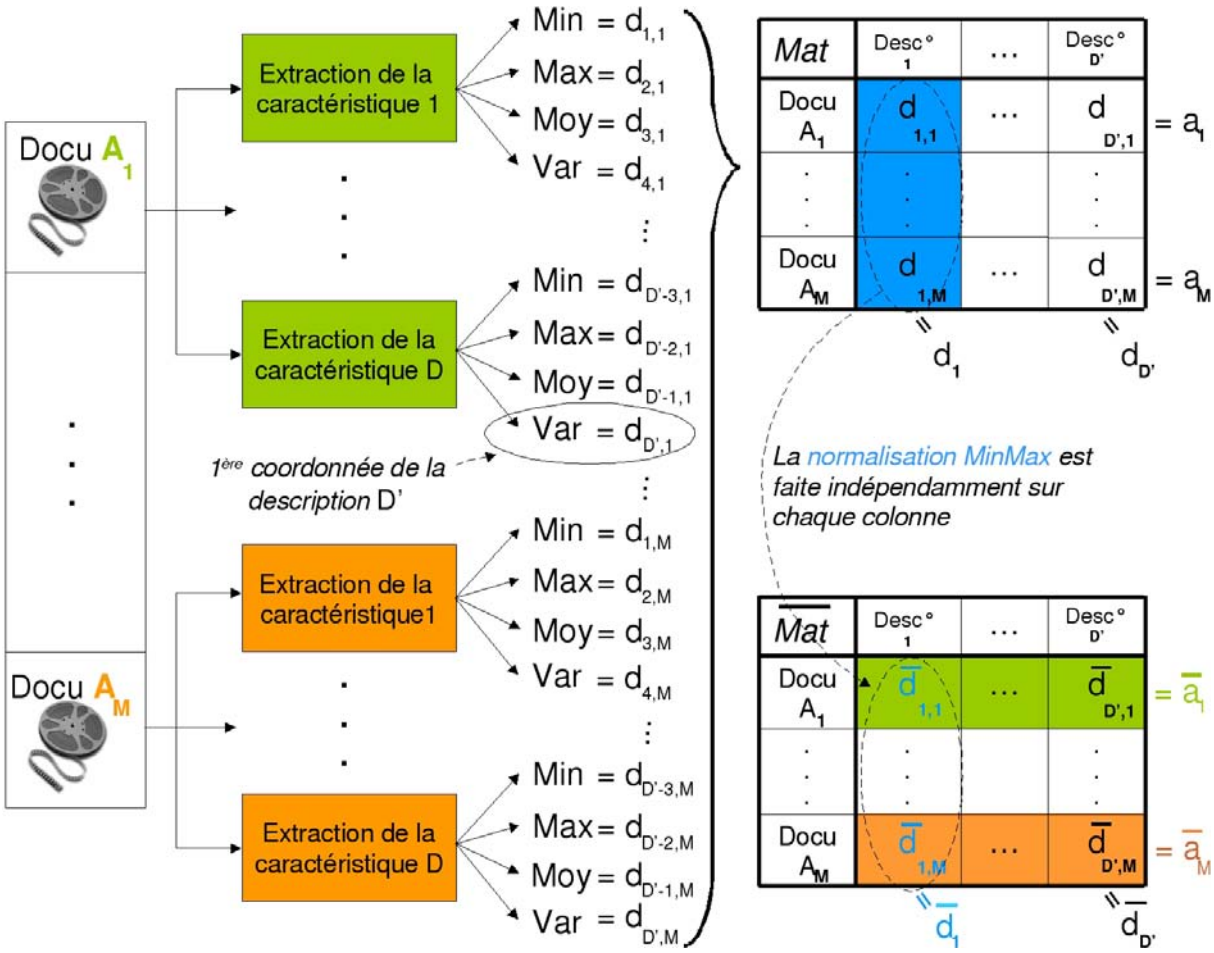


FIGURE 4.3 – Procédure d'extraction et de normalisation des variables exogènes

Pour deux contenus A_k et A_l, le but de notre modèle est de mettre en relation la distance $\delta(A_k, A_l)$ avec les vecteurs de contenus a_k et a_l. Nous avons choisi d'utiliser l'opérateur « $\oplus$ » de concaténation sur les vecteurs de contenus normalisés pour créer notre variable exogène. Pour mémoire, l'opérateur de concaténation se définit ainsi :

$$\begin{gathered}\forall (v, w) \in \mathbb{R}^m \times \mathbb{R}^n \ / \ v = \{v_1, \ldots, v_m\} \text{ et } w = \{w_1, \ldots, w_n\}, \\ v \oplus w = \{v_1, \ldots, v_m, w_1, \ldots, w_n\}\end{gathered} \tag{4.5}$$

Notez que la construction de ce vecteur doit se faire en mettant en correspondance les indices des contenus avec les indices de l'équation 4.4 de la valeur endogène (pour faire correspondre une mesure de distance calculée entre deux objets informationnels avec la paire de vecteurs de contenus correspondante).

4.2.5.3 Définition

Pour une classe donnée, soient $y_1, \ldots, y_N$ les variables endogènes associées au corpus d'apprentissage.

Les variables exogènes $x_1, \ldots, x_N$ correspondantes sont constituées des N concaténations de vecteurs normalisés $\overline{a_j}$, vecteurs issus des M contenus A_j du corpus d'apprentissage.

Pour chaque x_i, les indices k et l des vecteurs concaténés $\overline{a_k}$ et $\overline{a_l}$ correspondants, sont les mêmes indices k et l que pour $y_i = \overline{\Lambda_{kl}}$:

$$x_1 = (\overline{a_1} \oplus \overline{a_2}), x_2 = (\overline{a_1} \oplus \overline{a_3}), \ldots, x_N = (\overline{a_{(M-1)}} \oplus \overline{a_M}) \tag{4.6}$$

4.2.6 Remarques

4.2.6.1 Sur la normalisation des variables

Au-delà de son aspect normatif, cette méthode de normalisation abonde dans le sens de l'utilisateur omnipotent. Prenons comme exemple l'acquisition des variables endogènes lorsque la procédure d'apprentissage vient d'être lancée (le même raisonnement peut être tenu pour les variables exogènes).

Si nous considérons l'ensemble de la base documentaire, les extrema obtenus sont locaux : il va de soi que, n'ayant pas consulté tous les contenus de la base, les distances maximum et minimum définies par l'utilisateur sur son corpus d'apprentissage à un instant t ne peuvent refléter la totalité des mesures de la base. Il peut alors être reproché à cette méthode son manque de représentativité, car l'apprentissage résultant ne pourra extrapoler au-delà de ces informations locales une réelle valeur globale maximum ou minimum de la variable endogène.

Toutefois, ce potentiel défaut n'en est pas un pour nous, car il cadre avec notre volonté de placer l'utilisateur au centre du processus. Sous l'hypothèse que les contenus servant de corpus d'apprentissage ont plus de chance d'être visionnés par l'utilisateur que les contenus qui subiront la mesure de similarités, il est tout à fait légitime de leur attribuer plus de crédit. Nous concevons de ce fait les extrema évoqués lors de l'apprentissage à l'instant t comme globaux au regard du corpus manipulé, plutôt que comme locaux sur l'ensemble de la base.

Un nouveau contenu ne peut intervenir dans l'élaboration de la mesure que s'il est ancré classé (AC), c'est-à-dire explicitement caractérisé par l'utilisateur. Toutefois, même si sa position est contrainte par les extrema calculés à l'instant t, elle n'est pas aberrante. La figure 4.4 illustre ce constat. Son contexte est le suivant :

- les objets informationnels sont des films ;
- tous ces contenus appartiennent à une même classe ;
- la tâche organisationnelle est un ordonnancement des films en fonction des émotions qu'ils ont procurés à l'utilisateur (plus il a été ému, plus les documents sont à gauche).

Au temps $t = t_0$ (figure 4.4a), seuls les contenus A,B et C sont présents. Le film jugé le plus fort en émotions à cet instant par l'utilisateur est le A. La mesure est calculée et le système propose deux nouveaux contenus D et E, qu'il positionne proche de A : une interprétation possible de cette réponse est que le système suggère D et E plus émouvants que B et C.

L'utilisateur peut se contenter de ce résultat, ou considérer que la proposition du système implique aussi que D est moins fort en sensations que A, ce qu'il réprouve. Il décide alors d'affiner son ordonnancement et ancre D plus à gauche que A, créant ainsi une nouvelle mesure de similarités dont les extrema globaux au temps $t = t_1$ remettent en question les précédents (la flèche verte de la figure 4.4b). Fort de cette nouvelle information, le document E est autorisé à franchir l'ancienne barrière de l'extremum imposé par A au temps $t = t_0$, et se retrouve positionné par le système entre A et D au temps $t = t_1$ (la flèche rouge de la figure 4.4b).

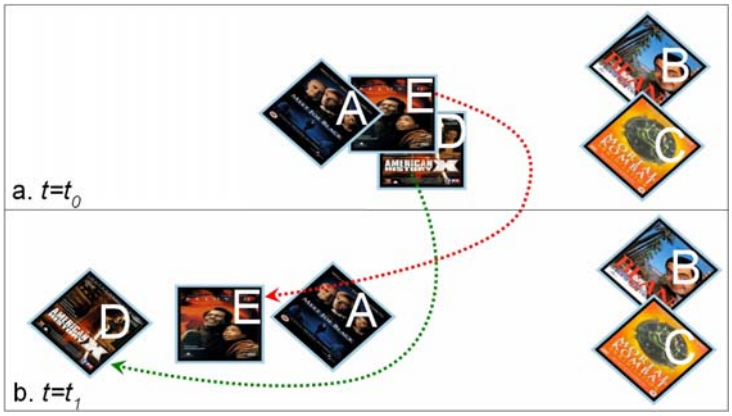


FIGURE 4.4 – Illustration du principe des extrema globaux à l'instant t

C'est une volonté de ne pas inférer à l'instant $t = t_0$ une mesure permettant d'outrepasser les extrema calculés. L'aide que nous apportons doit rester dans le cadre des limites des connaissances de l'utilisateur : nous suggérons mais n'imposons pas notre point de vue (section 2.3.2.1).

4.2.6.2 Sur la cardinalité du corpus d'apprentissage

Nous voyons ici une particularité de notre système. Nous travaillons sur des comparaisons entre éléments :

- si $M = 0$ ou $M = 1$, aucun calcul de distances n'est possible, et l'apprentissage ne peut avoir lieu ;
- si $M = 2$, avec A_1 et A_2 les éléments de ce corpus, $min(\delta(A_1, A_2)) = max(\delta(A_1, A_2))$ et la normalisation n'a pas de sens (division par zéro).

Un apprentissage n'a donc d'intérêt que pour $M \geq 3$. Ce cas est aussi valable pour la classe C_0, et implique que des éléments $\overline{AC}$ ne pourront être affectés par une mesure de similarités que si au moins trois classes existent à l'instant t.

4.2.7 Choix du modèle de régression

4.2.7.1 Une fonction non-linéaire

Notre volonté d'utiliser toute forme de donnée numérique sans conserver d'information sur leur comportement (contrainte $\mathscr{C}_{6.1}$) ne nous permet pas d'anticiper sur le type de dépendances qui pourrait exister entre l'agencement des contenus et leurs descriptions (autrement formulé, entre variables endogènes et exogènes). Nous devons accorder un maximum de flexibilité à notre espace d'hypothèses, et emploierons pour ce faire une fonction non-linéaire afin de piloter notre modèle.

4.2.7.2 Nature de la fonction

Nous avons choisi d'utiliser la régression par machines à vecteur de support, du fait de ses nombreux avantages :

- la rapidité de l'apprentissage en fait un candidat parfait pour notre contrainte $\mathscr{C}_2$;
- la ε-SVR est, par définition, une méthode parcimonieuse, car sa paramétrisation ne se fait qu'en utilisant un échantillon réduit de valeurs suffisamment pertinentes aux yeux de ε (les vecteurs de support). Cette propriété peut être vue comme un élément de réponse à la contrainte $\mathscr{C}_{7.2}$ portant sur l'imprécision apportée par l'utilisateur lors de l'agencement des objets sur l'espace dynamique : seules les distances réellement significatives interviendront dans le modèle, permettant ainsi de potentiellement réduire une partie du bruit apporté en entrée ;
- l'algorithme est la solution d'un problème régularisé, ce qui permet d'aborder le problème de sur-apprentissage de manière directe, à travers le paramètre de régularisation et la technique du *weight decay* ;
- la popularité de cette technique a permis son implémentation efficace dans de nombreuses bibliothèques de développement, citons parmi les plus utilisées *lib-svm* [23] et *svm-light* [24].

Ce choix détermine le contexte du modèle de régression :

- l'espace d'hypothèses est supposé préhilbertien. Son équation 3.18 est celle de l'espace $\mathcal{H}$ défini dans la section 3.4.3.2, que nous formalisons de la façon suivante :

$$\mathcal{H} = \left\{ f : f(x) = \sum_{i=1}^{n} v_i K(x, x'_i), \text{tel que } v = \{v_1, \ldots, v_n\} \text{ et } v_i \in \mathbb{R} \right\} \tag{4.7}$$

  avec x'_i les vecteur supports et v_i leurs coefficients.
- la fonction de perte associée est L_ε ;
- le terme de régulation, défini par l'expression de son espace d'hypothèse, s'écrit :

$$Reg(f) = \sum_{i,j} v_i v_j K(x_i, x_j) \tag{4.8}$$

23. http ://www.csie.ntu.edu.tw/ cjlin/libsvm/
24. http ://svmlight.joachims.org/

Nous avons choisi d'utiliser un noyau gaussien, qui se comporte le mieux dans de nombreux problèmes étudiés. Nous rappelons ici son équation :

$$K(x_1, x_2) = e^{-\gamma \frac{\|x_1 - x_2\|^2}{2}} \text{ avec } \gamma = 1/\sigma^2 \text{ et } \sigma > 0 \tag{4.9}$$

Le critère de performance choisi pour le modèle est l'erreur quadratique moyenne :

$$EQM(f) = \sum_{i=1}^{N} \frac{(f(x_i) - y_i)^2}{N} \tag{4.10}$$

4.2.7.3 Choix des hyper-paramètres

Plusieurs paramètres dépendent directement du choix de la ε-SVR, et entrent en compte dans le processus global d'apprentissage. Pour ne pas les confondre avec les paramètres θ de l'expression générale de f, ces attributs sont appelés hyper-paramètres.

L'hyper-paramètre associé au noyau gaussien est γ [25]. Ceux associés à la fonction de régression sont le coût $\mathbb{C}$ et la tolérance ε.

De nombreuses techniques de recherche existent dans la littérature (voir [FI05]). Nous avons opté pour les choix suivants :

1. $\mathbb{C}$ et γ sont recherchés par exploration de grille ou *grid search* : nous testons tous les couples possibles, chaque hyper-paramètre décrivant, un pas donné, un intervalle arbitraire. Nous avons choisi de faire varier les paramètres de manière exponentielle et d'utiliser deux types de grilles [Sta02] :

   (a) une grille initiale, qualifiée de « large », qui utilise un pas de 2×2 :
   $log_2(\mathbb{C}) = -5, -3, \ldots, 15$ et $log_2(\gamma) = -15, -13, \ldots, 3$.

   (b) une fois le couple optimal $(\mathbb{C}^*, \gamma^*)$ trouvé par grille large, nous construisons une grille « réduite » au voisinage du couple, avec un pas de $2^{\frac{1}{4}}$) :
   $log_2(\mathbb{C}) = (log_2(\mathbb{C}^*) - 2), (log_2(\mathbb{C}^*) - 1.75), \ldots, (log_2(\mathbb{C}^*) + 2)$
   et $log_2(\gamma) = (log_2(\gamma^*) - 2), (log_2(\gamma^*) - 1.75), \ldots, (log_2(\gamma^*) + 2)$.

2. Concernant ε, nous avons opté pour une valeur arbitraire, fixée à 0,1. Des techniques permettent de contrôler ε : la ν-SVR [SSWB00] par exemple, détermine le nombre de vecteurs supports utilisés pour l'apprentissage (et donc indirectement la largeur du tube), en faisant varier l'hyper-paramètre ν entre 0 (aucun vecteur d'apprentissage n'est utilisé comme support) et 1 (tous les vecteurs d'apprentissage sont des vecteurs supports). Il est également possible d'intégrer la recherche de ε dans la grille.

   Toutefois, prendre en compte l'optimisation de cet hyper-paramètre ralentit considérablement le calcul de la fonction de régression.

25. Si l'analogie avec une fonction gaussienne est faite, γ correspond à l'inverse du carré de la dispersion des données.

L'intégration de ces hyper-paramètres dans le processus global d'apprentissage n'entre pas en compte dans l'optimisation du problème convexe caractérisé par la maximisation de la marge. Ils sont à déterminer *a priori* et ce choix à une grande incidence sur les performances du système.

4.2.8 Algorithme de réduction de la dimensionnalité

Nous avons choisi de ne pas utiliser d'algorithme d'extraction d'attributs, pour ne pas « déformer » l'espace de description. Cela nous permet de continuer à en tirer avantage pour caractériser l'influence de telle ou telle description en fonction du contenu et de la tâche.

Nous avons opté pour l'algorithme séquentiel sous-optimal SFFS (chapitre 3 section 3.4.6.3) :
- cet algorithme détermine de lui-même le nombre de passes avant et arrière à effectuer.
- l'échafaudage itératif de la variable exogène par l'ajout ou la suppression de valeurs descriptives permet de théoriquement garder à l'écart des descriptions qui viendraient perturber le modèle de régression (contrainte $\mathscr{C}_{7.2}$) ;
- nous l'utilisons comme un *wrapper*, ce qui implique que :
  - la fonction d'évaluation est, de fait, la mesure de performances liée à la méthode d'apprentissage, à savoir l'erreur quadratique moyenne ;
  - la méthode d'évaluation est celle utilisée pour déterminer les hyper-paramètres du modèle. Nous avons opté pour la validation croisée (voir [DKP03] pour une évaluation des performances des méthodes les plus courantes). Nous l'avons choisie tripartite car :
    - le nombre de contenus minimum pour démarrer un apprentissage est de trois (voir section 4.2.6.2) ;
    - un grand nombre de parties permet une meilleure évaluation, mais nécessite beaucoup plus de temps de calcul.
- son critère d'arrêt est également l'erreur quadratique moyenne : s'il n'y a plus d'amélioration au sens du critère de performances du modèle, l'algorithme se termine.

Notons que la fonction de régression f est commutative et associative vis-à-vis de l'opérateur de concaténation : $f(a_i \oplus a_j) = f(a_j \oplus a_i)$. Sans cette propriété, il faudrait reconsidérer tous les arrangements possibles de valeurs descriptives qui composent la variable exogène, cet algorithme perdrait alors tout intérêt.

Notons également qu'il s'agit d'un algorithme de sélection d'attributs dit « glouton », l'ajout d'un contenu au corpus d'apprentissage augmente le nombre de variables endogènes de N, ce qui peut conduire à considérablement augmenter le temps de calcul. Nous verrons toutefois que, sous la contrainte $\mathscr{C}_4$ du nombre réduit de contenus, ce temps d'apprentissage reste raisonnable.

4.2.9 Synthèse : algorithme du moteur d'apprentissage

Nous expliquons ici, à travers un algorithme en pseudo-code commenté, comment nous avons implanté notre fonction de régression, résultat du moteur d'apprentissage. Pour plus de lisibilité, nous décrivons une version SFS de cet algorithme : seules les étapes *forward* interviennent dans le processus.

Algorithm 1 Construction de la fonction de régression (version SFS)

Données :
$E = \{A_1, \ldots, A_M\}$ un corpus d'apprentissage relatif à une classe ;
p le numéro de la passe algorithmique, $p \in \{0, \ldots, D'\}$;
$S = \{(x_i^p, y_i) | i = 1, ..., N\}$ les couples de variables endogènes et exogènes correspondants,
tels que $y_1 = \overline{\Lambda_{12}}, \ldots, y_N = \overline{\Lambda_{(M-1)M}}$,
avec $\Lambda_{kl} = \delta(A_k, A_l)$,
et $x_1^p = (\overline{a_1} \oplus \overline{a_2}), \ldots, x_N^p = (\overline{a_{(M-1)}} \oplus \overline{a_M})$,
avec $\overline{a_k} = (\overline{d_{1,k}}, \ldots, \overline{d_{p,k}})$ le vecteur normalisé du contenu A_k,
f^p la fonction de régression construite sur la variable exogène courante ;
EQM le critère d'évaluation, servant pour l'optimisation par grille et comme critère d'arrêt ;

Initialisation :
$x_i^0 = x_i^{D'}$, pour tout i ; *// l'initialisation est faite sur le vecteur de description complet*
Construire f^0 en recherchant $(\mathbb{C}, \gamma)$ optimal sur grille large par validation croisée tripartite ;
$f = f^0$; $p = 1$; $x_i^p = ()$, pour tout i ;

Etape *forward*
Pour chaque description d, $d \in \{1, \ldots, D'\}$:
| $x_i^p = (x_i^p \oplus x_{d,i})$, pour tout i ; *// concaténation de la description d au vecteur x_i^p courant*
| Construire une fonction de régression f_d^p relative à la description d ;
| Calculer $EQM(f_d^p)$ par validation croisée tripartite ;
Conserver $f_{d^*}^p$ telle que $d^* = \arg\min_d EQM(f_d^p)$;
Si $EQM(f_{d^*}^p) > EQM(f)$
| $\mathbb{C}^* = \mathbb{C}$, $\gamma^* = \gamma$;
| Aller à l'**Etape *sortie*** ;
Sinon
| $f = f_{d^*}^p$
| Optimiser f en recherchant $(\mathbb{C}, \gamma)$ sur grille large par validation croisée tripartite ;
| Retirer d^* de l'ensemble des descriptions ; $D' = D' - 1$;
| $p = p + 1$; $x_i^p = (x_i^{p-1} \oplus x_{d^*,i})$, pour tout i ;
| Aller à l'**Etape *forward***

Etape *sortie*
// cas particulier : la fonction d'initialisation est meilleure que la fonction construite avec une description
Si $p = 1$, $x_i^1 = x_i^{D'}$, pour tout i ;

// optimisation des paramètres en sortie
Construire $f = f^p$ en recherchant $(\mathbb{C}^{**}, \gamma^{**})$ optimal sur grille réduite par validation croisée tripartite au voisinage de $(\mathbb{C}^*, \gamma^*)$;

La version implantée pour notre prototype utilise la variante flottante de cette méthode (la SFFS, explicitée dans l'annexe A de ce manuscrit). Pour obtenir la version SFFS, il est

nécessaire de rajouter les étapes *backward* destinées à éliminer les descriptions parasites. Dans ce cas, le nombre de descriptions conservées n'est pas forcément égal au nombre de passes algorithmiques, comme dans la version SFS. Lorsque nous évoquerons cet algorithme par la suite, nous sous-entendrons qu'il s'agit de sa version SFFS.

La particularité de cet algorithme concerne les phases d'optimisation des hyper-paramètres de la fonction de régression :

- les hyper-paramètres sont initialisés sur le vecteur de description complet (sans sélection d'attributs).
- nous n'opérons une nouvelle phase d'optimisation que si l'apport ou le retrait d'une description améliore les performances, au sens du critère d'évaluation.

Sa complexité est polynomiale ($O(N^2)$). Cet algorithme fournit une réponse en temps acceptable (contrainte $\mathscr{C}_2$), pour un nombre relativement faible de contenus utilisés pour l'apprentissage (contrainte $\mathscr{C}_4$) (pour plus de détails, voir l'évaluation 4.5.1).

4.3 Autres choix technologiques

Nous présentons dans cette section les autres principaux choix technologiques que nous avons pris pour concevoir notre prototype. Ceux-ci concernent les mécanismes centraux d'interaction avec l'utilisateur et le moteur de visualisation des similarités sous forme de distances.

4.3.1 Interaction

4.3.1.1 Initialisation

Utilisé en complément de l'interface d'exploration de l'Ina « Archives pour tous », les contenus choisis pour initialiser le système sont les documents consultés par l'utilisateur et mis dans l'historique de navigation (le bordereau de la figure 1, vignette ⑨), sous contrainte qu'il y en ait un minimum de trois (voir section 4.2.6.2).

Dans d'autres cas (moins de trois documents sélectionnés avec l'interface « Archives pour tous », consultation d'une base personnelle), nous avons choisi de présenter trois contenus pris au hasard dans la base documentaire.

Par défaut, nous choisissons de disposer ces trois premiers contenus dans une classe : supposons que les premiers contenus apparaissent à l'écran non classés ; un apprentissage ne pourra démarrer que si l'un des scénarios suivants est appliqué :

- l'utilisateur regroupe tous les contenus dans une même classe (puis les ancre) ;
- l'utilisateur crée trois classes, une pour chaque contenu.

Nous avons choisi d'anticiper en implantant la première de ces deux alternatives. Même si cela nous fait déroger à la règle du classement uniquement piloté par l'utilisateur, cette option nous semble pertinente car elle lui permet de directement commencer un apprentissage.

La conséquence directe de ce choix est que tant qu'une seule classe C_1 est représentée dans l'espace dynamique, les contenus rapatriés sont automatiquement classés dans C_1. L'intérêt des documents non classés ne pourra être révélé que si un minimum de trois classes sont présentes (nous rappelons que trois classes génèrent trois représentants, assimilés lors de l'apprentissage à des ancrés classés pour une classes C_0 (voir section 4.2.1)).

4.3.1.2 Rapatriement

Dans le but d'aider à organiser des contenus, il est important de ne pas surcharger l'utilisateur d'informations. Présenter trop de contenus en une fois peut nuire à la lisibilité générale.

L'apport incrémental de contenus, ou rapatriement comme nous l'appelions dans le chapitre 2, est la technique que nous avons choisie pour l'exploration progressive de la base de contenus : à chaque procédure d'apprentissage, l'utilisateur est chargé de déterminer s'il veut, ou non, rapatrier un ou plusieurs contenus. L'utilité de ce choix vis-à-vis de notre mesure de similarités sera discuté dans la section 4.4.2.

Ces contenus peuvent êtres sélectionnés par l'utilisateur, ou le choix peut être laissé à la discrétion du système. Nous expliquons ici la stratégie utilisée pour sélectionner automatiquement un contenu. Pour plus de lisibilité, nous considérons qu'il n'y a qu'une classe présente à l'écran et qu'un seul document rapatrié (illustré par la figure 4.5). Soit :

- $S = \{A_1, \ldots, A_M\}$ une base documentaire ;
- $S' = \{A_1, \ldots, A_{M'}\}$ le corpus des contenus ancrés ;
- $S'' = \{A_{M'+1}, \ldots, A_{M''}\}$ le corpus des contenus non ancrés ;
- $S''' = \{A_{M''+1}, \ldots, A_M\} = S \setminus (S' \cup S'')$, le corpus qui n'est pas encore représenté dans l'espace dynamique.

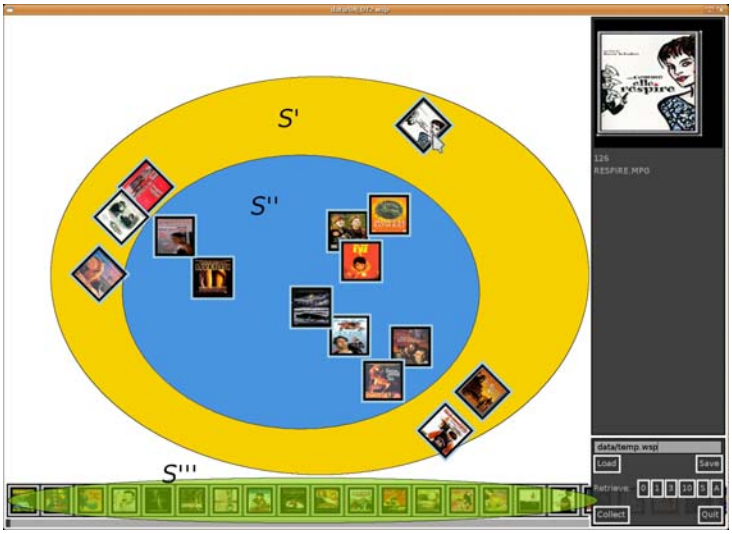


FIGURE 4.5 – Extrait annoté de l'interface du prototype, permettant de visualiser les différents corpus

Soit f la fonction de régression construite sur S'. Nous rapatrions le contenu $A_{j^*} \in S''$ tel que :

$$j^* = \arg\max_j \sum_{i=1}^{M'} \frac{f(a_j \oplus a_i)}{M'} \tag{4.11}$$

Nous considérons la moyenne de la somme des similarités entre un élément A_j et les éléments de S' comme une fonction d'énergie, résultat du processus de prédiction f et dépendante de A_j.

En choisissant l'élément A_j^* qui maximise cette énergie, nous supposons qu'il s'agit de l'élément de S'' qui est le plus susceptible de perturber l'équilibre des similarités calculées, et, par conséquent, des distances représentées. Nous reviendrons sur ce choix dans l'expérience 4.5.4.

Notons que le corpus ainsi obtenu, relatif à une classe et constitué des contenus d'apprentissage, des contenus non ancrés, et des contenus rapatriés entrant en jeu dans la dite classe, s'appelle le **corpus de réorganisation**.

Nous conclurons sur une remarque concernant le temps de calcul nécessaire au rapatriement d'un élément. Ce délai est, bien évidemment, proportionnel au nombre de contenus présents dans la base documentaire, et dépend de la fonction de régression elle-même. Toutefois, à la lumière des nombreuses expériences pratiquées dans cette étude, nous considérons ce temps comme négligeable sur des corpus d'une taille inférieure à 500 contenus [26].

4.3.2 Moteur de visualisation

Nous présentons dans cette section le moteur de visualisation que nous avons implanté. Les différentes notations sont ici propres à celui-ci. De ce fait, la considération du temps est sans rapport avec celle que nous avons évoquée jusqu'alors : elle ne concerne plus les différents lancements d'apprentissages initiés par un utilisateur, mais les différentes passes algorithmiques relatives au modèle d'énergie.

4.3.2.1 Principe

Une fois le corpus de réorganisation établi et la matrice de similarité calculée, il reste à transcrire cette dernière sous la forme d'une matrice de distances pour qu'elle soit visualisable sur l'espace dynamique.

Dans un premier temps, les échelles des similarités sont changées par une normalisation $MinMax$ inverse (notée $MinMax^{-1}$), afin de faire coïncider le maximum (resp. minimum) des valeurs estimées avec la valeur maximum (resp. minimum) des valeurs de distances du corpus d'apprentissage (voir section 4.2.6.1). Notons que la nouvelle matrice de similarités obtenue est symétrique (par propriété du noyau gaussien de f) et positive (par la transformation

26. À titre indicatif, sur des expériences demandant le rapatriement d'un contenu parmi 500, et pour une fonction de régression construite sur un corpus de 18 contenus, le temps d'attente avant une réponse du système est en moyenne inférieur à la seconde.

$MinMax^{-1}$).

Pour obtenir une matrice de distance visualisable, il reste à garantir l'inégalité triangulaire. La résolution de ce problème est laissée au modèle d'énergie (voir section 2.3.1.3). Son principe est simple : une matrice de distances est calculée à partir des positions initiales des sommets du graphe ; cette matrice est modifiée à intervalles de temps réguliers dans le but de se rapprocher le plus possible de la matrice de similarités, sous la contrainte des différentes forces agissant entre les sommets.

4.3.2.2 Différentes forces

Soit V l'ensemble des sommets du graphe et $v \in V$ un sommet. L'énergie d'un système masse-ressort est égale à la somme des travaux des forces s'exerçant sur chaque masse :

$$Energie = \sum_{v \in V} \int \overrightarrow{F_v} \cdot d\overrightarrow{x_v} \tag{4.12}$$

où $\overrightarrow{F_v}$ est la somme des forces qui s'exercent sur le sommet v pendant le temps $d\overrightarrow{x_v}$.

Une force de tension est considérée entre deux sommets voisins, tension qui représente l'attraction exercée entre deux particules, ainsi qu'une force de répulsion entre tout couple de sommets du graphe. Nous définissions deux fonctions T et F qui simulent respectivement cette tension et cette répulsion.

$$T : E \longrightarrow \mathbb{R}^2 \tag{4.13}$$
$$F : V \times V \longrightarrow \mathbb{R}^2 \tag{4.14}$$

Pour un modèle de force naturelle, c'est-à-dire que la force d'attraction suit la loi de Hooke [27] et que deux sommets se repoussent à la manière de deux particules électriques, nous obtenons les formules suivantes [Thi06] :

$$\overrightarrow{T}(u,v) = k \cdot (|uv| - l_0) \cdot \frac{\overrightarrow{uv}}{|uv|}, \text{ avec } (u,v) \in E \tag{4.15}$$
$$\text{et} \qquad \overrightarrow{F}(u,v) = g.m_u.m_v \cdot \frac{\overrightarrow{uv}}{|uv|^3}, (u,v) \in V \times V \tag{4.16}$$

avec :
- k le coefficient de raideur de l'arête ;
- l_0 sa longueur au repos ;
- g la constante de gravitation (négative pour une force de répulsion) ;
- m_u et m_v les masses ou charges respectives de u et v.

Ces deux fonctions dépendent de la distance entre u et v, comme pour tout autre modèle de forces. Ainsi, nous pouvons simplifier l'expression des forces en considérant les fonctions $t(x)$

27. loi formulée en 1968 qui stipule que « l'allongement est proportionnel à la force ».

et $f(x)$. $t : \mathbb{R} \longrightarrow \mathbb{R}$ et $f : \mathbb{R} \longrightarrow \mathbb{R}$ pour respectivement la tension d'une arête de longueur x et la répulsion entre deux sommets à distance x.

$$t(x) = k \cdot (x - l_0) \tag{4.17}$$

$$f(x) = \frac{g \cdot m_u \cdot m_v}{x^2} \tag{4.18}$$

Pour intégrer ce modèle à notre prototype, nous avons suivi les principes d'un modèle physique de particules électriques, et considérons l'application des différentes forces de la manière suivante :

- il y a une attraction entre deux particules si une relation existe entre elles ; ces relations, déterminées par le formalisme d'organisation, n'existent qu'entre des contenus libres et ancrés ;
- toutes les particules se repoussent les unes des autres ; nous avons restreint cette propriété aux particules d'une même classe, pour respecter la notion d'espace autonome propre à une classe définie dans notre formalisme.

En plus de celles-ci, nous avons introduit une force ponctuelle de « viscosité » $\overrightarrow{R}$, qui joue le rôle d'une force de frottement, inversement proportionnelle à la vitesse de la particule. Elle permet d'atténuer les oscillations des particules et de converger vers un état de stabilité énergétique quelle que soit la configuration du système. Si $\overrightarrow{V}^t(u)$ est la vitesse de la particule u à l'instant t, alors :

$$\overrightarrow{R}^t(u) = \iota \times \overrightarrow{V}^t(u), \text{ avec } \iota < 0 \tag{4.19}$$

avec ι l'intensité de la viscosité.

La figure 4.6 récapitule les différentes forces exercées sur un contenu non ancré.

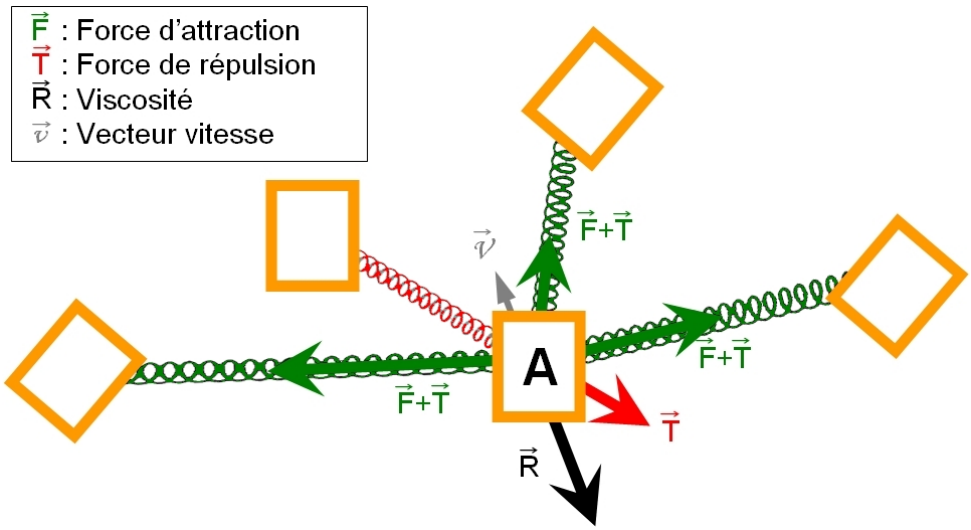


FIGURE 4.6 – Forces exercées sur un contenu A

4.3.2.3 Calcul des positions des sommets du graphe

Pour calculer les nouvelles positions des sommets du graphe, nous avons choisi d'utiliser l'algorithme de Runge-Kutta à l'ordre 4 (RK4), couramment employé dans le domaine de la physique.

Il s'agit d'un schéma dit « d'intégration explicite », qui calcule la position d'une particule à l'instant suivant $(t + 1)$ à partir de l'instant courant (t). L'intégration temporelle consiste en une approximation de la solution d'une équation différentielle du premier ordre à l'aide de 4 points intermédiaires. La notion de « pas temporel » est interprétée en considérant un pas de discrétisation de l'abscisse des points. Nous l'avons fixé à 10 pour notre prototype.

L'intensité de la vitesse d'une particule au temps $t + 1$ est remise à jour en fonction de sa nouvelle position.

4.3.2.4 Stabilité énergétique

Pour une classe donnée, soit $S = \{A_1, \ldots, A_M\}$ un corpus de réorganisation. Pour simuler l'état de stabilité énergétique, permettant ainsi d'arrêter le processus de réorganisation automatique, nous calculons l'énergie dépensée du temps t au temps $t + 1$ de la manière suivante :

$$E = \sum_{i=1}^{M} \frac{\delta(A_i^t, A_i^{t+1})}{M} \tag{4.20}$$

avec A_i^t la position de l'objet au temps t et $\delta(u, v)$ la distance euclidienne entre u et v.

Nous supposons l'état de stabilité atteint lorsque l'énergie passe n fois sous un seuil arbitraire κ. Pour notre prototype, nous avons fixé n à 3 et κ à 0,01. Sous ces paramètres, le temps nécessaire pour atteindre un état stable oscille entre 5 et 10 secondes. L'utilisateur peut également arrêter manuellement le processus.

4.3.2.5 Invariance

Ordonner des éléments avec notre interface implique une visualisation de cet ordre sous la forme d'une orientation dans un espace en deux dimensions : disposer des éléments sur une courbe virtuelle sous-entend une volonté d'ordonnancement d'un bout à l'autre de cette courbe.

Nous travaillons sur de l'interprétation de distances. Nous ne nous soucions pas des positions des contenus ancrés dans l'espace dynamique pour réorganiser les libres, mais uniquement des écarts entre ceux-ci. Pour notre système, l'orientation (de gauche à droite, de haut en bas, etc.) de la forme géométrique servant de support à l'ordre n'a aucune influence sur nos résultats.

De la même manière, les changements d'échelles et de translations entre les objets ne nous dérangent pas, tant qu'ils restent homogènes pour tous les éléments ancrés d'une même classe (cette propriété nous a servi lors de l'implantation du zoom de notre GUI).

Nous qualifions de ce fait notre système d'invariant aux rotations, aux translations et aux homothéties.

4.4 Synthèse générale : la mesure de similarités

Concrètement, notre mesure de similarités est le résultat de la conjonction :
- de la fonction de régression d'une part, issue du modèle numérique de similarité (décrit dans le chapitre 3 et appliqué dans la section 4.2 de ce chapitre),
- de l'acquisition et de la restitution des données (chapitre 1) dans l'espace dynamique d'autre part, processus soumis au formalisme d'organisation (présenté dans le chapitre 2).

Nous résumons, dans cette section, les principales notions évoquées dans ce manuscrit permettant de créer et de visualiser cette mesure.

4.4.1 Vision schématique

Le tableau 4.1 résume les trois principales phases du processus de création de la mesure de similarités : considérons un utilisateur qui positionne quelques contenus sur l'espace dynamique et ancre certains d'entre eux. Il lance la procédure d'apprentissage, en demandant ou non le rapatriement d'un ou plusieurs contenus (choisis ou laissés à la discrétion du système) :

TABLE 4.1 – Création de la mesure de similarités

1. Apprentissage	(a) Création des variables endogènes ; (b) Création des variables exogènes ; (c) Apprentissage de la fonction de régression ;
2. Prédiction	(a) Création du corpus du réorganisation ; (b) Calcul de la matrice de similarités ;
3. Visualisation	(a) Initialisation du modèle d'énergie avec la matrice de similarité ; (b) Réorganisation automatique des contenus non ancrés ;

Le schéma 4.7 résume la manière dont les différentes briques de ce système ont été assemblées. Il illustre l'interdépendances existant entre les trois étapes.

4.4.2 Une mesure adaptative

Revenons sur l'apport incrémental de contenus. Même si ce processus couplé à notre système ressemble à de l'apprentissage actif ou de la rétroaction (appelé *relevance feedback* en anglais), c'est-à-dire à des méthodes pour lesquelles l'apport de nouveaux éléments contribue à la spécialisation du modèle, il n'en reste pas moins différent.

La finalité est certes la même : l'affinage de la mesure de similarités va dans le sens où nous cherchons à obtenir une meilleure compréhension de la tâche. Toutefois, notre système ne peut se permettre de considérer que la tâche restera la même d'un bout à l'autre du processus

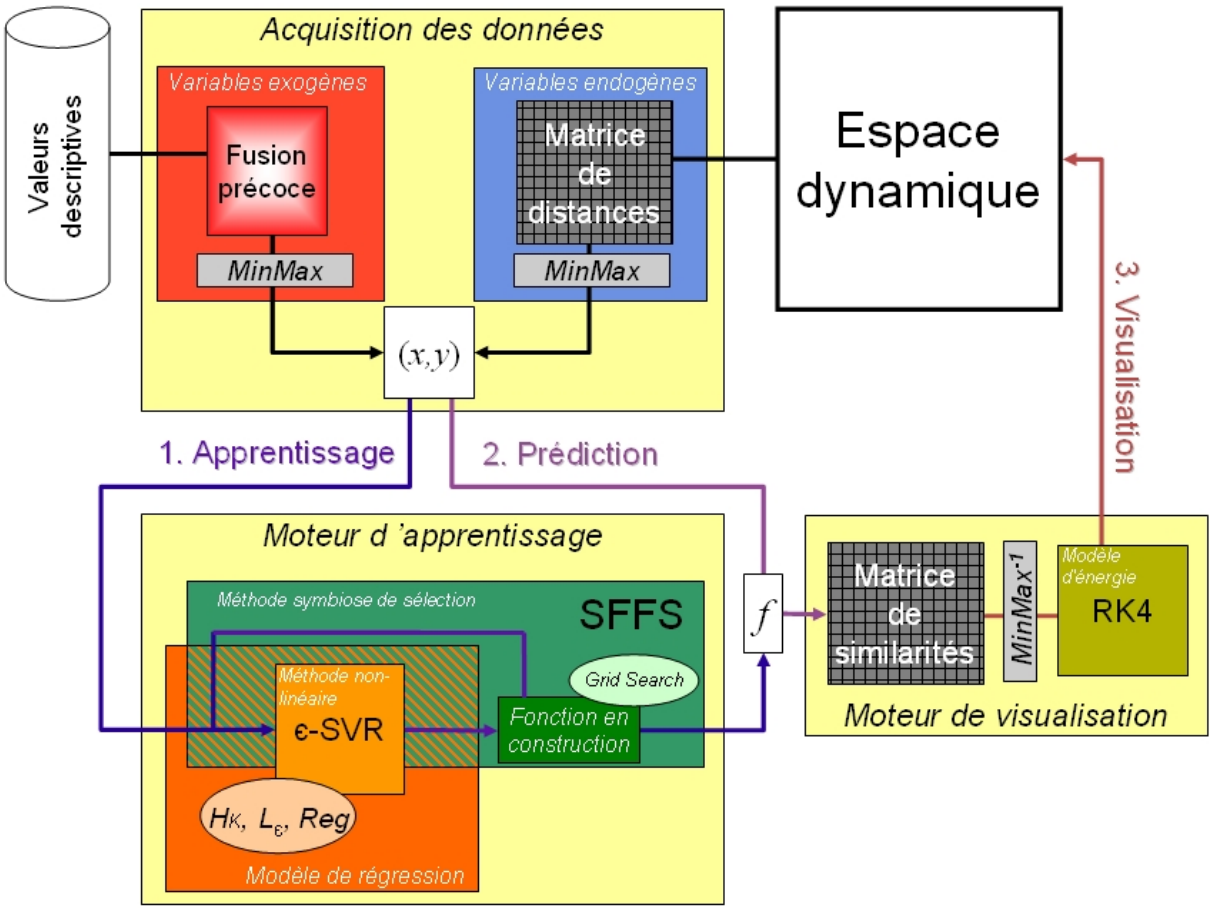


FIGURE 4.7 – Schéma du système

d'organisation global (contrainte $\mathscr{C}_5$) : l'utilisateur peut vouloir ordonner ses contenus, puis changer d'avis et les classer, pour n'en réordonner une partie, etc.

Chacune de ces actions est une sous-tâche organisationnelle pouvant être traitée par spécialisation du modèle. Toutefois la tâche globale doit être vue comme évolutive, et la mesure de similarités qui la caractérise se doit de l'être également.

D'une manière générale, chaque apprentissage initié par l'utilisateur engendre une nouvelle fonction, indépendante de la précédente. Ce choix a le désavantage d'être plus coûteux qu'un processus qui tirerait directement profit des apprentissages précédents. Cependant, c'est à notre avis un mal nécessaire pour permettre à l'utilisateur d'organiser librement sa base de contenus.

Nous avons donc pensé notre mesure pour qu'elle puisse à la fois permettre de spécialiser l'apprentissage sur la tâche en cours et se remettre totalement en question si la tâche devait est modifiée. En ce sens, nous parlons de notre mesure de similarités comme d'une mesure adaptative. Nous étudierons cet aspect à travers l'expérience 4.5.2.

4.4.3 Performances théoriques et pratiques

La décomposition en trois phases du processus général de construction de la mesure (voir tableau 4.1), nous permet de constater trois types d'erreurs engendrées par notre système :

1. une **erreur d'apprentissage**, caractérisée par le moteur d'apprentissage, relative aux performances du modèle de similarités ;
2. une **erreur de prédiction**, dépendante de l'interprétation de la matrice de distances théorique par une matrice de similarités ;
3. une **erreur de visualisation**, caractérisée par le moteur de visualisation, relative aux contraintes imposées par le modèle d'énergie.

La seule erreur sur laquelle nous ayons une main mise est l'erreur d'apprentissage, qui reste théorique. Même si une erreur faible implique une fonction de régression à bon compromis biais/variance, la tâche réelle pensée par l'utilisateur peut être mal comprise par la machine. Dans ce cas, la performance théorique ne peut témoigner de l'écart entre la fonction de régression, considérée comme bonne par le système, et la mise en œuvre pratique de la mesure, jugée mauvaise par l'utilisateur.

Appelons **erreur d'interprétation** l'erreur qui témoigne de l'écart entre ce qui est attendu par l'utilisateur (les distances espérées sur l'espace dynamique) et ce qui est restitué par le système (les distances observées une fois la stabilité énergétique atteinte).

La réelle difficulté inhérente à nos travaux peut se traduire par l'inexistence de corrélation directe entre erreur d'apprentissage et erreur d'interprétation, car cette dernière dépend également des erreurs de prédiction et de visualisation. Ce phénomène et ses conséquences feront l'objet d'une étude dans l'expérience 4.5.3.

4.4.4 Liens avec nos contraintes

Tout au long de ce manuscrit, nous avons décrits nos choix comme des réponses aux différentes contraintes définies dans l'introduction (section 2.4).

Le tableau 4.2 récapitule ceci, en précisant à quelle contrainte correspond le choix adopté :

TABLE 4.2 – Rappel des contraintes et réponses apportées par un modèle de régression

	Rappel des contraintes	**Propriétés du système**
$\mathscr{C}_0$	La tâche est-elle modélisable ?	**Hypothèse** générale.
$\mathscr{C}_1$	La notion de similarité est-elle utilisable ?	Le **formalisme d'organisation** nous aide à extraire les valeurs nécessaires à la construction des **variables endogènes**.
$\mathscr{C}_2$	Rapidité d'apprentissage du modèle ?	La **méthode ε-SVR est rapide** pour un nombre raisonnable de contenus d'apprentissage.
$\mathscr{C}_3$	Gestion du multi-grain ?	Le **formalisme d'organisation** nous aide à travers la **construction de classes** et la **notion de représentant**.
$\mathscr{C}_4$	Nécessite peu de contenus pour l'apprentissage ?	Le modèle de régression est un **modèle prédictif**. Il permet d'extrapoler la tâche à partir de peu de données.
$\mathscr{C}_5$	Possibilité de remettre la tâche en question ?	La **mesure de similarités est adaptative** et le **processus incrémental de contenus** aide à son exploitation.
$\mathscr{C}_{6.1}$	Gestion de valeurs descriptives inconnues ?	Le **schéma de fusion précoce** n'est pas regardant sur la provenance des valeurs numériques. La **nature non-linéaire de la régression** permet d'envisager un panel étendu de relations possibles entre variables exogènes et endogènes.
$\mathscr{C}_{6.2}$	Gestion de valeurs descriptives perturbatrices ?	La parcimonie apportée par l'**algorithme de sélection** aide à gérer partiellement ce problème.
$\mathscr{C}_{7.1}$	Corpus d'apprentissage inconnu ?	La **régression est non-linéaire** et permet d'exploiter une marge d'erreur souple pour gérer l'imprécision provenant de valeurs de distances approximatives.
$\mathscr{C}_{7.2}$	Tâche organisationnelle mal définie ?	La **ε-SVR est parcimonieuse**. À travers le choix des vecteurs supports les distances superflues sont élaguées du modèle.
$\mathscr{C}_{7.3}$	Utilisateur inconnu ?	Le **formalisme d'organisation** a été conçu pour aider un utilisateur *Lambda* à organiser ses contenus.

4.4.4.1 Conclusion

Nous avons pris un peu de recul dans cette section, afin d'apprécier la manière dont les différents composants de notre système s'imbriquent les uns aux autres, pour constituer une mesure de similarités adaptative. Les trois étapes permettant sa conception sont interdépendantes, et les erreurs générées par chacune d'elle le sont également. Elles vont nous aider, dans la suite de ce chapitre, à analyser les points forts et les faiblesses de ce système, à travers l'évaluation de notre prototype sur différentes expériences.

4.5 Validation des choix technologiques

Nous décrivons dans cette section plusieurs séries d'expériences qui nous ont permis d'analyser, à travers l'étude de notre prototype, les points théoriques et pratiques évoqués au cours de ce manuscrit.

Nous avons développé ce prototype en C/C++, sur un ordinateur possédant 2 Go de mémoire vive et un processeur à double cœur cadencé à 1,6 GHz.

Le modèle de régression a été implanté grâce aux bibliothèques *libsvm* [28] et *gsl* [29]. L'interface visuelle a été développée en *OpenGL* [30] grâce à la bibliothèque graphique *Clutter* [31].

Les contenus exploités dans ces expériences sont des documents d'une durée inférieure à trois minutes. Les descriptions utilisées sont les valeurs moyenne, variance, minimum et maximum calculées sur les descripteurs « bas niveau » présentés dans le chapitre 1. Nous avons exploité 32 descriptions audio et 36 vidéo, soit un total de 68 descriptions.

Le tableau 4.3 dresse un récapitulatif en fonction du média concerné :

TABLE 4.3 – Descripteurs utilisés

Média	**Descripteur**	**Nombre de descriptions**
Audio	« Modulation de l'énergie à 4Hz », « Modulation de l'entropie », « ZCR », « Flux spectral », « Centroïde spectral », « *RollOff point* », « Fréquence fondamentale », « Énergie »	$4 \times 8 = 32$
Vidéo	« 1^{ere} teinte dominante », « 2^{ieme} teinte dominante », « 1^{ere} saturation dominante », « 2^{ieme} saturation dominante », « 1^{ere} valeur dominante », « 2^{ieme} valeur dominante », « Luminance moyenne », « Contraste », « Taux d'activité »	$4 \times 9 = 36$

4.5.1 Durées des apprentissages

Nous nous sommes proposés de satisfaire la contrainte du temps réel en présentant une réponse du système avant qu'un délai de 10 secondes ne soit dépassé.

Au cours de nos expériences, ce délai était totalement respecté pour des apprentissages utilisant de 3 à 7 contenus, partiellement respecté entre 8 et 15 contenus, non respecté au-delà.

Nous avons également fixé un deuxième seuil de tolérance plus lâche de 20 secondes, et notons que ce délai est totalement respecté en dessus de 12 contenus, partiellement respecté entre 13 et 18 contenus, non respecté au-delà.

28. http ://www.csie.ntu.edu.tw/ cjlin/libsvm
29. http ://www.gnu.org/software/gsl
30. http ://www.opengl.org
31. http ://www.clutter-project.org

La figure 4.8 illustre la durée nécessaire à l'apprentissage d'une fonction de régression via notre algorithme 1, en fonction du nombre de contenus présents dans le corpus. L'étude a été menée sur l'ensemble de nos expériences, soit un minimum de 50 valeurs de durées, par cardinalité différente.

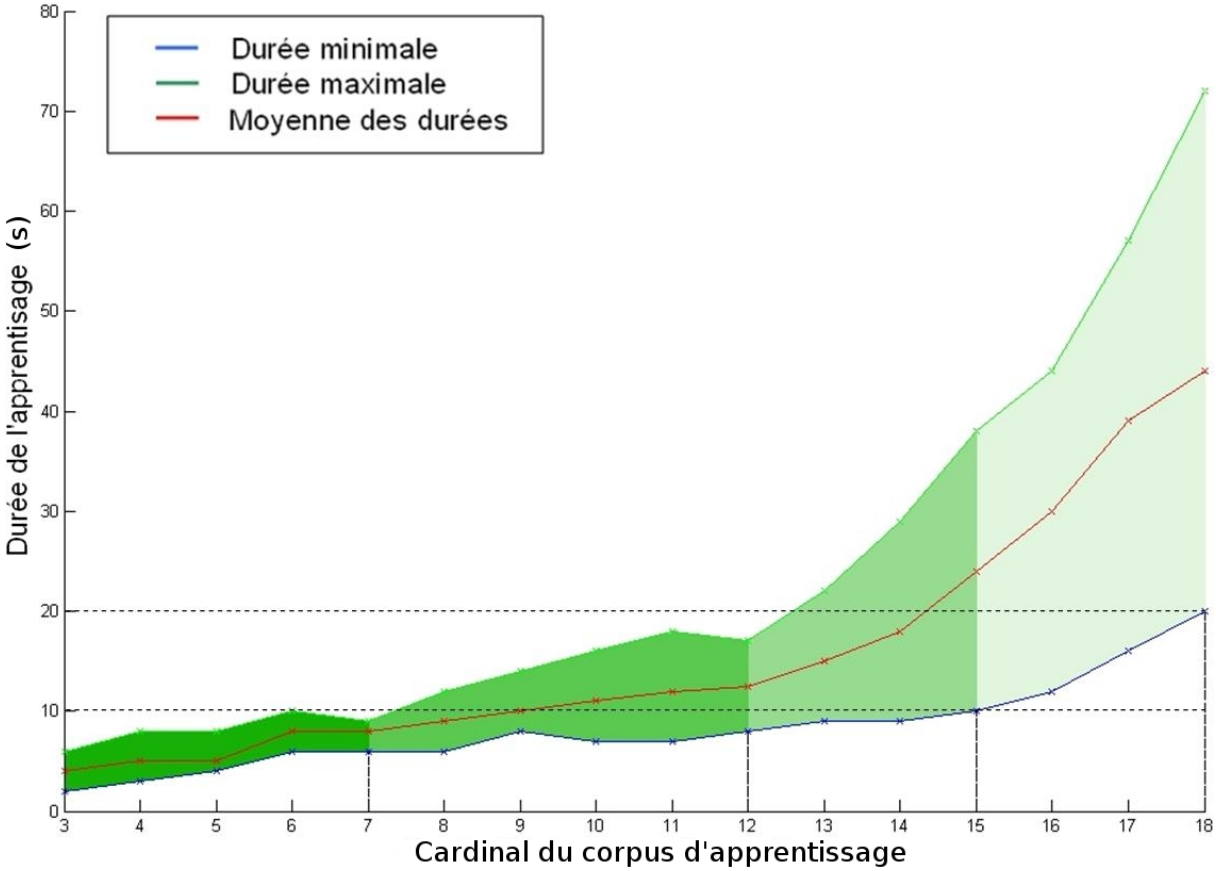


FIGURE 4.8 – Observation de la durée d'un apprentissage en fonction du cardinal du corpus

Pour un même nombre de contenus, la variation des durées d'une expérience à l'autre est essentiellement due au nombre de passes algorithmiques nécessaires à l'élaboration de la fonction, ainsi qu'à la valeur du coût qui, s'il devient grand (2^{13}, 2^{15}) peut ralentir le processus.

Nous identifions ainsi quatre intervalles de temps représentés par les zones vertes de la figure 4.8, délimités par différentes cardinalités du corpus d'apprentissage, permettant de plus ou moins satisfaire l'utilisateur en terme de temps d'attente. Nous ferons sciemment l'amalgame entre ces délais et le nombre de contenus utilisés pour l'apprentissage, et utiliserons dans la suite de ce manuscrit des qualificatifs relatifs à ces zones :

- pour un apprentissage utilisant un corpus de 3 à 7 contenus, nous qualifions le temps d'attente correspondant de **court** ;
- de 8 à 12 contenus, nous le qualifions de **raisonnable** ;
- de 13 à 15 contenus, nous le qualifions de **long** ;
- de 16 à 18 contenus, nous le qualifions de **très long**.
- au-delà de 18 contenus, nous le qualifions de **trop long**.

Les valeurs 3, 7, 12, 15 et 18 seront appelées **valeurs limites**.

Dans le cadre d'un usage réel de notre application, nous rappelons qu'un contenu d'apprentissage, avant d'être placé puis ancré, a très probablement été visionné et qu'il n'y a pas de limite à sa durée. La charge cognitive pesant sur ce type d'objet est donc suffisamment lourde pour considérer qu'un corpus de 18 éléments constitue un ensemble d'apprentissage conséquent.

4.5.2 Évaluation des performances théoriques du modèle

Cette série d'expériences est relative aux performances théoriques de notre méthode de construction de régression ε-SVR par sélection SFFS, fondée sur une optimisation partielle des hyper-paramètres par exploration de grilles, évaluées par validations croisées tripartites.

Nous l'avons confrontée à des méthodes de construction par optimisation que nous qualifions de « complètes » : pour une passe algorithmique donnée, soit D le nombre de descriptions déjà concaténées et D' le nombre total de descriptions ; chaque passe *forward* (resp. *backward*) entraîne $(D' - D)$ (resp. D) phases d'optimisations des hyper-paramètres pour savoir s'il est intéressant de concaténer (resp. supprimer) une description au modèle.

Rappelons que pour chacun de ces tests, notre méthode, que nous qualifions de « partielle », se contente d'utiliser les mêmes hyper-paramètres trouvés lors de la passe algorithmique précédente. Son initialisation se fait par optimisation de γ et $\mathbb{C}$ sur l'ensemble complet des descriptions. Le paramètre ε est fixé à 0,1.

Pour la première méthode complète, ε est fixé à 0,1 et seuls γ et $\mathbb{C}$ sont optimisés. Pour la deuxième, nous avons cherché à optimiser ε en jouant sur le paramètre ν d'une ν-SVR : une grille d'intervalle]0,1] et de pas 0,1 a été utilisée.

Ces méthodes finissent toutes trois par une optimisation de γ et $\mathbb{C}$ sur grille réduite.

4.5.2.1 Protocole expérimental $\mathcal{P}_1$

Type de tâche : Regroupement ;

Corpus documentaire : 43 macrosegments vidéo extraits automatiquement d'une émission télévisuelle intitulée « *Morning Café* ». Ces extraits durent entre 30 secondes et 3 minutes ;

Descriptions utilisées : Toutes (32 audio et 36 vidéo) ;

But : Comparer les performances de notre algorithme d'apprentissage à des méthodes plus fiables d'un point de vue théorique.

Séries d'expériences A

Les échantillons sont étiquetés en deux groupes : 18 contenus « plateaux » et 25 contenus « reportages ». À chaque groupe correspond un couple de coordonnées précis dans l'espace de représentation.

Pour chaque expérience, M contenus sont sélectionnés de manière aléatoire, ancrés et superposés sur les coordonnées du groupe correspondant, sous contrainte d'avoir un minimum de 1 élément dans le groupe le plus petit.

Nous avons pratiqué trois séries de dix expériences pour chaque valeur de M variant entre 3 et 7 contenus :

- la première série utilise une régression construite avec la méthode d'optimisation partielle ;
- la deuxième porte sur la méthode complète avec ε fixé à 0,1 ;
- la dernière emploie la méthode complète avec la prise en charge de l'optimisation de ε par grille.

4.5.2.2 Résultats

Observons dans un premier temps l'erreur d'apprentissage, qui est le critère de performance du modèle. La figure 4.9 illustre les résultats recueillis lors de cette série de tests (se référer à l'annexe B pour plus de détails). Nous remarquons que, pour un nombre de contenus d'apprentissage donné, les moyennes des erreurs commises sur les trois méthodes sont relativement proches. Les améliorations significatives des performances théoriques du modèle ne sont pas dues à la méthode utilisée mais bien au cardinal du corpus : en moyenne, plus de contenus sont fournis en entrée, meilleurs sont les résultats.

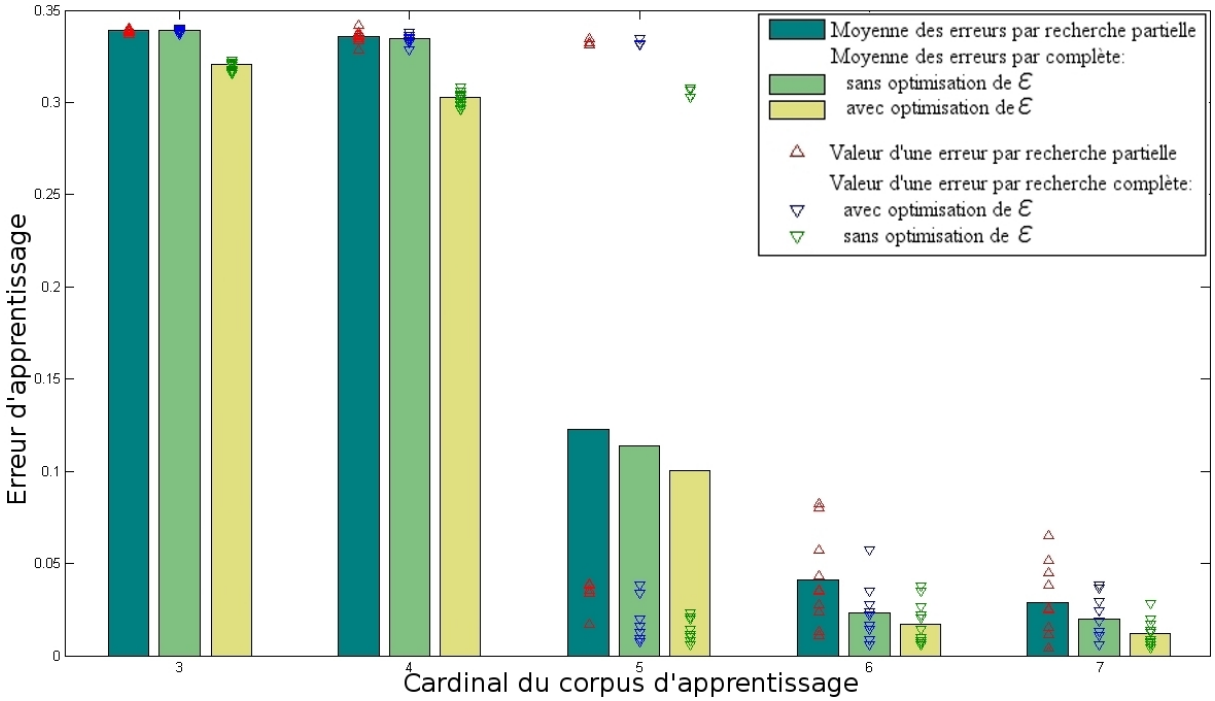


FIGURE 4.9 – Observation des moyennes des erreurs d'apprentissage en fonction du nombre de contenus du corpus d'apprentissage

Du point de vue du temps d'apprentissage, la méthode partielle se révèle extrêmement avantageuse (le tableau 4.4 donne un récapitulatif).

Une autre observation intéressante concerne les descriptions retenues par les algorithmes de sélection d'attribut des différentes méthodes.

TABLE 4.4 – Observation des moyennes des durées d'apprentissage (en secondes) en fonction du cardinal de l'ensemble d'apprentissage.

Cardinal	3	4	5	6	7
Partielle	6,7	7,4	7,9	8,5	8,9
Complète sans ε	189,3	253,3	245,8	333,1	423,6
Complète avec ε	587,6	955,8	1047,9	1545,3	1823,5

Nous verrons par la suite que même si la modélisation de telle ou telle description engendre une faible erreur d'apprentissage, elle n'implique pas forcément une interprétation correcte de la tâche organisationnelle.

Nous notons toutefois que, pour un même corpus d'apprentissage, la méthode partielle retrouve régulièrement des descriptions retenues par la méthode complète sans optimisation de ε, ce qui est un point théorique positif pour cette alternative moins coûteuse en temps de calculs (le tableau 4.5 illustre ce phénomène). Nous avons noté qu'en moyenne :

- dans plus de la moitié des tests (56%), les deux méthodes ont au moins une description en commun (la ligne Intersection du tableau 4.5) ;
- plus du tiers (34%) des descriptions retenues par la partielle se retrouvent dans la complète (la ligne Inclusion du tableau 4.5) ;
- les descriptions sont les mêmes pour les deux méthodes dans plus d'un quart (28%) des expériences (la ligne Égalité du tableau 4.5).

TABLE 4.5 – Comparaison des descriptions retenues par les deux méthodes

Cardinal	3	4	5	6	7	Moyenne
Intersection	8/10	4/10	5/10	5/10	6/10	56%
Inclusion	4/10	4/10	4/10	2/10	3/10	34%
Égalité	4/10	3/10	3/10	2/10	2/10	28%

Nous pouvons terminer cette série de tests en notant que **le nombre de passes algorithmiques pour construire notre régression est toujours supérieur à 1**, quelle que soit la méthode utilisée. Cela dénote de l'intérêt d'une fonction construite grâce à une méthode de sélection séquentielle d'attributs, par rapport à une fonction qui conserverait l'ensemble de ses paramètres.

4.5.2.3 Conclusion

Aux vues de ces résultats, nous pensons qu'il est pertinent d'utiliser notre méthode partielle, comparativement à des méthodes plus fiables, mais dont le rapport « gain théorique / coût de mise œuvre » n'est absolument pas avantageux.

4.5.3 Pertinence visuelle de la mesure de similarités

Notre prototype exploite une matrice de similarités créée par notre moteur d'apprentissage, pour la transformer en matrice de distances grâce au moteur de visualisation. En proposant cette architecture, nous supposons qu'il est pertinent de visualiser sous forme de distances, via un modèle d'énergie masse-ressort, des similarités prédites à partir d'une fonction de régression non-linéaire.

Ces nouvelles séries de tests nous permettent de vérifier cette hypothèse (se référer à l'annexe C pour plus de détails sur les résultats).

4.5.3.1 Protocole expérimental $\mathcal{P}_2$

Type de tâche : Ordonnancement ;

Corpus documentaire : 96 échantillons de musique mono instrumentale. Un échantillon correspond à une note d'une durée de 4 secondes. 12 notes sont jouées, une note par demi-ton en montant du La bémol jusqu'au Sol. L'acquisition a été faite sur 8 instruments différents : accordéon, flûte de pan, guitare acoustique, ocarina, violon, violoncelle, piano, piano électrique ;

Descriptions utilisées : Uniquement les descriptions audio (32) ;

Buts : La mesure de similarités est correctement traduite en terme de distances dans l'espace dynamique (série B) ; Quel est son comportement si la(les) description(s) qui pilotait la tâche disparaît (série C) ;

Série d'expériences B

Les contenus sont disposés sur une ligne virtuelle de manière à témoigner de la hauteur de leur note. Un couple de coordonnées dans l'espace dynamique correspond donc à une hauteur de note. Une même couleur est attribuée aux contenus de note identique (voir la figure 4.10).

FIGURE 4.10 – Couleurs associées aux contenus en fonction de la hauteur de note

Pour chaque test, M contenus d'apprentissage sont sélectionnés de manière aléatoire, et ancrés sur leur position d'origine. Nous avons pratiqué une série de dix expériences pour chaque valeur de M, M prenant les valeurs 3, 4, 5, 6, 7, 12, 15 et 18, soit un total de 80 expériences.

Nous souhaitons observer si le moteur de visualisation rend bien compte de la mesure de similarités calculée (les résultats sont-ils visuellement cohérents ?) ;

Série d'expériences C

Nous pratiquons deux nouvelles séries de 80 tests sur le thème de l'expérience précédente :
- dans la première série, nous enlevons la description qui a été la plus sélectionnée dans la série B ;
- dans la deuxième série, nous enlevons le descripteur correspondant (c'est-à-dire les 4 descriptions : moyenne, variance, minimum et maximum).

Nous avons pensé l'ordre en fonction de la hauteur de la note, et nous disposons d'un bon tuteur dans le descripteur de la fréquence fondamentale. Nous cherchons à étudier ici le comportement de la mesure de similarités lorsque ce guide disparaît.

4.5.3.2 Résultats de l'expérience B

Nous avons choisi d'observer l'évolution des quatre types d'erreurs définies dans la section 4.4.3 :

- l'erreur d'apprentissage (le critère de performance du modèle) ;
- l'erreur de prédiction, entre la matrice de position initiale et la matrice de similarité ;
- l'erreur de visualisation, entre la matrice de similarité et la matrice de position finale ;
- l'erreur d'interprétation, entre la matrice de position initiale et la matrice de position finale.

Notons que l'ordre de grandeur est le même pour les trois dernières erreurs, qui sont calculées sur les valeurs de pixels à l'écran. Cette ordre est environ 10^5 fois plus grand que pour l'erreur d'apprentissage, calculé sur des valeur d'EQM normalisées.

Il se peut que certains tests aient une mauvaise erreur d'interprétation alors que les résultats sont visuellement corrects. La figure 4.11 illustre ce phénomène : bien qu'ayant des erreurs d'interprétation très proches, le résultat 1 est visuellement bien meilleur que le 2. Le résultat 3, qui montre une erreur correcte pour un résultat visuel convenable, sert de témoin.

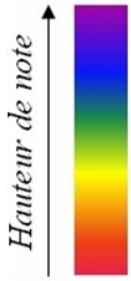
	Résultat 1	Résultat 2	Résultat 3
Résultats visuels après réorganisation 			
Cardinalité du corpus d'apprentissage	4	6	7
Erreur d'interprétation	93386,1	106843,8	60211,7

FIGURE 4.11 – Différents résultats ainsi que leur erreur d'interprétation associée

Ce phénomène est lié à la normalisation $MinMax$, qui contraint les contenus à s'organiser entre les notes extrêmes de l'intervalle d'apprentissage (cas pratique de la discussion faite en

section 4.2.6.1).

La figure 4.12 illustre l'évolution des moyennes des différentes erreurs. Nous notons dans un premier temps une baisse significative des erreurs d'interprétation, de prédiction et d'apprentissage en fonction du nombre de contenus présents dans le corpus d'apprentissage. De plus, l'erreur de visualisation est relativement faible, ce qui nous conforte dans notre choix du modèle de placement de graphe par énergie. Notons pour finir que les erreurs de prédiction et d'interprétation semblent très proches l'une de l'autre.

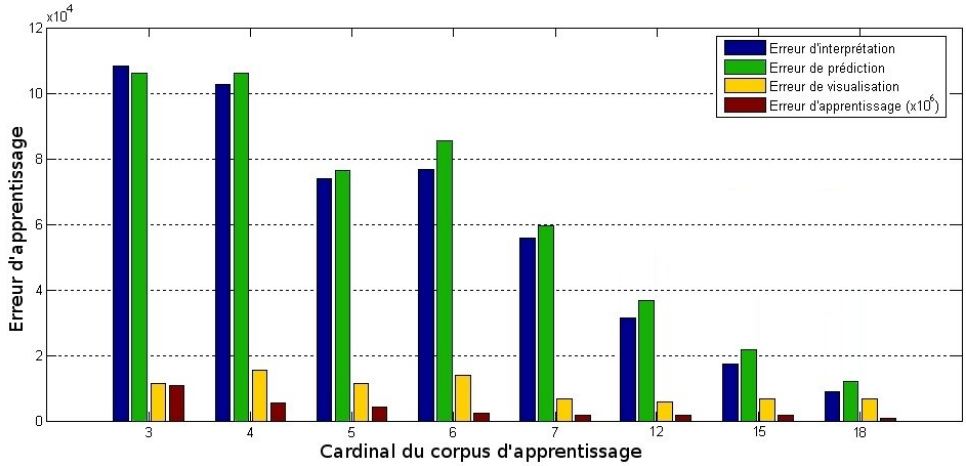


FIGURE 4.12 – Observation des moyennes des différentes erreurs en fonction du cardinal de l'ensemble d'apprentissage

Nous avons ensuite analysé les erreurs pour essayer de trouver d'éventuelles liens entre elles.

Une première analyse nous confirme que l'on ne peut pas déduire clairement les performances pratiques du système (incarnées par l'erreur d'interprétation) à partir des performances théoriques du modèle (représentées par l'erreur d'apprentissage) (voir la figure 4.13). Aucune relation directe exploitable ne semble exister entre ces deux erreurs.

Par contre, nous pouvons relever un phénomène intéressant entre les erreurs de prédiction et d'interprétation : le coefficient de corrélation linéaire entre les vecteurs construits sur ces deux erreurs est de 0,98 (voir la figure 4.14).

La mesure de similarités calculée par le prototype est donc indirectement corrélée à la distance représentée à l'écran, via la matrice de similarités. Une grande (resp. petite) erreur de prédiction de la tâche organisationnelle par la machine entraîne de mauvais (resp. bons) résultats visuels.

4.5.3.3 Résultats de l'expérience C

La figure 4.15 illustre le comportement des différentes erreurs d'interprétations sur les trois séries d'expériences.

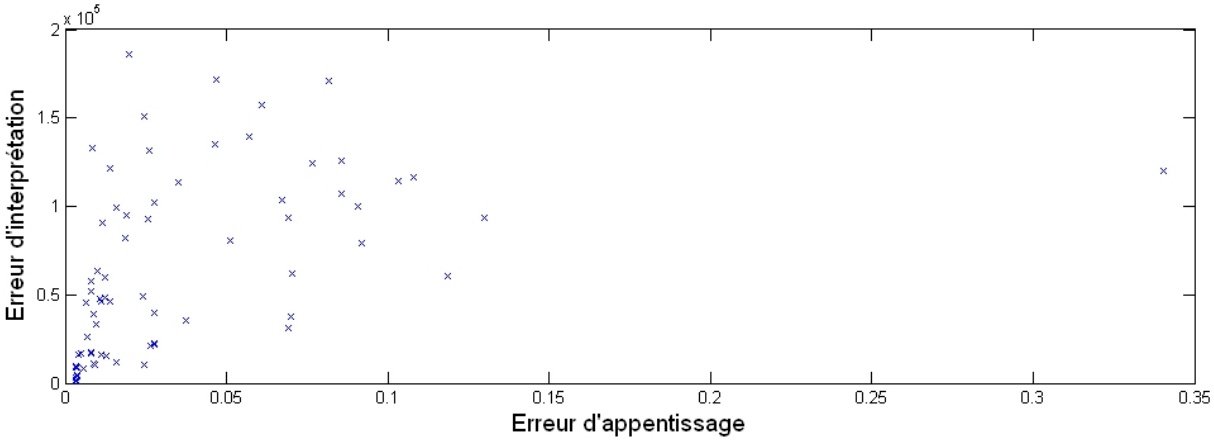


FIGURE 4.13 – Observation des erreurs d'interprétation en fonction des erreurs d'apprentissage

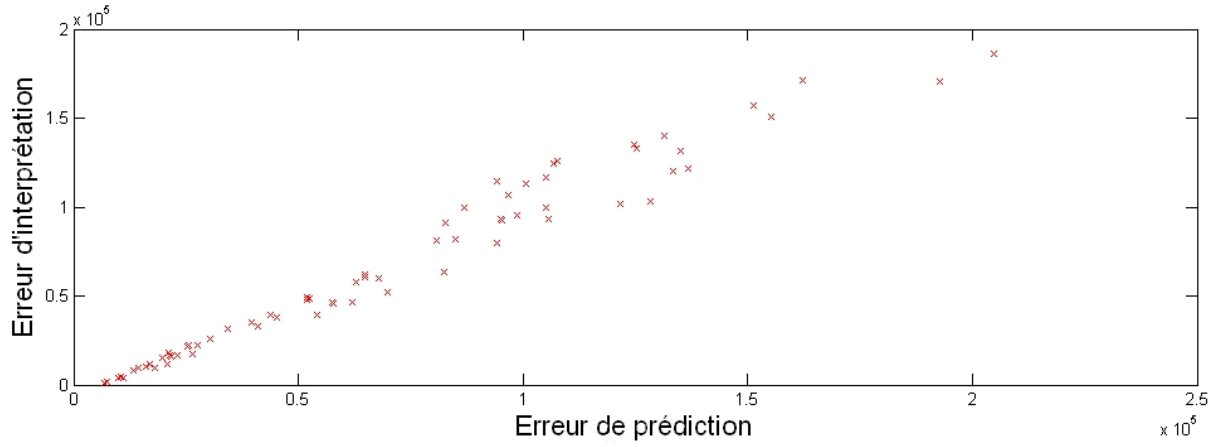


FIGURE 4.14 – Observation des erreurs d'interprétation en fonction des erreurs de prédiction

Dans la première série, la description « maximum de la fréquence fondamentale » est sélectionnée dans 81.2% des cas (65 apparitions sur 80 expériences). L'erreur d'interprétation est correcte, la tâche a été saisie et bien interprétée par le système, l'ordonnancement des contenus sur leur hauteur de note se fait selon le bon attribut.

Privée de celui-ci, le relais est pris par la description « variance de la fréquence fondamentale », qui intervient dans la construction de 72,5% des modèles (58 apparitions sur 80). Il s'agit toujours du même descripteur, mais il témoigne moins bien de la tâche imposée. L'interprétation en pâtie quelque peu, mais les résultats sont toujours corrects.

Sans le descripteur « fréquence fondamentale » (c'est-à-dire sans aucune des quatre descriptions ayant un rapport avec ce paramètre), 65% des modèles (52 sur 80) s'appuient sur le minimum de l'énergie, entraînant de mauvais résultats.

La figure 4.16 nous montre les meilleurs résultats de chacune de ces trois séries. Ces illustrations vont dans le sens des conclusions tirées précédemment. Les deux premières séries

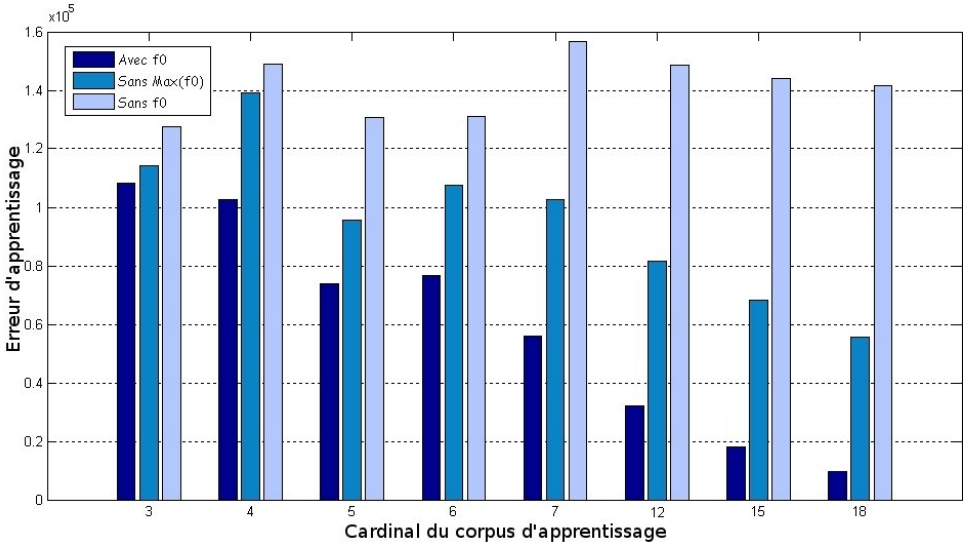


FIGURE 4.15 – Observation des moyennes des erreurs d'interprétations en fonction du cardinal de l'ensemble d'apprentissage

d'expériences permettent d'obtenir des résultats satisfaisants, tandis que même pour le meilleur résultat de la troisième, l'ordre n'a pas été compris par le système.

La mise en défaut notre prototype nous semble légitime, et oriente la question d'un gain en performance sur la modélisation des descriptions plutôt que sur une éventuelle amélioration du système d'apprentissage. Une solution envisageable serait d'utiliser plus de valeurs descriptives ou de nouveaux descripteurs, afin d'élargir le domaine de compétence du système et pouvoir ainsi combler d'éventuelles lacunes.

4.5.3.4 Conclusion

Ces résultats sont pour nous concluants. Ils nous permettent d'envisager sereinement l'utilisation de notre prototype pour traduire visuellement des similarités sous forme de distances.

Toutefois, nous pensons que l'amélioration des résultats d'ordre « pratiques » ne doit pas se focaliser sur l'optimisation de la méthode de prédiction ou sur le modèle d'énergie, mais sur des mécanismes transversaux permettant d'adapter la conjonction de ces méthodes à la compréhension de la réelle tâche organisationnelle. La solution peut se trouver dans une meilleure modélisation des valeurs descriptives ou dans l'utilisation de procédés plus interactifs, tel que le processus de rapatriement, que nous analyserons dans la série d'expériences suivante.

4.5.4 Performances du processus d'apprentissage incrémental

Revenons sur les expériences précédentes. Quels que soient le protocole et le type de la tâche, nous remarquons une amélioration globale des résultats en fonction du nombre de contenus utilisés lors de l'apprentissage. Ce constat justifie l'intérêt que représente pour nous le mécanisme d'apport incrémental de contenus.

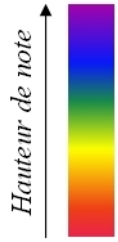
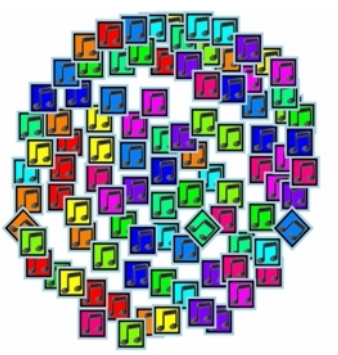
	Apprentissage avec toutes les descriptions	Apprentissage sans Max (f0)	Apprentissage sans f0
Résultats visuels après réorganisation 			
Cardinal du corpus d'apprentissage	18	18	3
Descriptions sélectionnées	Max(f0) ⊕ Min(énergie)	Var(f0) ⊕ Moy(f0) ⊕ Min(énergie)	Var(entropie) ⊕ Max(entropie) ⊕ Max (énergie)
Erreur d'apprentissage	0,003378	0,008574	0,105470
Erreur d'interprétation	3569,0	8188,6	99718,1
Erreur de prédiction	10988,7	12132,6	81970,2
Erreur de visualisation	7366,8	3888,2	22524,7

FIGURE 4.16 – Meilleurs résultats pour les expériences B et C en terme d'erreur d'interprétation

Il arrive toutefois que certains contenus rapatriés desservent le modèle. Notre système se devant d'apporter une réponse satisfaisante dans un délai relativement court (et donc un cardinal de corpus d'apprentissage limité !), il faut que les contenus proposés lors d'un rapatriement aident à faire converger le système le plus rapidement possible.

De plus, n'oublions pas que l'utilisateur a la possibilité de remettre en question la tâche à tout moment.

Dans cette nouvelle série d'expériences, nous étudions notre processus d'apprentissage incrémental, utilisé avec différentes méthodes de rapatriement. Ces tests servent d'argumentaire à l'emploi de la méthode décrite dans la section 4.3.1.2 (appelée max par la suite).

Soit $S = \{A_1, \ldots, A_M\}$ un corpus d'apprentissage, et $S' = \{A_{M+1}, \ldots, A_{M'}\}$ l'ensemble des contenus restants :

– la méthode max rapatrie le contenu A_{j^*} de S' tel que :

$$j^* = \arg\max_j \sum_{i=1}^{M} \frac{f(a_j \oplus a_i)}{M} \tag{4.21}$$

– la méthode min rapatrie le contenu A_{j^*} de S' tel que :

$$j^* = \arg\min_j \sum_{i=1}^{M} \frac{f(a_j \oplus a_i)}{M} \tag{4.22}$$

– la méthode témoin $rand$ rapatrie le contenu A_{j^*}, choisi de manière aléatoire dans S'.

4.5.4.1 Protocole expérimental $\mathcal{P}_3$

Type de tâche : Regroupement ;

Corpus documentaire : Le même corpus que pour $\mathcal{P}_1$ (« plateaux » / « reportages ») ;

Descriptions utilisées : Toutes (32 audio et 36 vidéo) ;

But : Validation de la technique de rapatriement max ;

Série d'expériences D

Trois contenus sont choisis aléatoirement. Nous les superposons ancrés aux couples de coordonnées qui correspondent aux groupes « plateaux » et « reportages », sous contrainte d'avoir un minimum d'un élément par groupe. Le reste des contenus est laissé dans la base documentaire, hors de l'espace dynamique. La procédure d'un test est la suivante :

1. nous rapatrions tous les contenus de la base dans l'espace dynamique, et attendons que l'état de stabilité énergétique soit atteint ;
2. nous notons les performances de classification, évaluée par la méthode du plus proches voisins [32] (les contenus étant superposés sur les coordonnées du même groupe) ;
   (a) si les deux clusters sont linéairement séparables, nous arrêtons le processus ;
   (b) sinon :
      i. nous « rangeons » tous les contenus libres de l'espace dynamique dans la base documentaire, ne laissant sur l'écran que les contenus ancrés ;
      ii. nous constituons le corpus de réorganisation avec la méthode choisie et rapatrions n contenus ;
      iii. nous positionnons ces n contenus sur les coordonnées correspondant à leurs groupes, puis nous les ancrons et recommençons la procédure à son début.

Nous pratiquons deux séries d'expérience :
– une première série de cinq tests pour laquelle $n = 1$;
– une deuxième série de cinq tests pour laquelle $n = 3$.

32. À l'initialisation, $s = 0$. Nous comptons $s = s + 1$ pour chaque contenu s'il est rapatrié à une distance plus proche de son groupe que de l'autre. Pour m contenus à rapatriés, le pourcentage p de classification correcte est donné par la formule $p = s \times 100/m$.

4.5.4.2 Résultats

L'annexe D détaille les résultats numériques propres à ces séries d'expériences.

La figure 4.17 illustre les résultats que nous avons obtenus lors de la première série d'expériences. Les techniques comparées sont min, max et $rand$. Il n'y a pas de technique qui se révèle radicalement meilleure qu'une autre. Toutes donnent des résultats plutôt satisfaisants au sens du plus proche voisin.

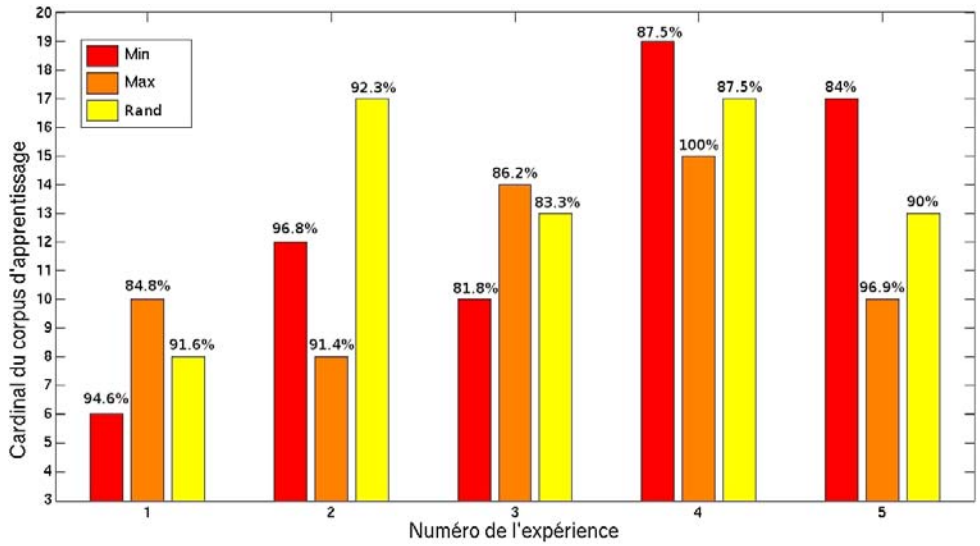


FIGURE 4.17 – Performances du processus sur une tâche de classification : apport incrémental d'un contenu

La méthode max se démarque quelque peu de min de par sa constance en terme de nombre de contenus nécessaires à l'apprentissage, donnant de bons résultats en des temps raisonnables à longs. Cependant, ce n'est pas la qualité des résultats obtenus qui nous a fait choisir la méthode max, mais la philosophie qui s'y rattache.

La réelle difficulté de cette tâche de regroupement est venue du caractère particulièrement hétérogène des contenus « reportage » : nous y avons observé des images fixes commentées, des interviews, des séquences sportives, etc. La série d'expériences numéro 4 a donné des résultats plus mauvais que les autres (en nombre de contenus finaux), car les contenus du corpus d'apprentissage initial n'ont pas su témoigner de cette hétérogénéité.

Observons les figures 4.18 et 4.19, qui, respectivement, retracent cette expérience pour les méthodes min et max. Ces figures montrent l'évolution parallèle de l'erreur d'apprentissage et des performances de classification en fonction du cardinal du corpus.

Pour chaque nouveau rapatriement, nous avons également regardé si le contenu choisi par la méthode était bien (resp. mal) classé (au sens du plus proche voisin). Nous les avons annotés + (resp. -) sur ces figures.

L'analyse de l'erreur d'apprentissage et des performances de classification nous montre plusieurs types de comportements :

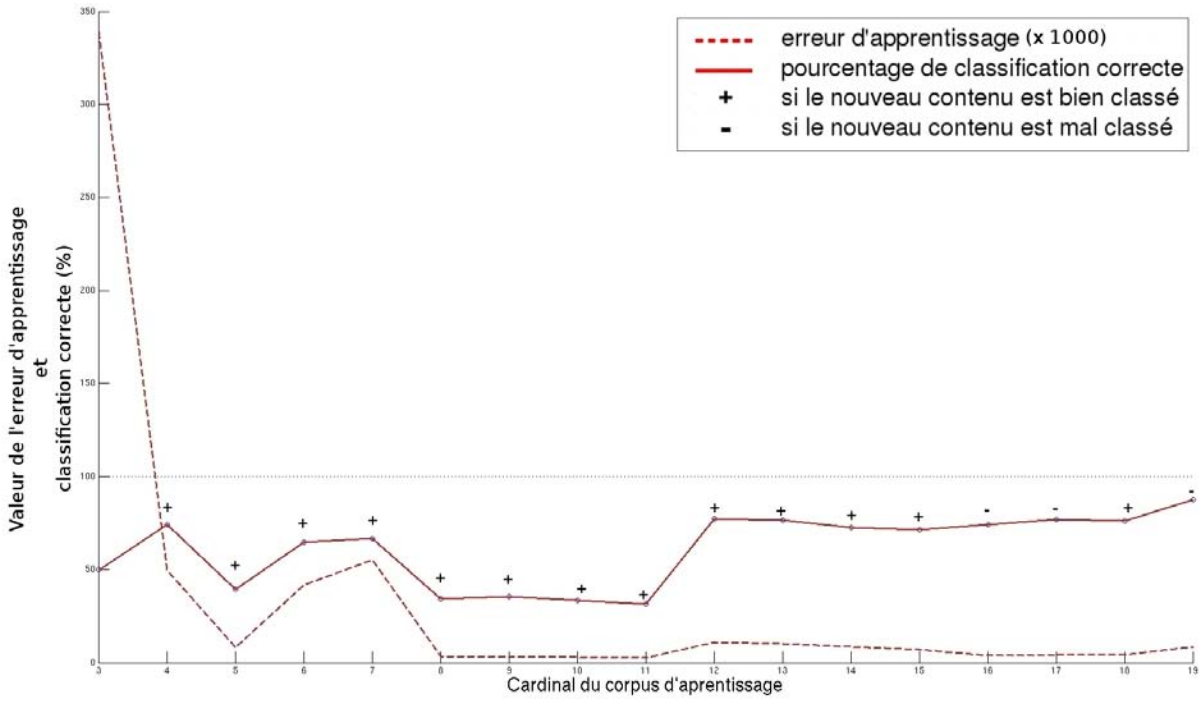


FIGURE 4.18 – Exemple sur le processus min

- des **paliers** les deux courbes évoluent très peu ;
- des **ruptures conjointes** : les deux courbes montent ou diminuent significativement dans une même direction ;
- des **ruptures disjointes** : les deux courbes montent ou diminuent significativement dans des directions opposées ;

La méthode min a tendance à privilégier les ruptures conjointes aux disjointes. Sur la figure 4.18, ce sont les passages de 4 à 5 contenus, de 5 à 6, de 7 à 8, et de 11 à 12.

Nous interprétons ces ruptures comme des phases de remise en question négatives : soit le modèle devient sûr de lui et les performances de classification baissent, soit le modèle semble perdu et les performances de classification s'améliorent.

Nous avons considéré les paliers comme des phases de spécialisation du modèle : l'apport de nouveaux contenus ne fait pas évoluer la mesure ; ils sont majoritairement bien classés et n'apportent donc aucune information déterminante ; le système se spécialise sur ce qu'il a cru comprendre, que cette compréhension soit bonne ou mauvaise.

L'avantage de la méthode min est qu'elle semble générer de nombreux paliers. Dans cet exemple, le problème est que cet avantage a été mal géré, car les paliers ont succédé aux phases de ruptures conjointes, confortant le modèle dans son erreur.

La méthode max semble, quant à elle, privilégier les ruptures disjointes. Sur la figure 4.18, ce sont tous les ajouts de contenus compris dans les intervalles 3 à 6 et 7 à 12. Nous interprétons ces ruptures comme des phases de remise en question positives : soit le modèle semble se perdre

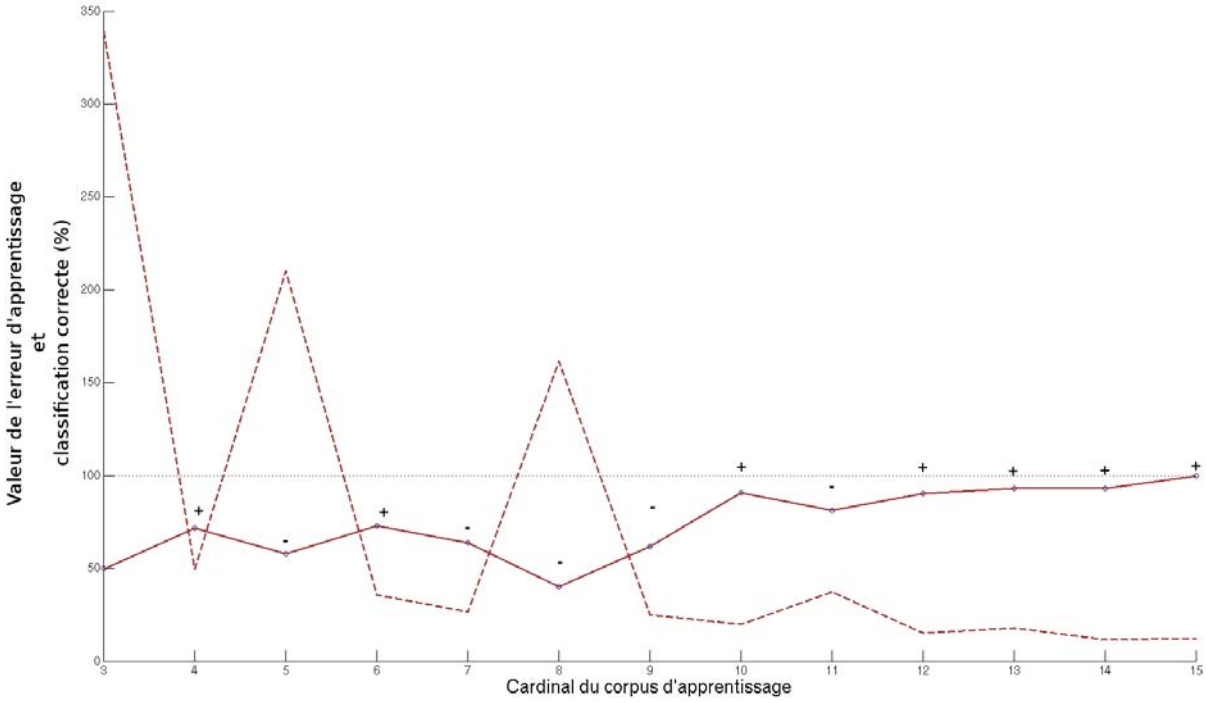


FIGURE 4.19 – Exemple sur le processus max

et les performances de classification baissent, soit il semble comprendre et les performances de classification s'améliorent. L'efficacité du modèle et de la compréhension de la tâche par le système évoluent de concert.

Le problème, dans cet exemple, provient du fait que ces phases ont été trop nombreuses, ce qui a entraîné la convergence vers une spécialisation (de 12 à 15 contenus) trop tardive.

La figure 4.20 présente les résultats finaux de ces deux exemples. Le cluster jaune identifie les contenus étiquetés « reportage ». La vignette de chaque objet informationnel est l'image médiane du contenu vidéo quelle illustre.

La figure 4.21 illustre la deuxième série d'expériences, qui teste les rapatriements de trois contenus à la fois. En proposant une association des méthodes max et min à travers le triplet (max, max, min), nous voulions tirer profit de ces deux techniques : favoriser les phases de ruptures disjointes, par l'apport de deux éléments provenant de max, et apporter un contenu issu de min permettant de spécialiser plus rapidement le modèle.

Les résultats ne sont pas significativement meilleurs en utilisant le procédé (max, max, min). Nous notons toutefois que la technique (max, max, max) se démarque des autres de manière positive : les performances de classifications sont globalement les mêmes que les autres, et les classes deviennent linéairement séparables bien plus vite.

4.5.4.3 Conclusion

Même si la méthode min peut être intéressante dans le sens où elle tend à conforter le système dans ses choix (ce qui a conduit à de très bons résultats dans certains cas), nous

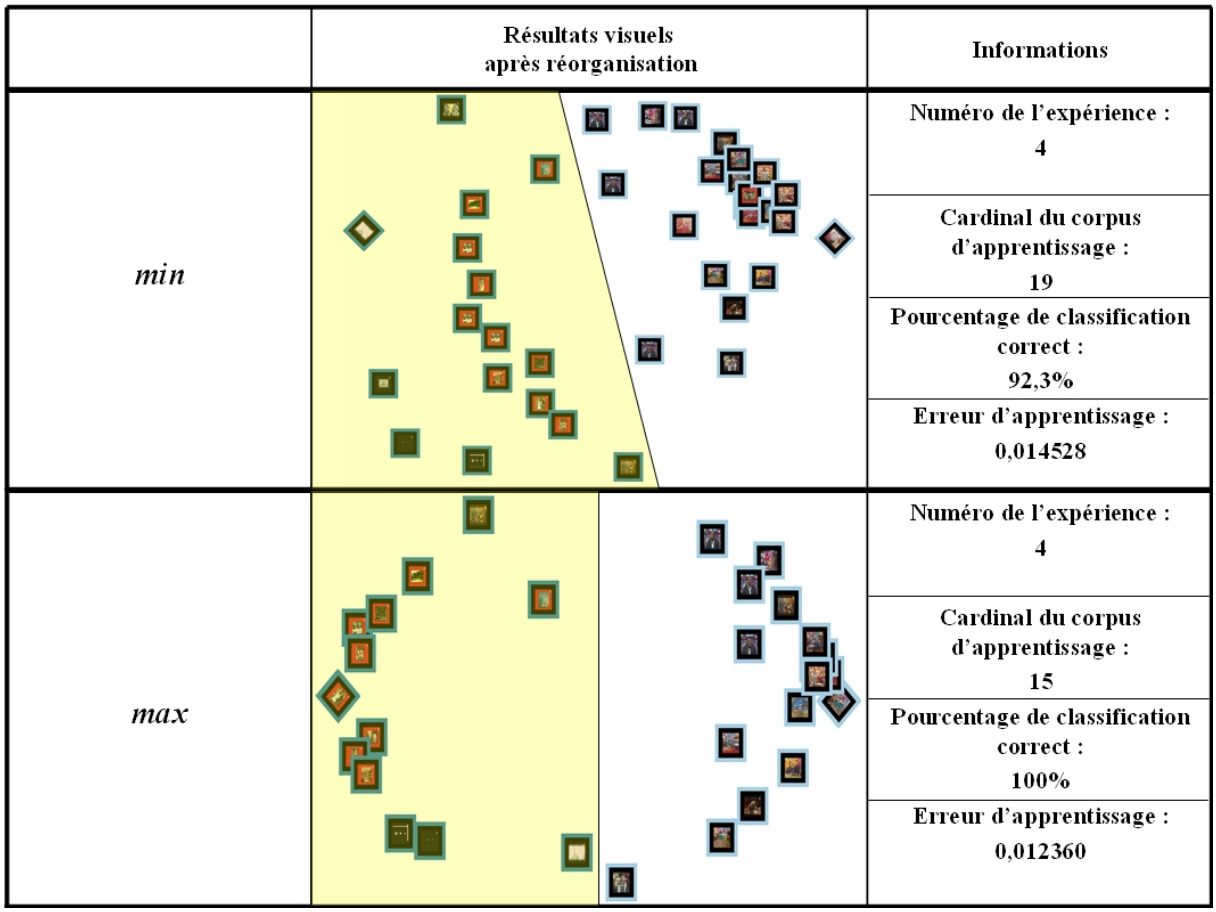


FIGURE 4.20 – Illustration des expériences de D ayant les résultats finaux les plus mauvais en terme de nombre de contenus d'apprentissage

préférons utiliser la méthode max qui, en générant plus des ruptures disjointes, va dans le sens de la philosophie de notre système.

L'élément rapatrié par max est le contenu le plus susceptible de perturber l'équilibre des similarités calculées, et de remettre en question le modèle de manière bénéfique pour la tâche organisationnelle, que celle-ci reste constante ou qu'elle soit modifiée.

4.5.5 Organisation multi-grains

Penchons-nous maintenant sur la question de la multi-granularité. Introduire de telles fonctionnalités dans un mécanisme d'organisation semi-automatique comme le nôtre soulève de nombreuses questions. Le principal problème est identifié par la possibilité d'interagir avec des grains de niveaux différents sur une même interface.

Nous n'avons organisé, dans les précédentes expériences, que des contenus de la granularité la plus faible (des documents). La prochaine série concerne les relations intra-grains (entre contenus de mêmes granularité) de différents niveaux et les relations inter-grains (entre contenus de granularité différentes).

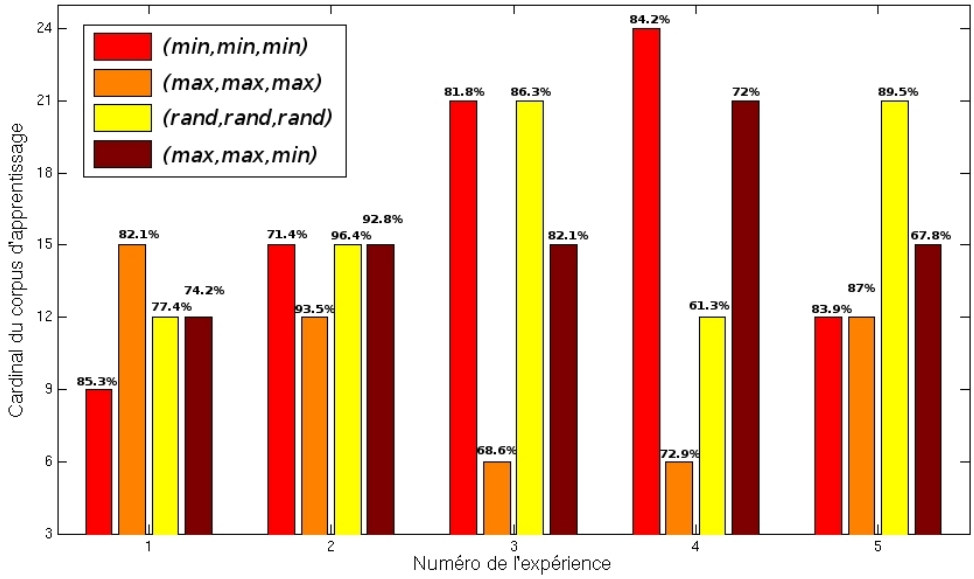


FIGURE 4.21 – Performances du processus sur une tâche de classification : apport incrémental de trois contenus

Pour ce faire, nous avons utilisé notre corpus d'instruments de musique au travers de la taxonomie hiérarchique de Peeters [Pee03] (figure 4.22) : les différentes couleurs identifient dorénavant les instruments (celles présentées dans la figure).

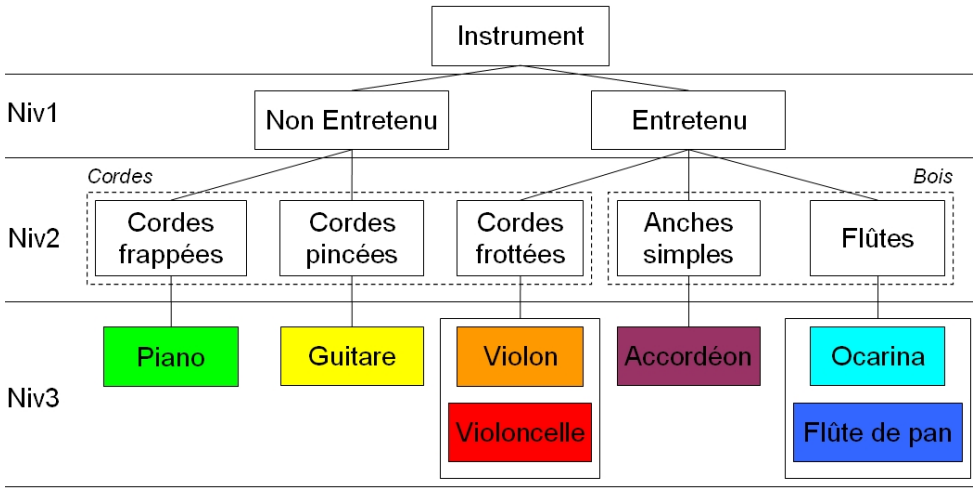


FIGURE 4.22 – Nos contenus musicaux au sein de la taxonomie de Peeters

Cette hiérarchie va nous permettre de définir des grains bien identifiés. Dans cette dernière série d'expériences, nous avons focalisés notre étude sur les valeurs limites du corpus d'apprentissage (se référer à l'annexe E pour plus de détails sur les résultats).

4.5.5.1 Protocole expérimental $\mathcal{P}_4$

Type de tâche : Regroupement ;

Corpus documentaire : Le même corpus que pour $\mathcal{P}_2$, auquel nous avons retiré les notes de piano électrique (qui ne rentrait pas dans la taxonomie), ce qui ramène le nombre d'échantillons sonores à 84 ;

Descriptions utilisées : Toutes les sonores (32) ;

But : Comportement de la mesure sur une tâche intra-grains (série E) et sur une tâche inter-grains (série F) ;

Séries d'expériences E : séries portant sur l'organisation intra-grains, relatives au niveau 1 de la taxonomie.

Les contenus sont regroupés dans une même classe et étiquetés en deux groupes : 16 contenus « Non Entretenus » (constitués des échantillons de guitare et de piano) et 48 contenus « Entretenus » (le reste). Un groupe est identifié par un couple de coordonnées dans l'espace de représentation. Nous avons pratiqué trois séries d'expériences :

- pour la première série (E1), le grain documentaire est la « note » : M contenus d'apprentissage sont sélectionnés de manière aléatoire, ancrés et superposés sur les coordonnées du groupe correspondant, sous contrainte d'avoir un minimum de 1 élément dans le groupe le plus petit.

- la deuxièmes série (E2) porte sur les représentants de groupes restreints. Nous appelons ce grain « notes » : pour chaque instrument, nous constituons 4 classes de 3 notes choisies aléatoirement, ce qui engendre $81/4 = 21$ nouveaux contenus (les représentants des 21 classes). Les M contenus sont choisis et placés de la même manière que précédemment.

- la série (E3) porte sur les représentants des instruments. Nous appelons ce grain « instrument » : toutes les notes d'un même instrument sont regroupés dans une même classe, créant ainsi 7 nouveaux contenus (les 7 représentants de classes). Les M contenus sont choisis et placés de la même manière que précédemment.

Nous avons pratiqué 10 expériences pour chaque valeur de M, et fait varier M sur différentes valeurs, en fonction de la série (voir figure 4.23).

Séries d'expériences F : séries portant sur l'organisation intra-grains, relatives au niveau 2 de la taxonomie.

Les contenus sont étiquetés en cinq groupes : 12 contenus « Cordes frappées » (piano), 12 contenus « Cordes pincées » (guitare), 24 contenus « Cordes frottées » (violon et violoncelle), 12 contenus « Anches simples » (accordéon) et 24 contenus « Flûtes »(ocarina et flûte de pan). Un groupe est identifié par un couple de coordonnées précis dans l'espace de représentation. Nous avons pratiqué deux séries d'expériences :

– la première série (F1) est identique à son homologue (E1) transposée à 5 groupes, avec M testée sur des valeurs limites adaptées (la borne inférieure de 3 contenus a été ramenée à 5 pour respecter le nombre minimum de 1 contenu par groupe). Comme précédemment, tous les contenus sont de même grain (regroupés dans une même classe).
– pour la deuxième série (F2) nous avons créé 5 classes (une classe par groupe). La procédure de tests est identique à la précédente.

4.5.5.2 Résultats de la série de expériences E

Nous souhaitons observer le comportement de la mesure de similarités sur des contenus d'une même granularité, et ce à différents niveaux de grains. Nous nous sommes intéressés aux performances de classifications obtenues au sens du plus proche voisin.

La figure 4.23 illustre les résultats obtenus lors de cette série d'expériences. Compte tenu du nombre réduit de contenus des expériences (E1) et (E2), nous avons adapté le nombre de contenus dans leur corpus d'apprentissage.

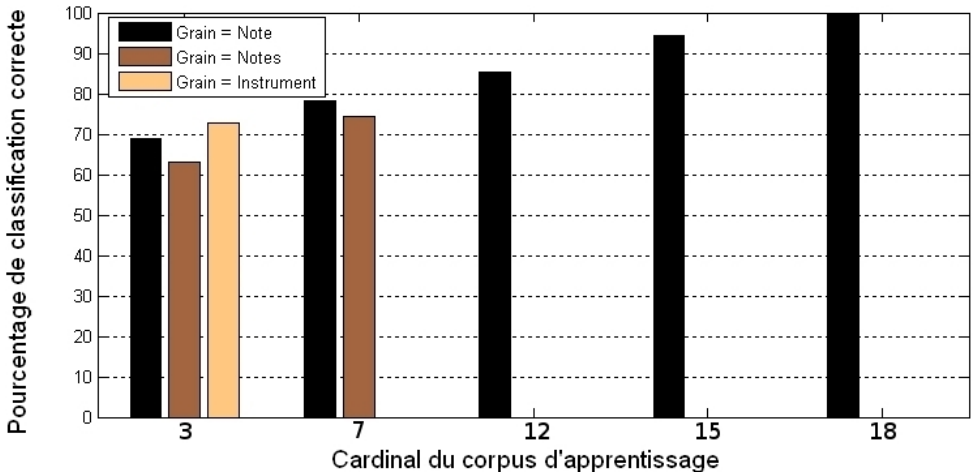


FIGURE 4.23 – Observation des performances sur une tâche d'organisation intra-grains

Quel que soit le grain choisi, pour un même nombre de contenus, les performances sont proches et satisfaisantes. Cela nous encourage à penser l'organisation inter-grains comme pertinente, pour différentes granularités d'un même corpus.

La figure 4.24 illustre les meilleurs résultats de chacune de ces trois séries (le cluster identifié en jaune porte l'étiquette « Entretenu »).

4.5.5.3 Résultats de l'expérience F

À travers (F2), nous souhaitons observer le comportement des contenus libres non classés par rapport à des représentants de classes et ainsi vérifier si une organisation sur deux niveaux de granularité est envisageable. La série (F1) sert de comparatif.

La figure 4.25 montre les résultats obtenus lors de cette deuxième série d'expériences. Les performances de la série de tests (F1) augmentent significativement à mesure que le corpus

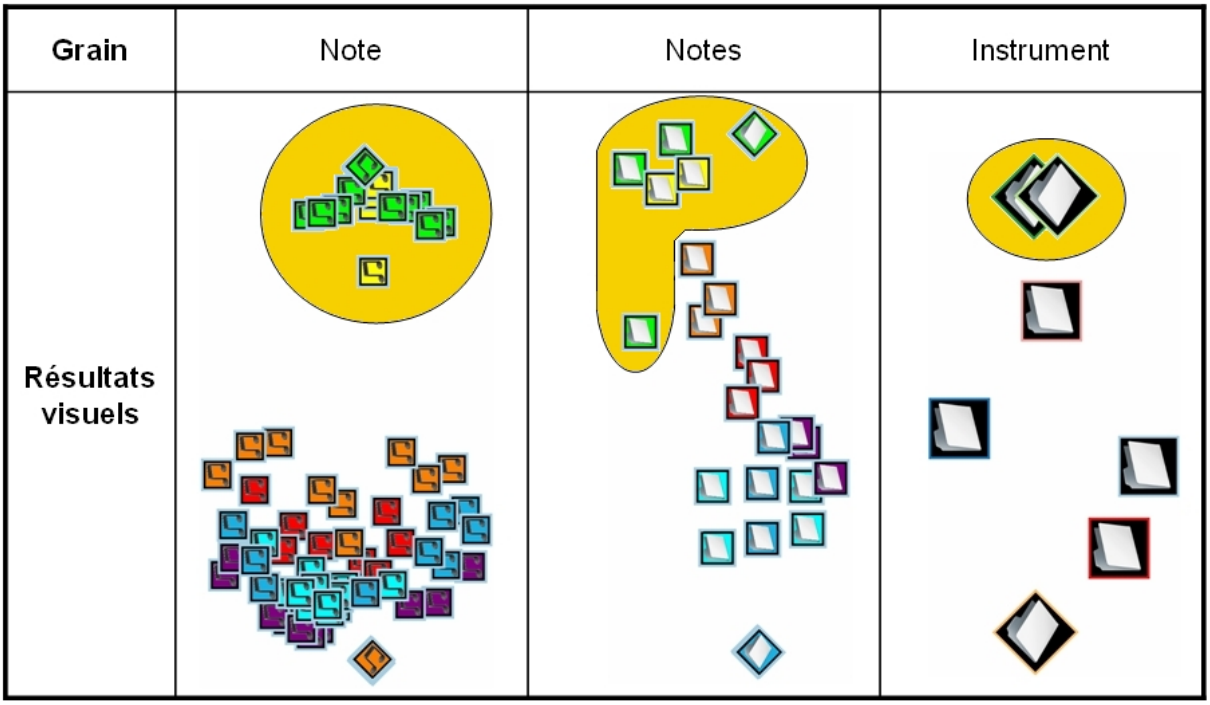


FIGURE 4.24 – Meilleurs résultats pour les expériences E1, E2 et E3 en terme d'erreur de performance de classification

d'apprentissage croît. Celles obtenues avec (F2) évoluent certes, mais très faiblement et restent mauvaises.

Nous pouvons noter les valeurs très proches des performances de ces deux séries lorsque le cardinal du corpus est égal à 5 : dans ce cas, chaque classe de (F2) ne comporte qu'un seul élément, le représentant de la classe n'est autre que ce contenu et nous nous retrouvons dans la même configuration que pour (F1).

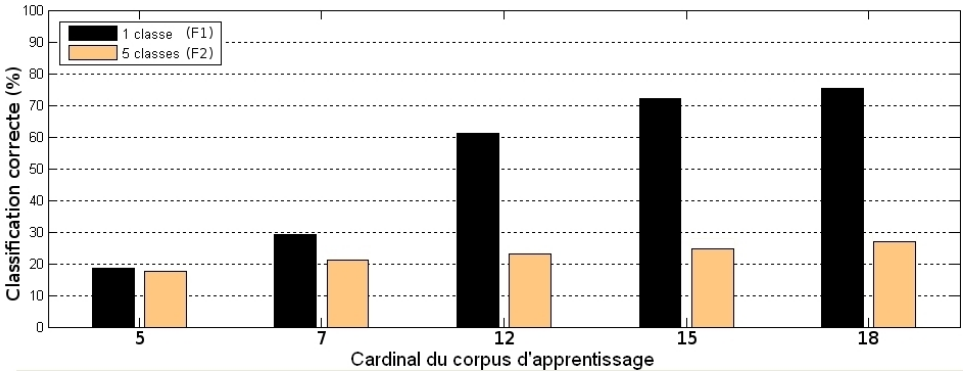


FIGURE 4.25 – Comparaison de performances entre une organisation intra-grains et inter-grains

La figure 4.26 montre les meilleurs résultats (au sens du plus proche voisin) de ces deux séries de tests. Les vignettes circulaires ont été rajoutées pour identifier les différents groupes

en fonction de leurs couleurs (les hachurés rassemblent deux instruments). En terme de visualisation, bien que des rassemblements puissent être observés dans les résultats obtenus avec (F2), ils n'ont rien de comparables avec les groupes identifiables dans la série (F1).

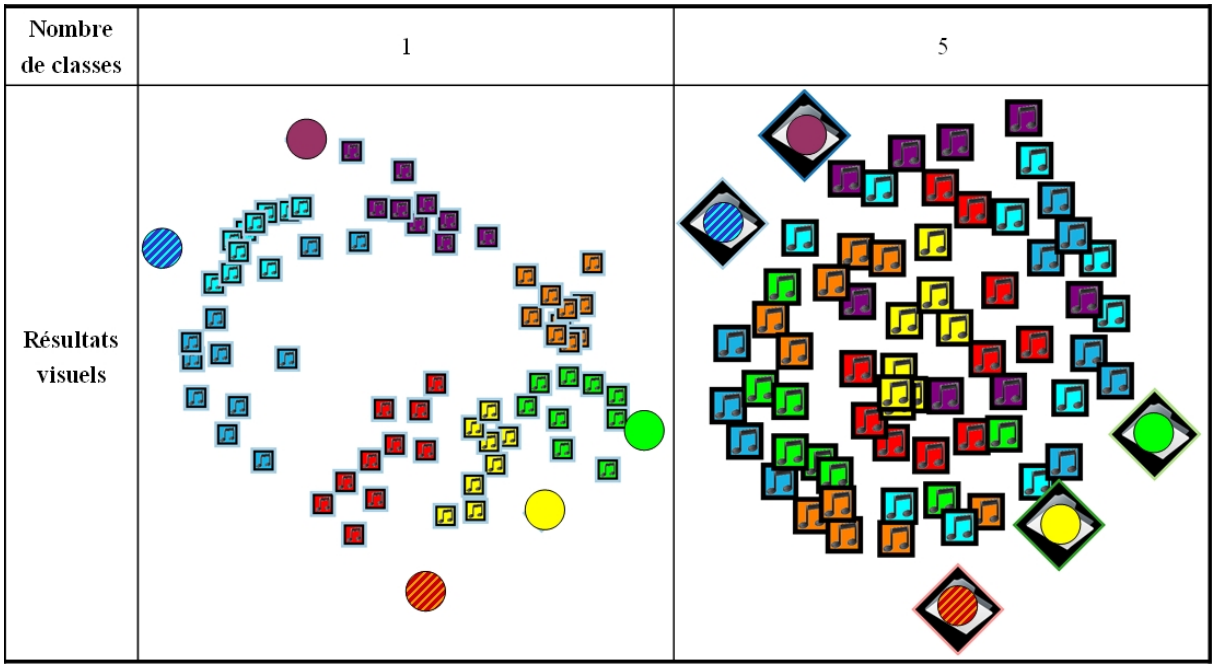


FIGURE 4.26 – Meilleurs résultats pour les expériences F1 et F2 en terme d'erreur de performance de classification

L'explication de la différence de résultats entre ces deux séries vient de notre modélisation du représentant de classe et du formalisme que nous avons conçu :

lorsque nous rajoutons un élément à un corpus de M contenus :

- dans (F1), M nouvelles valeurs de distances sont générées. L'apprentissage se fait sur plus d'élément ce qui explique l'affinage de la mesure de similarités et l'amélioration croissante de (F1), phénomène amplement observé au cours de toutes les précédentes expériences ;
- dans (F2), le nouveau contenu n'intervient que pour modifier le vecteur des descriptions du représentant de la classe correspondante. Il n'a pas d'influence sur le nombre de distances entrant en jeu dans la construction de la mesure de similarités qui influence les contenus non classés (ce nombre reste constamment à 5). Changer les valeurs des descriptions d'un élément ne prodigue pas suffisamment d'informations pour améliorer significativement la mesure.

Nous illustrons par cette expérience la difficulté qu'il peut y avoir à vouloir faire de l'organisation inter-grains. En effet, la mise en œuvre du représentant de classe tel que nous l'avons pensé implique la double assimilation suivante :

1. un groupe d'objet est traité comme un objet ;
2. un ensemble de valeurs descriptives est traité comme une unique valeur descriptive.

Dans la section 1.2.3.3 du premier chapitre, nous qualifions de « cohérente » la démarche de comparer des contenus provenant d'un même gain, intuition confirmée par les résultats de la série E.

Cela laissait toutefois sous-entendre qu'une confrontation de contenus issus de granularités différentes pouvait ne pas garantir cette cohérence, démonstration faite lors de cette dernière série d'expériences.

4.5.5.4 Conclusion

La conception d'un représentant de classe comme la moyenne des descriptions des contenus de grains inférieurs peut être intéressante pour de l'organisation intra-grains, à condition que les contenus soient relativement homogènes.

Le problème de la classification inter-grains, reste quant lui, bien entier, et un représentant décrit sous cette forme ne permet pas de l'appréhender correctement.

Nous restons également dubitatifs sur son utilisation au sein d'une hiérarchie trop poussée ou avec des contenus trop hétérogènes, la fonction de moyenne risquant de beaucoup trop diluer l'information portée par les descriptions des grains inférieurs.

Nous pensons que l'organisation inter-grains est envisageable aidée du concept de représentant, si toutefois la modélisation de ses descriptions est repensée de manière plus fine. À notre avis, l'exploration d'une telle piste constitue à elle seule une charge de travail équivalente à celle fournie pour le reste des sujets abordés dans ce manuscrit.

4.5.6 Évaluation utilisateur

Cette dernière série d'expériences porte sur l'évaluation des fonctionnalités de notre système, utilisé sur un corpus moins artificiel que les précédents. Nous avons intégré des extraits vidéo provenant de l'offre Internet « Archives pour tous » de l'Ina, et avons proposé à une vingtaine de personnes d'utiliser notre outil pour les organiser.

Pour ce faire, nous leur avons présenté succinctement les principales fonctionnalités tout en omettant volontairement de prononcer certains termes par trop révélateurs. Cela nous a permis d'analyser leur comportement et de relever si les concepts fondamentaux propres à notre système sont compréhensibles sans que nous ayons à les introduire.

Nous avons choisi des personnes hors de notre environnement scientifique et industriel, sans avoir connaissance de leur catégorie socioprofessionnelle.

4.5.6.1 Protocole expérimental $\mathcal{P}_5$

Type de tâche : Organisation ;

Corpus documentaire : 100 contenus issus du corpus de l'Ina d'une durée totale de 306 minutes ; ils proviennent de cinq collections différentes (voir figure 4.27) :

- collection 1 : le documentaire d'investigation « **Cinq colonnes à la une** » ;
- collection 2 : l'émission d'informations « **Les actualités françaises** » ;
- collection 3 : l'émission de divertissement « **La minute de Monsieur Cyclopède** » ;
- collection 4 : le magazine thématique « **Reflets de Cannes** » ;

– collection 5 : l'émission de sport « **Spécial sport** » ;
Nous avons utilisé 20 contenus de chacune de ces collections ;

Descriptions utilisées : Toutes (32 audio et 36 vidéo) ;

But : Organiser des collections de contenus de l'Ina et apprécier la compréhension de différents concepts propres au système ;

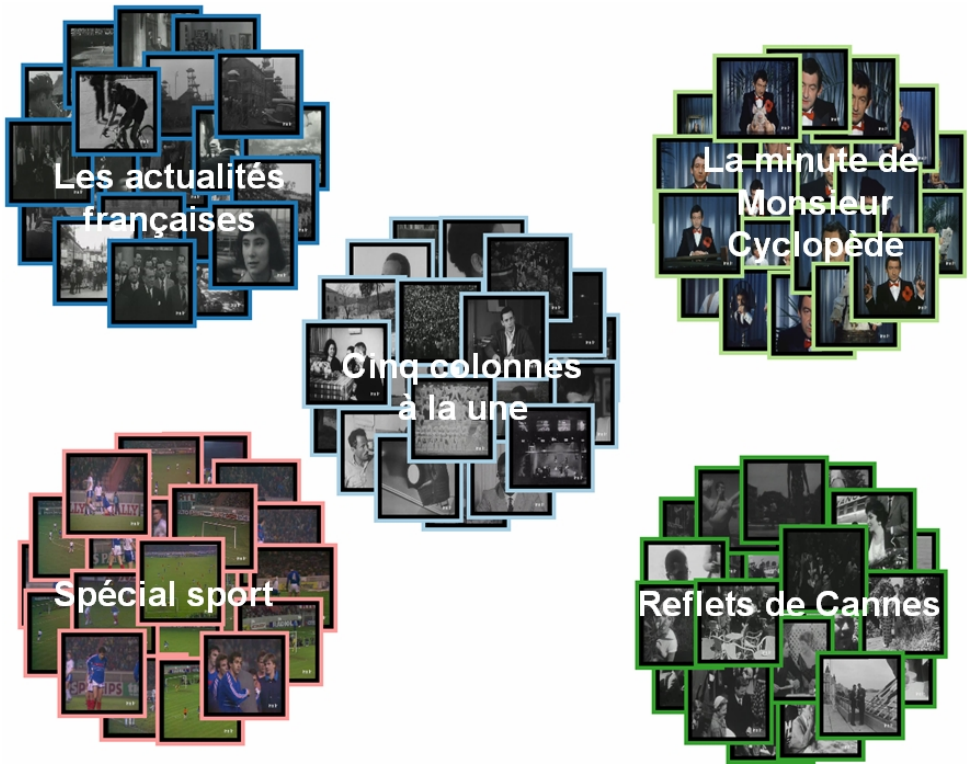


FIGURE 4.27 – Aperçu des différentes collections

Expérience G

Trois contenus sont choisis aléatoirement, présentés dans l'espace dynamique, puis regroupés dans une même classe. Nous avons testé l'outil sur leur organisation pour évaluer la faisabilité de la tâche.

Séries d'expériences H

Les trois mêmes contenus sont présentés classés dans l'espace dynamique. Nous avons demandé à 20 utilisateurs de tester notre système. Après leur avoir présenté les différentes fonctionnalités, nous leur avons demandés d'organiser en moins de 30 minutes (soit environ un dixième de la durée totale nécessaire à leur visionnage) la base documentaire de contenus de l'Ina dans le but de créer des DVD. Aucune autre information ne leur a été confiée à propos du corpus (se référer à l'annexe F pour plus de détails).

Nous avons pris soin de ne pas évoquer les termes suivants : similarité, classe, représentant de classe, grain, granularité, contenus audiovisuel.

Nous n'avons pas précisé que les classes évoluaient de manière indépendante les unes des autres, et nous n'avons rien dit au sujet des corpus étudiés (nombre d'extraits par collections, genres, thématiques, etc.). Quelques remarques :

- la zone d'informations est gardée vide, ne reste que la vignette ;
- le terme « Dynamique » utilisé dans l'interface et employé pour caractériser l'activation du mode assisté, a été remplacé par le mot « Aide » pour l'expérience ;
- la possibilité de faire des dossiers a été enlevée ;
- lorsque plusieurs classes sont créées, les procédures d'apprentissages liées à la création de leurs régressions sont séquentielles (non parallélisées), le temps de réponse du système est affecté en conséquences.

4.5.6.2 Résultats de l'expérience G

Dans l'état actuel d'avancée de notre prototype, le scénario idéal, illustré par la figure 4.28, est le suivant :

étape 0 : prise de connaissance de la base :
- visionnage des trois contenus proposés par l'ordinateur ;
- parcours rapide de la base documentaire en faisant défiler le bordereau de droite à gauche ;

étape 1 : une première tâche paraît évidente sans consulter d'autres contenus : séparer les extraits *couleur* des *noir et blanc* :
- dans une même classe, séparer certains contenus en couleurs (collections 3 et 5) des noir et blanc (collections 1, 2 et 4) ; les ancrer ; rapatrier des contenus ; recommencer jusqu'à satisfaction ;
- rapatrier l'ensemble de la base ;
- créer une classe constituée des contenus en couleurs (résultat figure 4.28 étape 1) ;

étape 2 : dans la classe des contenus en couleur, séparer les contenus de chacune des deux collections, facilement différentiables grâce à la vignette représentative (sans nécessité de visionnage) ;
- séparer quelques contenus ; les ancrer ; réorganiser les contenus libres ; recommencer jusqu'à satisfaction ;
- constituer une sous-classe avec une des deux collections (résultat figure 4.28 étape 2) ; ne laisser ancrer qu'un seul contenu dans chacune des classes, de manière à contraindre géographiquement les deux groupes tout en évitant de nouveaux calculs de similarité intra-grains inutiles ; déplacer les deux classes dans un endroit isolé ;

étape 3 : pour la classe des contenus noir et blanc, la vignette n'a plus d'utilité. À l'écoute, nous nous apercevons vite de la différence de qualité sonore entre les contenus, bien plus faible pour les « actualités françaises » que pour les deux autres collections. De plus, le présentateur intervient systématiquement et sa voix est extrêmement caractéristique de cette collection.
- dans la classe des contenus noir et blanc, écouter un extrait ; le disposer en fonction du son ; l'ancrer ; réorganiser les contenus libres ; recommencer jusqu'à satisfaction ;
- constituer une sous-classe des actualités françaises (résultat figure 4.28 étape 3) ; désancrer tous les contenus sauf un ; déplacer la classe dans un endroit isolé ;

étape 4 : la dernière séparation est moins triviale. Nous n'avons pas noté de caractéristiques propres au signal permettant de segmenter ces contenus. Nous avons donc joué sur les sujets évoqués. Ceux provenant de « Cinq colonnes à la une » sont extrêmement variés, mais la thématique du festival de Cannes, abordée sur l'ensemble de la collection « Reflets de Cannes », a grandement facilité la séparation. La procédure est la même que pour l'étape 3.

Les résultats de l'aide automatique sont moins bons qu'avec les autres collections, à cause de la forte ressemblance de ces contenus d'un point de vue « signal ». Nous nous sommes donc arrêtés de l'utiliser après avoir ancré une douzaine de contenus (résultat figure 4.28 étape 4) et avons dû visionner partiellement (jusqu'à évocation du thème dans l'archive) tous les contenus de ces deux collections pour les séparer correctement.

Nous avons mis 21 minutes pour organiser ces documents, soit environ quinze fois moins de temps qu'il n'en faudrait pour visionner l'intégralité du corpus. Un tel scénario ne peut aboutir qu'en pleine connaissance des atouts et des failles de notre système. Nous n'espérons pas qu'un utilisateur arrive à réaliser une telle performance, mais cela donne une idée de la fonctionnalité de notre outil dans un cadre moins théorique que dans les expériences précédentes.

4.5.6.3 Résultats de la série d'expériences H

Le tableau 4.6 résume les différentes mesures que nous avons relevées durant ces expériences. Ces résultats servent de support à la compréhension des comportements que nous avons pu analyser, et qui seront expliqués à la suite de ce tableau.

Déroulement global de l'expérience

L'expérience a été très majoritairement vécue en deux phases :

1. une phase d'exploration de l'outil ;
2. une phase d'exploitation des fonctionnalités pour atteindre l'objectif.

Durant la phase d'exploration, l'aide est sollicitée via des rapatriements assistés et non assistés d'un petit nombre de contenus (un à trois maximum). Les premiers contenus sont visionnés, déplacés et ancrés. Aucune nouvelle classe n'est créée autre que celle proposée par la machine. La prise de conscience des fonctionnalités propres à l'aide s'est souvent faite de manière brutale, suite à une activation du mode dynamique succédant à un oubli d'ancrage (tous les contenus sont alors mélangés au centre de l'écran) ou à un ancrage de trop de contenus (le temps d'attente est alors trop long). Une fois cette phase de transition passée, les actions se sont tournées vers la création de nombreuses classes et le rapatriement massif de documents (dix par dix / sélection de beaucoup de contenus / rapatriement de l'ensemble de la base).

La tâche de constituer des DVD a été correctement interprétée par les utilisateurs qui ont tous associé la notion de DVD à la notion de classe. Les deux collections en couleur ont été retrouvées et complétées facilement (le rapatriement par sélection a beaucoup aidé dans ce cas précis, car les vignettes étaient très significatives). Les actualités françaises ont été mieux classées que les autres contenus en noir et blanc. Ces dernières ont énormément évolué à mesure

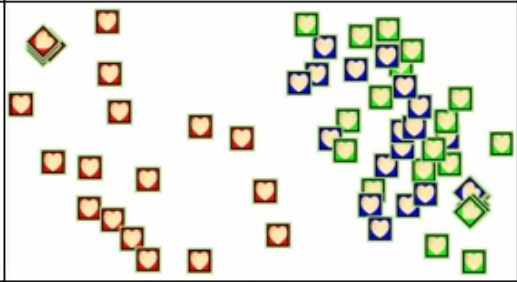
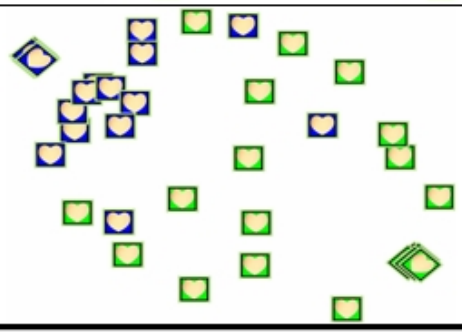
Étape	Résultats visuels	Informations
1		**Cardinal du corpus d'apprentissage**: 8 **Descriptions conservées:** Mean(1ère valeur dominante) Mean (2ème valeur dominante)
2		**Cardinal**: 6 **Descriptions**: Min (2ème valeur dominante) Min (1ère teinte dominante)
3		**Cardinal**: 10 **Descriptions**: Var (*RollOff point*) Var (Mod. de l'énergie à 4Hz) Min (Taux d'activité)
4		**Cardinal**: 12 **Descriptions**: Var (Mod. de l'entropie) Mean (Mod. de l'entropie) Max (contrast) Mean (Mod. de l'énergie à 4Hz) Min (2ème valeur dominante)

FIGURE 4.28 – Résultats obtenus à la fin des différentes étapes (dans les deux dernières étapes, les vignettes ont été changées par des couleurs pour mieux discerner les différentes collections)

que l'utilisateur prenait connaissance de la base, les changement d'avis remettant régulièrement en question l'organisation.

Nous avons d'ailleurs pu apprécier des organisations variées, différentes de celles attendues, qui s'engageaient pleinement dans l'esprit avec lequel nous avions originellement pensé (et non évalué) notre outil. Par exemple, certains épisodes de l'émission « Reflet de Cannes » utilisant la métaphore de la guerre pour présenter les différents artistes (« *Mais voici qu'une autre bombe se profile à l'horizon. Cet avion qui atterrit contient dans son flanc Suzanne Hayward [...]* », « *Pour*

TABLE 4.6 – Mesures moyennes observées au cours de la série d'expériences H

Donnée analysée	**Valeur moyenne**
Temps écoulé avant de créer une classe	9'17"
Nbre de contenus visionnés	12,3
Nbre de contenus visionnés et ancrés	10,1
Nbre de contenus rapatriés	85,1
Nbre de contenus rangés dans la base	0,2
Nbre de contenus ancrés	17,5
Nbre de contenus désancrés	0,1
Nbre de contenus classés	74,2
Nbre de contenus déclassés	3,1
Nbre de contenus changés de classes	24,6
Nbre de classes créées	6,6
Pourcentage de contenus rapatriés convenablement classés	72,2%

ne pas demeurer en reste avec l'aviation atomique de Yul Brynner, la marine française [...] », « *[...] les Russes préparent leur première offensive [...]* » en parlant des diverses personnalités soviétiques du festival, etc.) ont été regroupés avec des « Actualités françaises » et des « Cinq colonnes à la une » portant sur cette même thématique.

Satisfaction liée à l'organisation de l'espace de travail

Les fonctionnalités les plus appréciées furent celles liées à l'organisation de l'espace de travail. L'ancrage a plus été exploité comme un outil de contrainte géographique que comme un guide pour aider au rangement. La possibilité de créer et de modifier facilement des classes a plu : les trois quarts des contenus ont été classés ; un quart des documents manipulés ont transité d'une classe à une autre. Très peu de contenus ont été déclassés ou rangés dans la base de données.

Notons pour finir quelques autres comportements observés que nous espérions voir émerger :

- les contenus mal positionnés suite à un rapatriement sont manipulés en priorité pour être ancrés au bon endroit ;
- les documents ancrés sont majoritairement ceux qui sont visionnés, confirmant ainsi l'hypothèse que nous avions avancée sur l'importance de la charge cognitive qui pèse sur ce type de contenus ;
- certains utilisateurs ont ancré des contenus non classés (volontairement déclassés ou rapatriés tels quels) lorsqu'ils avaient des doutes sur l'appartenance d'un élément à une

classe constituée.

Frustration liée à l'organisation de l'espace des données

D'une manière générale, les mécanismes d'apprentissages sous-jacents n'ont été saisis que partiellement. Le problème vient essentiellement du fait que l'organisation des contenus se comporte différemment suivant qu'il y ait une ou plusieurs classes à l'écran :

- lorsqu'une unique classe est présente, les contenus rapatriés sont automatiquement classés dans celle-ci et le système se comporte bien ; cela correspond à la phase d'exploration, pendant laquelle l'utilisateur s'est fait une idée légitime du comportement du système, à travers les mécanismes liés à l'organisation intra-grains ;
- lorsqu'elle pense avoir compris la machine (à tort ou à raison), la majorité des utilisateurs crée de multiples classes et demande un rapatriement de nombreux contenus ; nous nous trouvons alors dans une configuration d'organisation inter-grains qui, comme nous l'avons souligné dans la série d'expériences F, fonctionne mal. Le représentant n'a pas suffisamment de poids pour attirer à lui les éléments non classés qui lui sont similaires. La mesure de similarités créée manque de pertinence, ce que pensait avoir compris l'utilisateur disparaît. De plus, nous ne fournissons pas d'indicateur permettant de savoir si le système a mal compris la proposition ou s'il est simplement perdu.

Ce problème a également porté préjudice au processus de rapatriement incrémental : notre algorithme de choix d'un nouveau contenu n'est réellement utile que si une seule classe est présente dans l'espace dynamique. Au delà, l'efficacité de cette méthode n'a pu être révélée.

Nous avons noté qu'une fois dérouté, seul un utilisateur a pensé revenir sur ses décisions, en désancrant des contenus et en rangeant des éléments dans la base documentaire pour y voir plus clair. D'autres personnes ont tenté de modifier leurs comportement vis à vis du système, mais eurent de mauvaises intuitions (créer plus de classes, associer des éléments dissimilaires, etc.).

Discussions autour du système

Une fois l'expérience terminée, nous nous sommes renseignés sur ce que les utilisateurs avaient compris ou pas du système, sans les aiguiller sur les réponses. Le tableau 4.7 recense le pourcentage de personnes ayant évoqué spontanément les notions attendues :

- la notion de similarité est revenue dans la bouche de 85% des utilisateurs, essentiellement vécue avec la création des classes ; six d'entre eux se sont favorablement servis de cette similarité pour établir des relations interclasses, en les regroupant soit par thèmes soit par caractéristiques physiques ;
- un quart des personnes a relevé les phénomènes liés à l'indépendance des classes, même si tous ont compris cet aspect de l'outil (du moins concernant l'espace de travail) ;
- 15% des utilisateurs ont noté le caractère adaptatif de la mesure, évoqué à travers la notion d'intelligence artificielle ;
- un dixième des personnes a évoqué la granularité à travers le terme de hiérarchie ;
- aucun n'a pensé à interpréter une classe comme un unique élément, de par le manque de consistance de la mesure sur de l'organisation inter-grains.

TABLE 4.7 – Termes recensés après l'expérience

Concepts compris spontanément	**Nombre moyen de personnes**	**Termes employés** (nbre de personnes)
Similarité	85%	proche (15), loin (14), ressemblant (8), semblable (3), différent (6)
Classes indépendantes	25%	bouger séparément (1), régions autonomes (1), bouger ensemble (3)
Représentant	0%	
Granularité	10%	hiérarchie (2)
Dynamique (dans le sens adaptatif)	15%	adapter (1), évoluer (2)

4.5.6.4 Conclusion

Avec l'expérience G, nous avons montré que notre système propose une aide non négligeable pour organiser des contenus audiovisuels qui possèdent certaines spécificités propres à leur signal et fonctionne moins bien lorsqu'il s'agit d'établir des relations plus sémantiques entre les éléments.

La série H nous a permis de tester notre prototype avec des utilisateurs du grand public. Les fonctionnalités propres à l'espace de travail ont été bien appréciées, mais les mécanismes d'apprentissage sous-jacents n'ont majoritairement pas été saisis. Il est primordial d'aller dans le sens d'une homogénéisation des comportements du système dans les organisations intra-grains et inter-grains pour remédier à ce problème.

4.6 Conclusion

Ce chapitre a débuté par une présentation détaillée de notre prototype. Son moteur d'apprentissage utilise une régression ε-SVR univariée, fondée sur la concaténation de valeurs descriptives sélectionnées par un algorithme de type SFFS. Ses hyper-paramètres sont optimisés grâce à une méthode de recherche partielle par grille. Son moteur de visualisation est constitué d'un algorithme de placement de graphe par modèle d'énergie utilisant RK4, agrémenté d'un mécanisme de stabilisation énergétique artificiel.

Nous avons ensuite testé la mesure de similarité qu'il est capable de produire sur plusieurs séries d'expériences en lien avec les différentes problématiques relevées dans ce document.

Ces tests ont débuté par l'évaluation des performances théoriques de notre moteur d'apprentissage. Les résultats sont concluants en termes de temps de calcul (sous contrainte d'un nombre limité de contenus d'apprentissage), d'optimisation des hyper-paramètres et d'attributs sélectionnés.

La relation de linéarité existant entre erreur d'interprétation et erreur de prédiction nous encourage à considérer la régression non-linéaire univariée comme un bon outil pour modéliser la similarité extraite d'une tâche organisationnelle.

En ce qui concerne le moteur de visualisation, les erreurs liées à la transformation de la matrice de similarités (résultat de la prédiction) en matrice de distances sont relativement faibles, ce qui nous conforte dans le choix du modèle d'énergie pour relier ces deux univers.

La conception de notre mesure s'est faite sur un principe d'apprentissage « sans mémoire » (la mémoire de travail est effacée à chaque nouvel apprentissage). En ce sens, nous avons exploité l'apport incrémental de contenus comme mécanisme d'aide à la compréhension d'une tâche organisationnelle évolutive. Les résultats obtenus sont intéressants et la méthode de rapatriement implantée nous garantit de bons résultats pour une taille réduite du corpus d'apprentissage.

Nous avons défriché le problème de la multi-granularité à l'aide du représentant de classe et constaté de bons résultats concernant une organisation intra-grains sur différents niveaux d'une hiérarchie peu profonde, sachant les contenus de natures relativement homogènes. Toutefois, le problème de l'organisation inter-grains n'a pu être traité à l'aide de notre proposition, mais ne paraît pas insoluble et mériterait une attention particulière.

Enfin, l'étude du comportement de différents utilisateurs sur une tâche d'organisation de contenus de l'Ina nous a permis de vérifier si la théorie pouvait être mise en pratique. Les différents points forts et faibles révélés dans les précédentes expériences ont été mis à jour, nous confortant ainsi sur certaines de nos décisions et nous aiguillant sur les priorités des futures améliorations de notre système.

Conclusion et perspectives

1 Conclusion générale

Nous avons abordé dans ce document la problématique de l'organisation de contenus audiovisuels par l'apprentissage de similarités. Pour ce faire, nous avons créé un système expérimental semi-automatique qui illustre notre réflexion sur la manière de concilier la perception que peuvent avoir un humain et une machine d'un contenu audiovisuel. Notre prototype s'inspire de principes provenant directement de différents domaines : Indexation audiovisuelle, Statistiques, Interface Homme Machine, Apprentissage (semi-)supervisé, Fouille de donnés... Il se fonde sur la combinaison d'un formalisme de visualisation, d'une régression non-linéaire univariée et d'un algorithme de placement de graphe par modèle d'énergie.

En vue de proposer de nouvelles utilisations des archives de l'Ina, nous avons pris le risque d'aborder une problématique plutôt généraliste, s'appuyant sur des usages réels difficilement définissables et mesurables, mais dont l'étude s'est révélée extrêmement intéressante. Notre contribution se situe sur plusieurs plans.

Nous avons tout d'abord reconsidéré le document audiovisuel pour en faire un contenu, dérogeant ainsi à la propriété de linéarité temporelle. Le principe d'organisation vu sous un angle global n'a jamais, à notre connaissance, été abordé dans la littérature.

Nous avons créé un formalisme de visualisation structuré par un graphe, qui utilise la notion de similarité pour s'adapter à des tâches de regroupement et d'ordonnancement potentiellement instables. La proposition que nous avons faite concilie tous les aspects propres au contenu audiovisuel que nous avons pu mettre en saillance. Un des aspects pointés lors de ces travaux est l'exploration multi-granularité. Nous avons abordé cette problématique par la mise en valeur de la notion de généralisation sous la forme du représentant de classe.

Nous avons ensuite été amenés à définir un modèle numérique de similarité. La prédiction de similarités fondée sur une régression univariée constitue une approche intéressante de l'utilisation de cet outil statistique dans le cadre de l'organisation de contenus audiovisuels. Son étude a débouché sur un algorithme fonctionnel, fondé sur une optimisation partielle des hyperparamètres d'une ε-SVR par une approche itérative pilotée par sélection séquentielle d'attributs.

Enfin, nous avons implanté un prototype afin de mettre en œuvre les différents points techniques et théoriques abordés dans ce manuscrit. Les résultats obtenus sur nos différentes expériences sont encourageants, et constituent une bonne base de connaissances exploitable pour de futurs travaux sur le sujet.

2 Perspectives

Nous avons conscience que le choix des différentes briques qui composent notre outil peut paraître arbitraire. Toutefois, ce sont les travaux menés sur la mise en relation de ces parties de prime abord très distinctes qui ont stimulé notre étude, et c'est avec regret que nous laissons presque inexplorées les différentes pistes mises en valeur dans ce manuscrit.

Nous disons « presque », car les idées ont eu le temps de germer même si elles ne furent pas toutes mises en œuvre dans cette étude. Nous en présentons quelques unes dans cette dernière section.

2.1 Moteur d'apprentissage

Nous pouvons, dans un premier temps, rester sur une vision très pragmatique du problème, et penser l'organisation comme un problème d'optimisation statistique encore irrésolu.

Nous pensons que sa modélisation sous la forme d'une régression non-linéaire univariée est une bonne piste, et la ε-SVR un bon outil de base pour créer une mesure de similarités pertinente. Cependant, les technologies que nous avons employées pour construire notre algorithme sont critiquables. Une meilleure solution doit pouvoir se trouver avec une méthode techniquement plus performante que la nôtre.

Un algorithme glouton de sélection d'attributs reste coûteux en terme de complexité. Il serait bien venu de tester d'autres méthodes sous-optimales plus rapides même si elle sont moins efficaces. Nous avons pensé utiliser des algorithmes d'extractions simples (comme *PCA* ou *ICA*) qui pourraient offrir de meilleurs résultats que SFFS (au moins en temps de calcul).

Une expérience simple et intéressante serait de tirer profit des deux méthodes en les combinant de la manière suivante :

- choisir les descriptions principales avec un algorithme de sélection plus rapide qui ne déforme pas la topologie de l'espace de description (afin de conserver une certaine forme de compréhension de l'espace de description) ;
- compenser ensuite les lacunes de celui-ci en opérant une extraction sur le sous-ensemble final des descriptions.

Notre prototype utilise un autre algorithme glouton pour optimiser les hyper-paramètres. Même si elle garantit de bons résultats théoriques, cette recherche exhaustive reste coûteuse, et ne permet pas l'incorporation décente d'un troisième attribut (ε dans notre cas). Essayer d'autres méthodes de la littérature (méthodes de descente du gradient, algorithmes évolutionnaires) pourrait être intéressant.

Toujours concernant les hyper-paramètres de ce type de régression, nous aurions aimé avoir le temps de nous attarder sur l'influence que pouvait avoir chacun d'entre eux. C'est ce que nous avons initié en étudiant brièvement ε par rapport à $\mathbb{C}$ et γ, mais la valeur de 0,1 reste arbitraire et contestable. De même, nous avons observé un temps de calcul bien plus grand lorsque $\mathbb{C}$ se rapproche de la borne supérieur de la grille, mais n'avons pas étudié les performances de la régression pour un $\mathbb{C}$ fixé sur une valeur faible, ou variant sur une grille moins large.

Au delà de notre problématique, nous avons remarqué que l'erreur d'apprentissage d'une ε-SVR était étroitement liée au nombre de vecteurs supports retenus pas le modèle : meilleure est la régression, moins elle a besoin de vecteurs pour généraliser l'ensemble des échantillons en garantissant un biais faible. Nous avons noté cette parcimonie bienfaisante, mais nous ne l'avons pas exploitée. L'incorporation de cette propriété dans la conception du critère de performance du modèle serait une idée intéressante à creuser.

2.2 Valeurs descriptives

Passons maintenant au domaine de l'indexation, incarné par les valeurs descriptives que nous avons utilisées.

2.2.1 Des valeurs numériques

Nous rappelons que, schématiquement parlant, nous avons considéré qu'un nombre représentait une information suffisamment pertinente pour caractériser une vidéo.

Nous avons choisi d'implanter la moyenne, la variance, le minimum et le maximum de séries temporelles, mais pourquoi ces valeurs seraient-elles plus représentatives que n'importe quelle autre valeur statistique, comme la médiane par exemple ? Un travail intéressant peut relever du point soulevé ici : comment créer un nombre qui représente l'évolution d'une suite de valeurs ? Une réponse peut exister dans la création d'une forme de code, comme une signature numérique, ou bien dans l'étude des relations qui lient les termes de la suite ou les distinguent. Là encore, le champ des possibles est vaste.

La volonté première a été d'utiliser des descripteurs bas-niveau, car plus facilement manipulables dans un modèle statistique. Intégrer des données issues de descriptions de nature plus « sémantique » nous aurait certainement apporté des informations intéressantes.

C'était dans cette direction que nous comptions initialement aller en incluant des descriptions textuelles, mais nous n'avons pu nous pencher suffisamment sur la question pour réellement tirer profit de ce média. Nous pensions extraire et tester des valeurs, telles que le nombre de mots présents dans une notice documentaire associée à un contenu, le nombre d'occurrences du mot le plus présent dans la notice, ou encore le nombre des mots référençant la thématique dominante. Il serait également intéressant d'exploiter l'ontologie audiovisuelle de l'Ina et d'utiliser un code numérique pour identifier la position d'un terme représentatif d'un contenu dans celle-ci.

2.2.2 L'organisation inter-grains

Ce problème de valeur descriptive numérique unique a été au centre de notre questionnement concernant le représentant de classe et, de fait, l'organisation multi-granularité. Bien que la mesure construite se comporte de manière correcte sur de l'intra-grains, le formalisme tel que nous l'avons pensé ne permet pas au représentant d'être suffisamment bien exploité pour de l'inter-grains.

Revoir les arêtes des graphes serait une solution rapide à mettre en oeuvre, par exemple en reliant un contenu libre non classé avec tous les classés de l'espace dynamique. Or, le concept de classes indépendantes assimilées à des « univers » évoluant selon leurs propres règles en serait modifié. Pour schématiser, chaque modification du nombre de contenus d'un dossier (une classe) remettrait en question tous les libres non classés.

Nous pensions revenir sur une propriété du représentant pour résoudre ce problème. Construit sur l'ensemble des contenus d'une classe, il est un représentant statistique du groupe. Oublier le représentant et relier les contenus libres et non classés à tous les ancrés des différentes classes construites permettrait d'augmenter de manière raisonnable le nombre d'arêtes.

Des tests seraient facilement praticables, pour voir si le temps de calcul ne pâtit pas trop de ces nouvelles relations créées.

Une solution intéressante, du point de vue du vecteur de descriptions du représentant, serait de travailler de manière hiérarchique : si nous regroupons n contenus dans une même classe, au lieu de faire une moyenne des n descriptions $\{d_i\}_{i=1...n}$ pour construire celle correspondant au représentant, nous pourrions générer quatre nouvelles descriptions, à l'image du niveau inférieur : la moyenne, la variance, le minimum et le maximum.

Il serait alors possible de créer une relation g entre ces deux niveaux, par exemple sous la forme d'une régression. Chaque valeur descriptive du représentant d'une classe ne serait plus une moyenne, mais une valeur issue d'une prédiction :

$$\widehat{d_i} = g(mean(\{d_i\}_{i=1...n}) \oplus var(\{d_i\}_{i=1...n}) \oplus min(\{d_i\}_{i=1...n}) \oplus max(\{d_i\}_{i=1...n})) \quad (1)$$

Plus il y aurait de contenus, meilleure serait g, et mieux serait représentée la classe. Il serait intéressant de voir si le temps de calcul de ces fonctions reste raisonnable.

Une autre idée sur laquelle nous aurions aimé travailler concerne la relation entre les mesures de similarités sur différents grains : en aidant à l'organisation d'une classe, nous créons une mesure spécifique à celle-ci. Étudier la répercussion directe des effets de cette mesure sur sur les grains inférieurs ou supérieurs pourrait aider à mieux appréhender l'organisation inter-grains.

2.3 La visualisation

Tout d'abord, au sujet de notre moteur, le choix d'un modèle masse-ressort est tout à fait contestable, car il engendre des difficultés qui lui sont propres, comme le fait que certaines particules trop rapprochées bloquent le passage à d'autres, ou le fait que la position initiale joue un rôle non négligeable dans le déplacement d'un contenu, puisque ce sont ces coordonnées qui servent à initialiser le modèle d'énergie et à calculer la première direction du vecteur vitesse par Runge-Kutta 4. Une alternative à mettre en oeuvre, ne serait-ce qu'à titre comparatif, serait d'implanter un algorithme de type *Multi Dimensional Scaling*.

En vue d'aider le placement de graphe par modèle d'énergie, nous avions implanté dans un précédent prototype une option que nous avions baptisée « shaking ». En cliquant sur un bouton, l'utilisateur bouleversait entièrement les longueurs des ressorts en leur attribuant des valeurs aléatoires. Au bout de quelques secondes de réorganisation les valeurs de similarité étaient restituées et la recherche de l'équilibre énergétique reprenait son cours normal, une idée que nous n'avons pas reprise dans notre dernier prototype, faute de temps.

Le modèle masse-ressort est intéressant dans le sens où les particules restent manipulables, même pendant une phase dynamique. Cela pose la question de la consistance de la mesure lorsque des contenus ancrés sont déplacés par l'utilisateur sans qu'une nouvelle régression ne soit calculée. Une étude plus approfondie de notre moteur pourrait déterminer les avantages à jouer sur cet aspect uniquement visuel, sans avoir à repasser par de l'apprentissage.

Concernant la conception d'une classes, nous avons opté pour une identification de ses éléments en leur attribuant une couleur.

Nous avions dans un premier temps envisagé de créer une forme avec une consistance physique, comme une sphère englobante centrée sur le représentant ou encore l'enveloppe convexe des éléments de la classe. Au terme d'une réorganisation, un contenu libre pourrait buter sur cette forme englobante et déformer sa surface, de manière à symboliser la proposition de classement de cet élément par le système. Toutefois, ces formes pourraient contraindre les mouvements des contenus libres, en les empêchant de contourner la classe par exemple.

Un autre problème concernerait la symbolique de la taille de ces formes. La métaphore visuelle de la taille d'un objet informationnel sur une interface est clairement associée à son poids. Pour une classe, cela correspond au nombre de ses contenus. Nous pourrions cependant imaginer une forme large ne contenant que quelques éléments très éloignés et une forme petite abritant un millier de contenus. Comment forcer une interprétation correcte de cette métaphore ? La question reste ouverte.

Une idée que nous n'avons pas eu le temps de mettre en place portait sur une extension de notre outil : contraindre l'organisation des contenus sur des formes géométriques prédéfinies, des *patterns* de visualisation. Une recherche sur l'aspect de ces formes en fonction du type de corpus serait intéressante :

- une organisation classique de journaux télévisés par date serait contrainte dans un rectangle tel que la largeur corresponde aux années et la hauteur aux mois ;
- une organisation plus exotique d'une collection d'émissions culturelles pourrait se présenter sous différentes formes ; en voici deux exemples :
  - un cercle : plus un contenu se trouve au centre, plus le propos abordé par celui-ci est en lien avec un thème particulier, comme l'actualité du jour par exemple ;
  - une étoile : chaque branche correspond à une discipline artistique, avec des intervenants marginaux à ses extrémités et plus consensuels au centre.

Pour finir, une suggestion intéressante a été formulée par deux fois lors de nos tests utilisateurs et concerne les contenus groupés : une fois qu'une classe est estimée complète, elle reste dans l'espace dynamique, influençant toujours les autres contenus. Une proposition intéressante serait d'exploiter la métaphore des dossiers dont nous parlions dans le chapitre 2, et de permettre de ranger ces dossiers dans le bordereau de la base de données. Ce bordereau deviendrait alors un deuxième espace d'organisation que nous pourrions qualifier de statique.

2.4 L'évaluation

Pour finir, nous prendrons quelques lignes pour parler d'une question qui est restée suspendue tout au long de ces trois années : l'évaluation.

Une des failles de notre outil réside dans l'incapacité d'évaluer si le système comprend ou non la tâche réelle.

Nous avions implanté, dans un précédent prototype, une fonctionnalité permettant de prendre connaissance de l'erreur de visualisation commise par le système sur le placement de chaque contenu, en calculant la moyenne des différences d'énergie entre un contenu et tous les autres : l'énergie entre deux contenus vaut la différence entre leur mesure de similarités et la distance calculée après stabilité.

Si nous estimons qu'une énergie minimale vaut 0 (mesure de similarité = distance à l'écran), le contenu est parfaitement placé. Par une normalisation de type *MinMax* de cette énergie, nous pouvons visualiser l'erreur sous forme de couleur (vert = bien placé, rouge = mal placé).

Toutefois, nous n'avons pas trouvé de corrélation entre erreur de visualisation et erreur d'interprétation dans notre étude, mais nous pensons qu'il doit être possible de tirer profit de pareilles données pour informer l'utilisateur de l'état de compréhension du système.

La dernière réflexion concerne la nature du processus d'évaluation lui-même. Nous venons d'une communauté scientifique dans laquelle les outils d'évaluation sont clairement définis, où les tâches sont précisément identifiées. Cependant, nous avons pensé cet outil dans un but ludique. Il peut n'avoir comme finalité que d'apporter une motivation à ranger ses données personnelles, en permettant de le faire de différentes manières, en fonction de ses humeurs, sans protocole particulier.

Il devient alors particulièrement difficile de fournir un critère d'évaluation objectif, tel que peut l'être un rapport « rappel / précision », sur des tâches comme « *ordonner les photos d'anciens collègues du lycée en fonction de l'effet que je provoquais sur l'humeur de leurs parents (de admiratifs à agressifs en passant par désespérés)* », « *regrouper ces publicités en trois tas : pour les gentils, pour les méchants et pour les autres* », voire même « *organiser ma base comme je le souhaite* ». La principale cause de ce mal est que nous avons pensé une tâche comme une action évolutive, reposant sur des fondations subjectives pouvant être mal définies (voire indéfinissables).

Une évaluation objective a pour intérêt d'éclairer sur certains aspects d'un phénomène, mais occulte le sens même de la subjectivité. En focalisant sur le phénomène, le chercheur se détache paradoxalement de la subjectivité vécue par celui qui expérimente le phénomène. Nous avons choisi de pratiquer des évaluations théoriques en fournissant des résultats objectifs, et ne nous sommes pas laissés guidés par ce pour quoi cet outil était pensé à la base, de peur de ne pouvoir donner suffisamment de crédits à nos travaux. Nous espérons que l'expérience acquise au long de ces trois années nous permettra dorénavant de nous affirmer sur ce plan, quitte à déroger à d'implicites règles que nous avons perçues comme trop restrictives et partiellement inadaptées à notre problématique.

Annexe A

Pseudo code de l'algorithme SFFS

Cet algorithme, écrit en pseudo-code, décrit comment construire un processus de sélection d'attributs avec la variante flottante de la méthode SFS.

Algorithm 2 Algorithme SFFS

Données : $S := \{x_i | i = 1, ..., N\}$
// S : ensemble initial de variables
// C : critère à minimiser (par exemple)
// better : booleen servant au test d'arrêt final
Sortie : $S_n := \{y_i | i = 1, ..., n\}$
// S_n : sous-ensemble de variables de taille n, n $\leq$ N
Initialisation :
$n = 0$
$S_n := \emptyset$
Etape *forward*
better := $false$
$x_+ := \underset{x_i \in S \setminus S_n}{\arg\min}\, C(S_n \cup \{x_i\})$
$S_{n+1} := S_n \cup \{x_+\}$
$n := n + 1$
si $C(S_n) < C(S_{n-1})$ **alors**
| better := $true$
fin si
Etape *backward*
$x_- := \underset{y_i \in S_n}{\arg\min}\, C(S_n \setminus \{y_i\})$
si $C(S_n \setminus \{x_-\}) < C(S_{n-1})$ **alors**
| $S_{n-1} := S_n \setminus \{x_-\}$
| $n := n - 1$
| Aller à l'**Etape *backward***
sinon
| **si** better == $true$ **alors**
| | Aller à l'**Etape *sortie***
| **sinon**
| | Aller à l'**Etape *forward***
| **fin si**
fin si
Etape *sortie*

Annexe B

Compléments sur l'expérience portant sur les performances théoriques du modèle

Les tableaux B.2 et B.3 présentent les résultats correspondant à la série d'expériences A, portant sur la comparaison des performances obtenues avec les trois méthodes d'optimisation des hyper-paramètres.

La légende est la suivante :
- **N° Exp** : Numéro de l'expérience ; à chaque expérience correspond une série de trois tests (un par méthode) réalisés avec un même corpus.
- **TYPE** : Méthode utilisée :
  - **P** : Partielle ;
  - **C** : Complète sans optimisation de ε ;
  - **C+**ε : Complète avec optimisation de ε ;
- **Nb** : Cardinal du corpus d'apprentissage ;
- **EQM** : Erreur d'apprentissage ;
- **DESC** : descriptions retenues pour construire la régression (complément de légende figure B.1).

Lum. moy.	1ère teinte dom.	2ème teinte dom.	1ère sat. dom.	2ème sat. dom.	1ère valeur dom.	2ème valeur dom.	Contrast	Taux d'activité
[0][1][2][3]	[4][5][6][7]	[8][9][10][11]	[12][13][14][15]	[16][17][18][19]	[20][21][22][23]	[24][25][26][27]	[28][29][30][31]	[32][33][34][35]

Mod. énergie 4hz	Mod. entropie	Energie	ZCR	Flux sp.	Centroïde sp.	*RollOff point*	Fréquence fond.
[36][37][38][39]	[40][41][42][43]	[44][45][46][47]	[48][48][50][51]	[52][53][54][55]	[56][57][58][59]	[60][61][62][63]	[64][65][66][67]

[a][b][c][d] :	a = Moyenne b = Variance c = Minimum d = Maximum

FIGURE B.1 – Légende portant sur les numéros des descriptions audio et vidéo

N° Exp	TYPE	Nb	EQM	DESC	N° Exp	TYPE	Nb	EQM	DESC
	P	3	0.339598	[36][28]		P	4	0.334801	[54][43][33][21][1][15]
1	C	3	0.339598	[28][36]	11	C	4	0.334616	[20][13]
	C+ε	3	0.320512			C+ε	4	0.309265	
	P	3	0.337968	[18][27]		P	4	0.341877	[7][55][59]
2	C	3	0.337968	[18]	12	C	4	0.337702	[59][7]
	C+ε	3	0.313672			C+ε	4	0.298376	
	P	3	0.338576	[6][27]		P	4	0.333634	[22][6]
3	C	3	0.338545	[6][4]	13	C	4	0.333634	[22]
	C+ε	3	0.311567			C+ε	4	0.289032	
	P	3	0.339550	[6][18]		P	4	0.334838	[51][18]
4	C	3	0.339509	[42][6]	14	C	4	0.332130	[57]
	C+ε	3	0.312930			C+ε	4	0.302930	
	P	3	0.339275	[0][7]		P	4	0.328460	[6][7]
5	C	3	0.339265	[35]	15	C	4	0.328480	[13][14]
	C+ε	3	0.320056			C+ε	4	0.300261	
	P	3	0.338932	[26][6]		P	4	0.336133	[57][19]
6	C	3	0.338932	[26]	16	C	4	0.336133	[57][19]
	C+ε	3	0.314342	[35]		C+ε	4	0.290991	
	P	3	0.339549	[10][6]		P	4	0.338249	[5][18][34]
7	C	3	0.339518	[10][11]	17	C	4	0.335084	[24]
	C+ε	3	0.311314			C+ε	4	0.300217	
	P	3	0.339997	[6][27]		P	4	0.335742	[25][5][27]
8	C	3	0.339976	[6][35]	18	C	4	0.335181	[67]
	C+ε	3	0.321673			C+ε	4	0.289093	
	P	3	0.338883	[35][6]		P	4	0.337474	[17][34]
9	C	3	0.338883	[35]	19	C	4	0.336189	[5][27][42]
	C+ε	3	0.318293			C+ε	4	0.309293	
	P	3	0.336733	[4][5]		P	4	0.334604	[7][20][18][27]
10	C	3	0.336702	[4][6]	20	C	4	0.334604	[7][20][18][27]
	C+ε	3	0.321009			C+ε	4	0.297629	

FIGURE B.2 – Résultats de la série d'expériences A (tableau 1)

N° Exp	TYPE	Nb	EQM	DESC	N° Exp	TYPE	Nb	EQM	DESC
	P	5	0.033770	[6][13][4]		P	6	0.080045	[16][4]
21	C	5	0.009316	[15]	31	C	6	0.022075	[24][26]
	C+ε	5	0.002301			C+ε	6	0.031029	
	P	5	0.334672	[44][47][18]		P	6	0.023729	[4][6][17][22]
22	C	5	0.334672	[44][47]	32	C	6	0.023729	[4][6][17][22]
	C+ε	5	0.309365			C+ε	6	0.024122	
	P	5	0.033616	[6][4][7]		P	6	0.035203	[26][1]
23	C	5	0.019928	[14]	33	C	6	0.035203	[26][16]
	C+ε	5	0.002019			C+ε	6	0.033781	
	P	5	0.035235	[2][14]		P	6	0.013361	[4][51][20][28]
24	C	5	0.015697	[11]	34	C	6	0.057356	[4][35]
	C+ε	5	0.030768			C+ε	6	0.035726	
	P	5	0.038360	[7][24]		P	6	0.057356	[4][35]
25	C	5	0.038360	[7][24]	35	C	6	0.016641	[0][20][33]
	C+ε	5	0.310023			C+ε	6	0.008277	
	P	5	0.016941	[6][22]		P	6	0.035639	[6][35][4][7][43]
26	C	5	0.007810	[14][23][22]	36	C	6	0.014503	[26][40]
	C+ε	5	0.002412			C+ε	6	0.002839	
	P	5	0.038584	[64][30]		P	6	0.043561	[20][67]
27	C	5	0.012667	[24]	37	C	6	0.008560	[7]
	C+ε	5	0.002215			C+ε	6	0.006627	
	P	5	0.332112	[6][20][22][14][5]		P	6	0.027596	[5][55]
28	C	5	0.332089	[22][14]	38	C	6	0.027596	[5][55]
	C+ε	5	0.300723			C+ε	6	0.007325	
	P	5	0.331441	[24][59][7]		P	6	0.010725	[28][9]
29	C	5	0.331000	[19]	39	C	6	0.006062	[20][15][50]
	C+ε	5	0.001312			C+ε	6	0.004892	
	P	5	0.033955	[6][13][22][4]		P	6	0.082651	[64][16][18]
30	C	5	0.033967	[22][6][4]	40	C	6	0.022364	[64][29]
	C+ε	5	0.000976			C+ε	6	0.002909	

FIGURE B.3 – Résultats de la série d'expériences A (tableau 2)

Annexe C

Compléments sur l'expérience portant sur la mesure de similarité

Les tableaux C.2, C.3 et C.4 présentent les résultats correspondant à la série d'expériences B, portant sur l'évolution des différents types d'erreurs en fonction du cardinal du corpus apprentissage. La figure C.1 décrit les numéros correspondant aux descriptions conservées pour construire la fonction de régression.

Mod. énergie 4hz	Mod. entropie	Energie	ZCR	Flux sp.	Centroïde sp.	*RollOff point*	Fréquence fond.
[0][1][2][3]	[4][5][6][7]	[8][9][10][11]	[12][13][14][15]	[16][17][18][19]	[20][21][22][23]	[24][25][26][27]	[28][29][30][31]

[a][b][c][d] :	a = Moyenne b = Variance c = Minimum d = Maximum

FIGURE C.1 – Légende portant sur les numéros des descriptions (audio uniquement)

Le tableau C.5 présente les résultats correspondant à la série d'expériences C, portant sur l'évolution de l'erreur d'interprétation en fonction de la présence ou de l'absence du guide « fréquence fondamentale ».

La légende du tableau est la suivante :
- **#** : Cardinal du corpus d'apprentissage ;
- **Erreur** : Erreur d'interprétation ;
- **DESC** : descriptions retenues pour construire la régression.

Cardinal du corpus d'apprentissage	Erreur d'apprentissage	Erreur de prédiction	Erreur de visualisation	Erreur d'interprétation	Descriptions conservées
3	0.067234	128311.630447	29297.089504	103338.129243	[31][10]
3	0.118648	64733.816371	4999.390539	60755.194190	[31][29][10]
3	0.035074	100706.103200	15433.528473	113385.559046	[15][10]
3	0.076474	106775.746180	8158.115130	124419.940974	[24][25][23]
3	0.051241	80841.148176	11846.601029	80971.210270	[29][28]
3	0.340548	133432.782075	8312.030664	120085.587200	[31][10]
3	0.103429	94292.590380	20250.806826	114266.020620	[25][5][0]
3	0.129865	95013.929815	3663.621626	93386.189458	[31]
3	0.060752	151408.607056	5152.653815	157158.424632	[2][6]
3	0.107935	105010.911291	7147.875295	116347.332688	[5][7][28]
4	0.069824	45241.739066	9157.338918	37697.784777	[31][10]
4	0.011503	82705.022975	12175.149884	90861.193949	[18][31]
4	0.091910	94208.052336	19637.127587	79492.269876	[31][10]
4	0.046554	124600.097312	2776.121286	135400.527073	[12][13][27][15]
4	0.069162	105752.996218	18155.409260	93471.386853	[31][10]
4	0.046731	162305.750140	47681.166278	171466.579203	[8][24][4][17]
4	0.027699	121721.090755	13002.861348	102126.316248	[31][10]
4	0.019888	204756.543025	9849.609428	186303.289898	[2][10]
4	0.090613	86855.494637	18881.290582	99833.119451	[5]
4	0.068987	34345.620164	3366.006053	31324.642586	[31][29]
5	0.009925	82561.346402	16079.792617	63578.053694	[31][10]
5	0.024390	16005.658442	5433.094685	10487.290251	[31][10]
5	0.085785	107644.260946	9999.271960	125658.185278	[6][2][10]
5	0.057075	131446.815863	4261.836230	139760.184301	[15][28][8][9]
5	0.085676	96633.480694	23637.458222	106843.795578	[0][19][24][10]
5	0.027740	25568.136128	3692.144203	22067.667606	[31][10]
5	0.024334	155301.248370	10249.420388	150926.363634	[27][10]
5	0.015913	20958.273836	8813.595457	11867.129740	[31][10]
5	0.013844	62041.259806	24865.512599	46567.046492	[29][10]
5	0.070540	64712.874269	7337.790938	62213.911210	[31][29][27]

FIGURE C.2 – Résultats de la série d'expériences B (tableau 1)

Cardinal du corpus d'apprentissage	Erreur d'apprentissage	Erreur de prédiction	Erreur de visualisation	Erreur d'interprétation	Descriptions conservées
6	0.008135	69877.883082	18249.667522	52247.650116	[29][18][31]
6	0.013847	136755.742582	11616.909179	121783.910891	[28]
6	0.025693	95339.797558	2537.206574	92724.427885	[31][29]
6	0.081910	192736.738728	64817.703829	170892.322305	[9][10]
6	0.011278	57467.716439	12073.641770	46217.938174	[31][10]
6	0.024225	51839.446698	4822.650410	49273.397306	[31][28]
6	0.015751	105163.226525	11096.176234	99629.038445	[31][23][29][12][10]
6	0.019097	98732.082432	4670.040716	95311.399487	[28][10]
6	0.012658	19665.736899	4424.826754	15374.072306	[31][10]
6	0.027640	27570.174559	5090.516103	22367.225568	[31][10]
7	0.012380	67900.905558	7154.569006	60211.763475	[31]
7	0.008844	16963.544750	5517.068712	11463.588895	[31][10]
7	0.008290	125241.314632	18369.348210	133291.463626	[6][13]
7	0.026592	25377.168920	4176.307851	21361.215966	[31][10]
7	0.027735	43864.192982	5627.770100	39482.773266	[28][31][29]
7	0.009133	15963.152259	5471.346052	10342.819724	[31][10]
7	0.018466	85078.387777	4909.568339	82155.750421	[31][12]
7	0.026231	135031.336372	2210.422609	131771.402161	[16][21][8]
7	0.037483	39635.206366	4908.597090	35271.575994	[28][31]
7	0.009755	41093.675251	8114.693666	33062.357492	[31][10]
12	0.006944	30290.892690	4494.051321	25916.722657	[29][31]
12	0.011000	21474.943475	4826.019742	16464.392503	[31][10]
12	0.007847	23061.827034	6489.371170	16614.938781	[31][10]
12	0.010930	51993.841990	4600.217058	47650.158147	[31][10]
12	0.007986	62767.925679	6641.854887	57692.688334	[29][10]
12	0.007902	21011.262128	3154.890573	17774.311455	[31][10]
12	0.008846	54173.352148	15199.739810	39096.845928	[31][10]
12	0.012154	52457.827843	3639.904636	48697.657169	[31][10]
12	0.006531	57757.504499	12112.258281	45712.357813	[31][10]
12	0.005786	13264.500176	4794.078888	8353.846371	[31][10]

FIGURE C.3 – Résultats de la série d'expériences B (tableau 2)

Cardinal du corpus d'apprentissage	Erreur d'apprentissage	Erreur de prédiction	Erreur de visualisation	Erreur d'interprétation	Descriptions conservées
15	0,0062496	29436,954	2993,74604	19168,386	[31][10]
15	0,0099	36865,017	14436,3528	18941,878	[29][10]
15	0,0070623	20480,719	3454,5094	7908,7662	[31][10]
15	0,009837	14181,948	11503,2454	2446,3543	[31][10]
15	0,0071874	44247,537	4550,97494	33249,831	[31][10]
15	0,0071118	22417,036	4265,94875	7316,6055	[31][10]
15	0,0079614	21459,288	4581,31875	9488,0418	[31][10]
15	0,0109386	37014,655	6161,50261	18933,72	[31][10]
15	0,0058779	15353,83	4366,80621	3880,345	[29][31]
15	0,0052074	54063,583	6306,36214	35080,268	[31][10]
18	0,004028	21689,96932	5722,88979	16219,42968	[31][10]
18	0,003378	10988,74292	7366,882057	3569,075859	[31][10]
18	0,003649	10519,25889	5917,888492	4628,313082	[31][10]
18	0,003478	18144,43878	8834,104285	9235,960922	[31][10]
18	0,003279	7526,387176	5581,161799	1892,851221	[31][10]
18	0,004924	26501,7563	9433,985241	17112,32574	[31][10]
18	0,003417	14429,87922	5183,516009	9350,432416	[31][29]
18	0,003697	18071,08655	8859,896435	9239,940415	[31][10]
18	0,003567	10039,5682	6215,25678	3857,934708	[31][10]
18	0,003504	6951,93518	5734,600627	1193,343562	[31][10]

FIGURE C.4 – Résultats de la série d'expériences B (tableau 3)

	Avec f0		Sans max(f0)		Sans f0	
#	Erreur	DESC	Erreur	DESC	Erreur	DESC
3	103338,13	[31][10]	111441,65	[29][25][10]	104029,74	[13][2][6]
3	60755,19	[31][29][10]	86180,07	[29][27]	127897,87	[7][2][10]
3	113385,56	[15][10]	113385,56	[15][10]	113385,56	[15][10]
3	124419,94	[24][25][23]	124419,94	[24][25][23]	124419,94	[24][25][23]
3	80971,21	[29][28]	80971,21	[29][28]	123683,41	[7][10]
3	120085,59	[31][10]	134479,55	[25][10]	134479,55	[25][10]
3	114266,02	[25][5][0]	114266,02	[25][5][0]	114266,02	[25][5][0]
3	93386,19	[31]	104192,62	[28][10]	177738,45	[18]
3	157158,42	[2][6]	157158,42	[2][6]	157158,42	[2][6]
3	116347,33	[5][7][28]	116347,33	[5][7][28]	99718,14	[5][7][11]
4	37697,78	[31][10]	102787,85	[7][16][18][10]	102787,85	[7][16][18][10]
4	90861,19	[18][31]	217359,27	[18][10]	217359,27	[18][10]
4	79492,27	[31][10]	117941,51	[29][10]	141951,91	[3][10]
4	135400,53	[12][13][27][15]	135400,53	[12][13][27][15]	135400,53	[12][13][27][15]
4	93471,39	[31][10]	95647,78	[29][28]	108702,75	[16][10]
4	171466,58	[8][24][4][17]	171466,58	[8][24][4][17]	171466,58	[8][24][4][17]
4	102126,32	[31][10]	211155,97	[18][10]	211155,97	[18][10]
4	186303,29	[2][10]	186303,29	[2][10]	186303,29	[2][10]
4	99833,12	[5]	99833,12	[5]	99833,12	[5]
4	31324,64	[31][29]	52087,98	[29][10]	114930,69	[0][10]
5	63578,05	[31][10]	99326,67	[29][10]	114354,32	[15][2][21][10]
5	10487,29	[31][10]	125675,33	[0]	125675,33	[0]
5	125658,19	[6][2][10]	125658,19	[6][2][10]	125658,19	[6][2][10]
5	139760,18	[15][28][8][9]	139760,18	[15][28][8][9]	140719,18	[15][0][8][9][17]
5	106843,80	[0][19][24][10]	106843,80	[0][19][24][10]	106843,80	[0][19][24][10]
5	22067,67	[31][10]	75146,67	[29][10]	172140,54	[2][10]
5	150926,36	[27][10]	150926,36	[27][10]	150926,36	[27][10]
5	11867,13	[31][10]	11867,13	[29][10]	126351,22	[6][10]
5	46567,05	[29][10]	46567,05	[29][10]	131569,40	[0]
5	62213,91	[31][29][27]	74043,32	[29][27]	111282,39	[6][10]
6	52247,65	[29][18][31]	52600,41	[29][18]	148591,01	[11][16][18]
6	121783,91	[28]	121783,91	[28]	117764,29	[16][1]
6	92724,43	[31][29]	150911,53	[29][10]	118292,08	[13][10]
6	170892,32	[9][10]	170892,32	[9][10]	170892,32	[9][10]
6	46217,94	[31][10]	105832,84	[29][27][24][13]	123539,81	[25][2][15][10]
6	49273,40	[31][28]	49615,62	[29][28]	102299,74	[13][12]
6	99629,04	[31][23][29][12][10]	99658,81	[15][29][10]	115990,14	[15][10]
6	95311,40	[28][10]	95311,40	[28][10]	118725,06	[27][16][10]
6	15374,07	[31][10]	80077,55	[28][29][10]	145424,29	[9][8][10]
6	22367,23	[31][10]	150009,36	[6][27][2]	150009,36	[6][27][2]
7	60211,76	[31]	141280,91	[12][10]	141280,91	[12][10]
7	11463,59	[31][10]	122781,22	[0]	122781,22	[0]
7	133291,46	[6][13]	133291,46	[6][13]	133291,46	[6][13]
7	21361,22	[31][10]	53524,42	[29][10]	132589,86	[25][10]
7	39482,77	[28][31][29]	57636,97	[28][29]	150235,83	[2]
7	10342,82	[31][10]	31170,53	[29][10]	267012,94	[27][10]
7	82155,75	[31][12]	120872,07	[29][12][25][10]	116226,33	[12][24][10]
7	131771,40	[16][21][8]	131771,40	[16][21][8]	131771,40	[16][21][8]
7	35271,58	[28][31]	183063,53	[6][10]	183063,53	[6][10]
7	33062,36	[31][10]	50488,91	[29][10]	189464,86	[20][23][10]

FIGURE C.5 – Résultats de la série d'expériences C

Annexe D

Compléments sur l'expérience portant sur le processus d'apprentissage incrémental

Les tableaux D.1 et D.2 présentent les résultats respectifs de la première et de la deuxième série de l'expérience D relatives à l'évaluation au processus incrémental d'apprentissage.

La légende est la suivante :

- **N°** : Numéro de l'expérience ; à chaque expérience correspond une série de tests, un par méthode, réalisés avec un même corpus ;
- **TYPE** : Méthode utilisée :
  - **Min** : rapatriement du (des trois) contenu(s) qui minimise(nt) l'énergie des similarités ;
  - **Max** : rapatriement du (des trois) contenu(s) qui maximise(nt) l'énergie des similarités ;
  - **Min2Max** : correspond à la méthode (max, max, min)) ;
  - **Rand** : rapatriement d'un (de trois) contenu(s) au hasard ;
- **?** : indique si le contenu rapatrié est bien classé (+) ou mal classé (-) au sens du plus proche voisin ;
- **?/3** : indique combien des trois contenus rapatriés sont bien classés au sens du plus proche voisin ;
- **%** : pourcentage de classification correcte au sens du plus proche voisin ;
- **EQM** : Erreur d'apprentissage ;
- Les chiffres (indices des colonnes) indiquent le nombre de contenus qui constituent le corpus d'apprentissage.

N°	TYPE		3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	Min	?		+	+	+													
		%	75,000	82,050	42,100	94,590													
		EQM	0,340	0,143	0,008	0,037													
	Max	?		-	-	+	-	-	-	+									
		%	75,000	56,410	44,730	35,130	52,770	31,420	76,470	84,840									
		EQM	0,340	0,336	0,061	0,033	0,213	0,043	0,009	0,006									
	Rand	?		+	+	+	+	+											
		%	75,00	35,89	81,57	78,37	30,55	91,36											
		EQM	0,340	0,050	0,214	0,032	0,039	0,035											
2	Min	?		+	+	+	-	+	+	+	+	+							
		%	45,00	46,10	42,10	43,20	44,40	42,85	41,17	33,30	34,37	93,77							
		EQM	0,340	0,054	0,008	0,043	0,049	0,046	0,037	0,024	0,028	0,008							
	Max	?		+	+	+	+	-											
		%	45,00	64,10	57,89	56,75	58,33	91,42											
		EQM	0,340	0,347	0,035	0,036	0,031	0,023											
	Rand	?		+	+	+	+	+	+	+	+	-	+	-	+	+	+		
		%	45,00	43,58	63,15	91,89	88,80	71,42	85,29	81,81	84,37	67,74	66,66	75,86	78,57	74,07	92,30		
		EQM	0,340	0,054	0,217	0,075	0,048	0,052	0,040	0,029	0,024	0,035	0,020	0,024	0,016	0,013	0,015		
3	Min	?		-	+	+	+	+	+	+									
		%	67,50	82,05	81,57	83,78	83,33	80,00	79,41	81,81									
		EQM	0,339	0,139	0,049	0,035	0,025	0,021	0,015	0,022									
	Max	?		-	-	+	+	+	+	+	+	-	+	+					
		%	67,50	53,84	81,57	81,08	91,66	94,28	85,29	87,87	81,25	77,41	80,00	86,20					
		EQM	0,339	0,306	0,049	0,010	0,011	0,057	0,025	0,017	0,006	0,021	0,041	0,022					
	Rand	?		-	-	+	+	+	+	+	+	+	+						
		%	67,50	35,89	60,52	62,16	80,55	60,00	82,35	75,75	87,50	87,09	83,33						
		EQM	0,339	0,311	0,057	0,047	0,033	0,024	0,187	0,011	0,015	0,022	0,018						
4	Min	?		+	+	+	+	+	+	+	+	+	+	+	+	-	-	+	-
		%	50,00	74,30	39,40	64,80	66,60	34,28	35,29	33,33	31,25	77,40	76,66	72,40	71,32	74,00	76,90	76,00	87,50
		EQM	0,339	0,050	0,008	0,042	0,055	0,003	0,003	0,003	0,003	0,011	0,010	0,008	0,007	0,004	0,004	0,004	0,009
	Max	?		+	-	+	-	-	-	+	-	+	+	+	+				
		%	50,00	71,79	57,80	72,97	63,80	40,00	61,70	90,90	81,25	90,30	93,30	93,10	100,00				
		EQM	0,339	0,050	0,210	0,036	0,027	0,162	0,025	0,020	0,037	0,015	0,018	0,012	0,012				
	Rand	?		-	-	-	-	+	+	+	+	+	+	+	+	+			
		%	50,00	61,50	65,78	64,86	66,60	57,14	88,20	81,81	72,50	73,66	78,20	80,54	81,36	87,50			
		EQM	0,339	0,336	0,129	0,103	0,062	0,061	0,024	0,032	0,032	0,029	0,027	0,025	0,023	0,021			
5	Min	?		-	-	+	+	-	+	+	+	+	+	+	+	+	+		
		%	67,50	64,10	81,57	64,86	55,55	68,50	88,23	90,90	90,60	74,19	73,30	82,75	88,80	88,46	84,00		
		EQM	0,340	0,310	0,054	0,056	0,118	0,290	0,016	0,006	0,007	0,017	0,019	0,007	0,004	0,004	0,018		
	Max	?		+	-	+	-	-	+	+									
		%	67,50	53,80	42,10	48,64	61,10	28,57	47,05	96,96									
		EQM	0,340	0,009	0,207	0,009	0,046	0,042	0,060	0,011									
	Rand	?		+	+	+	+	+	-	+	+	+	-						
		%	67,50	53,80	52,63	54,05	50,00	48,57	38,23	48,48	46,87	45,16	90,00						
		EQM	0,340	0,009	0,006	0,005	0,004	0,003	0,025	0,031	0,017	0,016	0,010						

FIGURE D.1 – Résultats de la première série de l'expérience D

N°	TYPE		3	6	9	12	15	18	21	24
1	Min	? / 3		3	1					
		%	75	35.13	85.29					
		EQM	0.339598	0.047706	0.009918					
	Max	? / 3		2	0	2	2			
		%	75	70.27	85.29	70.96	82.14			
		EQM	0.339598	0.028363	0.008962	0.041255	0.021393			
	Rand	? / 3		2	3	0				
		%	75	54.05	52.94	77.4				
		EQM	0.339598	0.050898	0.006577	0.020511				
	Min2Max	? / 3		2	0	2				
		%	75	37.83	76.47	74.19				
		EQM	0.339598	0.038808	0.017357	0.019125				
2	Min	? / 3		3	3	2	3			
		%	45.0	43.24	35.29	58.06	71.42			
		EQM	0.339774	0.045916	0.004223	0.023983	0.012714			
	Max	? / 3		3	2	2				
		%	45.0	56.75	61.76	93.54				
		EQM	0.339774	0.092996	0.015427	0.024656				
	Rand	? / 3		1	3	0	1			
		%	45.0	48.64	55.88	61.29	96.42			
		EQM	0.339774	0.016658	0.018423	0.234869	0.024503			
	Min2Max	? / 3		2	0	2	2			
		%	45.0	59.45	58.82	77.41	92.8			
		EQM	0.339774	0.048823	0.046623	0.038403	0.007977			
3	Min	? / 3		0	3	3	3	3	1	
		%	67.5	51.35	64.7	64.5	67.85	72	81.81	
		EQM	0.338576	0.162750	0.055059	0.020011	0.011585	0.016825	0.011135	
	Max	? / 3		0						
		%	67.5	68.63						
		EQM	0.338576	0.164524						
	Rand	? / 3		0	0	1	3	1	2	
		%	67.5	64.86	50	51.61	71.4	48	86.36	
		EQM	0.338576	0.176848	0.038538	0.040583	0.019642	0.038087	0.021606	
	Min2Max	? / 3		0	3	0	2			
		%	67.5	70.2	47.05	80.64	82.14			
		EQM	0.338576	0.054661	0.262631	0.004745	0.008873			
4	Min	? / 3		3	3	2	3	2	3	2
		%	50	48.64	41.17	74.19	71.42	76	72.72	84.21
		EQM	0.339275	0.041930	0.003027	0.010492	0.007273	0.003601	0.003290	0.004586
	Max	? / 3		2						
		%	50	72.97						
		EQM	0.339275	0.043681						
	Rand	? / 3		2	3	3				
		%	50	75.67	73.52	61.29				
		EQM	0.339275	0.024382	0.012143	0.007754				
	Min2Max	? / 3		2	2	2	3	2		
		%	50	48.64	44.11	54.8	75	72		
		EQM	0.339275	0.046188	0.009983	0.020018	0.015034	0.011879		
5	Min	? / 3		0	3	1				
		%	67.5	32.43	38.23	83.87				
		EQM	0.339549	0.052704	0.038896	0.026480				
	Max	? / 3		3	0	2				
		%	67.5	51.35	38.23	87				
		EQM	0.339549	0.005586	0.048298	0.021888				
	Rand	? / 3		1	3	3	1	2		
		%	67.5	51.35	47.05	41.93	86.53	89.54		
		EQM	0.339549	0.004724	0.003790	0.002429	0.02824	0.017234		
	Min2Max	? / 3		1	2	3				
		%	67.5	64.86	41.17	61.29	67.85			
		EQM	0.339549	0.038809	0.007944	0.003804	0.003593			

FIGURE D.2 – Résultats de la deuxième série de l'expérience D

Annexe E

Compléments sur l'expérience portant sur l'organisation multi-grains

Les tableaux E.1 et E.2 présentent les résultats respectifs des expériences E et F, portant sur l'organisation multi-granularité.

Note			
#	%	#	%
3	64,90	15	94,20
3	46,40	15	96,50
3	72,00	15	100,00
3	75,40	15	90,50
3	83,50	15	88,90
3	64,20	15	100,00
3	66,50	15	100,00
3	73,10	15	93,50
3	66,80	15	92,10
3	77,00	15	91,50
7	79,20	18	100,00
7	81,30	18	100,00
7	84,40	18	100,00
7	78,80	18	100,00
7	48,00	18	100,00
7	77,70	18	100,00
7	93,00	18	100,00
7	73,60	18	100,00
7	87,10	18	100,00
7	81,94	18	100,00
12	85,00		
12	88,70		
12	91,00		
12	84,00		
12	62,20		
12	82,90		
12	98,20		
12	80,00		
12	95,00		
12	85,50		

Notes	
#	%
3	62,90
3	71,60
3	44,40
3	70,80
3	54,70
3	70,10
3	68,40
3	60,50
3	58,30
3	62,90
7	65,30
7	75,20
7	80,00
7	81,70
7	43,50
7	75,00
7	84,60
7	71,80
7	79,30
7	84,30

Instru.	
#	%
3	71,43
3	71,43
3	71,43
3	85,70
3	71,43
3	57,14
3	57,14
3	85,70
3	71,43
3	71,43

#: Cardinal du corpus d'apprentissage d'apprentissage

%: Pourcentage de classification correcte

FIGURE E.1 – Résultats de la série d'expériences E

#	% (F1)	% (F2)
5	25,6	24,8
5	34,4	18,6
5	14,1	21,5
5	19,6	18,8
5	17,8	15,7
5	15,9	14,4
5	14,8	19,8
5	13,9	20,3
5	17,8	13,3
5	17,4	14,9
7	36,9	21,1
7	31,8	26,0
7	25,0	24,8
7	25,7	21,9
7	24,9	22,0
7	31,4	20,2
7	35,6	17,4
7	19,5	25,2
7	25,9	15,7
7	36,5	19,1
12	78,0	29,4
12	55,8	41,3
12	64,5	16,9
12	56,4	23,5
12	68,8	25,1
12	52,0	19,1
12	59,4	16,4
12	59,0	16,7
12	72,0	21,4
12	44,7	20,9

#	% (F1)	% (F2)
15	84,2	32,0
15	60,3	22,9
15	69,7	18,9
15	60,9	23,1
15	74,3	28,2
15	78,4	21,0
15	64,2	21,7
15	69,0	26,2
15	77,8	29,5
15	75,8	21,7
18	76,1	38,9
18	57,1	19,3
18	66,0	27,0
18	78,1	27,8
18	72,9	31,0
18	75,0	17,8
18	81,2	38,4
18	89,3	19,5
18	87,1	31,5
18	67,4	24,8

: Cardinal du corpus d'apprentissage

% (Fx): Pourcentage de classification correcte portant sur l'expérience Fx

FIGURE E.2 – Résultats des séries d'expériences F1 et F2

Annexe F

Présentation du logiciel

Nous complétons dans cette annexe les aspects de notre outil que nous n'avons pu expliquer plus tôt et qui portent essentiellement sur son interface. Les parties en italique correspondent au texte que nous avons lu à nos utilisateurs lors de la présentation que nous leur avons faite avant qu'ils débutent l'expérience 4.5.6 (le terme de « document » , et non de contenu, a été employé à l'oral et conservé ici).

Aperçu général

Voici un outil destiné à vous aider à organiser rapidement des documents audiovisuels. Nous allons évaluer ensemble son fonctionnement. Nous n'évaluerons pas l'interface en elle-même, je manipulerai donc la souris et le clavier pour éviter de focaliser sur ces usages.

Notre outil, illustré par la figure F.1, *se compose de quatre zones indépendantes.*

FIGURE F.1 – GUI

1. Un bordereau, situé en bas de l'interface, représentant la base documentaire dans son ensemble, que l'on peut visualiser en bougeant les contenus de droite à gauche.

Il est possible de les faire défiler de droite à gauche à l'aide des flèches du clavier (le *slider* n'a pas été implanté). Lorsqu'un contenu est retiré de la base, les autres se décalent vers la

gauche pour combler le vide créé. Lorsqu'un contenu est rajouté à la base, il se positionne à la suite de ceux déjà présents. Un passage de la souris sur un contenu permet de le mettre en valeur en agrandissant sa taille. Il reprend sa taille normale lorsque la souris ne pointe plus dessus.

2. *Une fenêtre d'organisation de la base*, illustré par la figure F.2, *occupe la majeure partie de l'interface.*

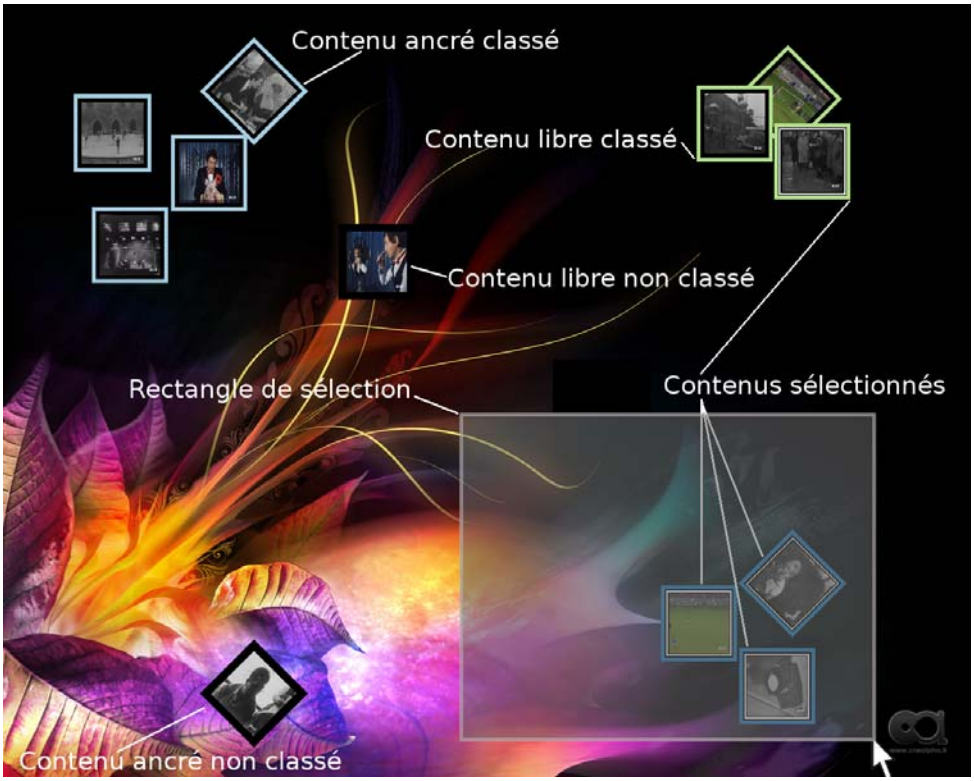


FIGURE F.2 – Fenêtre d'organisation

Comme pour la base documentaire, si la souris passe devant une vignette, celle-ci grandit puis rapetisse si la souris la quitte.

Cet outil offre de multiples fonctionnalités :
- ***sélectionner / désélectionner*** *un ou plusieurs documents ;*
  Le bouton gauche de la souris est dédié à la sélection :
  - il est possible de sélectionner un contenu en cliquant dessus ;
  - cliquer sur un point non occupé de la fenêtre et laisser le bouton appuyé permet de faire apparaître un rectangle d'opacité réduite ; bouger la souris modifie la taille de ce rectangle ; lorsque le bouton est relâché, tous les contenus situés dans la zone grisée du rectangle sont sélectionnés et le rectangle disparaît.
  - doucle-cliquer sur un point non occupé de la fenêtre sélectionne l'ensemble des contenus.
- ***déplacer*** *des documents sélectionnés ;*
- ***augmenter / diminuer*** *la taille des documents ;*

actionner la molette de la souris permet d'agrandir ou de diminuer la taille de toutes les vignettes ;

Cliquer sur le bouton droit de la souris permet de faire apparaître un menu contextuel, portant sur les les contenus sélectionnés, illustré par la figure F.3. La suite des fonctionnalités sont propres à ce menu.

– ***consulter*** *un ou plusieurs documents ;*
  Nous utilisons le lecteur de média VLC de VideoLAN [33] pour consulter les contenus.
– ***créer des groupes*** *: une couleur apparaît autour des contenus groupés. On peut créer jusqu'à huit groupes différents ;*
– ***dégrouper*** *les documents : ils sont entourés d'une couleur noir et n'appartiennent plus à aucun groupe ;*
– ***ancrer / libérer*** *des documents sélectionnés (pivot de 45°).*

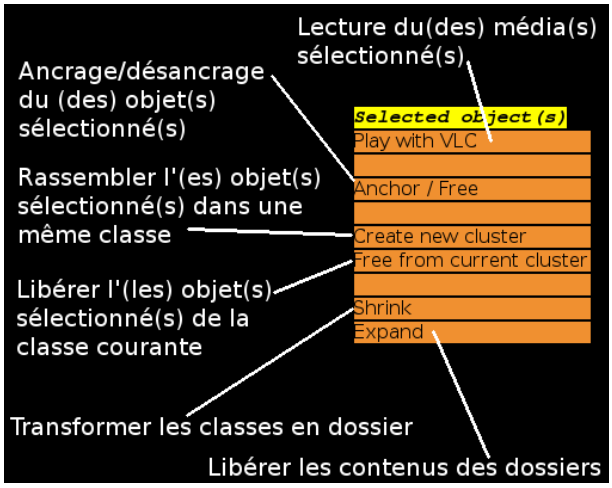


FIGURE F.3 – Menu

Notons qu'un contenu présent dans la fenêtre d'organisation ne peut être présent dans la base documentaire et inversement.

3. Une zone d'information.
Il s'agit d'un bandeau vertical (à droite de la fenêtre) sur lequel s'affichent des informations portant sur le contenu pointé par la souris ; sont visibles la vignette, le titre et la description du contenu (informations désactivées au cours de l'expérience) ;

4. Une zone de fonctions générales (figure F.4), *qui permet de :*
– ***rapatrier*** *des documents depuis la base documentaire jusque dans la fenêtre d'organisation. Nous pouvons rapatrier 1, 3 ou 10 documents. Nous pouvons également rapatrier une sélection de documents (en cliquant dans la base) ou de tous les documents. Ils arrivent tous empilés au milieu de l'écran ;*

33. http ://www.videolan.org/

– ***ranger*** *des documents sélectionnés depuis la fenêtre d'organisation jusque dans la base ;*
– ***activer / désactiver*** *l'aide.*

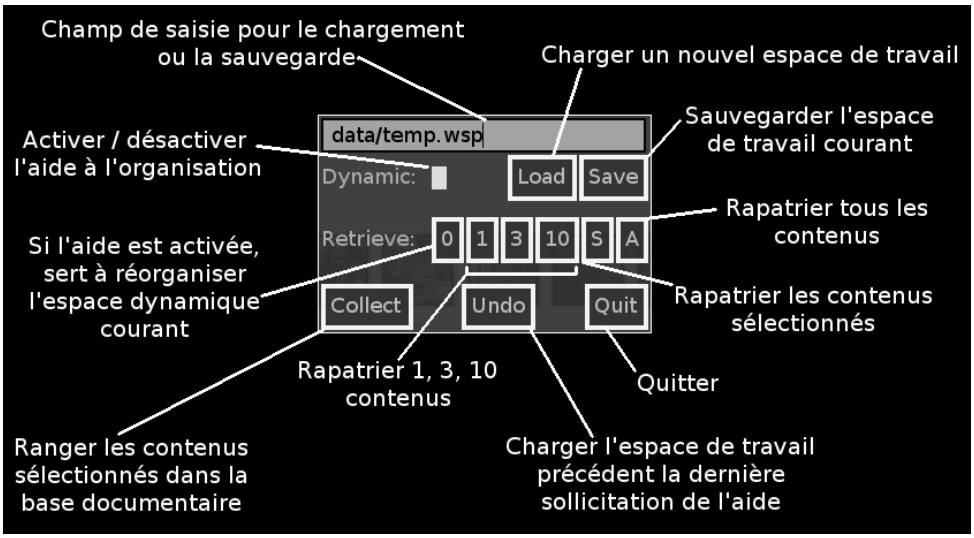


FIGURE F.4 – Fonctions

Présentation du mode assisté

Le système peut m'aider à organiser la base en déplaçant des documents automatiquement. Il y a deux types de documents : les ***libres*** *et les* ***ancrés****.*

Quand je suis dans un groupe, les documents ancrés vont servir de référence et resteront immobiles. L'ordinateur va chercher à comprendre, à travers leur position, comment je souhaite organiser les documents libres de son groupe et les déplacer en conséquence : si je veux en regrouper certains, en ordonner d'autres, etc...

Il faut qu'au moins trois documents soient ancrés dans un même groupe pour que l'aide fonctionne correctement.

En basculant en mode assisté, le système m'aide :
– *à choisir un, trois ou dix, documents (renouvelables) qu'il placera automatiquement ;*
– *à remettre en ordre les documents libres de la fenêtre d'organisation.*

Demander de l'aide nécessite un temps d'attente, qui grandit à mesure que le nombre de documents ancrés augmente. Toutefois, plus il y a de documents ancrés, plus le système a de chance de comprendre mon organisation.

Les documents qui n'ont pas de groupe fonctionnent différemment :
– *les libres se déplacent en fonction de la position des autres groupes ;*

– *les ancrés restent immobiles.*

Pour ces documents, il faut qu'au moins trois groupes soient créés pour que l'aide fonctionne correctement.

Lorsqu'un contenu est rapatrié, il est classé ou non, en fonction du nombre de classes présentes à l'écran. La figure F.5 récapitule les différentes configurations possibles (la classe de niveau 0 est l'ensemble des contenus non classés).

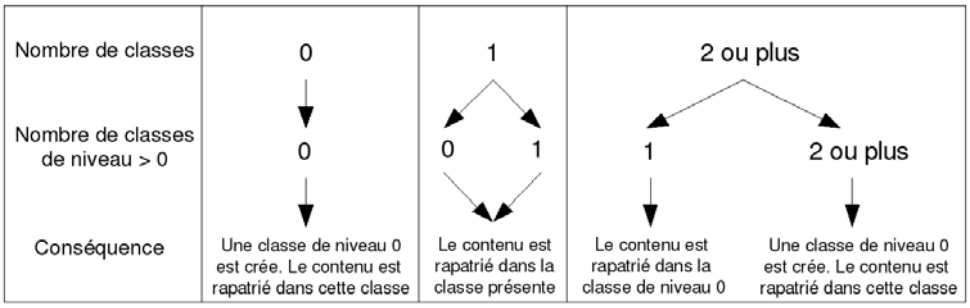


FIGURE F.5 – Attitude du système suite au rapatriement d'un contenu en fonction du nombre de classes et de leurs niveaux

Lorsque le mode dynamique est désactivé, le rapatriement de un, trois ou dix contenus se fait de manière aléatoire. Les contenus rapatriés sont reliés aux contenus de la même classe (ou aux contenus non classés).

Si la mode dynamique est activé, cela permet :

– en cliquant sur 1, de rapatrier le contenu qui maximise l'énergie des similarités prédites ;
– en cliquant sur 3, de rapatrier les trois contenus qui maximisent cette énergie ;
– en cliquant sur 10, de rapatrier les cinq contenus qui maximisent cette énergie et les cinq contenus qui la minimisent.

D'une manière générale, chaque rapatriement entraîne une réestimation des mesures de similarités, suivie d'une réorganisation de contenus non ancrés. Il est possible de provoquer artificiellement ces deux actions en cliquant sur le bouton 0 si le mode dynamique est activé.

Exemple de scénario classique

Le système me présente 3 documents groupés. Je les déplace et les ancre. Si je veux que le système m'aide, j'active le mode aide et demande au système de rapatrier quelques documents. En fonction des documents rapatriés, je peux rester dans le même groupe ou en créer d'autres, les déplacer les ancrer et demander le rapatriement de quelques autres encore, etc...

Aide

Nous avons founi une feuille aux utilisateurs, illustrée par le tableau F.1, avec un récapitulatif des différentes fonctions.

TABLE F.1 – Aide mémoire

Fonctions générales	**Fenêtre d'organisation**
sélectionner / désélectionner	rapatrier 1, 3, 10, une sélection ou tous les documents
déplacer	ranger des documents sélectionnés
augmenter / diminuer la taille	activer/désactiver l'aide
consulter	
créer des groupes	
dégrouper	
ancrer / libérer	

Bibliographie

[AB95] DW Aha and RL Bankert. A comparative evaluation of sequential feature selection algorithms. *Learning from Data : Artificial Intelligence and Statistics V*, pages 199–206, 1995.

[ABR64] M.A. Aizerman, E.M. Braverman, and L.I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25(6) :821–837, 1964.

[AD91] H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. In *9th National Conference on Artificial Intelligence*, pages 547–552. MIT Press, 1991.

[AD94] H. Almuallim and T. G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2) :279–306, 1994.

[AK99] U. Anders and O. Korn. Model selection in neural networks. *Neural Networks*, 12(2) :309–323, 1999.

[AKA91] D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1) :37–66, 1991.

[AQ07] S. Ayache and G. Quénot. Indexation de documents multimédia par réseaux d'opérateurs. In *Conférence francophone en Recherche d'Information et Applications (CORIA)*, pages 385–400, 2007.

[AQG07] S. Ayache, G. Quenot, and J. Gensel. Classifier Fusion for SVM-Based Multimedia Semantic Indexing. *LECTURE NOTES IN COMPUTER SCIENCE*, 4425 :494, 2007.

[Ari08] José Anibal Arias. *Méthodes spectrales pour le traitement automatique de documents audio*. PhD thesis, Université Paul Sabatier, Toulouse, France, 2008.

[Bas89] A. Baskurt. *Compression d'images numeriques par la Transformation Cosinus Discrete*. PhD thesis, Université de Lyon, 1989.

[BB96] G. Brassard and P. Bratley. *Fundamentals of algorithmics*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1996.

[Bel61] R. Bellman. Adaptive Control Processes : A Guided Tour. , 1961.

[BHHSW06] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a Mahalanobis Metric from Equivalence Constraints. *JOURNAL OF MACHINE LEARNING RESEARCH*, 6(1) :937, 2006.

[Bis92] CM Bishop. Curvature-Driven Smoothing in Backpropagation Neural Networks. In *Theory and Applications of Neural Networks : Proceedings of the First British Neural Network Society Meeting, London*. Springer Verlag, 1992.

[Bis00] G. Bisson. La similarité : une notion symbolique / numérique. *CEPADUES*, 2 :169–201, 2000.

[BL97] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence journal*, pages 245–271, 1997.

[BM92] K.P. Bennett and OL Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1(1) :23–34, 1992.

[Boi00] R. Boite. *Traitement de la parole*. Presses polytechniques et universitaires romandes, 2000.

[Bre94] C.A. Brewer. Color Use Guidelines for Mapping and Visualization. *Visualization in Modern Cartography*, 2 :123–148, 1994.

[BSW98] C. Bishop, M. Svensen, and C. Williams. Gtm : The generative topographic mapping. 1(10) :215–234, 1998.

[BW00] D.A. Bell and H. Wang. A Formalism for Relevance and Its Application in Feature Subset Selection. *Machine Learning*, 41(2) :175–195, 2000.

[Cal89] Calliope. *La parole et son traitement automatique*. Masson, Paris, France, 1989.

[Car07] J. Carrive. International workshop on content-based multimedia indexing (cbmi 2007). In *Document Description for Audiovisual Archiving, Corpora, Technologies and Uses*, pages 93–98, Bordeaux, France, 2007.

[CBT04] S. Chelcea, P. Bertrand, and B. Trousse. Un nouvel algorithme de classification ascendante 2-3 hiérarchique. In *Reconnaissance des Formes et Intelligence Artificielle (RFIA 2004)*, volume 3, pages 1471–1480, Toulouse, France, 2004.

[CC01] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. CRC Press, 2001.

[CE94] A. Cavallaro and T. Ebrahimi. Greedy attribute selection. In *11th Int. Conference on Machine Learning*, pages 28–36, 1994.

[CFJV03] Vania Conan, Isabelle Ferrané, Philippe Joly, and Christophe Vasserot. Klimt : Intermediations technologies and multimedia indexing. In *Third International Workshop on Content-Based Multimedia Indexing (CBMI'03)*, pages 11–18. INRIA, septembre 2003.

[CJHA97] V. Chalana, M.Y. Jaisimha, D.R. Haynor, and E. Arbogast. Medplus : a medical image analysis and browsing environment. In *The International Society for Optical Engineering*, volume 3031. Medical Imaging, 1997.

[Com94] P. Comon. Independent component analysis, a new concept. *Signal Processing*, 36(3) :287–314, 1994.

[CP00] M. Carre and P. Philippe. Indexation Audio : un état de l'art. In *Annales des télécommunications*, volume 55, pages 507–525. Lavoisier, 2000.

[CS06] E. Courses and T. Surveys. An Approximate Version of Kernel PCA. In *Machine Learning and Applications, 2006. ICMLA'06. 5th International Conference on*, pages 239–244, 2006.

[CV95] C. Cortes and V. N. Vapnik. Support vector networks. *Machine Learning*, 20 :1–25, 1995.

[CVAR07] M. Crampes, J. Villerd, Emery A., and S. Ranwez. Automatic playlist composition in a dynamic music landscape. In *SADPI '07 : Proceedings of the 2007 international workshop on Semantically aware document processing and indexing*, pages 15–20, New York, NY, USA, 2007. ACM.

[DBM07] T. Delavallade and B. Bouchon-Meunier. *Evaluation des risques de crise, appliquée à la détection des conflits armés intra-étatiques*. Paris, France, 2007.

[DH96] Ron Davidson and D. Harel. Drawing Graphs Nicely Using Simulated Annealing. *ACM Transactions on Graphics*, 15(4) :301–331, 1996.

[DHS73] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification and scene analysis*. Wiley New York, 1973.

[Did91] E. Diday. Des objets de l'analyse des données à ceux de l'analyse des connaissances. *CEPADUES*, 1991.

[DK82] P.A. Devijver and J. Kittler. Pattern recognition : A statistical approach. *Prentice Hall*, 1982.

[DKP03] K. Duan, S.S. Keerthi, and A.N. Poo. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 51(1) :41–60, 2003.

[DL97] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1 :131–156, 1997.

[dRKO+03] D. de Ridder, O. Kouropteva, O. Okun, M. Pietikainen, and R.P.W. Duin. Supervised Locally Linear Embedding. *Lecture Notes in Computer Science*, pages 333–341, 2003.

[DVSSU98] R.D. De Veaux, J. Schumi, J. Schweinsberg, and L.H. Ungar. Prediction Intervals for Neural Networks via Nonlinear Regression. *TECHNOMETRICS*, 40 :273–282, 1998.

[Ead84] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42(149160) :194–202, 1984.

[ET93] B. Efron and R.J. Tibshirani. An Introduction to the Bootstrap. *Monographs on Statistics and Applied Probability*, 57 :1–177, 1993.

[Fen82] J.P Fenelon. *Qu'est-ce que l'analyse des données ?* Lefonen, 1982.

[FI05] F. Friedrichs and C. Igel. Evolutionary tuning of multiple SVM parameters. *Neurocomputing*, 64 :107–117, 2005.

[Fis89] D.H Fisher. Knowledge acquisition via incremental conceptual clustering. 2 :139–172, 1989.

[Foo97] J.T. Foote. A simiarity measure for automatic audio classification. In *Proc. of AAAI 1997 Spring Symposium on Intelligent Integration and Use of Text, Image, Video, and Audio Corpora*, Stanford, 1997.

[FR91] T.M.J. Fruchterman and E.M. Reingold. Graph Drawing by Force-directed Placement. *Software- Practice and Experience*, 21(11) :1129–1164, 1991.

[GBD92] S. Geman, E. Bienenstock, and R. Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1) :1–58, 1992.

[GBV08] H. Goeau, O. Buisson, and M. L. Viaud. Image collection structuring based on evidential active learner. In *CBMI*, 2008.

[GE03] I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3 :1157–1182, 2003.

[GLF89] J. Gennari, P. Langley, and D. Fisher. Model of incremental concept formation. 40 :11–61, 1989.

[GM98] D. Gentner and A.B. Markman. Structure mapping in analogy and similarity. *Mind Readings : Introductory Selections on Cognitive Science*, 1998.

[GO98] T. Graepel and K. Obermayer. Fuzzy topographic kernel clustering. In *5th GI Workshop Fuzzy Neuro Systems*, pages 90–97, 1998.

[Gol94] R. Goldstone. The role of similarity in categorization : Providing a groundwork. *Cognition Journal*, 2(52) :125–157, 1994.

[GR06] A. Globerson and S. Roweis. Metric Learning by Collapsing Classes. *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, 18 :451, 2006.

[Guj04] D.N. Gujarati. *Économétrie*. De Boeck Université, 2004.

[GZZ05] X. Geng, D.C. Zhan, and Z.H. Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 35(6) :1098–1107, 2005.

[Hai05] S. Haidar. *Comparaison des documents audiovisuels par matrice de similarité*. PhD thesis, Université Paul Sabatier, 2005.

[Han90] S. Hanson. Conceptual clustering and categorization : Bridging the gap between induction and causal model. 3 :235–268, 1990.

[HJM92] S. Huet, E. Jolivet, and A. Messean. *La regression non-linéaire. Methodes et applications en biologie*. INRA Editions, 1992.

[HLLM06] S.C.H. Hoi, W. Liu, M.R. Lyu, and W.Y. Ma. Learning distance metrics with contextual constraints for image retrieval. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[HS85] T. Houtgast and HJM Steeneken. A review of the MTF concept in room acoustics and its use for estimating speech intelligibility in auditoria. *The Journal of the Acoustical Society of America*, 77 :1069, 1985.

[HSW89] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5) :359–366, 1989.

[HUN92] R. Haeb-Umbach and H. Ney. Linear discriminant analysis for improved large vocabularycontinuous speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, 1992.

[IN03] G. Iyengar and HJ Nock. Discriminative model fusion for semantic concept detection and annotation in video. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 255–258. ACM New York, NY, USA, 2003.

[JJ02] C. Jacquemin and M. Jardino. Une interface 3d multi-échelle pour la visualisation et la navigation dans de grands documents xml. In *IHM '02 : Proceedings of the 14th French-speaking conference on Human-computer interaction (Conférence Francophone sur l'Interaction Homme-Machine)*, pages 263–266. ACM, 2002.

[JML06] M. Jones, W.T. Maddox, and B.C. Love. The role of similarity in generalization. In *Proceedings of the 28th Annual Meeting of the Cognitive Science Society*, pages 405–410, 2006.

[Jol86] IT Jollie. Principal Component Analysis, 1986.

[Jol96] P. Joly. *Consultation et Analyse des Documents en Image Animée Numérique*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, juillet 1996.

[Kel99] CT Kelley. *Iterative Methods for Optimization*. Society for Industrial Mathematics, 1999.

[KFS05] K.I. Kim, M.O. Franz, and B. Schölkopf. Iterative Kernel Principal Component Analysis for Image Modeling. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, pages 1351–1366, 2005.

[KG86] Y. Kodratoff and J.G. Ganascia. Improving the generalization step in learning. *Machine Learning : An Artificial Intelligence Approach*, 2 :215–244, 1986.

[KJ97] R. Kohavi and G. John. Wrappers for feature selection. *Artificial Intelligence*, 97(1-2) :273–324, 1997.

[KK89] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1) :7–15, 1989.

[KO95] R. R. Korfhage and K. A. Olsen. Image organization using vibe, a visual information browsing environment. In *SPIE - The International Society for Optical Engineering*, volume 2606. Digital Image Storage and Archiving System, Philadelphia, 1995.

[Koh82] T. Kohonen. Analysis of a simple self-organizing process. 2(44) :135–140, 1982.

[Kon94] I. Kononenko. Estimating attributes : Analysis and extensions of relief. pages 171–182. Springer Verlag, 1994.

[KR92a] K. Kira and L. A. Rendell. The feature selection problem : Traditional methods and a new algorithm. In *10th National Conference on Artificial Intelligence*, pages 129–134. MIT Press, 1992.

[KR92b] K. Kira and L. A. Rendell. A paractical approach to feature selection. In *Proceedings of the 9th International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 1992.

[KT82] H.W. Kuhn and A.W. Tucker. Nonlinear programming. *2nd Berkeley Symposium on Mathematical Statistics and Probabilistics*, pages 481–492, 1982.

[KT03] J.T. Kwok and I.W. Tsang. Learning with Idealized Kernels. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, volume 20, page 400, 2003.

[LB84] R. A. Olshen C. J. Stone L. Breiman, J. H. Friedman. Classification and regression trees. 1984.

[LS96] H. Liu and R. Setiono. FEATURE SELECTION AND CLASSIFICATION-A PROBABILISTIC WRAPPER APPROACH. In *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems : Proceedings of the Ninth International Conference, Fukuoka, Japan*, 1996.

[LSG07] A. Lovett, E. Sagi, and D. Gentner. Analogy as a mechanism of comparison. *Proceedings of Analogies : Integrating Multiple Cognitive Abilities*, 2007.

[LSPM+03] E. Loisant, R. Saint-Paul, J. Martinez, G. Raschia, and N. Mouaddib. Browsing clusters of similar images. *Actes des 19e Journées Bases de Donnéees Avancéees (BDA'2003*, 2003.

[Lut94] S. Luttrell. A bayesian analysis of self-organizing maps. 5(6) :767–794, 1994.

[LW67] G.N. Lance and W.T. Williams. A general theory of classification sorting strategies. In *Computer Journal*, volume 9, pages 373–380, 1967.

[Mar02] J. Mariani. *Analyse, synthèse et codage de la parole*. Hermes, 2002.

[MB89] N. Morgan and H. Bourlard. *Generalization and Parameter Estimation in Feedforward Nets : : Some Experiments*. International Computer Science Institute, 1989.

[McC83] G.P. McCormick. *Nonlinear programming : theory, algorithms, and applications*. John Wiley & Sons, Inc. New York, NY, USA, 1983.

[Mic02] C. Michel. Ordre, agrégation et répétition : des paramètres fondamentaux dans les comparaisons d'objets informationnels. *In Actes du Congrès de la Société française de Bibliométrie appliquée*, pages 279–306, 2002.

[Mit82] TM Mitchell. Generalization as Search. *ARTIFICIAL INTELLIG.*, 18(2) :203–226, 1982.

[Mod89] R. Moddemeijer. On estimation of entropy and mutual information of continuous distributions. *Signal Processing*, 16(3) :233–248, 1989.

[MS78] D.L. Medin and M.M. Schaffer. Context theory of classification learning. *Psychological Review*, 85(3) :207–238, 1978.

[MS84] R. S. Michalski and R. E. Stepp. Learning from observation : Conceptual clustering. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning : An Artificial Intelligence Approach*, pages 331–363. Springer, Berlin, Heidelberg, 1984.

[MTL+04] B. Moghaddam, Q. Tian, N. Lesh, C. Shen, and T.S. Huang. Visualization and user-modeling for browsing personal photo libraries. *International Journal of Computer Vision*, 56(1-2) :109–130, 2004.

[NF77] P.M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *TC*, 26(9) :917–922, September 1977.

[Nib88] T. Niblett. A Study of Generalization in Logic Programming. *Proceedings of the 3rd European Workingsessions on Learning (EWSL-88)*, pages 131–138, 1988.

[Nie93] J. Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[Nor94] D. Norman. « cognitives artefacts». *Raisons pratiques*, pages 15–34, 1994.

[Osh90] D.N. Osherson. Category-Based Induction. *Psychological Review*, 97(2) :185–200, 1990.

[PAO05] E. Plaza, E. Armengol, and S. Ontañón. The explanatory power of symbolic similarity in case-based reasoning. *Artificial Intelligence Review*, 24(2) :145–161, 2005.

[PC84] R.R. Picard and R.D. Cook. Cross-validation of regression models. *J. AM. STAT. ASSOC.*, 79(387) :575–583, 1984.

[Pee03] G. Peeters. Automatic classification of large musical instrument databases using hierachical classifiers with inertia ratio maximization. In *AES 115th Convention*, New-York, USA, Octobre 2003.

[Pin04] J. Pinquier. *Indexation sonore : recherche de composantes primaires pour une structuration audiovisuelle*. PhD thesis, Université Paul Sabatier, 2004.

[PNK94] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11) :1119–1125, 1994.

[Pol07] J.P. Poli. *Structuration automatique de flux télévisuels*. PhD thesis, Université Paul Cézanne, 2007.

[PPJ06] J. Philippeau, J. Pinquier, and P. Joly. Intervenant classification in an audiovisual document. In *International Conference on Signal Processing and Multimedia Applications (SIGMAP)*, pages 185–188. INSTICC Press, 2006.

[Qui86] JR Quinlan. Induction of decision trees. *Machine Learning*, 1(1) :81–106, 1986.

[Qui92] J. R. Quinlan. C4.5 : Programs for machine learning. *Morgan Kaufmann*, 1992.

[RGCV05] A. Rakotomamonjy, K. Gasso, S. Canu, and P. Vannoorenberghe. Prévisions de concentrations d'ozone : Comparaison de différentes méthodes statistiques de type « boîte noire ». *Journal européen des systèmes automatisés*, 39(4) :533–552, 2005.

[RH02] Le Hay V. Ackermann W. Le Roux B. Rouanet H., Lebaron F. Régression et analyse géométrique des données : réflexions et suggestions. *Mathématiques et Sciences Humaines*, pages 13–45, 2002.

[RI92] Michael M. Richter and Fachbereich Informatik. Classification and learning of similarity measures. In *In Proceedings*, pages 1–8. Springer Verlag, 1992.

[RLGW94] S. Rännar, F. Lindgren, P. Geladi, and S. Wold. A PLS kernel algorithm for data sets with many variables and fewer objects. Part 1 : Theory and algorithm. *Journal of chemometrics*, 8(2) :111–125, 1994.

[RS00] S.T. Roweis and L.K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding, 2000.

[Sap06] G. Saporta. *Probabilités, analyse des données et statistique*. Editions TECHNIP, 2006.

[Sau96] J. Saunders. Real-time discrimination of broadcast speech/music. In *Proc. ICASSP 96*, pages 993–996, 1996.

[SBS99] B. Scholkopf, C.J.C. Burges, and A. Smola. Advances in kernel methods support vector learning. Cambridge, 1999. MIT Press.

[She57] R.N. Shepard. Stimulus and response generalization : A stochastic model relating generalization to distance in psychological space. *Psychometrika*, 22(4) :325–345, 1957.

[SJ00] S. Santini and R. Jain. Integrated browsing and querying for image databases. *IEEE Multimedia*, 7 :26–39, 2000.

[SJ04] M. Schultz and T. Joachims. Learning a Distance Metric from Relative Comparisons. In *Advances in Neural Information Processing Systems 16 : Proceedings of the 2003 Conference*. Bradford Book, 2004.

[SKG+05] M. Scholz, F. Kaplan, C.L. Guy, J. Kopka, and J. Selbig. Non-linear PCA : a missing data approach. *Bioinformatics*, 21(20) :3887–3895, 2005.

[SOK06] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06 : Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[Sol98] S. Soltani. *Application de la transformée en ondelettes en Reconnaissance des Formes*. PhD thesis, Université de Compiègne, 1998.

[SPNP99] P. Somol, P. Pudil, J. Novovičová, and P. Paclik. Adaptive floating search methods in feature selection. *Pattern Recognition Letters*, 20(11-13) :1157–1163, 1999.

[SS93] W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Handbook of Pattern Recognition & Computer Vision*, 1993.

[SS97] E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *Proc. ICASSP 97*, pages 1331–1334, Munich, Germany, 1997.

[SS02] B. Scholkopf and A.J. Smola. *Learning with kernels*. MIT Press Cambridge, Mass, 2002.

[SS03] A. Smola and B. Scholkopf. A tutorial on support vector regression. *Statistics and Computing*, 14 :199–222, 2003.

[SSM97] B. Schoelkopf, A.J. Smola, and K.R. Mueller. Kernel Principal Component Analysis. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 583–588, 1997.

[SSWB00] B. Scholkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5) :1207–1245, 2000.

[Sta02] C. Staelin. Parameter selection for support vector machines. *HP Laboratories, Israel, Technical Report HPL-2002-354*, 2002.

[Ste76] S. Stearns. On Selecting Features for Pattern Recognition. In *International Conference on Pattern Recognition*, pages 71–75, 1976.

[Sto02] P. Stockinger. *Le document audiovisuel : Procédures de description et exploitation*. Lavoisier, 2002.

[SW05] G.A.F. Seber and CJ Wild. *Nonlinear Regression*. Wiley, 2005.

[SWS05] C.G.M. Snoek, M. Worring, and A.W.M. Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402. ACM New York, NY, USA, 2005.

[SYET07] ME Sargin, Y. Yemez, E. Erzin, and AM Tekalp. Audiovisual Synchronization and Fusion Using Canonical Correlation Analysis. *Multimedia, IEEE Transactions on*, 9(7) :1396–1403, 2007.

[Thi06] J. Thièvre. *Cartographies pour la recherche et l'exploration de données documentaires*. Université de Montpellier 2 Sciences et Techniques du Languedoc, 2006.

[TLN+03] BL Tseng, C.Y. Lin, M. Naphade, A. Natsev, JR Smith, I.B.M.T.J.W.R. Center, and NY Hawthorne. Normalized classifier fusion for semantic visual concept detection. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, 2003.

[TSL00] J.B. Tenenbaum, V. Silva, and J.C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction, 2000.

[Tve77] A. Tversky. Features of similarity. In *Psychological Review*, volume 84, pages 327–352, 1977.

[TW06] A. Tobudic and G. Widmer. Relational IBL in classical music. *Machine Learning*, 64(1) :5–24, 2006.

[VA98] TJ VanderNoot and I. Abrahams. The use of genetic algorithms in the nonlinear regression of immittance data. *Journal of Electroanalytical Chemistry*, 448(1) :17–23, 1998.

[VD97] A. Van Dam. Post-WIMP user interfaces. 2(40) :63–67, 1997.

[Voß94] A. Voß. Similarity concepts and retrieval methods. *FABEL, Report 13, Gesellschaft fur Mathematik und Datenverarbeitung mbH, Forschungsbereich Kunstliche Intelligenz*, 1994.

[WBKW99] E. Wold, T Blum, D. Keislar, and J. Wheather. Classification, search and retrieval of audio. *CRC Handbook of multimedia computing*, 1999.

[WLCS04] Y. Wu, C.Y. Lin, EY Chang, and JR Smith. Multimodal information fusion for video concept detection. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 4, 2004.

[XNJR03] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance Metric Learning with Application to Clustering with Side-Information. *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 521–528, 2003.

[XNZ08] S. Xiang, F. Nie, and C. Zhang. Learning a Mahalanobis distance metric for data clustering and classification. *Pattern Recognition*, 2008.

[YC04] W. Yiming and KL Chan. An Extended ISOMAP Algorithm for Learning Multi-Class Manifold. In *Proceedings of IEEE International Conference on Machine Learning and Cybernetics (ICM LC2004). Shanghai, China*, 2004.

[ZC99] T. Zhang and Kuo C. Heuristic approach for generic audio data segmentation and annotation. In *Proc. of ACM Multimedia*, pages 67–76, 1999.

[ZK98] T. Zhang and C-C. J. Kuo. Hierarchical system for content-based audio classification and retrieval. *Conference on Multimedia Storage and Archiving Systems III*, 3527 of SPIE :398–409, 1998.

[ZS04] H. Zargayouna and S. Salotti. Mesure de similarité sémantique pour l'indexation de documents semi-structurés. *12eme Atelier de Raisonnement à Partir de Cas*, 189, 2004.

Résumé

Dans une optique d'adaptation aux nouveaux usages de consultation d'archives audiovisuelles, nous souhaitons aider un utilisateur issu du grand public à organiser des contenus audiovisuels, c'est-à-dire les classifier, les caractériser, les identifier ou les ordonner.

Pour ce faire, nous proposons d'utiliser un vocabulaire autre que ce que l'on pourrait trouver dans une notice documentaire de l'Ina, afin de répondre à des envies qui ne sont pas facilement définissables avec des mots. Nous avons conçu pour cela une interface graphique qui s'appuie sur un formalisme de graphe dédié à l'expression d'une tâche organisationnelle.

La similarité numérique est un bon outil au regard des éléments que nous manipulons, à savoir des objets informationnels présentés sur un écran d'ordinateur et des valeurs descriptives de « bas niveau » audio et vidéo extraites de manière automatique. Nous avons choisi de prédire la similarité entre ces éléments grâce à un modèle statistique. Parmi les nombreux modèles existants, la prédiction statistique basée sur la régression univariée nous semble adaptée.

Afin de mettre en œuvre les différents points théoriques abordés dans cette étude, nous avons implanté un prototype d'aide à l'organisation.

Son moteur d'apprentissage utilise une régression ε-SVR non-linéaire, fondée sur la concaténation de valeurs descriptives sélectionnées par un algorithme de type séquentiel. Ses hyperparamètres sont optimisés grâce à une méthode de recherche partielle par grille.

Son moteur de visualisation est constitué d'un algorithme de placement de graphe par modèle d'énergie utilisant Runge-Kutta 4, agrémenté d'un mécanisme de stabilisation énergétique artificiel.

Nous avons ensuite testé la mesure de similarité qu'il est capable de produire sur plusieurs séries d'expériences en lien avec les différentes problématiques relevées dans ce document.

Mots-clés: apprentissage automatique, organisation d'objets informationnels, description de contenus audiovisuels, modèle numérique de similarités, mesure de similarité adaptative, régression univariée, ε-SVR, interface homme-machine, modèle physique masse-ressort