Chapitre 1

Représentation et visualisation des données volumineuses pour l'apprentissage supervisé

Sommaire

1 Représentation et visualisation 3
1.1 Définition 3
1.2 Intérêt 5
2 Méthodes de représentation en contexte supervisé 7
2.1 Méthodes fondées sur l'ACP 7
2.2 Méthodes fondées sur les graphes de voisinage 10
2.3 Arbres phylogénétiques 17
2.4 Cartes auto-organisatrices 21
3 Récapitulatif des méthodes de représentation 22
4 Problèmes de l'apprentissage face aux données volumineuses . 23
4.1 Espace de Grandes Dimensions 24
4.2 Volumétrie des exemples 31
4.3 Validation des découvertes 32
4.4 Contributions aux problèmes énoncés 33

Représentation et visualisation des données volumineuses pour l'apprentissage supervisé

Comme nous l'avons souligné dans le chapitre introductif, il existe plusieurs méthodes d'apprentissage supervisé. Au cours de cette thèse, nous nous sommes plus particulièrement intéressés à celles qui établissent une représentation des données en vue de la prédiction. La représentation n'est pas un concept clairement établi en apprentissage automatique, c'est pourquoi nous commençons par en donner une définition précise en section 1, ainsi que les principaux avantages que nous entendons en tirer. La définition donnée est suffisamment générale pour être implémentée de diverses manières par les méthodes existantes. Nous dressons un état de l'art des stratégies possibles dans la section 2, accompagné d'une synthèse en section 3. Enfin, dans la section 4, nous analysons les limites de ces stratégies face à des données volumineuses.

1 Représentation et visualisation

1.1 Définition

Pour définir les représentations en apprentissage supervisé, nous nous appuyons d'abord sur une définition générale, issue des sciences cognitives. Nous l'adapterons ensuite aux besoins de l'apprentissage supervisé.

1.1.1 Définition générale

Bien que l'apprentissage automatique ait souvent recours à la notion de représentation, la définition de celle-ci est rarement au centre des préoccupations. Elle intéresse en revanche le domaine connexe des sciences cognitives où la question de l'existence des représentations dans la cognition suscite de nombreux débats. C'est pourquoi nous avons choisi ce domaine pour tirer une première définition, d'ordre général, émise par Palmer (1978). Pour lui, une représentation est complète si elle définit :

- ce qu'est le domaine représenté,
- ce qu'est le domaine de représentation,
- quels aspects du domaine représenté sont modélisés,
- quels aspects du domaine de représentation réalisent le modèle et
- quelles sont les correspondances entre les deux domaines.

Suivant cette définition, une représentation contient donc, en plus des informations sur elle-même, les informations sur ce qu'elle représente et les correspondances des unes

aux autres. Ainsi, la représentation peut se substituer au domaine représenté (sans qu'il y ait pour autant équivalence entre l'un et l'autre) de sorte que toute action dirigée vers le domaine représenté puisse être réalisée à partir de la représentation. L'avantage de la représentation est alors de simplifier la manipulation de l'information. Cette définition souligne également qu'une représentation nécessite de modéliser le domaine à représenter. À cet égard, elle se doit d'expliciter ce qu'elle modélise et comment elle le modélise. Si la représentation n'a pas vocation à considérer tous les aspects du domaine à représenter, elle se doit en revanche de définir précisément comment chacun de ses aspects est pris en compte.

1.1.2 Définition en apprentissage supervisé

Dans le domaine de l'apprentissage supervisé, il est possible de préciser cette définition. Ici, le domaine à représenter est connu, il s'agit des données d'apprentissage. À notre avis, deux aspects de ce domaine sont à modéliser, d'une part les proximités entre les exemples et d'autre part le lien entre l'étiquette des exemples et leur position dans l'espace d'apprentissage. À partir de là, les autres caractéristiques d'une représentation (à savoir le domaine de représentation, comment il réalise la modélisation et les correspondances entre les deux domaines, représenté et représentant) dépendent du mode de représentation choisi. Celui-ci peut prendre plusieurs formes en apprentissage automatique, formes que nous présentons dans la section suivante.

Au-delà de la définition formelle de Palmer, l'usage des représentations en apprentissage automatique est souvent ciblé vers la visualisation des données. Ainsi, dans sa thèse, Muhlenbach (2002) définit la représentation comme le « fait de mettre quelque chose à l'esprit et, plus concrètement, de rendre cette chose présente à la vue par une image, un signe ou un symbole ». En apprentissage, les données d'intérêt étant des points dans des espaces de taille supérieure à 3, il est difficile pour un utilisateur de s'en faire une idée. La visualisation devient alors le passage obligé pour permettre à un être humain de s'approprier ces données (le domaine représenté). C'est un aspect important des représentations puisqu'il permet de lier un utilisateur à l'apprentissage, ce qui n'est pas sans intérêt pour la fouille de données (voir sous-section suivante).

Pour résumer, nous entendons donc par représentation en apprentissage supervisé : un modèle qui d'une part assure une *correspondance* avec les données et d'autre part permet la *visualisation à la fois des proximités entre exemples* et *du lien entre la position des exemples dans l'espace d'apprentissage et leur étiquette.*

1.1.3 Justification

Le choix de représentation que nous avons fait modélise deux aspects du domaine à représenter, d'une part les proximités entre les exemples et d'autre part le lien entre l'étiquette des exemples et leurs positions.

La modélisation des proximités définit pour deux exemples leur degré de proximité. Elle est nécessaire pour contextualiser un exemple (quels exemples lui sont proches) et pour naviguer à travers les exemples en se déplaçant de proche en proche. Elle sert également de base à la visualisation qui organise ces proximités dans le plan. Enfin, cette modélisation transpose le problème d'apprentissage de son espace initial à l'espace représentationnel, ce qui autorise à travailler directement sur ce dernier.

En plus des proximités, la représentation doit également modéliser le lien entre l'étiquette des exemples et leur position. Une solution différente aurait été d'intégrer l'étiquette dans la modélisation des proximités (deux exemples sont d'autant plus éloignés qu'ils sont d'étiquette différente, comme c'est le cas pour le modèle LASSO (Midenet et Grumbach, 1994), voir section 3.2, chapitre 2) plutôt que de modéliser indépendamment son lien avec la position des exemples. Cette *autre représentation* aurait à coup sûr un pouvoir prédictif intéressant. En revanche, elle perdrait certaines propriétés. Par exemple, ce ne serait plus une modélisation du problème d'apprentissage initial mais au contraire un apprentissage du problème. De même, cette représentation ne donnerait plus accès à un exemple proche de classe différente (donc une perte de la contextualisation) mais rendrait proches des exemples de même classe. Ainsi, en introduisant un biais relatif à la classe, ce type de représentation limiterait son intérêt à une meilleure prédiction au détriment des aspects de contextualisation ou de navigation.

À la modélisation de la proximité entre exemples et du lien entre leur position et leur étiquette, nous avons ajouté la nécessité de visualiser la représentation. Pour nous, cette visualisation constitue un lien entre l'utilisateur et ses données, lien à partir duquel il est possible de concevoir une interaction entre l'utilisateur et la tâche qu'il souhaite réaliser. En ce sens, la représentation est alors l'occasion de mettre à contribution la cognition de l'utilisateur au cours de l'apprentissage. Cela nous semble particulièrement intéressant dans le cas de données complexes que la numérisation a souvent du mal à décrire parfaitement.

1.2 Intérêt

L'intérêt d'une représentation dans le processus d'extraction des connaissances a plusieurs aspects.

En tant que modélisation des proximités, la représentation établit une métrique qui mesure l'importance des relations de voisinage entre les exemples. Elle offre ainsi une

structure qui relie les exemples entre eux. Cette structure facilite le déplacement d'un exemple vers un autre exemple similaire, autorisant ainsi la navigation à travers les données.

En modélisant le lien entre la position des exemples et leur étiquette, la représentation conserve le problème d'apprentissage initial sous son aspect géométrique. En effet, plus il existe un lien fort entre la position des exemples dans l'espace d'apprentissage et leur classe, plus il sera facile de trouver un modèle de prédiction efficace des différentes étiquettes. Il est donc possible de valider les capacités prédictives de la représentation en mesurant la force de ce lien directement sur la représentation. En outre, si cette dernière est de qualité, cette mesure témoigne des résultats en prédiction que l'on peut attendre à partir de ces données (des exemples de cette stratégie sont donnés en section 2.2 de ce chapitre pour les graphes de voisinage et au chapitre 2 pour les cartes auto-organisatrices).

En tant qu'outil de visualisation, la représentation aide l'utilisateur à considérer les données dans leur ensemble. Au niveau des pré-traitements, l'utilisateur peut se servir de la représentation pour effectuer des opérations de contrôle comme la détection des exemples atypiques ou mal étiquetés. La visualisation est également intéressante pour aider à la compréhension des données. Ainsi, une visualisation peut amener à se rendre compte que les proximités sont essentiellement établies à partir de certains attributs, au détriment des autres, et conduire ainsi à reconsidérer l'espace d'apprentissage. Par ailleurs, la visualisation peut servir de point d'entrée pour une interaction entre l'utilisateur et le processus d'apprentissage. En facilitant cette interaction, elle permet de tirer parti des connaissances de l'utilisateur.

Grâce à ces caractéristiques, la représentation trouve son utilité tout au long du processus d'extraction des connaissances.

Lors des étapes de transformation et de prétraitement des données, la représentation permet de mettre à contribution l'utilisateur. En offrant une visualisation, elle l'aide à s'approprier les données pour y rechercher des exemples atypiques. Cette interaction est également l'occasion de remarquer certains biais liés au choix des attributs (sur-représentation de certains aspects des données par rapport à d'autres, attributs inutiles ou à l'inverse attributs manquants), ce qui serait impossible en travaillant directement sur une matrice attributs/valeurs.

Une fois les données validées, elles sont soumises à un algorithme d'apprentissage dont l'exécution est souvent longue et coûteuse. En offrant la possibilité de contrôler ses capacités prédictives, la représentation détermine par avance si l'état des données garantit un apprentissage de qualité suffisante pour la tâche visée par l'utilisateur. Dans le cas contraire, il est toujours possible de retourner à l'étape de sélection des données.

Au terme de l'apprentissage, la représentation trouve un intérêt lors de la prédiction. Grâce à elle, l'étiquette prédite pour un nouvel exemple est associée à une explication qui

se compose d'un côté des plus proches exemples d'étiquettes similaires et de l'autre de ceux d'étiquettes différentes. Ces éléments aident l'utilisateur à porter un regard critique sur la qualité de la prédiction proposée par le modèle.

Enfin, en dehors de l'aspect prédictif, la représentation reste un point d'accès (qui plus est visuel) aux données. À tout moment, elle sert à rechercher une information, soit à partir d'une donnée pour en trouver les données similaires, soit en naviguant de proche en proche dans la base d'exemples.

Pour ces raisons, le recours à une représentation des données en apprentissage supervisé est un atout non négligeable. Dans la section suivante, nous présentons les principales méthodes de représentation en apprentissage automatique en accord avec la définition que nous venons d'en donner. Comme nous le verrons, ces stratégies rencontrent des problèmes de passage à l'échelle que nous détaillons dans la section 4. Ces problèmes identifiés, la recherche menée au cours de ce doctorat a visé à leur dépassement pour proposer une méthode de représentation en apprentissage supervisé adaptée aux données volumineuses et qui assure une bonne qualité de prédiction.

2 Méthodes de représentation en contexte supervisé

Dans cette section, nous décrivons succinctement les principales méthodes d'apprentissage conduisant à une représentation des données, à savoir les méthodes fondées sur l'ACP (Sect. 2.1), les méthodes fondées sur les graphes de voisinage (Sect. 2.2), les arbres phylogénétiques (Sect. 2.3) et les cartes auto-organisatrices (Sect. 2.4).

2.1 Méthodes fondées sur l'ACP

2.1.1 Présentation

Issue de l'analyse de données, l'Analyse en Composantes Principales (ACP, ou encore transformation de Karhunen-Loève) est une méthode d'interprétation des données couramment utilisée (voir Lebart et al. (1997) pour plus de détails sur son exploitation en statistique exploratoire). À partir de l'espace d'apprentissage initial, l'ACP recherche un nouveau repère orthogonal dont les axes sont ordonnés suivant leur capacité à expliquer la dispersion des données. Les axes sont déterminés de manière incrémentale de telle sorte que chacun maximise la variance des exemples dans l'espace de projection. La figure 1.1 illustre ce que serait le choix d'un tel axe pour des données en deux dimensions. Le résultat de l'ACP n'est pas seulement un plan de projection des données, mais un ensemble d'axes explicatifs qui peuvent être combinés pour se faire une idée précise de l'organisation des données.

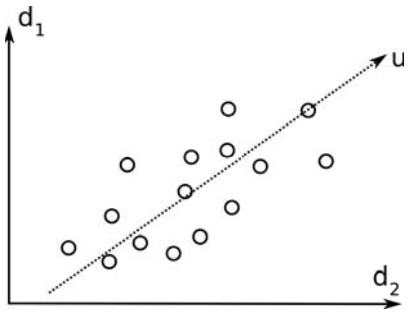


FIG. 1.1: Illustration d'une ACP sur des données en dimension 2, d_1 et d_2. L'axe u représente le premier axe renvoyé par l'ACP sur ces données. Le choix de cet axe tire parti de la corrélation qui semble exister entre d_1 et d_2 pour résumer l'information.

2.1.2 Algorithme

En pratique, l'ACP est simple à mettre en œuvre. En effet, il se démontre que le vecteur qui supporte l'axe maximisant la variance des exemples n'est autre que le premier vecteur propre de la matrice des corrélations entre les attributs. Qui plus est, l'axe qui lui est perpendiculaire et qui continue à maximiser la variance n'est autre que le deuxième vecteur propre de cette matrice, et ainsi de suite jusqu'au dernier vecteur propre. Par ailleurs, la part de variance expliquée par chaque axe est quantifiée par le rapport entre la valeur propre correspondant au vecteur et la somme de toutes les valeurs propres de la matrice.

2.1.3 Intérêt

À l'issue d'une ACP, une bonne visualisation des données est donc assurée par le plan issu des deux premiers vecteurs propres. Cette représentation peut être complétée à l'aide des vecteurs propres suivants afin d'accéder à différents aspects des données. Un autre avantage de l'ACP est qu'elle quantifie automatiquement la part d'information expliquée par la représentation, ce qui permet à l'utilisateur de juger rapidement de la pertinence des informations qui lui sont données. Enfin, si l'ACP permet la visualisation des exemples, elle permet également de visualiser la position des dimensions initiales dans le même plan de projection. Grâce à cela, l'utilisateur est en mesure de décider de l'interprétation à donner aux axes par rapport aux attributs initiaux. Il lui est alors possible de donner une signification aux positions des exemples. Pour ces qualités, le recours à l'ACP est fréquent lorsqu'il s'agit de visualiser des données multi-dimensionnelles. Comme elle est simple à mettre en œuvre, l'ACP est souvent le premier outil à utiliser pour traiter des données d'apprentissage.

2.1.4 Choix des proximités

Si l'ACP est un très bon outil d'analyse et de visualisation, elle ne modélise pas explicitement les proximités entre exemples. Ainsi, une fois l'ACP réalisée, une multitude de règles sont possibles pour déterminer le contexte d'un exemple. Suivant la définition que nous avons donnée d'une représentation, l'ACP doit être couplée à une définition de la proximité entre exemples qui détermine en particulier dans quel espace les proximités sont calculées et ce que signifie être proche pour 2 points de cet espace.

Classification Ascendante Hiérarchique. Une première solution consiste à construire une typologie des exemples en parallèle à l'ACP. En analyse des données, une Classification Ascendante Hiérarchique (CAH) est une solution couramment utilisée (on trouve le détail de cette méthode dans Jain et Dubes (1988)). Cette méthode d'apprentissage non-supervisé regroupe les exemples en classes. À partir d'un état initial où chaque exemple est une classe, elle fusionne les classes deux à deux jusqu'à obtenir une classe unique recouvrant tous les exemples.

Le choix des classes à fusionner dépend du critère optimisé par la CAH. Ainsi, le critère de Ward minimise la variance intra-classes (ou encore maximise la variance inter-classes) en fusionnant à chaque étape les classes C_i et C_j, composées respectivement de n_i et n_j exemples, qui minimise la distance :

$$D(C_i, C_j) = \frac{n_i n_j}{n \times (n_i + n_j)} d\left(g_i, g_j\right)$$

où g_i et g_j représentent les centres de gravité des classes C_i et C_j et d la distance euclidienne dans l'espace d'apprentissage.

La typologie finale est obtenue lorsque le nombre de classe souhaité est atteint ou qu'un bon compromis entre le nombre de classes et le pourcentage d'inertie expliqué soit trouvé.

Graphes de voisinage. Une autre solution, pour définir les proximités à l'issue de l'ACP, consiste à utiliser les graphes de voisinage pour obtenir une structure des données qui détermine, pour un exemple, son voisinage. Les solutions de représentation fondée sur les graphes de voisinage ont fait l'objet de nombreuses études au sein du laboratoire Eric. Aussi nous consacrons la section 2.2 de ce chapitre à ces graphes.

Utilisation des axes factoriels. La structure de voisinage n'a pas nécessairement à être construite dans l'espace d'apprentissage puisque l'ACP fournit de nouveaux axes de description des données plus pertinents par rapport à la variance de ces données. Il peut donc être judicieux de se servir de ces axes. Une stratégie courante est alors de se limiter à un petit nombre de dimensions (par exemple à celles qui expliquent 80% de la variance

des données) pour réduire la taille du problème et définir la proximité entre exemples de manière plus robuste.

2.1.5 Complexité algorithmique de l'approche.

Réaliser une ACP revient à calculer les vecteurs propres et valeurs propres de la matrice symétrique des corrélations entre attributs, soit une complexité totale en $O(d^3 + n \times d^2)$, avec n le nombre d'exemples et d le nombre de dimensions.

Le calcul d'une corrélation entre deux attributs requiert 2 parcours des valeurs des exemples pour ces attributs, le premier pour estimer la moyenne des valeurs des attributs et le second pour estimer la valeur de la corrélation. Comme le calcul de la matrice revient à calculer les corrélations entre chacune des $\frac{d^2}{2}$ paires d'attributs, la complexité finale est en $O(n \times d^2)$.

Le calcul des vecteurs propres et valeurs propres est plus complexe, aussi nous n'en donnons pas le détail ici. Néanmoins, l'algorithme décrit dans Wilkinson et Reinsch (1971) et implémenté dans la plupart des librairies mathématiques a une complexité en $O(d^3)$.

La CAH, lorsqu'elle utilise le critère de Ward, requiert le calcul des distances euclidiennes entre les centres de gravité des classes. Le calcul d'une distance entre deux exemples (ou centres de gravité) nécessite un parcours des valeurs de chaque attribut pour les deux exemples. À la première étape, il faut calculer les distances euclidiennes entre les $\frac{n^2}{2}$ paires d'exemple, soit une complexité en $O(n^2 d)$ pour trouver la meilleure paire. La CAH réitère l'opération pour k regroupements successifs. Dans le pire des cas (on poursuit l'algorithme jusqu'à obtenir 2 classes), $k = n - 2$. Enfin, à chaque itération, le nombre d'exemples et/ou de centres de gravité diminu. Au final, la complexité est en $O(c \times n^2 \times d)$ à une constante c, près liée au nombre de classes recherchées.

2.2 Méthodes fondées sur les graphes de voisinage

2.2.1 Présentation

Les graphes de voisinage sont des structures qui relient les exemples entre eux sur la base de leurs proximités. Les exemples représentent les sommets du graphe reliés (ou non) par des arêtes. Lorsqu'une arête existe entre deux sommets, ils sont considérés comme voisins. La notion de voisinage peut prendre plusieurs formes, conduisant chacune à un type de graphe. Citons, à l'instar de Muhlenbach (2002), l'arbre recouvrant minimal (Kruskal, 1956; Prim, 1957), le graphe des voisins relatifs (Toussaint et Menard, 1980; Jaromczyk et Toussaint, 1992), le graphe de Gabriel (Gabriel et Sokal, 1969) ou encore le graphe des polyèdres de Delaunay. En théorie, ces graphes permettent tous un apprentissage supervisé

des données. En pratique néanmoins, leur utilisation dans ce contexte n'est pas forcément judicieuse. Par exemple, le graphe de Delaunay a une trop forte complexité algorithmique (exponentielle en fonction du nombre de dimensions) alors que l'arbre recouvrant minimal a une règle de prédiction très proche de la méthode des k plus proches voisins (k-PPV, Cover et Hart, 1967) avec $k \leq 2$. En revanche, les graphes de Gabriel et ceux des voisins relatifs donnent des résultats intéressants pour des complexités moindres (Muhlenbach, 2002). Pour ces raisons, nous nous contentons de présenter ici ces deux derniers graphes.

Enfin, pour se servir du graphe comme modèle de prédiction, il est nécessaire d'étiqueter chaque sommet en fonction de la classe de l'exemple qu'il représente. Ainsi, lorsqu'un nouvel exemple est à prédire, il suffit de rechercher ses voisins (au sens défini par les arêtes du graphe) pour lui attribuer l'étiquette majoritaire dans ce contexte.

Graphe des voisins relatifs de Toussaint. Dans le cas de ce graphe, deux sommets a et b sont reliés par une arête si et seulement si il n'existe pas de sommet c qui soit à la fois plus proche de a que b et de b que a, soit :

$$d(a,b) \leq \max_{c \neq \{a,b\}} (d(a,c), d(b,c)), \forall c \ . \tag{1.1}$$

avec $d(a,b)$ la distance entre a et b. La figure 1.2 illustre un tel graphe dans un espace à 2 dimensions. Graphiquement, l'équation 1.1 signifie que l'intersection des deux hypersphères centrées respectivement sur a et b et de rayon $d(a,b)$ (hachurée sur la figure 1.2) est vide.

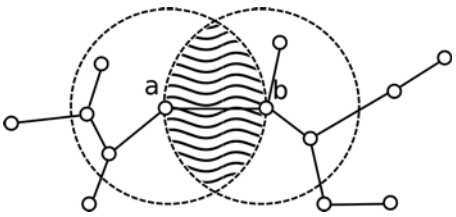


FIG. 1.2: Exemple de graphe des voisins relatifs dans $\mathbb{R}^2$. En pointillé sont représentés deux cercles de centre respectif a et b. Comme l'intersection de ces cercles (zone hachurée) est vide, l'équation 1.1 est vérifiée et a et b sont reliés par une arête.

Graphe de Gabriel. Dans le cas de ce graphe, deux sommets a et b sont reliés par une arête si et seulement si il n'existe pas de sommet c qui soit contenu dans l'hypershère de rayon $[a,b]$ et de centre μ, le milieu du segment $[a,b]$:

$$r = d(a,\mu) = d(b,\mu), \ d(c,\mu) \geq r, \forall c \ . \tag{1.2}$$

avec $d(a,\mu)$ la distance entre a et μ. La figure 1.3 illustre un tel graphe dans un espace à 2 dimensions.

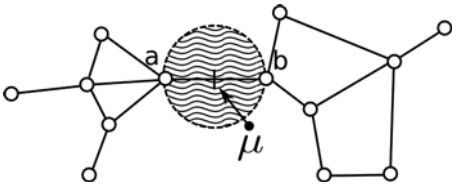


FIG. 1.3: Exemple de graphe de Gabriel dans $\mathbb{R}^2$. En pointillé est représenté le cercle de centre μ et de diamètre $[a, b]$. Comme l'aire de ce cercle (zone hachurée) est vide, l'équation 1.2 est vérifiée et a et b sont reliés par une arête.

Notons que pour un même ensemble de points, le nombre d'arêtes du graphe de Gabriel est au moins égal au nombre d'arêtes du graphe des voisins relatifs puisque si l'équation 1.1 est vérifiée pour deux points, alors l'équation 1.2 est aussi vraie pour ces points. En revanche, l'inverse n'est pas vérifié. Le voisinage d'un point sera donc plus dense en moyenne avec le graphe de Gabriel qu'avec le graphe des voisins relatifs.

Notons enfin que le graphe de Gabriel nécessite de connaître les coordonnées des points dans l'espace d'apprentissage alors que le graphe des voisins relatifs se construit aussi bien avec une matrice des distances entre exemples. En pratique, le recours au graphe des voisins relatifs est donc plus fréquent.

2.2.2 Algorithmes

Ces deux graphes sont générés sur la base d'un algorithme commun dont seule la fonction $arête(\alpha, \beta, \gamma)$ (qui détermine si un sommet γ empêche la formation d'une arête entre deux sommets α et β) varie. Nous présentons ici la racine commune de ces algorithmes (algorithme 1) ainsi que les fonctions $arête_{GVR}$ (algorithme 2) et $arête_{GG}$ (algorithme 3) qui implémentent la fonction $arête$ respectivement pour le graphe des voisins relatifs et le graphe de Gabriel. Cette fonction définit la notion de proximité entre exemples.

Algorithme 1 : Construction d'un graphe

entrées : **G**, l'ensemble des n exemples complètement interconnectés dans $\mathbb{R}^p$.
sorties : **G**, un graphe de voisinage des n exemples dans $\mathbb{R}^p$.

pour $\alpha \leftarrow 1$ **à** $n-1$ **faire**
  pour $\beta \leftarrow \alpha+1$ **à** n **faire**
    $\gamma \leftarrow 1$;
    $arête_{\alpha\beta} \leftarrow$ vrai ;
    tantque $\gamma \leq n$ **et** $arête_{\alpha\beta}$ **faire**
      si $\gamma \neq \alpha$ **et** $\gamma \neq \beta$ **alors**
        si non arête(α, β, γ) **alors**
          $arête_{\alpha\beta} \leftarrow$ faux ;
          $G \leftarrow G - \{(\alpha, \beta)\}$;
        fin-si
      fin-si
      $\gamma \leftarrow \gamma + 1$;
    fin-tq
  fin-pour
fin-pour
renvoie G;

Fonction $arête_{GVR}(X, \alpha, \beta, \gamma)$

entrées : **X**, l'ensemble des n exemples dans $\mathbb{R}^p$; $1 \leq \alpha, \beta, \gamma \leq n$
sorties : vrai si γ n'empêche pas la connexion entre α et β selon l'eq. 1.1, faux sinon.

si $\mathrm{d}(X_\alpha, X_\beta) \leq \max(\mathrm{d}(X_\alpha, X_\gamma), \mathrm{d}(X_\beta, X_\gamma))$ **alors renvoie** vrai ;
sinon renvoie faux ;

Fonction $arête_{GG}(X, \alpha, \beta, \gamma)$

entrées : **X**, l'ensemble des n exemples dans $\mathbb{R}^p$; $1 \leq \alpha, \beta, \gamma \leq n$
sorties : vrai si γ n'empêche pas la connexion entre α et β selon l'eq. 1.2, faux sinon.

$\mu \leftarrow \mathrm{centre}(\alpha, \beta)$;
si $\mathrm{d}(X_\mu, X_\alpha) \leq \mathrm{d}(X_\mu, X_\gamma)$ **alors renvoie** vrai ;
sinon renvoie faux ;

2.2.3 Projection dans le plan

Les graphes de voisinage ne satisfont pas à notre définition des représentations. Ils offrent bien une structure aux données (les relations de proximité entre exemples sont modélisées par les arêtes du graphe tandis que le lien entre la position dans l'espace et

l'étiquette est conservé) mais cette structure étant construite dans l'espace initial des données, elle ne permet pas de les visualiser. Cet aspect se corrige facilement par une méthode de projection des données de l'espace d'apprentissage vers un plan. Parmi les méthodes de projection usuelles, nous avons déjà présenté l'Analyse en Composantes Principales (ou ACP). Le caractère linéaire de l'ACP lui est fréquemment reproché. Dans certains cas, cette linéarité empêche de trouver une projection qui explique suffisamment la variance des exemples. Ce reproche est d'ailleurs d'autant plus vrai que la dimensionnalité des données augmente. Plusieurs approches réalisent une projection non-linéaire, comme l'Analyse en Composantes Curvilignes (Curviligne Composante Analysis, CCA, Demartines et Hérault (1997)), Isomap (Tenenbaum et al., 2000), la mise à l'échelle multi-dimensionnelle (*Multi-Dimensional Scaling*, Shepard et Carroll (1965)), t-SNE (van der Maaten et Hinton, 2008) et la mise en correspondance de Sammon (Sammon's mapping, Sammon (1969)). Nous présentons ici cette dernière technique dont l'utilisation est parfois couplée avec les cartes auto-organisatrices.

Mise en correspondance de Sammon (Sammon's mapping). La mise en correspondance de Sammon permet d'obtenir une projection non-linéaire des données. Cette technique vise à préserver la topologie initiale des données en s'assurant que :

- 2 exemples proches dans l'espace d'apprentissage soient également proches dans l'espace de projection et, réciproquement
- 2 exemples proches dans l'espace de projection correspondent à des exemples proches dans l'espace d'apprentissage.

En ce sens, cette méthode partage des similarités avec les cartes auto-organisatrices présentées en section 2.4. Pour réaliser cet objectif, Sammon a mis en place une fonction d'erreur en charge de quantifier la distorsion (ou tension) qui existe entre les deux espaces à l'issue de la projection :

$$E_{Sammon} = \frac{1}{\sum_{i}^{n-1}\sum_{j=i+1}^{n}\delta_{ij}} \sum_{i}^{n-1}\sum_{j=i+1}^{n} \frac{(\delta_{ij} - d_{ij})^2}{\delta_{ij}}$$

avec n le nombre d'exemples présents dans les données, d_{ij} la distance entre les exemples i et j dans l'espace d'apprentissage et δ_{ij} la distance entre i et j dans l'espace de projection.

Cette fonction prend ses valeurs dans l'intervalle $[0;1]$ (0 indiquant une projection de qualité médiocre). Pour maximiser cette fonction et trouver ainsi la meilleure projection possible, Sammon propose d'utiliser une descente de gradient à partir d'une configuration initiale choisie au hasard (voir par exemple l'implémentation donnée dans SOM-PAK par Kohonen et al. (1996)).

2.2.4 Qualité de la représentation

L'utilisation de graphes comme représentation nécessite deux étapes successives, puisqu'il faut d'abord modéliser les proximités entre exemples (construire le graphe en lui-même) puis les rendre visualisables (projection du graphe dans le plan). L'inconvénient évident de ce processus réside dans le temps nécessaire à la réalisation de ces deux étapes dont les complexités respectives ne sont pas triviales (voir la sous-section 2.2.6 pour leur étude). En revanche, l'avantage majeur est que la représentation repose sur une structure qui n'est pas soumise aux aléas de la projection : il n'y a ni variance inexpliquée, ni distorsion résiduelle dans le graphe de voisinage. Cette structure n'est pas nécessairement sans défauts (puisque, comme le souligne entre autres Muhlenbach (2002), le graphe reste sensible aux attributs non pertinents) mais elle correspond à une modélisation très proche des données initiales.

Des travaux menés au laboratoire ERIC ont donc cherché à valider la qualité de cette représentation pour se renseigner sur ses capacités prédictives. L'idée centrale, initiée par Sebban et Zighed (1996) et généralisée par Zighed et al. (2001), a été de questionner les arêtes du graphe, en particulier celles qui relient des sommets d'étiquettes différentes, pour construire une statistique corrélée avec l'erreur en généralisation du graphe. Ces travaux reposent sur l'hypothèse suivante : s'il est possible de distinguer parfaitement deux modalités de l'étiquette en fonction des descripteurs, alors le graphe construit sur ces descripteurs est tel que les voisins d'un sommet ont tendance à avoir la même étiquette que le sommet lui-même. À l'inverse, si les voisins d'un sommet ont tendance à avoir une étiquette différente du sommet lui-même, alors il est difficile de distinguer les différentes modalités de l'étiquette. Autrement dit, plus le nombre d'arêtes entre des sommets d'étiquettes différentes, appelées arêtes coupées, est faible, plus la qualité de la représentation est bonne. À partir de cette hypothèse, Lallich (2002) a fait le lien avec les outils de l'auto-corrélation spatiale (et en particulier avec ceux de la statistique produit-croisé initiée par Mantel (1967)), pour proposer la statistique du poids des arêtes coupées. Cette statistique correspond à la proportion d'arêtes coupées par rapport au nombre total d'arêtes sur le graphe. Le nombre d'arêtes coupées est le résultat d'un produit terme à terme entre deux matrices booléennes de taille $n \times n$, celle des proximités entre exemples au sens des descripteurs (deux exemples sont proches s'ils sont reliés par une arête sur le graphe) et celle des proximités entre exemples au sens des étiquettes (pour lesquelles deux exemples sont proches s'ils ont des étiquettes différentes). Grâce aux outils de l'auto-corrélation spatiale, la loi de cette statistique est connue sous l'hypothèse nulle d'indépendance entre les sommets du graphe et leur étiquette, ce qui permet d'accéder à la *p-value*[1] de cette statistique sous cette hypothèse. Enfin, les travaux de Zighed et al. (2002) ont montré le bien-fondé de cette statistique en la

[1] « Dans un test statistique, p est la probabilité d'obtenir au hasard un résultat au moins aussi extrême que la valeur réellement observée, sous l'hypothèse nulle. » (source : *wikipédia.fr*)

corrélant avec le taux d'erreur en généralisation de l'apprentissage par graphe de voisinage.

En pratique, pouvoir valider la qualité d'une représentation n'est pas sans intérêt, notamment dans les étapes de pré-traitement comme nous l'avons souligné dans la section 1. Ainsi, les travaux menés par Muhlenbach et al. (2003) se sont servis avec succès de cette statistique pour détecter les exemples atypiques à la fois dans le cas d'une étiquette catégorielle et dans celui d'une étiquette numérique (Lallich et al., 2003b). Dans le domaine du traitement de l'image, ces travaux ont permis de proposer une stratégie de segmentation d'image fondée sur les graphes de voisinage et la validation de leur représentation (Lallich et al., 2003a).

2.2.5 Interaction avec l'utilisateur

L'intérêt central de la représentation est de relier l'utilisateur au processus d'extraction grâce à la visualisation des données et à leur structuration. Scuturici et al. (2005) puis Hacid (2008) exploitent cette possibilité avec les graphes de voisinage pour permettre la recherche d'information dans des bases de données images. Grâce à la structure des données, ils sont bien sûr capables de renvoyer les images semblables à une requête, mais pas seulement. Cette structure permet à Hacid (2008) de propager des annotations sur des images cibles et fournit ainsi à l'utilisateur des connaissances supplémentaires sur sa requête. Enfin, à partir des résultats d'une requête, par exemple un petit sous-ensemble d'images proches d'une image cible soumise par l'utilisateur, celui-ci peut naviguer à travers les images qui lui paraissent répondre au mieux à sa requête en suivant les liens de voisinage pour trouver la réponse exacte à sa recherche.

2.2.6 Complexité algorithmique de l'approche.

Comme l'indique l'algorithme 1, construire un graphe requiert de déterminer l'existence d'une arête entre chacune des $\frac{n^2}{2}$ paires d'exemple. Pour savoir si une telle arête existe, il est nécessaire de vérifier qu'aucun des exemples restant ne l'empêche, ce qui nécessite un parcours supplémentaire des n exemples et de la valeur de leurs d attributs (pour calculer les distances entre exemples) dans chaque cas. Au final, la complexité est donc en $O(n^3 \times d) \sim O(n^3)$.

À cette complexité, il faut ajouter celle de la projection dans le plan réalisée soit avec une ACP (dont la complexité est donnée en section 2.1.5) soit avec une mise en correspondance de Sammon. De ce dernier cas, il est nécessaire de calculer les distances entre les $\frac{n^2}{2}$ exemples à la fois dans le plan de projection et dans l'espace d'apprentissage pour ensuite maximiser la formule donnée en section 2.2.3 au cours de k itérations d'une descente en gradient (avec $k < n$), soit une complexité finale en $O(k \times d \times n^2) \sim O(n^2)$.

2.3 Arbres phylogénétiques

2.3.1 Présentation

Parmi les stratégies de représentation que nous présentons ici, les arbres phylogénétiques sont sans conteste la stratégie la moins fréquemment exploitée, certainement en raison de leur double complexité à la fois algorithmique et d'implémentation. Ces arbres (ou X-arbres) sont néanmoins des outils intéressants pour la représentation des données puisque, contrairement aux graphes de voisinage, ils modélisent la proximité entre exemples dans une structure plane que l'utilisateur peut visualiser. Au-delà de l'intérêt naturel porté par la bio-informatique sur ces structures, notamment pour représenter l'évolution de données de mesures d'expression de gènes (Arigon, 2006), ils ont été utilisés assez tôt en analyse de données (voir Barthélemy et Luong (1987) pour une description des arbres phylogénétiques dans ce contexte) en particulier pour la représentation et l'analyse de corpus textuel (voir Barthélemy et Luong (1998) pour un tutoriel).

Description formelle. Dans ce paragraphe, nous définissons de manière formelle les X-arbres, en reprenant les notations de Clech (2004).

- **Graphe** : soit $(\Sigma,\ A)$ un graphe de sommets Σ et d'arêtes A. Le degré du sommet $\alpha \in \Sigma$, noté $|\alpha|$, est le nombre d'arêtes de ce sommet.
- **Feuille et nœud** : une feuille est un sommet de degré 1, un nœud un sommet de degré supérieur à 1.
- **X-arbres et étiquette** : un X-arbre est un couple, noté (H, Ett), composé d'un graphe $H = (\Sigma,\ A)$ et d'une fonction d'étiquetage Ett. La fonction étiquette définie de X (l'ensemble fini des étiquettes associées à H) à Σ est telle que $\forall \alpha \in (\Sigma - Ett(X)),\ |\alpha| \geq 3$.
- **Sommets réels *vs* latents** : les sommets de $Ett(X)$ sont dits réels alors que les sommets de $\Sigma - Ett(X)$ sont dits latents.
- **X-arbre libre** : un X-arbre (H, Ett) est libre s'il y a équivalence entre les feuilles de H et sa fonction $Ett(X)$.
- **X-arbre valué** : un X-arbre valué est un triplet (H, Ett, L) où L associe à chaque arête de H une valuation.
- **Distance additive** : sur un X-arbre (H, Ett, L), la distance additive, notée $\delta_{\alpha,\beta}$, entre deux sommets $\alpha,\ \beta \in \Sigma$ est égale à la somme des valuations des arêtes du chemin entre α et β.

Dans la suite, nous allons nous intéresser aux X-arbres libres et valués qui sont ceux utilisés en analyse de données. Ces X-arbres représentent l'ensemble des exemples de X par les sommets réels de l'arbre notés $Ett(X)$. Comme ces X-arbres sont libres, leurs sommets réels sont aussi les feuilles de l'arbre. Un schéma de X-arbre est donné par la figure 1.4.

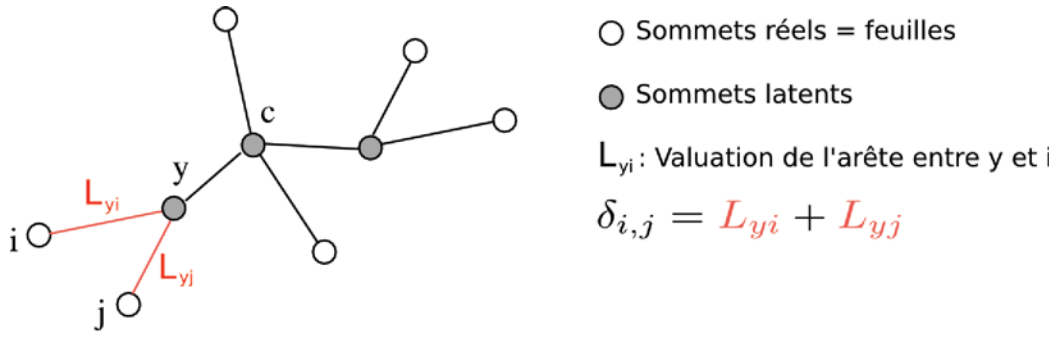


FIG. 1.4: Schéma d'un X-arbre libre valué.

Exemple. L'étude menée par Mayaffre (2003) sur l'évolution des discours politiques de Jacques Chirac et Lionel Jospin lors de leur cohabitation de 1997 à 2002 est un exemple typique de ce que les arbres phylogénétiques permettent. Le corpus de textes analysés représente 2100 discours segmentés en 12 parties regroupant chacune les discours tenus par l'un des deux hommes politiques pour une année donnée. Ce sont ces parties qui ont été analysées par Mayaffre (2003) afin de juger de l'évolution des discours de ces deux candidats à l'élection présidentielle de 2002. Grâce à la représentation visuelle de l'arbre phylogénétique (reproduite à l'identique en figure 1.5), on retrouve facilement l'existence de la séparation politique entre L. Jospin et J. Chirac mais, plus intéressant encore, l'évolution dans le temps de leurs discours respectifs. Comme le montre la figure 1.5, la séparation politique entre les deux hommes tend à s'effacer avec le temps (à l'exception des discours de L. Jospin en 1998, la distance d'un nœud de l'arbre au centre est d'autant plus courte qu'elle correspond à une année tardive dans la cohabitation). Enfin, l'analyse montre le recentrage final des deux candidats avant l'élection présidentielle.

2.3.2 Algorithme

Il existe plusieurs familles d'algorithmes permettant d'obtenir les X-arbres. Dans sa thèse sur l'analyse de données complexes, Clech (2004) cite notamment les méthodes de parcimonie, les méthodes de vraisemblance et les méthodes de distance. Ce sont ces dernières qui sont exploitées en fouille de données où il est facile d'obtenir la matrice carrée des distances entre individus. Nous présentons ici la méthode de jointure des voisins (*neighbor joining*, Saitou et Nei (1987)).

Cet algorithme convertit la matrice des distances entre exemples en leur représentation sous forme de X-arbre libre valué au cours d'un processus itératif dont l'objectif est de minimiser l'écart entre les distances additives (δ) et les distances réelles qu'elles représentent (d). Les étapes d'une itération sont schématisées par la figure 1.6 et reprises par l'algorithme 4. À la première étape, les exemples sont représentés sous la forme d'une arborescence

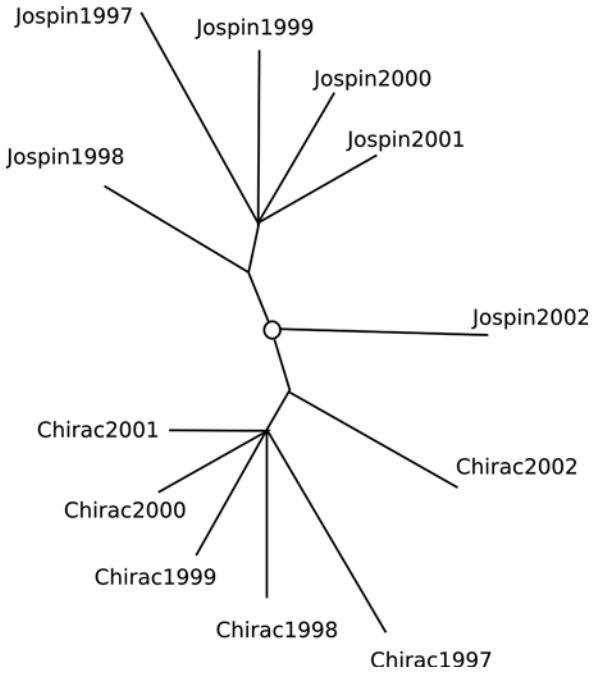


FIG. 1.5: Analyse arborée selon le vocabulaire du corpus cohabitation Chirac-Jospin (1997-2002). Illustration issue de Mayaffre (2003).

étoilée. Le sommet central (c sur la figure) est un sommet latent relié à toutes les feuilles de l'arbre. À partir de cette configuration, l'algorithme cherche à regrouper les sommets deux à deux en veillant à conserver une longueur d'arbre totale la plus petite possible. On note S_{ij} la longueur totale d'un arbre étoilé de centre c sur lequel la paire (i, j) est regroupée et on cherche quelle paire minimise cette longueur. La longueur totale d'un sous-arbre étoilé de centre c s'obtient à partir des distances entre exemples grâce à la formule :

$$S_0 = \sum_{i}^{\Sigma} L_{ic} = \frac{1}{n-1} \sum_{i<j}^{n} d_{ij}$$

Le calcul de S_{ij} est la somme des longueurs totales du sous-arbre de centre y (entouré de vert sur la figure 1.6-c, noté S_1), du sous-arbre de centre c (entouré de rouge sur la figure 1.6-c, noté S_2) et de l'arête entre c et y de valuation L_{cy}, soit :

$$S_{ij} = S_1 + S_2 + L_{cy} = \frac{1}{2(n-2)} \sum_{\substack{k=1 \\ k\neq i, k\neq j}}^{n} (d_{ik} + d_{jk}) + \frac{1}{2} d_{ij} + \frac{1}{n-2} \sum_{\substack{k<l \\ k,l\neq i, k,l\neq j}} d_{kl} \qquad (1.3)$$

Une fois déterminée la paire (i, j) à regrouper, les valuations de L_{yi}, L_{yj} et L_{yc} sont estimées, puis le tableau initial des dissimilarités est mis à jour, de manière à remplacer les lignes et les colonnes affectées aux exemples i et j par la paire (i, j). Pour cette paire, les distances avec l'exemple k sont telles que $d_{(i,j),k} = (d_{ik} + d_{jk})/2$. Le X-arbre se retrouve

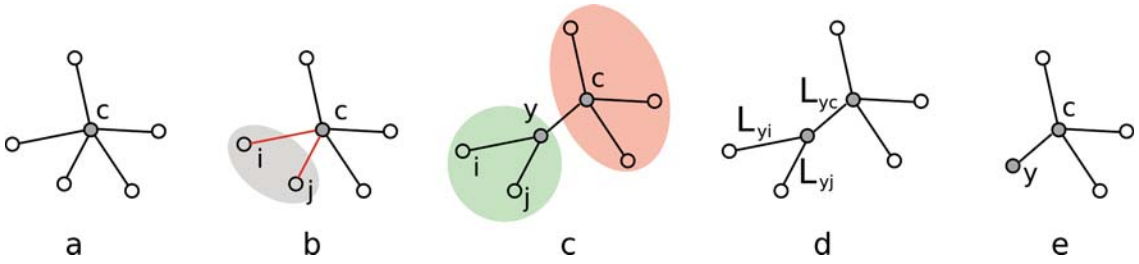


FIG. 1.6: Illustration des étapes du *neighbor joining* : (a) les données sont sous la forme d'une arborescence étoilée, (b) recherche de la meilleure paire d'exemples (i, j) qui minimise la longueur d'arête, (c) ajout du sommet y, (d) mise à jour des nouvelles valuations et (e) regroupement des exemples (i, j) dans la matrice des distances.

dans une configuration étoilée puisque les sommets i et j ont été remplacés par le sommet latent y, ce qui permet de commencer une nouvelle itération de l'algorithme.

Algorithme 4 : *Neighbor joining*

entrées : **D**, la matrice carrée de taille $n \times n$ des distances entre exemples. **L**, le vecteur vide des valuations de taille $n + 1$ d'un arbre en étoile. L_i est la valuation du sommet réel i avec le sommet centre c.

sorties : **L**, la matrice des valuations de taille $(n + l) \times l$ avec l le nombre de sommets latents. $L_{ij} > 0$ si i et j sont reliés par une arête et i est l'indice d'un sommet latent si $i > n$.

pour $k \leftarrow 1$ **à** $n - 2$ **faire**

- $(i, j) \leftarrow argmin_{i,j \in D}\ S_{ij}$; `/* Recherche de la meilleure paire */`
- `Etendre(`L`)` ; `/* Ajout du sommet` Y `*/`
- $y \leftarrow k + 1$
- $L_{ic} \leftarrow 0$, $L_{jc} \leftarrow 0$; `/* Suppression des anciennes arêtes */`
- `Maj` $(L_{cy}, L_{(n+y)c}, L_{iy}, L_{jy})$; `/* Mise à jour des valuations */`
- $D \leftarrow$ `Fusion(`D, i, j`)` ; `/* Fusion des lignes` i `et` j `dans` D `*/`

fin-pour

renvoie L

2.3.3 Intérêt

Comme le montre l'exemple donné en section 2.3.1, l'arbre phylogénétique est particulièrement adapté pour mettre en valeur une évolution des exemples les uns par rapport aux

autres. Ainsi, dans l'exemple, le X-arbre permet de visualiser le recentrage des discours des deux hommes politiques.

Cette caractéristique est aussi reconnue à l'ACP, cependant, en comparant sur les mêmes données une ACP au X-arbre de la figure 1.5, Mayaffre (2003) remarque que la représentation par arbre phylogénétique conserve plus fidèlement les rapports de proximités entre paires de points proches, lors du passage de l'espace d'apprentissage à l'espace de projection. En effet, l'arbre phylogénétique, pour minimiser l'écart entre les distances calculées dans l'espace d'apprentissage et les distances additives, réalise des ajustements locaux de la projection en regroupant les individus proches. À l'inverse, l'ACP procède de manière globale en cherchant des axes qui résument au mieux l'ensemble des exemples.

2.3.4 Complexité algorithmique de l'approche.

Le calcul de l'arbre phylogénétique commence par la recherche de la meilleure paire d'exemples à fusionner. Pour ce faire, les S_{ij} sont calculés pour chacune des $\frac{n^2}{2}$ paires d'exemples (i, j). Chaque S_{ij} requiert un nouveau parcours des exemples pour calculer les distances du premier terme de l'équation 1.3. La recherche de la meilleure paire a alors une complexité en $O(d \times n^3)$. Une fois cette paire déterminée, il faut étendre la matrice L des valuations et calculer les nouvelles valuations pour ajouter le sommet y. Chacune des 3 valuations ne requiert qu'un parcours des exemples, soit une complexité en $O(3n)$. Enfin, il faut fusionner deux lignes de la matrice des distances, ce qui requiert là encore un parcours des exemples pour calculer les nouvelles distances avec le noeud y. Ces opérations sont répétées pour les k fusions de paires, avec $k \leq n - 2$ puisqu'il y a au plus $n - 2$ fusions possibles (cas où, à chaque itération, un exemple fusionne toujours avec une fusion existante, excepté bien sûr à la première itération). Au final, la complexité est donc bornée par la recherche de la meilleure paire, soit en $O(k \times d \times n^3) \sim O(n^3)$ dans le cas général et en $O(n^4)$ dans le pire des cas.

2.4 Cartes auto-organisatrices

La dernière stratégie de représentation citée ici est celle des cartes auto-organisatrices (Kohonen, 1982). À l'instar des arbres phylogénétiques, elles réalisent conjointement la structuration des données et leur projection dans un plan en conservant la topologie locale de l'espace. Leur algorithme de faible complexité en fait un outil puissant, idéal pour la représentation et la visualisation. Comme elles sont au cœur des contributions présentées aux chapitres 2 et 3, nous reportons leur présentation détaillée au chapitre suivant.

3 Récapitulatif des méthodes de représentation

Le tableau 1.1 récapitule les propriétés afférentes aux méthodes de représentation que nous avons présentées dans ce chapitre. Les deux premières lignes indiquent l'algorithme sur lequel se fondent respectivement la modélisation des proximités et la visualisation des données. Sur ces aspects, seuls les arbres phylogénétiques et les cartes auto-organisatrices offrent une procédure unique.

	Méthodes				
Proximités	CAH	Graphes[1]	Graphes[1]	X-arbres	Cartes[2]
Visualisation	ACP	ACP	Sammon		
Projection	linéaire	linéaire	non-linéaire	non-linéaire	non-linéaire
Contexte	classe	voisins	voisins	voisins	classe
Navigation					
Validation	non	oui	oui	non	non
Complexité	$n^2 + d^3$	$n^3 + d^3$	$n^3 + n^2$	n^3	$n \times m \times d$

TAB. 1.1: Tableau récapitulatif des propriétés des différentes méthodes de représentation des données en apprentissage supervisé.

La propriété *projection* indique si le plan de projection est une combinaison *linéaire* des variables (ce qui est le cas seulement pour l'ACP) ou *non-linéaire*. Il s'agit d'un aspect important, en particulier face à des données volumineuses, comme nous le verrons en section 4.1.2, puisqu'une projection non-linéaire permet une projection plus souple face à des topologies complexes de données.

Tous les algorithmes présentés autorisent la *contextualisation* de l'information et la *navigation* à travers les exemples, mais suivant les algorithmes, deux stratégies sont possibles. Les graphes de voisinage et les X-arbres sont deux structures pour lesquelles les voisins d'un exemple sont connus. L'avantage est une précision assez fine lors de la navigation puisque seuls les proches voisins sont proposés. L'inconvénient est que la contextualisation requiert la reconstruction de cette structure pour connaître le voisinage d'un nouvel exemple. À l'inverse, pour les cartes et la CAH, la navigation et la contextualisation passent par l'identification de la classe d'un exemple. Pour un exemple connu, la navigation pourra proposer tous les exemples associés à la classe, ce qui sera plus ou moins précis en fonction de l'étendue de la classe. En revanche, lors de la contextualisation, il suffit de trouver la

[1] Concerne uniquement le graphe des voisins relatifs et le graphe de Gabriel

[2] m, le nombre de neurones de la carte. Cette complexité est détaillé en section 2.1 du chapitre 2

classe du nouvel exemple pour connaître son contexte.

La propriété *validation* indique s'il est possible de valider la qualité de la représentation suivant l'étiquette de classe. Pour cette propriété, nous avons répondu par l'affirmative seulement pour les graphes de voisinage. Néanmoins, comme nous le proposons dans le chapitre 2, la stratégie de validation associée aux graphes peut s'adapter aux cartes auto-organisatrices.

Enfin, la dernière ligne du tableau rappelle la *complexité algorithmique* pour l'apprentissage de la représentation. Les méthodes fondées sur les graphes ou les X-arbres ont une complexité en $O(n^3)$ alors que celles fondées sur l'ACP ont une borne inférieure en $O(d^3)$. Seuls les cartes ont donc une complexité qui n'est pas cubique en fonction des données, ce qui est un avantage sérieux face à des données volumineuses.

4 Problèmes de l'apprentissage face aux données volumineuses

Rappelons que nous entendons par données volumineuses des données décrites à la fois par un grand nombre d'individus et par un grand nombre d'attributs. Dans le cas d'un grand nombre d'attributs, les données sont qualifiées de données en grande dimension. Les données volumineuses résultent souvent de l'analyse d'objets associés à une sémantique que l'analyse numérique a du mal à prendre en compte comme le texte, l'image ou les puces à ADN. Elles peuvent également découler de l'agrégation d'une multitude d'objets simples. Dans ce dernier cas néanmoins, nous préférons parler d'objets complexes.

Comme nous l'avons souligné dans l'introduction, l'apparition de données volumineuses a engendré des besoins d'analyse sans pour autant que les méthodes soient adaptées à leurs particularités. À ce titre, le problème des données en grandes dimensions est devenu pour Donoho (2000) l'un des challenges du XXI^{e} siècle pour le domaine de l'ECD.

Plusieurs expériences, qu'elles soient théoriques ou pratiques, ont illustré les problèmes posés par l'analyse des données volumineuses. Cette section recense les points importants de ces expériences afin de mieux cerner comment ce type de données affecte les outils d'analyse actuels, en particulier l'obtention d'une représentation des données. Cette section est organisée de la manière suivante. Nous détaillons d'abord les problèmes spécifiques d'un grand nombre d'attributs (section 4.1) et d'exemples (section 4.2) sur les méthodes de fouille de données, en insistant sur leur impact sur la représentation. Ensuite, nous présentons l'impact de ces données lorsqu'il s'agit de valider les connaissances qui en ont été extraites (section 4.3). Enfin, nous ouvrons sur les solutions que nous avons étudiées en réponse à ces problèmes (section 4.4).

4.1 Espace de Grandes Dimensions

Chaque exemple d'une base de données se voit comme un point dans un espace dont les dimensions correspondent aux attributs de l'exemple. Lorsque le nombre d'attributs augmente, la dimensionnalité de l'espace augmente également. Au fur et à mesure que le nombre d'attributs augmente, les propriétés de l'espace dans lequel sont décrits les exemples changent. On parle alors d'Espace en Grandes Dimensions (EGD ou *High Dimensional Spaces* en anglais). Nous présentons ici certaines de ces particularités, en insistant sur l'impact qu'elles peuvent avoir en extraction des connaissances.

4.1.1 Particularités des Espaces de Grandes Dimensions.

Malédiction de la dimension. Pour caractériser les problèmes liés aux EGD, l'expression de malédiction (ou fléau) de la dimension (*curse of dimensionnality* en anglais) est souvent employée. Elle a été utilisée pour la première fois par Bellmann (1961)[(2)] lorsqu'il a étudié le coût d'optimisation d'une fonction par une recherche exhaustive suivant le nombre de dimensions d de l'espace. Si 10 points sont connus dans l'espace, le nombre de fonctions à évaluer est de 10^d. Il augmente donc exponentiellement avec le nombre de dimensions.

Le problème de l'apprentissage supervisé revient à trouver une (ou plusieurs) fonction(s) qui sépare(nt) un nombre fini d'exemples suivant leurs étiquettes. Le résultat de Bellman montre que le nombre de fonctions à évaluer pour déterminer la meilleure fonction augmente exponentiellement. En pratique, les algorithmes d'apprentissage ne procèdent jamais à une recherche exhaustive mais au contraire à une approximation. Le problème est alors reporté du coût de l'optimisation à sa précision. Trunk (1979) illustre ce problème par un exemple simple. Il considère un problème de dimension d à deux classes, ω_1 et ω_2, de probabilités égales $p(\omega_1) = p(\omega_2) = 1/2$ avec des distributions conditionnelles gaussiennes $p(X|\omega_i) \sim G(\mu_i, I), i = \{1, 2\}$. Les vecteurs moyennes μ_i sont définis de la manière suivante :

$$\mu_1 = -\mu_2 = \left(1, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{3}}, \ldots, \frac{1}{\sqrt{d}}\right)$$

et I est la matrice identité de taille d. Les vecteurs de moyennes sont choisis de telle sorte que leur pouvoir discriminant soit maximum sur la première composante et diminue sur les composantes suivantes. Ces vecteurs sont les seuls paramètres du modèle. Trunk considère alors deux cas de figure. Dans le premier, les paramètres sont connus, alors que dans le second, les paramètres sont estimés à partir d'un ensemble d'exemples de taille fini. Suivant ces deux cas, Trunk montre que le comportement de la probabilité d'erreur lors de la classification varie énormément puisque, dans le premier cas, cette probabilité tend vers 0 quand d tend vers l'infini, alors que dans le second, à l'inverse, elle tend vers $1/2$.

[2] à la page 97, « In view of all that we have said in the forgoing sections, the many obstacles we appear to have surmounted, what casts the pall over our victory celebration ? It is the curse of dimensionality, a malediction that has plagued the scientist from the earliest days. »

Cet exemple est une bonne illustration du fléau de la dimension sur les problèmes d'apprentissage. Il montre que l'augmentation du nombre de dimensions – qui devrait pourtant être de nature à rajouter de l'information sur la discrimination des classes – rend plus difficile l'estimation des paramètres du modèle, ce qui dégrade ensuite les chances de trouver une bonne séparation des classes.

Pour permettre l'estimation des paramètres la plus précise possible, il faut que le nombre d'exemples soit suffisamment grand pour bien décrire l'espace d'apprentissage. Il est donc intéressant de pouvoir déterminer le nombre d'exemples nécessaires à un bon apprentissage en fonction de la dimension de l'espace. Une pratique courante, proposée par Jain et Chandrasekaran (1982) (d'après Jain et al. (2000)), est de considérer qu'il faut un rapport de 10 entre le nombre d'exemples par classe et la dimension soit :

$$\frac{n}{d \times |Y|} \geq 10$$

avec $|Y|$, le nombre de modalités de la variable de classe.

Phénomène de l'espace vide (*empty space phenomenon*). Ce phénomène est une particularité statistique des EGD. Pour comprendre ces espaces, différents auteurs[3] ont cherché à décrire leur géométrie. Dans cette optique, Scott et Thompson (1983) se sont intéressés au volume d'une hypersphère et à son évolution lorsque le nombre de dimensions augmente. La figure 1.7 représente graphiquement le volume d'une hypersphère de rayon 1 en fonction du nombre de dimensions. Comme le montre cette figure, le volume de l'hypersphère augmente avec le nombre de dimensions pour une taille d'espace inférieure à 5. Au-delà, ce volume diminue pour être quasi nul lorsque le nombre de dimensions approche 20.

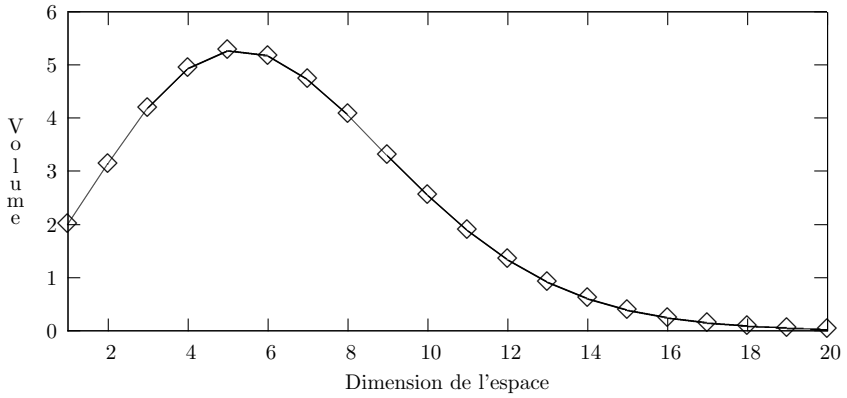


Fig. 1.7: Volume d'une hypersphère de rayon 1 en fonction de la dimension

[3]En plus des travaux présentés ici, notons également l'expérience rarement citée de Dunau et Hiriart-Urruty (2003) sur le cochonnet n-dimensionnel.

Pour mieux comprendre ce phénomène, Verleysen (2003) propose de s'intéresser, non plus simplement au volume d'une hypersphère, mais au rapport entre le volume d'une hypersphère de rayon $0,9$ et celui d'une hypersphère de rayon 1, dans le cas de données uniformes. La figure 1.8 donne l'évolution de ce rapport lorsque le nombre de dimensions augmente. Ce rapport est de $0,8$ pour un espace de dimension 2. Autrement dit, 80% du volume d'une sphère de rayon 1 est contenu dans une sphère de rayon $0,9$ (et de même centre), ce qui semble naturel à l'esprit. En revanche, lorsque le nombre de dimensions devient supérieur à 20, ce rapport devient inférieur à $0,1$. Dans ce cas, la grande majorité du volume d'une hypersphère (là où les points sont le plus probablement concentrés) se trouve sur les bords de celle-ci.

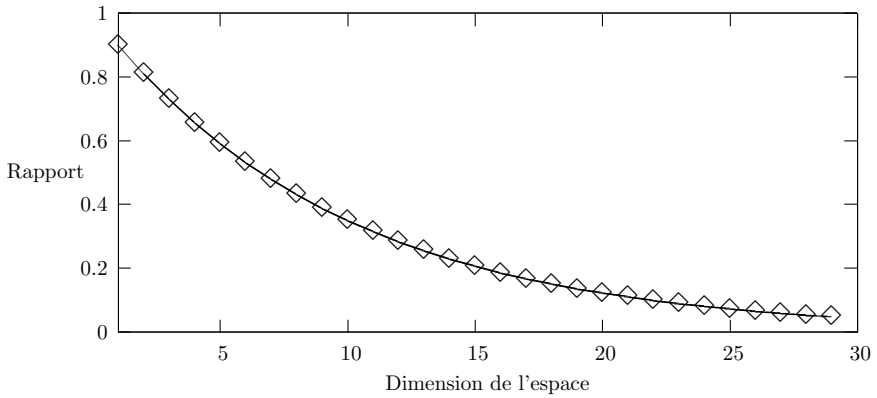


Fig. 1.8: Rapport entre les volumes d'une hypersphère de rayon $0,9$ et d'une hypersphère de rayon 1 en fonction du nombre de dimensions

En extraction des connaissances, l'hypersphère est fréquemment utilisée pour délimiter une région de l'espace. En recherche d'information ou en indexation, par exemple, certains algorithmes recherchent l'ensemble des exemples se trouvant à une distance h d'un point référence. Ils considèrent alors l'hypersphère de rayon h. Or, les expériences précédentes tendent à montrer que le volume de cette hypersphère risque d'être trop faible pour avoir une chance de contenir d'autres exemples.

Concentration des distances (*concentration of measure phenomenon* en anglais). La géométrie particulière des EGD, approchée par les expériences sur le volume des hypersphères, se répercute sur le calcul des distances, qu'il soit entre deux vecteurs ou entre deux points de l'espace.

- Distance entre vecteurs (sous l'hypothèse d'indépendance entre les dimensions) : au cours de sa thèse, Demartines (1994) a exprimé l'espérance mathématique et la variance de la norme de vecteurs à composantes aléatoires dans un espace à d dimensions. Si l'espérance croît en $\sqrt{d}$, la variance, elle, tend vers une constante quand d tend vers l'infini. L'in-

terprétation qu'en donne Demartines est que, pour d suffisamment grand, les vecteurs semblent normalisés et répartis à la surface d'une hypersphère de rayon constant. La distance entre deux vecteurs étant également un vecteur, elle devient elle aussi constante lorsque d augmente.

- Distance entre points : de nombreux algorithmes de fouille reposent sur la recherche des plus proches voisins dans l'espace d'apprentissage. Beyer et al. (1999) se sont intéressés au pouvoir discriminant des requêtes qui visent à récupérer les plus proches voisins d'un exemple dans les EGD. Ils ont ainsi démontré que, sous certaines conditions, notamment d'indépendance entre les dimensions et de répartition uniforme des données, la distance d'un exemple à son plus proche voisin tend à être égale à celle de ce même point à son voisin le plus éloigné lorsque le nombre de dimensions augmente. La notion de voisinage n'est donc plus celle, intuitive, que l'on trouve dans des espaces de petites dimensions.

4.1.2 Impact sur les algorithmes

La section précédente liste un ensemble de particularités des EGD en lien direct avec la fouille de données. Nous reportons dans cette section la manière dont ces particularités se traduisent sur certains algorithmes.

Apprentissage supervisé. L'impact des grandes dimensions a été relativement bien étudié pour certaines méthodes non-paramétriques qui expriment la frontière de décision entre les classes $f(x)$ sous la forme d'une combinaison linéaire de fonctions noyaux (telles que les *Supports Vecteurs Machines* (Vapnik, 1998) ou encore les réseaux RBF (Moody et Darken, 1989)) :

$$f(x) = b + \sum_{i}^{n} \alpha_i K_D(x, x_i)$$

avec K_D une fonction noyau et α_i des cœfficients déterminés lors de l'apprentissage. Dans la suite du paragraphe, les résultats que nous présentons sont associés à ce type de classifieur.

On estime que le nombre d'exemples requis pour obtenir une certaine précision en apprentissage supervisé est exponentiel en fonction de la dimension du problème (voir section 4.1.1). Härdle et al. (2004) explicitent ce problème pour les méthodes non-paramétriques à noyaux. En particulier, ils montrent que lorsque ces modèles sont utilisés comme estimateur de la densité, leur erreur quadratique minimum converge vers :

$$O(n^{\frac{-4}{4+d}})$$

Par exemple, pour $d = 10$ et $n = 100\ 000$, l'erreur est $1,5$ fois pire que pour $d = 1$ et $n = 100$ (Härdle et al., 2004).

Bengio et al. s'intéressent également à ce problème à partir de fonctions ayant un noyau Gaussien :

$$K_D(x, x_i) = \exp\left(-\frac{\|x - x_i\|}{\sigma^2}\right)$$

où σ représente un hyper-paramètre qui contrôle la localité du noyau. Dans un premier temps, ils imaginent un problème où la frontière de décision dans $\mathbb{R}^d$ croise une ligne droite $2k$ fois. Dans ce cas, ils montrent que les noyaux gaussiens nécessiteront k bases (k $\alpha_i \neq 0$) pour trouver cette frontière. L'idée de cette première expérience est de montrer que plus une fonction varie (*ie* plus elle croise la ligne droite), plus les noyaux gaussiens auront besoin de bases (*ie* d'exemples) pour l'approximer. Si ce résultat ne relie pas directement le nombre de bases au nombre de dimensions, les auteurs estiment néanmoins qu'une fonction sera d'autant plus variable que l'espace sera de grande dimension. Pour essayer de confirmer cette intuition, Bengio et al. ont cherché un exemple où la dimensionnalité de l'espace serait fonction du nombre de bases. À cette fin, ils ont utilisé le problème de la parité. Pour ce problème, les exemples sont des ensembles de bits de taille d, chacun associé à une étiquette qui vaut 1 si la somme des bits de l'exemple est paire, 0 sinon. Soit :

$$(b_1, \ldots, b_d) \in \{0,1\}^d \longmapsto \begin{cases} 1 & \text{si } \sum_{i=1}^{d} b_i \text{ est paire;} \\ 0 & \text{sinon.} \end{cases}$$

Ils ont montré que si $f(x) = b + \sum_i^{2^d} \alpha_i \exp\left(-\frac{||x-x_i||}{\sigma^2}\right)$ résolvait ce problème, alors au moins 2^{d-1} des α_i sont différents de 0, soit au moins 2^{d-1} bases.

Au final, ces auteurs ont donc (1) montré que le nombre d'exemples minimum pour estimer une fonction à l'aide de noyaux gaussiens augmentait linéairement avec les variations de cette fonction, (2) avancé l'idée que les variations sont d'autant plus grandes que la dimensionnalité est importante en s'appuyant sur le fait que (3) il existe des fonctions dont le nombre d'exemples requis pour être apprises par des noyaux gaussiens augmente exponentiellement en fonction de la dimension.

L'utilisation des noyaux gaussiens pose un autre problème lié cette fois à la répartition des exemples dans les EGD. En effet, Hérault et al. (2002), repris par Verleysen et al. (2003), rappellent que la densité de probabilité d'un point à une distance r du centre d'une distribution gaussienne de variance σ^2 est :

$$f(r) = \frac{r^{d-1} e^{-\frac{r^2}{2\sigma^2}}}{2^{\frac{d}{2}-1}\Gamma\left(\frac{d}{2}\right)}$$

La figure 1.9 trace cette fonction pour différentes valeurs de r et d. Comme on peut le voir, la probabilité n'est pas maximum au centre de la distribution ($r = 0$, comme la distribution elle-même), mais lorsque $\frac{r}{\sigma} = \sqrt{d-1}$. Or les modèles bâtis sur ces fonctions font justement

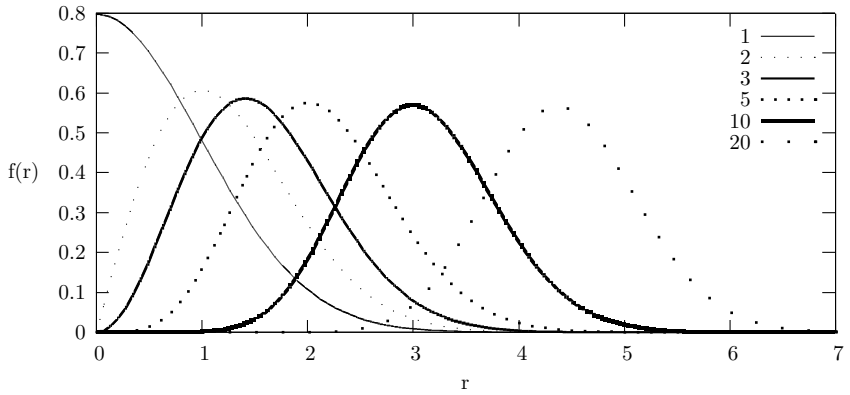


FIG. 1.9: Densités de probabilité de gaussiennes de moyenne $\mu = 0$ et d'écart-type $\sigma = 1$ pour les tailles d'espace : 1,2,3,5,10 et 20

l'hypothèse que la densité est maximum, au centre de la distribution gaussienne.

Le problème principal des EGD sur l'apprentissage supervisé concerne donc la précision de la prédiction (voir à ce sujet l'article de Friedman et Fayyad (1997) sur le compromis biais/variance pour des données en grandes dimensions). Néanmoins, leur topologie est également de nature à modifier l'apprentissage, au moins en ce qui concerne les méthodes non-paramétriques.

Classification non-supervisée. Il est possible de mettre en évidence l'existence de classes dans des données en traçant un graphique qui représente pour chaque distance entre deux exemples, la probabilité d'obtenir cette distance (Brin (1995) repris par Steinbach et al. (2004)). La figure 1.10.(a) représente la répartition des probabilités que l'on s'attend à trouver dans le cas où les données sont organisées en classe. La distribution des distances a deux pics. Le premier, pour des petites distances, est généré par la distance intra-classe et le second, pour des distances plus importantes, par la distance inter-classes. La figure 1.10.(b) représente la répartition des probabilités si les données ne contiennent aucune classe. Les distances ne semblent pas contrastées et un seul pic est présent autour de la distance inter-exemples moyenne.

Dans les EGD, le résultat de Demartines (1994) montre que les distances inter-points tendent à être constantes lorsque le nombre de dimensions augmente (et sous certaines hypothèses, voir section 4.1.1). On comprend alors que même si des classes sont présentes dans les données, la répartition des probabilités des distances se rapproche plus fortement de la figure 1.10.(b) que 1.10.(a) . Autrement dit, il sera plus difficile de retrouver ces classes sur une analyse des distances inter-exemples comme le font beaucoup d'algorithmes standard de classification.

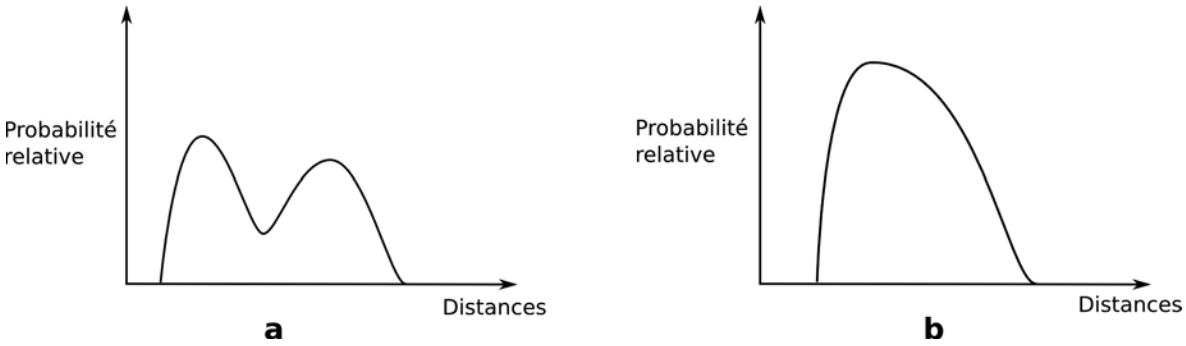


FIG. 1.10: Probabilités relatives attendues des distances inter-points dans les cas où (a) les données sont organisées en classes et (b) les données ne sont pas organisées en classes.

Pourtant, les données en grandes dimensions n'empêchent pas l'existence de classes. Cependant, leur organisation n'est plus forcément celle des espaces de plus petite taille. Parsons et al. (2004) prennent pour exemple la classification de documents décrits par un vecteur de mots. En recherchant dans une telle base l'ensemble des documents qui contiennent le mot *fils*, on s'attend à récupérer plusieurs catégories de documents, chacune en relation avec le contexte dans lequel est utilisé ce mot. Ce contexte est défini essentiellement par les mots co-occurrents avec le mot recherché. Dans l'exemple, il peut faire référence à la famille (avec des mots co-occurrents comme père, mère, ...), à l'électricité (rouge, vert, soudure, connecté, ...) ou encore au tricot (laine, aiguille, ...). Dans cet exemple – et souvent avec les données en grandes dimensions – les différentes classes d'exemples sont définies au sein de différents sous-espaces de l'espace d'apprentissage. Cette propriété est exploitée par toute une famille d'algorithmes de classification en sous-espaces qui en tire parti pour déterminer les classes (voir Parsons et al. (2004) pour une revue du domaine).

Recherche d'Information (RI), indexation. La recherche d'information dans des grandes bases de données est souvent fondée sur l'existence d'index qui permettent d'accélérer la recherche sans perdre en précision. Ces index reposent soit sur un découpage de l'espace en fonction de l'organisation en classes des données, soit sur un découpage hiérarchique de l'espace. Pour ces méthodes, et sous des conditions de distribution indépendante et uniforme des exemples dans l'espace, Weber et al. (1998) ont montré qu'un parcours séquentiel des exemples était plus rapide dès que la dimension de l'espace atteint 10. Pour contrer cet effet, ils proposent d'avoir recours à une méthode d'approximation (VA-File).

Représentation des données. La représentation des données est également sensible aux EGD. Parce qu'elle modélise les proximités entre exemples, la représentation est par-

ticulièrement concernée par le problème de concentration des distances. Cette observation a poussé certains auteurs à remplacer la distance euclidienne par d'autres mesures de distance moins sujettes à ce phénomène. Pour les cartes auto-organisatrices, citons par exemple l'utilisation de distances fractionnaires par Francois et al. (2005).

De même, l'organisation des données dans l'espace, notamment la notion de sous-espace, rend difficile la projection linéaire de ces données. Aussi, dans la plupart des cas, il est fortement conseillé d'utiliser des méthodes de projection non-linéaire pour réduire la dimensionnalité du problème (Verleysen, 2003).

Enfin, comme nous utilisons la représentation dans un contexte d'apprentissage supervisé, sa précision sera également sensible au fléau de la dimension. En effet, les méthodes de représentation que nous avons présentées sont des méthodes d'apprentissage non-paramétriques puisqu'elles ont recours aux exemples d'apprentissage lors de la prédiction. Ces méthodes font l'hypothèse que les propriétés en un point sont propagées à son voisinage. Cela est vrai lorsque l'espace est décrit avec précision par les exemples. Malheureusement, à cause du phénomène de l'espace vide présent dans les espaces de grandes dimensions, le voisinage d'un point peut se trouver éloigné du point lui-même. Dans ce cas, la propagation des propriétés au voisinage devient source d'erreur (Bengio et al.).

4.2 Volumétrie des exemples

Les problèmes liés à la volumétrie des exemples sont d'une nature différente. En effet, comme souligné dans la section 4.1.1, un grand nombre d'exemples aide à la description de l'espace d'apprentissage. Ils permettent de lutter contre le fléau de la dimension. *A priori*, la volumétrie des exemples n'est donc pas de nature à dégrader l'apprentissage. En revanche, cette volumétrie est à relier directement avec la complexité des algorithmes, que ce soit en termes de temps de calcul ou de stockage en mémoire des structures nécessaires au bon déroulement de l'algorithme. Ces observations sont particulièrement vraies lorsque l'apprentissage est fondé sur la représentation. En effet, nous avons défini la représentation comme une structuration des exemples, structure qu'il est d'autant plus difficile d'obtenir que le nombre d'exemples est important.

Le cas des graphes de voisinage est particulièrement symptomatique de ces problèmes. En termes de complexité algorithmique, la construction du graphe est en $O(n^3)$, c'est-à-dire qu'elle requiert un temps exponentiel en fonction du nombre d'exemples suivant l'algorithme standard proposé en section 2.2.2. Pour donner une idée, un processeur réalisant 27079 MIPS[4] (Million Instructions Per Seconds) mettrait 427 jours pour réaliser les 1 000 000[3] opérations nécessaires à la construction d'un graphe de 1 000 000 d'exemples (en considérant – ce qui est optimiste – que le processeur résout une itération de l'algo-

[4] ce chiffre correspond à la puissance du processeur Intel® X6800, bicœur cadencé à 2.93 GHz, produit en 2007, qui équipe de nombreux ordinateurs personnels actuels

rithme en une instruction). Quant à la structure du graphe, elle est souvent modélisée en mémoire par une matrice carrée de taille $n \times n$. En utilisant le type `float`[5] en `C++`, la place occupée par cette matrice pour représenter un graphe de 1 000 000 d'individus serait de $1\,000\,000 \times 1\,000\,000 \times 4$ bits $= 1\,000\,000 \times 500\,000$ octets $= 500\,000\,000\,000$ octets $= 465$ Go soit 100 fois la taille de la mémoire vive des ordinateurs personnels actuels.

Cet exemple illustre bien pourquoi la volumétrie des exemples devient critique dans le choix des stratégies de fouille. Cependant, il est toujours possible d'assurer le passage à l'échelle d'algorithmes dont le champ d'application est restreint par leur complexité, soit en proposant des stratégies d'échantillonnage de la base d'exemples (Chauchat et Rakotomalala, 2000), soit en se servant d'heuristiques pour simplifier la complexité initiale de l'algorithme (Hacid, 2008).

4.3 Validation des découvertes

Les deux précédentes sections (4.1 et 4.2) ont mis en évidence le danger des données volumineuses lors de la mise en œuvre d'algorithmes de fouille en soulignant leur impact au cours de l'apprentissage. Cette section s'intéresse quant à elle aux problèmes liés au nombre de connaissances extraites.

En fouille de données, l'extraction de la connaissance n'est satisfaisante que si elle est accompagnée d'une mesure de sa qualité. Cette mesure vient valider le résultat des algorithmes. Sans elle, il est impossible de dire quelle confiance accorder à la connaissance, ce qui rend son exploitation pour le moins incertaine. Les algorithmes de fouille sont donc souvent accompagnés de mesures statistiques de la validité de leur résultat ou, à défaut, de procédures de validation. Dans certains cas, les algorithmes délivrent un ensemble de connaissances. Citons, pour exemple, les algorithmes de détection d'exemples atypiques (Beckman et Cooks, 1983), de sélection des attributs pertinents (Kira et Rendell, 1992) ou encore d'extraction de règles d'association (Agrawal et al., 1993). L'étape de validation associe alors une mesure de pertinence à chaque connaissance (respectivement un exemple atypique, un attribut pertinent, une règle d'association). Si l'on prend le cas de la sélection d'attributs, une stratégie classique consiste à effectuer une ANOVA (pour *ANalyse Of VAriance*) à un facteur sur chaque attribut pour tester l'indépendance entre leurs valeurs et les différentes modalités de la classe. À l'issue de ces tests, cette procédure détermine pour chaque attribut sa *p-value*. L'utilisateur peut alors choisir les attributs à conserver en connaissant le risque pris pour chacun (ou risque de première espèce, ici celui de sélectionner un attribut indépendant de l'étiquette).

Il est évident que le nombre de connaissances extraites est d'autant plus important que les données sont volumineuses. Cela conduit à une augmentation automatique du nombre de

[5] ce type de donnée est codé sur 4 bits et peut représenter des nombres allant de -3.4×10^{-38} à 3.4×10^{38}

tests, ce qui est de nature à fausser le risque de première espèce. La conséquence est alors une augmentation des fausses découvertes. Pour éviter cet écueil, le contrôle du risque multiple aide à rétablir une situation normale pour se concentrer sur la découverte des connaissances les plus intéressantes, notamment en augmentant raisonnablement le niveau d'exigence de chaque test.

Dans le chapitre 4, nous présentons plusieurs procédures de contrôle du risque multiple ainsi qu'un exemple d'application en extraction des connaissances sur des données génomiques en très grandes dimensions.

4.4 Contributions aux problèmes énoncés

Ces problèmes ont été au cœur des préoccupations de cette thèse où notre objectif a été de proposer des méthodes robustes face à des données volumineuses.

Dans la lignée de travaux réalisés au laboratoire ERIC, une première partie de notre travail a consisté à proposer une méthode d'extraction de connaissances fondée sur la représentation. Comme nous l'avons vu, l'obtention d'une représentation des données est coûteuse. Elle demande donc à être choisie avec précaution sur des grands volumes de données. Dans le deuxième chapitre, nous étudions la capacité des cartes auto-organisatrices à répondre à cette attente. La section 2.4 de ce chapitre a montré qu'elles pouvaient servir de représentation en offrant à la fois une structure aux exemples et leur visualisation. Nous présentons donc les algorithmes associés à ces aspects. Nous poussons ensuite plus avant son emploi face à des données volumineuses, d'abord en vérifiant que les cartes auto-organisatrices assurent le passage à l'échelle lorsque le nombre d'exemples augmente (en terme de temps de calcul, bien sûr, mais aussi de précision), puis nous proposons une mesure pour valider statistiquement l'apprentissage en adaptant le modèle proposé pour les graphes de voisinage (explicité en section 2.2).

Les résultats obtenus nous permettent de conclure que les cartes auto-organisatrices peuvent manipuler un grand volume d'exemples tout en restant stables dans leur prédiction. Néanmoins, parce qu'elles reposent sur l'utilisation des distances, elles restent sensibles aux données en grandes dimensions. Le chapitre 3 propose une approche pour assurer la qualité de leur représentation dans des espaces de grandes dimensions. À cette fin, nous avons exploité les résultats obtenus sur les ensembles de classifieurs pour les appliquer aux cartes auto-organisatrices. Nous détaillons d'abord deux méthodes permettant d'obtenir un ensemble de cartes (soit de manière totalement non-supervisée, soit en faisant intervenir l'étiquette), méthodes qui reposent toutes deux sur la notion de diversité propre aux ensembles. Si les résultats obtenus en prédiction avec ces méthodes sont bons, leur inconvénient majeur est de faire coexister plusieurs représentations. Nous proposons alors une stratégie pour obtenir une représentation unique, issue de l'ensemble des représentations, et qui peut être fournie comme interface d'exploration à l'utilisateur.

Le problème de la multitude de tests n'est pas rencontré dans la stratégie fondée sur les ensembles de cartes où l'objectif a été d'utiliser l'ensemble des informations à disposition pour proposer un modèle de prédiction à l'utilisateur. Dans d'autres cas, l'objectif est plutôt de trouver une information pertinente dans un grand volume de données. Prenons pour exemple les tableaux d'expression de gènes. Ces tableaux recensent le niveau d'expression de milliers de gènes (les attributs de nos données) dans quelques centaines de tissus (les exemples) généralement répartis en deux classes. Pour le biologiste, qui a mesuré de manière presque exhaustive les niveaux d'expression de tous les gènes qu'il connait, l'objectif est de trouver des gènes, si ce n'est des réseaux de gènes, impliqués dans la différenciation des tissus suivant les deux classes. Ici, on subodore le rôle des tests statistiques qui vont devoir valider l'implication de tel ou tel gène, de tel ou tel réseau dans une classe de tissus plutôt que dans l'autre. Le chapitre 4 se penche sur ce problème en proposant une stratégie en deux étapes, avec à chacune un contrôle statistique des fausses découvertes. La première étape sélectionne statistiquement les gènes intéressants par rapport à la variable de classe (*ie* les classes de tissus). Ensuite, à partir des gènes sélectionnés, une deuxième étape extrait des règles d'association différentiellement exprimées suivant le sous-ensemble d'exemples sur lequel elles sont mesurées (avec un sous-ensemble par étiquette de classe). Nous montrons comment cette stratégie, à l'aide d'un contrôle du risque multiple adéquat, permet de cibler les connaissances extraites plutôt que de se trouver face à une trop grande quantité de découvertes, par là même difficilement exploitable.