Modèles de mélange pour la caractérisation topologique de données multidimensionnelles

Maxime Maillot

UT Compiègne

Laboratoire Analyse des Données
et Intelligence des Systèmes
CEA LIST

Discipline : Informatique

Thèse de doctorat

Soutenue le 31 mars 2015

Maxime MAILLOT

Modèles de mélange pour la caractérisation topologique de données multidimensionnelles

Directeur de thèse :	M. Gérard GOVAERT	Professeur HDR (UT Compiègne)
Co-directeur de thèse :	M. Michaël AUPETIT	Ingénieur-Chercheur HDR (CEA LIST)

Composition du jury :

Président du jury :	Yves GRANDVALET	Directeur de Recherche CNRS (UT Compiègne)
Rapporteurs :	Mohamed NADIF	Professeur (Université Paris Descartes)
	Christophe BIERNACKI	Professeur (Université Lille 1
Examinateur :	Djamel Abdelkader ZIGHED	Professeur (Université Lyon 2)

Table des matières

Table des figures

1

Introduction

L'analyse exploratoire de données est un ensemble de méthodes qui ont été développées pour traiter des données afin de rendre leur interprétation plus aisée pour l'être humain. Un premier exemple consiste à résumer des données en grandes dimensions, donc non visualisables, en dimension 2 ou 3. L'utilisateur est alors capable de visualiser les données et de les interpréter. Une des difficultés réside dans le fait qu'il faut perdre le moins d'information possible lors de cette réduction de dimension. Une autre qu'il faut retrouver une information qui n'est pas directement mentionnée dans les données mais qui peut y figurer de façon implicite. Un problème illustre ce cas de figure : il faut retrouver l'existence de deux populations différentes dans les données. Imaginons un jeu de données qui contiendrait comme seules informations la taille et le poids d'un ensemble de chiens (constitué de cockers et d'épagneuls) comme on peut le voir sur la figure 1.1. À aucun moment l'information explicite de la classe à laquelle appartient un chien n'est donnée. Pourtant, en traçant ces données en deux dimensions comme des points dans $\mathbb{R}^2$, il apparait deux groupes distinctement séparés, au sens où l'on pourrait tracer une droite pour les séparer, ce qui montre qu'il y a bien deux groupes de points, que l'on peut supposer correspondre à deux races de chiens différentes dans cet échantillon.

Bien entendu on peut faire ce genre de déduction parce qu'on a utilisé deux variables pertinentes, *i.e.* explicatives de la classe, et uniquement ces deux variables pour l'analyse. Mais ces variables peuvent très bien être les seules pertinentes parmi toutes les variables proposées par le jeu de données : la couleur des yeux ou du pelage, le nom du chien, voire le nom de son maître, sont toutes des variables qui peuvent se retrouver dans le jeu de données, mais qui n'apportent aucune information discriminante en ce qui concerne la race des chiens de l'échantillon.

Cet exemple représente l'idée de l'analyse exploratoire de données : trouver de l'information dans un jeu de données qui peut compter un grand nombre de variables, mais dont

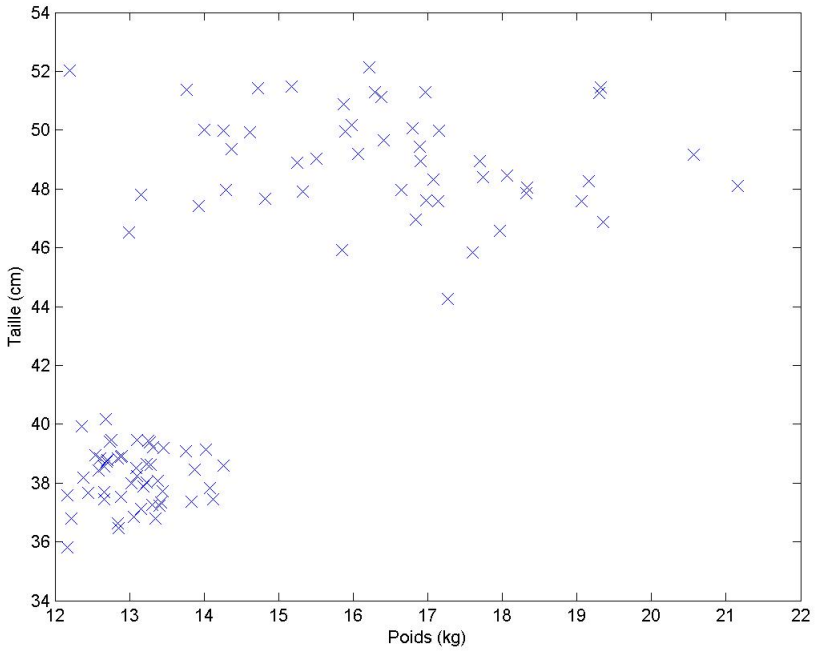


FIGURE 1.1 – Jeu de données simulées comportant deux populations, l'une composée d'épagneuls, l'autre de cockers, représentées par leurs poids et tailles. Un humain distingue à l'oeil nu ces deux populations dans la représentation graphique.

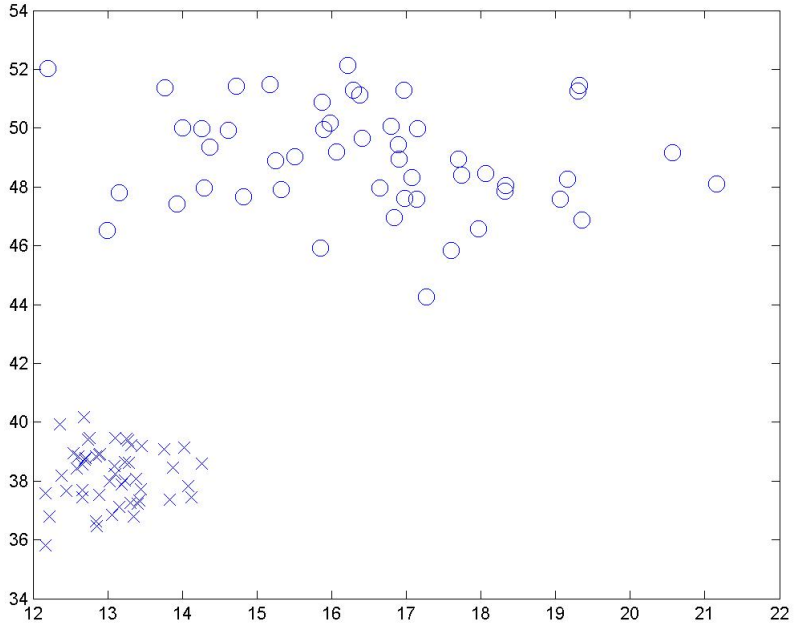


FIGURE 1.2 – Il s'agit du même jeu de données que précédemment, mais cette fois-ci une classe a été attribuée à chaque chien. C'est ce genre de classification qu'un humain sait faire à l'oeil nu sur des cas simples que l'on cherche à faire de manière automatique.

seulement certaines sont explicatives mais non identifiées. Si le nombre de ces variables pertinentes se limite à deux ou trois, il sera aisé de visualiser les données pour réaliser la classification. A partir de 4 ou plus, il faudra attaquer le problème sous un autre angle.

Les spécialistes de la visualisation de données tentent de projeter les données en dimension 2, en essayant d'être le plus fidèle possible aux données originales. Par exemple en cherchant la projection en 2D qui préservera au mieux les similarités : deux données proches dans $\mathbb{R}^D$ sont représentées proches dans $\mathbb{R}^2$ et réciproquement pour les données éloignées. On peut approcher cette question par l'extraction automatique des caractéristiques des données dans leur espace d'origine, quitte à ce que cette information soit moins détaillée que lors d'une visualisation, comme une estimation de la moyenne, de la variance des données ou le nombre de groupes.

C'est cette deuxième approche que nous étudierons dans cette thèse. Nous nous intéresserons en particulier à l'extraction automatique de caractéristiques de nature topologique d'un échantillon de la population étudiée.

1.1 Les origines du problème

Quand un phénomène peut être reproduit en conditions expérimentales, permettant de contrôler les variables dont dépend le phénomène, on peut alors proposer un modèle physique. L'équation des gaz parfaits, $PV = nRT$ ne contient que 4 variables et 1 constante universelle. Cette relation a donc pu être mise en évidence par des scientifiques grâce à des expériences qui sont aujourd'hui reproduites au lycée. Redécouvrir cette équation est aujourd'hui chose aisée, puisqu'on sait quel résultat atteindre. Les chercheurs eux, ont dû conduire plusieurs expériences pour éliminer certaines variables qui n'avaient aucune influence sur le modèle. Ensuite, il a fallu aussi démontrer que les relations entre les variables pertinentes étaient linéaires, alors qu'elles auraient pu être quadratiques, logarithmiques ou d'une toute autre nature encore.

Exclure les variables non pertinentes, trouver et expliciter les relations entre les variables d'intérêt, c'est ce à quoi sont confrontés les chercheurs. Certaines disciplines ont un certain contrôle sur l'environnement de leurs expériences. Cependant, dans le cas d'un phénomène réel impossible à reproduire expérimentalement, décrit par un grand nombre de variables dont on ne sait pas lesquelles sont pertinentes, l'expert peut utiliser des méthodes d'analyse exploratoire. Les plus simples et les plus anciennes vont chercher des relations de linéarité entre les différentes grandeurs, puisque c'est la relation la plus simple que l'on puisse imaginer entre deux variables. Il faut de plus intégrer des modèles de bruits puisque l'environnement est moins contrôlé et éventuellement prendre en compte des relations quadratiques ou d'ordre supérieur si le modèle linéaire ne suffit pas à expliquer la relation entre les variables. Si le nombre de variables est raisonnable, il est encore possible d'obtenir de bons résultats avec ces méthodes. On quitte toutefois le monde de la physique, où l'expert "comprend" le phénomène en question justement parce qu'il est spécialiste du domaine, pour entrer dans le monde du statisticien qui met en évidence des corrélations et autres grandeurs statistiques.

Un milliard de smartphones ont été vendus en 2013 (Rousseau, 2014). Ces appareils possèdent tous une puce GPS, un appareil photo numérique et des accéléromètres. L'historique de chaque utilisateur d'Internet est sauvegardé à chacun de ses passages sur un site web. Le stockage de données étant de moins en moins cher, toute cette information, les *logs*, est sauvegardée. La quantité d'information destinée à être analysée est amenée à croître de plus en plus vite. Retrouver une information donnée et précise est déjà un véritable challenge pour le scientifique dans un tel contexte.

Dans le cas des gaz parfaits, le nombre de paramètres est restreint : un gaz a peu de caractéristiques. Certains phénomènes sont beaucoup plus complexes, encore plus s'ils correspondent au comportement humain. Du fait de cette multiplication exponentielle des capteurs, il est fait dans l'ensemble de cette thèse l'hypothèse que l'espace d'observation E est plus grand que l'espace $\tilde{E}$ qui serait nécessaire pour décrire parfaitement le phénomène. Toujours dans le cadre ces travaux, E sera supposé euclidien. Les relations linéaires, simples et classiques ne suffisent plus, la représentation d'un phénomène n'est que rarement une ligne droite, mais une généralisation du concept d'espace linéaire qui permet d'appréhender des formes plus générales. Dans le cadre statistique, l'observation du système fournit un échantillon. C'est tout le but de cette thèse de faire ressortir les différentes caractéristiques de l'espace correspondant à un phénomène observé au travers de cet échantillon.

1.2 Le statisticien au service de l'expert

La plupart des analystes qui traitent des données sont des spécialistes de leur métier (géophysicien, biologiste, chimiste, etc.). Les données ne sont qu'un moyen pour eux de comprendre un phénomène essentiel de leur discipline, tel l'économiste qui va regarder l'évolution des PIB au cours du temps et vouloir le mettre en regard avec le taux de mortalité infantile d'un pays, le géophysicien qui va analyser des localisations de séismes ou le spécialiste e-commerce qui compare le prix du panier moyen avec l'heure d'achat ou l'âge de l'internaute. Ces personnes sont formées à leur domaine de compétence. Ils peuvent avoir des compétences en statistique mais sont avant tout experts dans leur domaine de prédilection.

A l'inverse, le statisticien sait comment traiter toutes ces données, mais ne sait pas forcément en tirer les conclusions : il peut mettre en évidence que le PIB et le taux de mortalité sont corrélés comme on le voit sur la figure 1.3, mais il ne sait pas l'expliquer. Quelles autres variables intermédiaires pourraient l'expliquer ? L'économiste saura interpréter cela comme un meilleur accès aux hôpitaux par exemple. Dans le domaine de la géophysique, le statisticien peut mettre en évidence que les séismes proviennent d'une région précise, mais seul le géophysicien peut dire s'il s'agit d'un point chaud ou de la rencontre de deux plaques tectoniques. Ce savoir ne fait pas forcément partie des connaissances du statisticien. Il lui faut donc fournir des outils au non-expert pour que ce dernier puisse tirer lui-même les conclusions pertinentes.

Ces outils se doivent de fournir une information intelligible à l'utilisateur non-expert. Gardons l'exemple du géophysicien : une méthode statistique lui indique un groupe de

séismes, modélisé par une distribution gaussienne avec une moyenne et une variance. Cela n'est pas forcément pertinent pour lui. En revanche, indiquer que ce groupe est une ligne ou une surface, voire le placer sur une carte, va tout de suite pouvoir l'aider à tirer une conclusion dans son domaine. La présentation et la mise en forme de l'information est essentielle pour que le non-expert puisse comprendre l'outil qu'il utilise. Nous pensons que la géométrie et dans une plus large mesure la topologie peuvent être beaucoup plus explicite pour le non-expert.

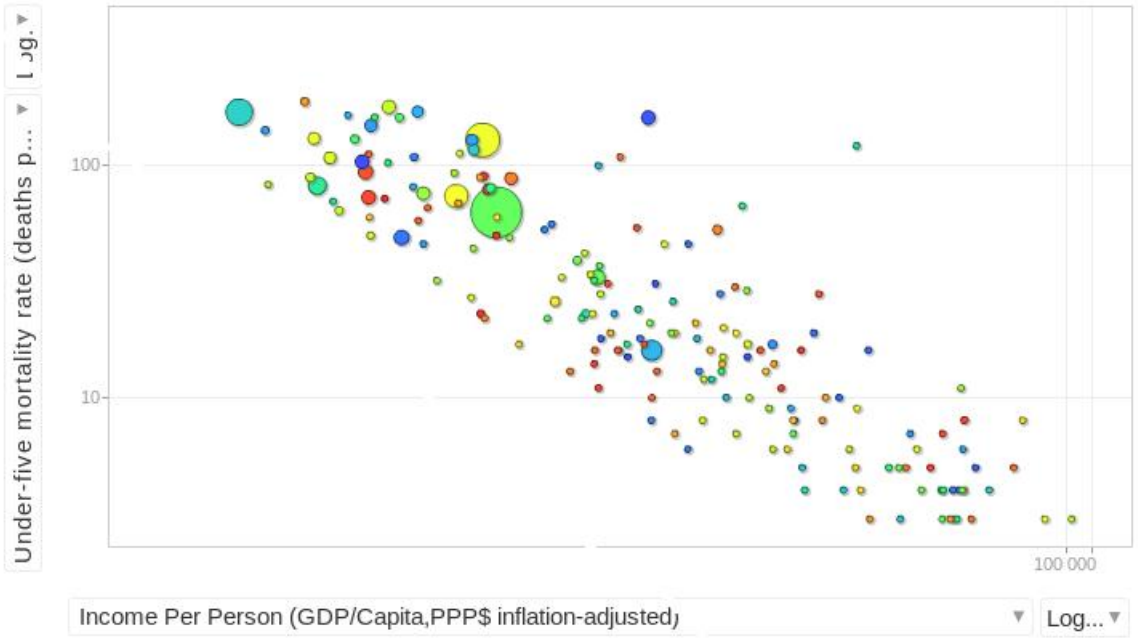


FIGURE 1.3 – Ce graphe met en évidence la corrélation entre PIB et mortalité infantile. Il faut néanmoins être économiste pour analyser les raisons ce phénomène. *Source : http ://www.apromiserenewed.org/Dashboard.html*

Dans le cadre de données en deux dimensions, avec une simple corrélation linéaire, on peut déjà tirer des conclusions avec un graphe en deux dimensions. Mais dès qu'on atteint des dimensions trop grandes pour être visualisées, ou que l'opération, même simple, doit être répétée un grand nombre de fois en peu de temps, *l'automatisation de la tâche devient nécessaire.*

1.3 Les outils statistiques

Outre les méthodes statistiques classiques, l'analyste de données dispose maintenant d'outils issus de l'apprentissage statistique, un domaine regroupant les méthodes qui permettent de découvrir l'information cachée dans un jeu de données. Ce domaine est très populaire puisque les données observées sont souvent bruitées, que cela soit dû au choix des variables d'observation ou au bruit inhérent au processus d'observation du phénomène.

Or l'apprentissage statistique permet de prendre en compte un modèle de bruit.

Comme en traitement du signal où l'on définit un rapport signal sur bruit, pour lequel le signal sera d'autant plus facile à interpréter que ce rapport est grand, il faut supposer que la position des données dans l'espace d'observation dépend plus du phénomène observé que du bruit. Les données sont générées par le phénomène. Ce qui est observé au final peut-être vu comme la composition de deux processus : premièrement une donnée est générée dans l'espace, sa position étant déterminée par les caractéristiques du phénomène, puis la donnée est bruitée, d'une part parce que le phénomène n'est peut-être pas parfaitement déterministe, et parce que le système de mesure des données est imprécis et les variables observées ne suffisent pas à exprimer tous les degrés de liberté du phénomène. Pour continuer sur l'exemple des chiens, le fait d'observer un épagneul ou un cocker explique la taille et le poids que l'on va retrouver, la position de l'espace dans laquelle se trouve la donnée d'un épagneul est donc définie par le fait qu'il soit un épagneul. On n'observe jamais de cocker qui font la taille et le poids d'un épagneul, et inversement. En revanche, tous les épagneuls ne font pas le même poids. Retrouver la classe à laquelle appartient un chien à partir des données observées est un problème de classification qui peut être résolu en estimant la densité de probabilité de chaque phénomène. Il faut ensuite décider de la classe d'un individu à partir de la densité estimée.

Ce problème de classification nous amène aussi à voir la notion de la dépendance. Quelle est la probabilité qu'un chien soit grand sachant que c'est un épagneul ? Quelle est la probabilité qu'un chien soit un épagneul sachant qu'il est grand ?

Si on note A l'évènement "être un épagneul" et B "être grand", alors les deux probabilités précédentes sont liées par la formule suivante issue du théorème de Bayes (Carlin and Louis, 1997) :

$$\mathrm{P(A|B)} = \frac{\mathrm{P(B|A)P(A)}}{\mathrm{P(B)}} \tag{1.1}$$

On appelle P(A) la probabilité *a priori* de A puisqu'elle ne dépend pas de B, alors que P(A|B) sera la probabilité *a posteriori* de A sachant qu'on a observé B.

Si l'on observe qu'un individu vérifie la condition B : il est grand. On exprimera alors la probabilité que cet individu soit un épagneul, sachant qu'il est grand, en fonction de la probabilité d'être grand, sachant que l'individu est un épagneul, ainsi que des probabilités d'être grand, et d'être un épagneul.

1.3.1 Classification

Comme on l'a vu avec l'exemple précédent, établir l'existence de groupes au sein d'un échantillon peut s'avérer très utile. L'exemple parle de lui-même pour la biologie : connaître l'existence ou non de différentes races ou différentes espèces au sein d'une population animale permet par la suite de les classer. Il y a d'autres domaines d'application, l'un des plus en vogue, parce que lucratif, est celui du marketing, notamment du webmarketing

(Mobasher et al., 2000) : quand on se connecte à une page web, le site visité va analyser rapidement notre historique, et déterminer à quelle classe d'âge ou quelle catégorie socio-professionnelle on appartient, quels sont nos désirs actuels, grâce à notre historique. D'où l'affichage plus ou moins pertinent de publicités ciblées sur le site web.

La classification est aussi appelée *apprentissage non supervisé*. Contrairement à la discrimination (*apprentissage supervisé*), où l'on cherche à ranger dans des classes que l'on connaît déjà de nouvelles données, en classification on se penche justement sur le problème de la découverte de ces classes, alors qu'on ne dispose d'aucune connaissance sur l'étiquette des données, ni du nombre de classes dans l'échantillon.

On dispose classiquement d'un jeu de N données $\mathbf{x} = (x_1, ..., x_N)$, représentées comme des points dans $\mathbb{R}^D$. Ces points sont généralement supposés être indépendants et identiquement distribués (*iid*) suivant une densité de probabilité p, définie sur $\mathbb{R}^D$.

On cherche à établir des groupes d'individus au sein du jeu de données : il faut donc identifier des classes, et attribuer chacune des données à une classe, les classes devant être le plus homogènes possible. La configuration idéale étant que ces classes soit facilement séparables : entre deux zones de grandes densité de données, chacune de ces zones correspondant à une classe, il existe une zone de densité très faible voire nulle. L'espace peut donc être facilement découpé entre la zone de la première classe, la zone qui ne correspond à aucune des deux classes, et la zone de la seconde classe.

Hartigan formalise cette solution en seuillant la densité de probabilité (Hartigan, 1975), rendant ainsi nulle la densité dans les endroits où elle est inférieure au seuil t. Il ne reste qu'à récupérer les composantes connexes pour identifier chacune des classes. C'est là tout l'objet de nombreuses méthodes de classification d'estimer correctement la densité, et de trouver la bonne valeur de t, Chazal et al. (2011b) utilisent une approche topologique pour déterminer le bon seuil.

1.3.2 L'estimation de la densité

On peut regrouper les méthodes d'estimation de la densité en deux grandes familles.

Les estimateurs paramétriques présupposent un type de fonction de densité. Le plus courant est le modèle de mélange fini (McLachlan and Peel, 2000). Dans ce dernier, on voit les données comme issues de plusieurs groupes différents, suivant chacun une loi de probabilité paramétrique de même nature, mais de paramètres différents pour chacun des groupes. On modélise alors la densité de probabilité p comme une somme pondérée finie de distributions de probabilité r_k, dépendant chacune d'un ensemble de paramètres θ_k :

$$f(\mathbf{x}, \theta) = \sum_{k=1}^{K} \pi_k r_k(x|\theta_k) \tag{1.2}$$

$$\sum_{k=1}^{K} \pi_k = 1 \tag{1.3}$$

On optimise alors les poids π_k et les paramètres θ_k, qui forment le vecteur de paramètres Θ de façon à maximiser un critère statistique, généralement la vraisemblance $L(\mathbf{x}, \Theta) = \prod_{n=1}^{N} f(x_n|\Theta)$. Cette méthode, le maximum de vraisemblance, a été développée par Fisher (Fisher, 1922). Quand le modèle est simple, il peut exister une solution analytique. La méthode d'estimation des paramètres que nous avons choisi d'utiliser est l'algorithme Espérance-Maximisation (*Expectation-Maximization*) (Dempster et al., 1977) qui permet de traiter des modèles avec de nombreux paramètres et ne nécessite par de réglage arbitraire de méta-paramètres. Ces estimateurs étant à la base de l'approche exposée dans cette thèse, ils seront développés plus avant dans les parties suivantes.

Les estimateurs non-paramétriques Parallèlement aux estimateurs paramétriques, se sont développés les modèles non-paramétriques. Une des premières approches développées est l'estimateur à noyaux rectangulaires de Rosenblatt $\hat{f}$ (Rosenblatt, 1956) défini par :

$$\hat{f}(x) = \frac{1}{2nh} \sum_{i=1}^{n} \mathbf{1}(-h < x - x_i \leq h) \tag{1.4}$$

Pour résumer, on construit des rectangles de largeur $2h$ et de hauteur $\frac{1}{2nh}$ autour de chacune des données x_i de l'échantillon. On a donc au final une fonction en escalier qui prend de plus fortes valeurs là où le plus de données ont été échantillonnées : on comprend l'approche qui consiste à estimer une densité de probabilité de cette façon. On montre d'ailleurs facilement que la probabilité totale fait bien 1 :

$$\int_{\mathbb{R}} \hat{f}(x)\mathrm{d}x = \int_{\mathbb{R}} \frac{1}{2nh} \sum_{i=1}^{n} \mathbf{1}(-h < x - x_i \leq h)\mathrm{d}x \tag{1.5}$$

$$= \frac{1}{2nh} \sum_{i=1}^{n} \int_{\mathbb{R}} \mathbf{1}(-h < x - x_i \leq h)\mathrm{d}x \tag{1.6}$$

$$= \frac{1}{2nh} \sum_{i=1}^{n} 2h \tag{1.7}$$

$$= \frac{1}{2nh} \times 2nh = 1. \tag{1.8}$$

Cette approche a été généralisée par la suite par Parzen (1962) :

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{1.9}$$

où h est la fenêtre et K est une fonction noyau qui doit vérifier certaines conditions : elle est positive, symétrique, radiale (elle ne dépend que de la distance de x à x_i) et être d'intégrale égale à 1. Une fois choisie la forme de K (rectangle, gaussienne etc.), l'unique paramètre à régler est h : trop petit et l'on obtient des distributions de Dirac sur chacune des données, trop grand et la fonction est trop lisse, ne permettant pas de discriminer des groupes de données.

$$\forall x, K(x) \geq 0 \tag{1.10}$$

$$\int_{\mathbb{R}^n} K(x)\mathrm{d}x = 1 \tag{1.11}$$

$$\forall x, K(x) = K(\|x\|) \tag{1.12}$$

1.4 Définition du problème : l'apprentissage topologique

On dispose maintenant d'approches permettant d'estimer la densité de probabilité correspondant à un échantillon donné $\mathbf{x}$. L'estimation de la densité utilise toute l'information disponible dans l'échantillon. Bien que nécessaire, l'étape d'estimation n'est pas forcément la finalité de l'analyse. Dans le cas où l'on voudrait aller plus loin, on peut appliquer un post-traitement à l'estimation, et tenter de recueillir une information d'une autre nature. Comme on peut le voir sur la figure 1.4, il peut exister deux distributions très proches d'un point de vue statistique, mais très éloignées d'un point de vue géométrique. A l'inverse, une simple rotation de l'ensemble des données au cours de la chaîne de pré-traitement des données peut en modifier la distribution sur chaque variable, alors que l'information sous-jacente propre aux données (celle qui est invariante par isométrie ou similitude) est la même. En faible dimension, le problème peut être traité à l'œil nu, et un utilisateur verra tout de suite quels jeux de données forment un groupe et lesquels sont à différencier. Bien entendu, pour les dimensions supérieures et pour les tâches répétitives, une méthode automatique serait préférable.

Puisqu'un critère statistique comme la vraisemblance par exemple n'est plus le critère discriminant, il faut utiliser un critère qui discriminera non pas les densités mais les formes. On supposera alors que les données se distribuent au voisinage de ces formes. Il faut donc se tourner vers la topologie. La topologie est la branche des mathématiques qui étudie les espaces et les déformations continues qu'on peut leur faire subir. La topologie algébrique établit une classification de ces espaces en leur attribuant des invariants.

Le premier invariant topologique le plus facile à appréhender est la notion de *dimension intrinsèque* (Fukunaga and Olsen, 1971; Camastra and Vinciarelli, 2002; Levina and Bickel, 2004), il permet de plus d'introduire rapidement d'autres concepts primordiaux de la topologie. Peu importe l'espace dans lequel elle est représentée, une ligne a une dimension intrinsèque de 1, on dit aussi que c'est une *variété* de dimension 1. Une surface a une dimension intrinsèque de 2 et un volume une dimension intrinsèque de 3.

On appelle homéomorphisme une application bijective continue dont la bijection ré-

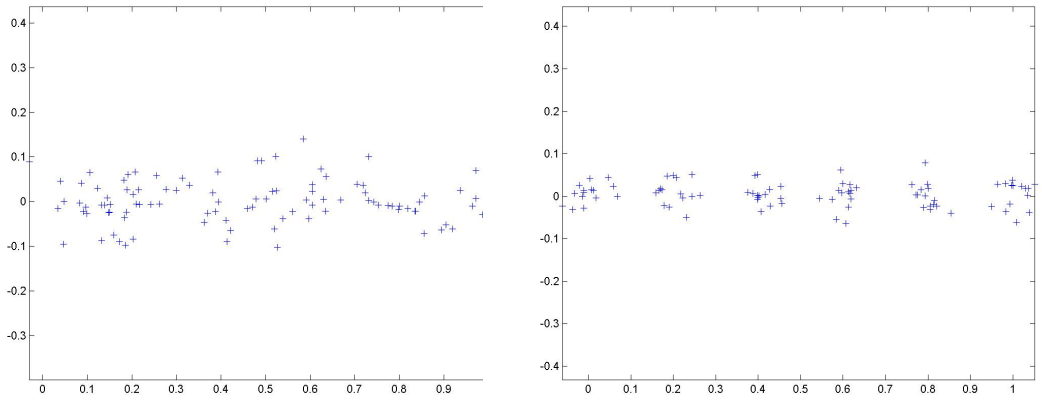


FIGURE 1.4 – Deux jeux de données très proches, mais le premier est issu d'un modèle avec une seule composante connexe, le second de 6 sommets différents.

ciproque est aussi continue. On dit que deux objets sont homéomorphes s'il existe un homéomorphisme qui transforme l'un en l'autre.

On dit que V est une variété topologique, si pour tout point x de V, il existe un voisinage ouvert U de x, et U' un ouvert de $\mathbb{R}^n$ tel que U et U' soient homéomorphes.

Montrer qu'une variété est homéomorphe à une autre est un problème très difficile à résoudre. C'est pour ça que les topologistes ont recherché des invariants topologiques plus simple à exprimer. On en est donc venu à comparer non pas les topologies, mais les homologies. La topologie algorithmique, qui est la branche "appliquée" de la topologie fournit des outils utilisables en mode *boîte noire* pour les informaticiens et les statisticiens (Zomorodian and Carlsson, 2005). Pour résumer, disons simplement que deux variétés qui ont des homologies différentes ne sont pas homéomorphes. Mais ce n'est pas parce que deux variétés ont la même homologie qu'elles sont homéomorphes. Nous utiliserons des invariants fondés sur les groupes d'homologie dans la suite.

1.4.1 Simplexes et complexes simpliciaux

Un complexe simplicial abstrait est une paire (V, Σ), où V est un ensemble fini et Σ est une famille non vide de sous-ensembles de V, tel que $\sigma \in \Sigma$ et $\tau \subseteq \sigma \rightarrow \tau \in \Sigma$, et σ est un simplexe. On peut associer à un complexe simplicial abstrait un espace topologique $|(V, \Sigma)|$. On définit d'abord :

$$\varphi : \left| \begin{array}{ccc} V & \longrightarrow & \{1, ..., N\} \\ s & \longmapsto & \varphi(s) \end{array} \right. \tag{1.13}$$

où N est le cardinal de V. On note $\{w_i\}_{i \in \{1,...,N\}}$ un ensemble de sommets de $\mathbb{R}^D$. Puis

on définit $|(V,\Sigma)|$ comme $\bigcup_{\sigma\in\Sigma} c(\sigma)$ où $c(\sigma)$ est l'enveloppe convexe de $\{w_{\varphi(s);s\in\sigma}\}$.

On appelle k-simplexe un simplexe de cardinal $k+1$. Dans l'espace topologique, un 0-simplexe est donc l'enveloppe convexe d'un point : c'est ce point lui-même. Un 1-simplexe est l'enveloppe convexe de deux points soit un segment, un 2-simplexe un triangle, un 3-simplexe un tétraèdre et ainsi de suite pour les dimensions supérieures (Carlsson, 2008).

1.4.2 L'homologie simpliciale et les nombres de Betti

Contrairement aux variétés continues, les complexes simpliciaux peuvent être représentés et manipulés par un ordinateur, ainsi certaines de leurs caractéristiques, comme l'homologie, peuvent être extraites algorithmiquement.

Soit un complexe simplicial $X = (V,\Sigma)$. On note Σ_k le sous-ensemble des k-simplexes de Σ. On définit $C_k(X)$, le groupe des k-chaines de X comme le groupe des sommes formelles linéaires sur Σ_k, ou de manière équivalente comme le groupe abélien libre sur Σ_k. En imposant un ordre total sur l'ensemble des sommets V, on peut définir l'opérateur :

$$d_i : \left| \begin{array}{ccc} \Sigma_k & \longrightarrow & \Sigma_{k-1} \\ \sigma & \longmapsto & \sigma - s_i \end{array} \right. \tag{1.14}$$

où s_i est le i-ème élément de σ donné par la relation d'ordre total. On définit l'opérateur $\partial_k : C_k(X) \to C_{k-1}(X)$:

$$\partial_k = \sum_{i=0}^{k} (-1)^i d_i \tag{1.15}$$

On peut observer que $\partial_k \circ \partial_{k+1} \equiv 0$. On en déduit que $Im(\partial_{k+1}) \subseteq Ker(\partial_k)$. On définit alors :

$$H_k(X,\mathbb{Z}) \cong Ker(\partial_k)/Im(\partial_{k+1} \tag{1.16}$$

$H_k(X,\mathbb{Z})$ est toujours isomorphe à l'homologie singulière de $|X|$ (Carlsson, 2008). Comme les $C_k(X)$ sont des groupes abéliens libres, ils sont équipés de bases, et l'opérateur ∂_k peut être représenté par une matrice, ce qui permet de trouver de manière algorithmique l'homologie d'un complexe simplicial (Dumas et al., 2003).

Une homologie se calcule pour un anneau ou un corps K donné. Le théorème des coefficients universels dit entre autre que l'homologie sur $\mathbb{Z}$ contient toutes les autres. Elle est cependant difficilement accessible, et on calcule plus souvent l'homologie sur $\mathbb{Q}$ ou des $\frac{\mathbb{Z}}{p\mathbb{Z}}$ (qu'on notera aussi $\mathbb{Z}_p$) , où p est premier. Dans ce travail, on utilisera l'homologie sur $\mathbb{Q}$ ce qui exclut notamment les termes de torsions (qui permettent de distinguer un ruban de Möbius d'un ruban classique).

On définit b_k le k-eme nombre de Betti sur un corps K comme le rang de $H_k(X, K)$.

Les premiers nombres de Betti peuvent être interprétés géométriquement. Par exemple pour $H_0(\Sigma)$, les structures de dimension 0 sont les sommets. Alors b_0 sera égal au nombre de composantes connexes de la variété. Deux sommets qui appartiendraient chacun à une composante connexe différente ne pourraient pas être déformé continûment l'un en l'autre, puisqu'il faudrait pour cela qu'ils quittent leur composante connexe, ce qui est une opération qui rompt la continuité.

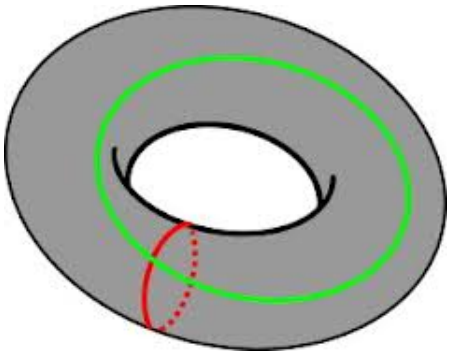


FIGURE 1.5 – Mise en évidence des deux cycles d'un tore. Il n'existe pas d'homéomorphisme transformant un des cycles en l'autre.

1.4.3 Formalisation du problème

On suppose des données $\mathbf{z}$ structurées, c'est-à-dire un échantillon d'un ensemble de variétés $\mathcal{M}$ inconnu, projeté de l'espace de $\mathcal{M}$, appelé E dans un espace d'observation $\tilde{E}$ par un processus f et dont la position est bruitée par un bruit ϵ de loi $\mathcal{E}$ et de paramètres θ eux aussi inconnus. On peut donc écrire les données observées :

$$x = f(z) + \varepsilon \text{ où } \varepsilon \sim \mathcal{E}(0, \theta) \tag{1.17}$$

Si on note $\tilde{\mathcal{M}}$ l'image de $\mathcal{M}$ par le processus f, on peut voir la densité de probabilité $p_{\tilde{\mathcal{M}}}$ qui a généré les données comme étant la convolution de $\tilde{\mathcal{M}}$ avec un bruit de loi $\mathcal{E}$. En ne connaissant que les données observées $\mathbf{x}$, comment retrouver $\mathcal{M}$?

Retrouver $\mathcal{M}$ exactement est un problème difficile, puisque f est inconnue. On cherchera donc à approcher $\tilde{\mathcal{M}}$ par un complexe simplicial, un ensemble de simplexes (sommets, segments, triangles, tétraèdres etc.), qui est la version discrète, donc plus facile à utiliser en algorithmique, d'une variété. Sous certaines conditions (de différentiabilités par exemple), une telle approximation existe toujours (Whitehead, 1940; Munkres, 1966). Pour s'approcher au plus près du modèle de génération des données, on convolue une loi de probabilité avec un complexe simplicial. On utilise par la suite un algorithme classique fondé sur le

principe du maximum de vraisemblance pour optimiser les différents paramètres de notre modèle (position du complexe simplicial, probabilités, variance etc.).

$\mathcal{M}$ et $\tilde{\mathcal{M}}$ sont deux variétés *a priori* différentes : elles vivent dans deux espaces différents, très certainement de dimensions différentes. Les données $\mathbf{z}$ et $\mathbf{x}$ qui leurs sont respectivement associées souffrent des mêmes différences : dimensions, moyennes, variances etc. Et pourtant ces deux variétés ne sont pas si éloignées que ça l'une de l'autre, en fait, elles sont liées par le fait que l'une est l'image de l'autre par la fonction f. Or si f est un homéomorphisme, $\mathcal{M}$ et $\tilde{\mathcal{M}}$ sont topologiquement équivalentes. Les invariants topologiques de l'une et de l'autre sont donc les mêmes. De plus, si l'on dispose de deux variétés $\tilde{\mathcal{M}}_1$ et $\tilde{\mathcal{M}}_2$ issues de deux homéomorphismes f_1 et f_2, chacune aura les mêmes invariants topologiques que la variété initiale. Tant que le processus d'observation est un homéomorphisme, ces invariants sont préservés. C'est pour cette raison que l'on cherche à extraire ces invariants, plus susceptibles d'avoir survécu au processus d'observation. Les invariants topologiques qui nous intéressent sont les nombres de Betti : ils permettent de discriminer certaines variétés, les trois premiers correspondent à des notions simples et des informations potentiellement utiles (nombre de composantes connexes, de cycles, de "trous"), et les algorithmes fournis par la topologie algébrique permettent de les calculer. C'est donc une partie de l'homologie qui est retrouvée par la technique que nous proposons par la suite.

1.4.4 La topologie pour l'analyse de données

Depuis une dizaine d'années, la complexité nouvelle des données due à de nouveaux moyens d'acquisition, mais aussi les nouvelles performances des ordinateurs ont rendu l'apprentissage topologique respectivement nécessaire et possible (Carlsson, 2008).

Bien que différente du domaine du traitement d'images, l'analyse topologique des données peut être couplée à des techniques classique du domaine comme la segmentation dans le domaine de l'imagerie médicale 2D ou 3D du cerveau (Allili et al., 2001), ou la classification de lésions hépatiques (Adcock et al., 2012). Toujours dans le domaine de la 3D, l'analyse des caractéristiques topologiques d'une protéine peut renseigner sur ses propriétés (Edelsbrunner et al., 2000). Pour le traitement du signal, les représentations à partir de fenêtre glissante font apparaître des caractéristiques topologiques (Perea and Harer, 2013), notamment dans le cas des systèmes dynamiques récurrents (Vejdemo-Johansson et al., 2012) ou des électro-encéphalogrammes (Carlsson, 2008). Enfin le fait d'être capable de détecter des cycles et des vides permet de vérifier la bonne couverture d'un réseau de capteur wi-fi (de Silva and Ghrist, 2007).

Comme le montrent ces exemples, la topologie propose déjà des méthodes pour l'analyse de données. A l'inverse, il peut être intéressant de voir quel pourrait être l'apport de l'apprentissage automatique à la topologie. Il convient donc d'étudier de manière plus avancée ces méthodes qui peuvent être utilisées dans le même cadre que les méthodes d'analyse topologique des données.

1.4.5 Organisation du mémoire

Le premier chapitre a introduit la problématique de cette thèse et le caractère novateur du domaine de l'apprentissage topologique. Dans la suite, nous proposons un nouveau modèle qui permet de répondre aux nouveaux problèmes soulevés par ce domaine. Il s'agit d'une généralisation d'un modèle (le Graphe Génératif Gaussien, GGG) initialement proposé par Aupetit (2005), puis développé par Gaillard (2008) qui permet d'extraire la connexité des variétés qui ont le plus vraisemblablement généré des données sous certaines hypothèses que nous détaillerons plus loin. Ce modèle généralisé s'appelle le *Complexe Simplicial Génératif* (CSG). Il peut être vu comme l'extension aux dimensions supérieures du GGG, ou le pendant génératif d'un modèle géométrique, le Witness Complex (De Silva and Carlsson, 2004).

Dans un premier temps sera établi un **état de l'art** des méthodes statistiques et géométriques qui ont mené à la création de ce modèle ou qui répondent à des problèmes équivalents ou proches de ceux résolus par le CSG. Cet état de l'art se base sur la complexité croissante des variétés traitées par les différents algorithmes, et présente à chaque fois, une méthode géométrique et son pendant génératif.

La deuxième partie formalise le problème de l'identification de la variété sous-jacente et propose une méthodologie pour l'aborder. Sont ensuite introduits le Complexe Simplicial Génératif et les méthodes qui permettent d'estimer ses paramètres : optimisation des paramètres par algorithme EM et sélection de modèle à l'aide du critère BIC.

La troisième partie est consacrée aux applications du CSG : caractériser l'homologie de la variété sous-jacente à un jeu de données et étudier la pertinence d'une méthode de projection de données. Le dernier chapitre de cette partie est une conclusion qui synthétise ces travaux.

1.4.6 Contributions

Les principales contributions de ce travail sont les suivantes :

- Un algorithme EM pour le Complexe Simplicial Génératif afin d'optimiser poids et variance du modèle.
- Une étude expérimentale des différents méta-paramètres qui influent sur l'optimisation des paramètres du modèle GSC : évaluation d'une intégrale multidimensionnelle par méthode de Monte-Carlo, taille de l'échantillon optimum pour le compromis précision-temps de calcul pour cette intégrale, critère d'élagage des composants du modèle,
- Un critère de sélection du nombre de sommets utilisés pour construire le CSG basé sur le critère BIC.

Ces travaux ont fait l'objet des publications suivantes :

Conférences avec actes

- (Maillot et al., 2012a) . Première version du Complexe Simplicial Génératif sans apprentissage de la variance, testée sur des variétés simples : boule et sphère.
- (Maillot et al., 2013) . Complexe Simplicial Génératif avec apprentissage de la variance, avec des résultats sur des variétés plus complexes : tore, sphère, et un jeu de données images.

Conférence sans actes

- (Maillot et al., 2012b) Complexe Simplicial Génératif testé sur une surface non-orientable, la bouteille de Klein, et sur un jeu de données images.

2

L'analyse topologique des données

Les données que nous allons chercher à analyser sont de type nuages de points : N points sont représentés dans $\mathbb{R}^D$. On notera cette échantillon $X = (x_1, ..., x_N)$. N est la taille des données, et D est la dimension des données. Le fait que l'un ou l'autre de ces nombres soit grand (et parfois les deux en même temps) rend l'analyse des données plus difficile. Nous ferons des hypothèses sur D plus tard. Quand N est grand, on peut chercher à représenter une sous-ensemble E du nuage de points, qui est donc lui-même un nuage de points, par un prototype qui résume certaines caractéristiques de E. Si E est un nuage de forme sphérique, alors le centre de cette sphère est un exemple de prototype que l'on peut donner pour E. C'est par exemple le but des K-moyennes. On peut aussi ajouter comme information un rayon autour de ce sommet, pour distinguer les petites et les grandes sphères, comme avec les Modèles de Mélange Gaussiens. Et puis petit à petit, on peut aller vers des représentations de prototypes de plus en plus complexes : segments, courbes, surfaces etc. Pour ces différents prototypes, on verra qu'il existe à chaque fois la même dualité entre modèle géométrique (comme les K-moyennes) et modèle génératif (comme les modèles de mélange gaussien). C'est-à-dire que pour un type de prototype donné, il existe une méthode qui va optimiser la position de ces prototypes suivant un critère géométrique, et une autre qui va utiliser le prototype comme base d'un modèle statistique, et dont la position sera optimisée selon un critère statistique. Nous allons présenter ces méthodes en les classant du modèle de prototype le plus simple au plus complexe, en commençant par le modèle géométrique, puis par son pendant génératif.

2.1 K-moyennes

La méthode des K-moyennes est une méthode descriptive et géométrique de partitionnement (MacQueen et al., 1967). Elle fait partie de la famille plus génériques des quantifications vectorielles. En tant que telle, elle permet de partitionner M données en K groupes : on initialise K vecteurs w_k de $\mathbb{R}^D$ appelés "moyennes" (*prototypes*), et on rattache chaque donnée au prototype le plus proche en lui attribuant l'étiquette de ce prototype. On obtient n_k données rattachées à chacun des w_k. Ces n_k données ont un centre géométrique, qui devient le nouveau prototype pour la catégorie k. On réitère le partitionnement avec ces moyennes mises à jour, jusqu'à atteindre la convergence.

Dans le cadre de ces travaux, cette méthode peut être vue comme une façon de représenter les données par une collection de variétés les plus simples possibles : des points.

L'objectif des K-moyennes est de minimiser la distorsion E :

$$E(\mathbf{w};\mathbf{x}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \varphi_k(\mathbf{w}, x_n)\,\|x_n - w_k\|^2 \tag{2.1}$$

avec

$$\varphi_k(\mathbf{w}, x_n) = \begin{cases} 1 \text{ si } w_k = \underset{w_l \in \mathbf{w}}{\operatorname{argmin}}\ \|x_i - w_l\| \\ \\ 0 \text{ sinon} \end{cases}$$

On actualise alors la position des prototypes w_k comme étant le barycentre des données associées à ce prototype.

$$w_k \leftarrow \frac{\sum_{n=1}^{N} \varphi(\mathbf{w}, x_n) x_n}{\sum_{n=1}^{N} \varphi(\mathbf{w}, x_n)} \tag{2.2}$$

Cette formule de mise à jour des prototypes assure une diminution de la distorsion à chaque étape. Le nombre de données et de prototypes étant fini, il existe un nombre fini d'états, et donc la distorsion est bornée. Une suite décroissante bornée est forcément convergente. En revanche rien ne garantit que ce minimum est global, il peut n'être que local. Cet optimum local dépend fortement de la position initiale des prototypes. Pour se rapprocher de l'optimum global, on utilisera différentes initialisations et on retiendra le résultat qui donne la distorsion la plus faible. On obtient un partitionnement des données qui correspond aux cellules de Voronoi des moyennes.

2.2 Modèles de mélanges gaussiens

Les K-moyennes ont un pendant génératif, les modèles de mélanges gaussiens (GMM pour *Gaussian Mixture Model* en anglais). Alors que les K-moyennes utilisent un critère purement géométrique (la distance à la moyenne), les GMM introduisent une dimension probabiliste : la densité de probabilité est approximée par K composants de densité élémentaire de type loi normale multivariée qui peuvent chacune avoir une variance différente et une probabilité différente dans le modèle.

C'est un modèle paramétrique qui permet de représenter une distribution a priori quelconque comme un mélange de distributions suivant un modèle de densité gaussienne identique, mais de paramètres différents. Dans ce modèle, on suppose que la population décrite est un ensemble de différentes sous-populations, chacune suivant sa propre distribution et ne représentant pas la même proportion de la population totale. La distribution la plus utilisée est la distribution gaussienne, c'est pourquoi on parle souvent de modèle de mélanges gaussiens, mais d'autres lois peuvent être envisagée. Une analogie peut être faite avec les fonctions classiques : de la même façon qu'on peut chercher à approximer une fonction par un polynôme (ce que l'on peut toujours faire sur un support compact d'après le théorème de Weierstrass), on peut approximer une distribution de probabilité par une somme pondérée de distributions gaussiennes.

K est le nombre de composantes dans le modèle, N le nombre de données, π_k est la probabilité *a priori* qu'une donnée appartiennent à la composante k du modèle, θ_k représente les paramètres de la composante k du modèle, c'est-à-dire moyenne et variance (w_k, Σ_k), $w_k \in \mathbb{R}^D$,$\Sigma_k \in \mathcal{M}_D(\mathbb{R})$, de chacune des composantes du modèle, et g_k une distribution gaussienne multivariée de paramètres (w_k, Σ_k). La probabilité d'une donnée x est alors :

$$p(x|\theta) = \sum_{k=1}^{K} \pi_k g_k(x|\theta) \tag{2.3}$$

Avec

$$g(x|w_k; \Sigma_k) = \frac{1}{(2\pi)^{(D/2)}|\Sigma_k|^{1/2}} \exp(-(x - w_k)^T \Sigma_k^{-1}(x - w_k)) \tag{2.4}$$

Soit x_n une donnée observée (avec $n = 1, ..., N$). On complète ces données en associant à chacune un vecteur z_n de dimension K, tel que $z_{kn} = 1$ si x_n a été généré par la composante k du mélange gaussien, et 0 sinon. A partir des données $\mathbf{x}$, on va chercher à estimer les z_n.

Le processus génératif du modèle est le suivant :

- On sélectionne d'abord un prototype $w_k \in \mathbf{w}$ selon les probabilités π
- On génère une donnée observée x suivant la loi du composant k choisi à la première étape.

Si l'on souhaite limiter le nombre de paramètres du modèle, on peut ajouter certaines contraintes :

- Sur les π_n par exemple, en les forçant à être tous égaux
- Sur les matrices de variance-covariance, en imposant des matrices isovariées ($\Sigma_n = \sigma_n^2 I$), voire en contraignant $\forall n, \sigma_n = \sigma$

Celeux and Govaert (1995) dénombrent en tout 28 modèles différents pour les modèles de mélange gaussien du plus contraint au plus générique. Ils sont notamment implémentés dans le logiciel Mixmod (Biernacki et al., 2006).

En utilisant le théorème de Bayes, la probabilité qu'une donnée x_n soit issue du groupe k est donnée par :

$$P(x_n \in G_k) = \frac{\pi_k g_k(x_n)}{\sum_{l=1}^{K} \pi_l g_l(x_n)}. \tag{2.5}$$

On utilise la méthode du *maximum a posteriori* pour décider, une fois les paramètres du modèle appris, quel groupe k^* a généré une donnée x_n :

$$k^* = \underset{1 \leq k \leq K}{\operatorname{argmax}} \frac{\pi_k g_k(x_n)}{\sum_{l=1}^{K} \pi_l g_l(x_n)}. \tag{2.6}$$

Une fois le modèle défini, il faut en apprendre les paramètres Θ.On cherche alors les paramètres qui vont maximiser la vraisemblance $\mathcal{L}$:

$$\mathcal{L}(\Theta; \mathbf{x}) = \prod_{n=1}^{N} p(x_n; \Theta). \tag{2.7}$$

On peut aussi utiliser la log-vraisemblance :

$$L(\Theta; \mathbf{x}) = \log(\mathcal{L}(\Theta; \mathbf{x})) = \sum_{n=1}^{N} \log(p(x_n; \Theta)). \tag{2.8}$$

Par définition, l'estimateur du maximum de vraisemblance est alors :

$$\hat{\Theta} = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \mathcal{L}(\theta; \mathbf{x}). \tag{2.9}$$

L'algorithme utilisé est appelé "Expectation-Maximization" et a été proposé plusieurs fois dans des cas particuliers avant d'être généralisé par (Dempster et al., 1977). L'algorithme est découpé en deux étapes, Expectation et Maximization. Le principe de l'algorithme suppose que les données observées $\mathbf{x} = (x_1, ..., x_N)$ sont incomplètes, et considère un vecteur de données manquantes $(z_1, ..., z_N)$ qui correspondent chacune à la classe à laquelle appartient la donnée x_n, l'ensemble $(\mathbf{x}, \mathbf{z})$ étant appelé données complétées. La vraisemblance est alors une fonction de $(\mathbf{x}, \underline{z})$ et θ. Si on note $f(z_i|x_i; \theta)$ la probabilité de z_i sachant x_i et les paramètres Θ, alors on peut écrire la log-vraisemblance complétée comme la quantité :

$$L((\mathbf{x},\mathbf{z});\Theta) = \sum_{n=1}^{N}(\log f(z_n|x_n,\Theta) + \log f(x_n;\Theta)). \tag{2.10}$$

et donc,

$$L(\mathbf{x};\Theta) = L((\mathbf{x},\mathbf{z});\theta) - \sum_{n=1}^{N}\log f(z_n|x_n,\Theta). \tag{2.11}$$

L'algorithme EM est une procédure itérative où l'on augmente la vraisemblance à chaque étape lors de la mise à jour du paramètre courant. En notant $\Theta^{(c)}$ ce paramètre, on peut écrire

$$E[L(\mathbf{x};\Theta)|\Theta^{(c)}] = E[L((\mathbf{x},\mathbf{z});\Theta))|\Theta^{(c)}] - E[\sum_{n=1}^{N}\log f(z_n|x_n,\Theta))|\Theta^{(c)}], \tag{2.12}$$

où l'espérance est prise sur $\mathbf{z}$.

Ou encore $L(\mathbf{x};\Theta) = Q(\Theta;\Theta^{(c)}) - H(\Theta;\Theta^{(c)})$, puisque $L(\mathbf{x};\Theta)$ est indépendant de $\mathbf{z}$, en posant $Q(\Theta;\Theta^{(c)}) = E\left[L((\mathbf{x},\mathbf{z});\Theta))|\Theta^{(c)}\right]$ et $H(\Theta;\Theta^{(c)}) = E\left[\sum_{n=1}^{N}\log f(z_n|x_n,\Theta))|\Theta^{(c)}\right]$.

Ensuite, on peut montrer que :

$$\Theta^{(c+1)} = \arg\max_{\Theta}(Q(\Theta,\Theta^{(c)}))$$

fait tendre $L(x;\theta^{(c+1)})$ vers un maximum local.

Pour la réalisation pratique, on calcule d'abord la probabilité *a posteriori* des données manquantes (étape *Expectation*) :

$$\tilde{z_{kn}} \leftarrow \frac{p(z_{kn})p(x_n|z_{kn},\theta)}{p(x_n;\theta)} \tag{2.13}$$

On met ensuite à jour les paramètres du modèle (étape *Maximization*)

$$\pi_k \leftarrow \frac{1}{N}\sum_{n=1}^{N}\tilde{z_{kn}} \tag{2.14}$$

$$w_k \leftarrow \frac{\sum_{n=1}^{N}\tilde{z_{kn}}x_n}{\sum_{n=1}^{N}\tilde{z_{kn}}} \tag{2.15}$$

$$\Sigma_k \leftarrow \frac{\sum_{n=1}^{N}\tilde{z_{kn}}(x_n-w_k)^T(x_n-w_k)}{\sum_{n=1}^{N}\tilde{z_{kn}}} \tag{2.16}$$

Les deux méthodes vues ci-dessus sont naturellement adaptées pour répondre aux problèmes de classification : les données sont facilement regroupées dans la classe d'un prototype qui représentera toute la classe, celui qui correspond au *maximum a posteriori* de la donnée.

Parmi les applications des GMM, on trouve la vérification et la reconnaissance vocale comme dans (Reynolds et al., 2000), le traitement d'images avec l'identification de l'arrière-plan (Stauffer and Grimson, 1999), ou la classification de signaux (electro-cardiogrammes (Huang et al., 2005), (Martis et al., 2009) (Bishop et al., 1998b)

2.3 Cartes auto-organisées

2.3.1 Cartes de Kohonen

Les cartes de Kohonen, aussi appelées *Self-Organizing Map* (SOM) sont un type d'apprentissage non supervisé. Elles ont été développées par Teuvo Kohonen, qui s'est inspiré des réseaux de neurones biologiques, pour cartographier un espace réel. (Kohonen, 1990).

Étant donné un ensemble P de prototypes disposés sur une grille de dimension 1 ou 2 et $\mathbf{x}$ les données. L'idée est d'associer chaque donnée $x_n \in \mathbf{x}$ à un prototype $p_k \in P$, le but étant que des données proches dans l'espace de départ soit associées à des prototypes proches sur la grille. Pour cela on définit la fonction δ :

$$\delta : \left| \begin{array}{ccc} \mathbf{x} & \longrightarrow & P \\ x_n & \longmapsto & \underset{p_k \in P}{\operatorname{argmin}} \ \|x_n - p\| = p_k^* \end{array} \right. \tag{2.17}$$

Une fois que les données sont associées à leur prototype le plus proche, la position des prototypes est mise à jour :

$$p_k \leftarrow p_k + \tau\varphi(p_k, p_k^*)(x_n, p_k) \tag{2.18}$$

où τ est un pas d'apprentissage qui décroit au cours du temps et φ une fonction de voisinage. Originellement cette fonction était discrète (1 si $p_k = p_k^*$, 0 sinon). Mais Lo and Bavarian (1991) montrent qu'il vaut mieux utiliser une fonction continue du type :

$$\varphi(p_k, p_k^*) = \exp(\frac{\|p_k - p_k^*\|}{2\sigma^2}) \tag{2.19}$$

Dans ce cas-là, c'est σ qui fait office de pas d'apprentissage (paramètre de voisinage). Kohonen conseille de faire décroître ce paramètre de voisinage sur des seules considérations pratiques.

Bien que populaires, les cartes de Kohonen ont toujours manqué de fondements théoriques : la convergence de l'algorithme n'est démontrée que dans le cas d'une carte unidimensionnelle (Hertz et al., 1991), il n'y a pas de critère objectif pour comparer différents modèles de cartes, il y a même des résultats invalidants certains fondements de l'algorithme (Erwin et al., 1992). Face à ces limitations, différentes solutions génératives ont été proposées.

2.3.2 Generative Topographic Mapping

Parmi les alternatives génératives aux cartes de Kohonen (Vlassis et al., 1997; Verbeek et al., 2003), le *Generative Topographic Mapping* (GTM) est certainement la plus populaire (Bishop et al., 1998a).

Contrairement aux cartes de Kohonen, le fait que le GTM soit un modèle génératif apporte tous les fondements théoriques qui vont avec ces modèles : optimisation par algorithme Expectation-Maximization, critère objectif de sélection du modèle, convergence de l'optimisation, interprétation probabiliste etc.

Le GTM définit la variété sous-jacente $\mathcal{M}$ comme un sous-espace euclidien, échantillonné par une grille de K sommets, projetée ensuite de manière non-linéaire dans l'espace des données par une fonction $f : \mathcal{M} \longrightarrow \mathbb{R}^D$. La distribution choisie sur $\mathcal{M}$ est une somme de Dirac dont les sommets z_n sont répartis le long de la grille :

$$p(z) = \frac{1}{K} \sum_{k=1}^{K} \delta(z - z_k) \tag{2.20}$$

L'espace originel ainsi discrétisé est équivalent à une carte de Kohonen. La fonction f est définie comme une somme de L gaussiennes pondérée par une matrice W de taille $D \times L$. D correspond à la dimension de l'espace des données observées, et L est un paramètre qui contrôle la régularité de f. La détermination de L a fait l'objet de plusieurs publications (Bishop et al., 1998a; Vellido et al., 2003). Le bruit des données est modélisé par une loi gaussienne g de variance σ^2 isotropique :

$$p(x|z;W) = g(x|f(z;W);\sigma^2) \tag{2.21}$$

$$= \frac{1}{K} \sum_{k=1}^{K} g(x|f(z_k,W);\sigma^2) \tag{2.22}$$

Si on pose $w_k = f(z_k;W)$, alors on peut voir $p(x|z;W)$ comme un modèle de mélange gaussien :

$$p(x|z;W) = \frac{1}{K} \sum_{k=1}^{K} g(x|w_k;\sigma^2) \tag{2.23}$$

Les paramètres à optimiser, W et σ^2 le sont grâce à l'algorithme EM.

2.4 Courbes principales

2.4.1 Modèle géométrique

Introduites pour la première fois par Hastie et Stuetzle Hastie and Stuetzle (1989), les courbes principales sont des courbes lisses qui approximent un nuage de points. La nouveauté réside dans le fait qu'elles ne sont pas forcément linéaires. Une fonction f définit une courbe qui est dite principale pour un nuage de point si elle vérifie la condition de consistance ;

$$f(z) = \mathbb{E}[x|g(x) = z] \tag{2.24}$$

où x est une donnée générée par une probabilité $P(x)$ et $g(x)$ la projection de x sur f :

$$g(x) = \sup\{\lambda : \|x - f(\lambda)\| = inf_{\mu} \|x - f(\mu)\|\} \tag{2.25}$$

La courbe principale minimise la distorsion E :

$$\mathbb{E}_{P(x)}[\|x - g(x)\|^2] \tag{2.26}$$

On peut donc la rapprocher d'une méthode de type "moindres carrés".

2.4.2 Modèle génératif

Ce modèle est d'autant plus important dans le cadre de cette thèse qu'il a apporté l'idée de faire porter la structure de la variété sous-jacente à autre chose qu'un point, en l'occurrence une courbe. Tibshirani (1992) propose une approche générative pour combler certaines lacunes de la première version de l'algorithme. De manière intuitive on peut le voir comme la convolution d'une fonction f avec une densité gaussienne isotropique, dépendant de la position sur la courbe :

$$p(x|z) = g(x|f(z); \sigma(z)^2) \tag{2.27}$$

Tibshirani utilise alors un théorème général sur les modèles de mélanges (Lindsay, 1983) pour montrer que résoudre son problème se ramène à estimer un modèle de mélange gaussien à nombre fini de composantes : il discrétise ainsi la fonction f et en dérive les équations de l'algorithme Expectation-Maximization correspondant pour estimer les paramètres $(\pi, \boldsymbol{\sigma^2}, \mathbf{w})$. Tibshirani introduit en même temps un modèle régularisé pour lisser la courbe.

Les courbes principales avaient dès leur origine une application pour paramétrer l'accélérateur linéaire de Stanford (Hastie and Stuetzle, 1989). On retrouve d'autres applications dans le traitement d'image (Banfield and Raftery, 1992; Stanford and Raftery, 2000)

La plupart des méthodes et techniques suivantes ont été utilisées dans un premier temps pour de la reconstruction 3D : à partir de données échantillonnées de plus en plus précisément, il s'agit de reconstruire un objet réel, c'est-à-dire de retrouver une information continue qui a été perdue lors de l'échantillonnage. Ce type de données et de problématiques sont devenues de plus en plus populaire avec l'apparition de scanners 3D de plus en plus précis (Bajaj et al., 1995; Bernardini et al., 1999).

De manière intuitive, si l'on est familier avec le théorème de Shannon en traitement du signal, on peut comprendre qu'il faut une certaine taille d'échantillon pour pouvoir reconstruire correctement l'objet initial et qu'il faut un échantillonnage plus fin dans les régions les plus complexes, celles qui ont la plus grande courbure locale par exemple.

Elles ont par la suite évolué pour s'adapter à des dimensions plus grandes, et pour tirer des informations de nature topologique. Une des informations les plus simples à extraire est le nombre de composantes connexes. Or, en faisant un parallèle entre une composante connexe d'une part, et un groupe d'autre part, un outil d'analyse topologique peut être utilisé pour faire de la classification. Le Complexe Simplicial Génératif qui est l'objet de cette thèse fait lui le cheminement inverse : à partir de modèles de mélanges et de l'algorithme EM (des méthodes statistiques donc), il va chercher à regrouper des classes (composants du modèle) pour former des composantes connexes.

Nous considérons le graphe de Delaunay comme structure élémentaire pouvant relier les sommets pour former des composantes connexes.

2.5 Le graphe de Delaunay

À l'origine du graphe de Delaunay, on trouve la triangulation de Delaunay d'un ensemble de points en 2D. Il s'agit de construire la triangulation d'un ensemble de point suivant un critère géométrique qui consiste à tester les intersections de cercles circonscrits.

Soit $\boldsymbol{X}$, un ensemble de points de $\mathbb{R}^2$. Soit a,b,$c \in \boldsymbol{X}$. On note (a, b, c) le triangle formé par ces trois points, $C(a, b, c)$ le cercle circonscrit à ce triangle, et $DT(\boldsymbol{X})$ la triangulation de Delaunay de $\boldsymbol{X}$. Alors :

$$\forall (a, b, c), a \neq b, b \neq c, c \neq a, \in DT(\boldsymbol{X}) \Leftrightarrow C(a, b, c) \cap \boldsymbol{X} = \{a, b, c\} \tag{2.28}$$

Autrement dit, les seuls points de $\boldsymbol{X}$ à l'intérieur du cercle circonscrit d'un triangle sont les sommets du triangle eux-mêmes.

Cette notion peut être généralisée aux dimensions supérieures avec le complexe de Delaunay. Il ne s'agit donc plus d'un ensemble de triangles, mais d'un ensemble de tétraèdres en dimension 3 par exemple, et de k-simplexes dans les dimensions supérieures. La condition se généralise : un k-simplexe appartient au graphe de Delaunay si sa k-sphère circonscrite ne contient pas de points autres que les sommets du k-simplexe.

Parmi les avantages du graphe de Delaunay, on note qu'il a tendance à ne pas contenir

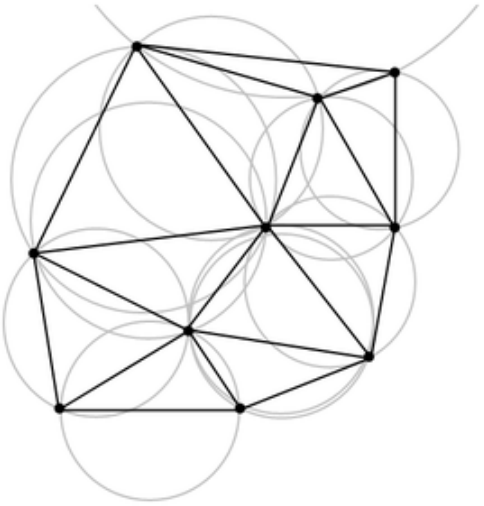


FIGURE 2.1 – Triangulation de Delaunay d'un ensemble de sommets avec mise en évidence des cercles circonscrits aux triangles : ils ne contiennent pas de sommets autre que ceux du triangle

de triangles trop aplatis (puisque leur cercle circonscrit est très grand, et a donc plus de chances de contenir d'autres sommets) et que la triangulation ne peut pas contenir des segments qui s'intersectent.

2.5.1 Calcul rapide du complexe de Delaunay

La complexité du calcul de la triangulation de Delaunay est exponentielle avec la dimension de l'espace ambiant (Boissonnat et al., 1998). Or nous faisons l'hypothèse que les données reposent sur une variété de dimension intrinsèque d inférieure à D, la dimension de l'espace ambiant. On peut calculer le d-squelette du complexe de Delaunay, grâce à une méthode reposant sur la programmation linéaire (Méndez and Lorenzo, 2012), qui construit le 1-squelette, le 2-squelette, ..., le d-squelette de Delaunay progressivement. La complexité dépend donc de d et non de D, comme cela peut-être le cas avec la méthode *delaunayn* dans Matlab. La méthode de Matlab "lifte" les points dans un espace de dimension $D+1$, en ajoutant une $D+1$-ème coordonnée pour créer un paraboloïde de coordonnées $(x_1, x_2, ..., x_D, \sum_{i=1}^{D} x_i^2)$. On recherche l'enveloppe convexe des points liftés, et les facettes de l'enveloppe convexe sont les simplexes de la triangulation de Delaunay.

2.5.2 Applications

En dimension 2 et 3, la triangulation de Delaunay est encore calculable. Il y a donc beaucoup d'applications dans la reconstruction de formes dans ces dimensions. Ainsi dans le domaine de l'imagerie médicale, on peut utiliser la triangulation de Delaunay pour

reconstituer en 3D des vues en coupe de certains organes humains pour l'aide à la chirurgie ou l'aide à la radiothérapie (Boissonnat and Geiger, 1993).

2.6 Triangulation de Delaunay restreinte

2.6.1 *Topology Representing Network*

Étant donné un ensemble de prototypes $\mathbf{w}$ et un jeu de données $\mathbf{x}$, on peut chercher à approximer une structure sous-jacente à ces données par un graphe de Delaunay. Or le graphe de Delaunay ne dépend que de la position des prototypes $\mathbf{w}$, et aucunement de la position des données $\mathbf{x}$. Cette information peut être utilisée pour positionner les prototypes (k-moyennes ou GMM par exemple). Mais elle peut encore être utilisée pour décider de garder ou non une arête dans le graphe de Delaunay. C'est ce qui est proposé notamment avec le Topology Representing Network (TRN) (Martinetz and Schulten, 1994) et l'algorithme Competitive Hebbian Learning (algorithme 1). L'ensemble initial des arêtes est l'ensemble vide. S'il existe un couple de prototypes qui sont les deux plus proches d'une donnée, alors on ajoute l'arête qui correspond à cette donnée. Cette donnée est le "témoin" de l'arête en question.

Algorithm 1 Competitive Hebbian Learning

Require: $\mathbf{w}, \mathbf{x}$
  $E \leftarrow \varnothing$
  for i = 1,...,M **do**
    $k = \underset{j}{\operatorname{argmin}} \{\|x_i - w_j\| \,|w_j \in \mathbf{w}\}$
    $l = \underset{j}{\operatorname{argmin}} \{\|x_i - w_j\| \,|w_j \in \mathbf{w} \setminus \{w_k\}\}$
    $E \leftarrow E \cup \{w_k, w_l\}$
  end for
Ensure: E

En sortie du *Topology Representing Network*, on obtient un sous-graphe du graphe de Delaunay, un graphe de Delaunay "élagué" pour correspondre au mieux aux données. Un des résultats importants apporté par Martinetz et Schulten est la preuve que le *Topology Representing Network* préserve la topologie de la variété sous-jacente à condition que l'échantillonnage des prototypes $\mathbf{w}$ soit dense dans la variété.

Gaillard (2008) dans sa thèse met en avant les limites du CHL :

- Il suffit d'une donnée pour qu'une arête soit conservée dans le *Topology Representing Network*. Il est possible de classer les arêtes selon le nombre de témoins qu'elles possèdent (Martinetz et al., 1993; Fritzke et al., 1995), et de seuiller le graphe pour ne garder que les arêtes qui ont le plus de témoins, mais se pose alors la question du choix de ce seuil qui n'a pas été résolue. D'où une grande **sensibilité au bruit**.
- Contrairement aux Courbes Principales, les données ne sont pas projetées directement sur le lien dont elles sont le témoins. La donnée qui permet l'existence d'un lien peut se retrouver très loin du-dit lien.
- Toujours par comparaison aux Courbes Principales, on ne retrouve pas le **principe de consistance** : certains liens ont une région d'influence qui ne les intersecte pas.
- Le fait d'être dans un cadre essentiellement géométrique ne permet pas la mesure de

qualité permise par les modèles génératifs via la vraisemblance.

2.6.2 Le Graphe Génératif Gaussien

Devant les limites du *Topology Representing Network*, Aupetit (2005) puis Gaillard (2008) introduisent un pendant génératif, le Graphe Génératif Gaussien (GGG). Le principe est proche de celui des Courbes Principales, puisqu'il reprend l'idée de convoluer une courbe avec une densité gaussienne, tout en contraignant cette courbe à être un segment.

Etant donné un ensemble de prototypes $\mathbf{w}$, le graphe génératif gaussien est un modèle de mélange gaussien qui contient, outre les gaussiennes centrées sur les prototypes, les segments gaussiens correspondants aux arêtes du graphe de Delaunay des prototypes. Un segment gaussien a une densité de probabilité g^1 associée qui correspond à l'intégrale d'une gaussienne classique g^0 le long d'un segment $[w_a, w_b]$ de longueur L_n :

$$g^1(x_i|[w_a, w_b]; \sigma^2) = \frac{1}{L_n} \int_{w_a}^{w_b} g^0(x_i|t; \sigma^2) \mathrm{d}t \tag{2.29}$$

Sommets et segments gaussiens ont chacun une probabilité associée dans le modèle de mélange. Ces probabilités, ainsi que la variance peuvent être optimisées de façon à maximiser la vraisemblance par un algorithme EM, et un algorithme GEM (Dempster et al., 1977) pour la position des sommets. L'algorithme GEM consiste à exécuter l'étape M de l'algorithme même si ce n'est pas celle qui maximise la vraisemblance, tant qu'elle l'augmente même faiblement. S'ensuit une sélection de modèles (sur le nombre de sommets ainsi que sur le nombre de composantes à garder à nombre de sommets fixé) à l'aide du critère BIC.

Cette étape est cruciale : c'est ici que se joue l'apprentissage de la connexité de la variété sous-jacente. De la "survie" d'un segment dans le modèle dépend l'existence ou non d'un lien entre les deux prototypes w_a et w_b qui constituent les extrémités de ce segment. *In fine*, cela permet de déterminer les différentes composantes connexes de la variété sous-jacente.

2.7 Complexe de Vietoris-Rips

Le complexe de Vietoris-Rips (VR) a été introduit pour la première fois par Vietoris (1927) pour étendre la théorie de l'homologie à tout espace métrique. Bien plus tard, Rips l'a utilisé dans l'étude des groupes hyperboliques, ce qui a conduit Gromov à l'appeler complexe de Rips (Gromov, 1987). Ce n'est qu'en 1995 que Hausmann et al. (1995) lui donnent le nom de complexe de Vietoris-Rips.

Soient S un ensemble de sommets dans $\mathbb{R}^n$, d la distance métrique euclidienne, un réel positif ε, alors le complexe de Vietoris-Rips $\mathcal{V}_\varepsilon$ est défini par :

$$\mathcal{V}_\varepsilon = \{\sigma \subseteq S | d(u,v) \leq \varepsilon, \forall u \neq v \in \sigma\} \tag{2.30}$$

Le complexe de Vietoris-Rips dépend donc de la distance ε : une arête appartient au simplexe si les deux sommets qui la composent sont à une distance inférieure à ε. Pour les simplexes de dimension supérieure il faut que chacune des arêtes qui le compose soient inférieures à ε. Pour ε assez grand, tous les simplexes appartiendront donc au complexe. Pour chaque simplexe on peut définir une fonction de poids :

$$w(\sigma) = \begin{cases} 0 \text{ si } dim(\sigma) = 0 \\ d(u,v) \text{ si } \sigma = (u,v) \\ \max\limits_{\tau \subset \sigma} w(\tau) \text{ dans les autres cas} \end{cases}$$

Cette fonction de poids prendra tout son sens quand on parlera de persistance homologique dans la section 2.9. Le complexe de Vietoris-Rips est construit directement sur l'ensemble de points étudiés. Avec la taille et la dimension des données, calculer un complexe de Vietoris peut devenir très coûteux en temps et en mémoire. De nouvelles implémentations ont donc vu le jour pour accélérer ces temps de calcul (Zomorodian, 2010).

2.8 Witness Complex

Le *Witness Complex* (WitC) part du constat que le complexe de Vietoris-Rips, bien que permettant de retrouver une topologie correcte, est obligé de travailler sur l'intégralité des données, ce qui peut devenir problématique quand la taille et la dimension des données augmente (De Silva and Carlsson, 2004). Il se fonde sur une définition d'un complexe de Delaunay faible, déjà introduite par De Silva (2003). L'idée est de "résumer" les données par des prototypes $\mathbf{w}$ et de construire un complexe simplicial en utilisant ces prototypes comme sommets. Le critère pour garder ou non un simplexe dans le complexe simplicial dépendra du nombre de données qui en sont les "témoins" (d'où le nom de Witness Complex).

Soit w_1 et w_2 deux prototypes. L'arête (w_1, w_2) appartient au Witness Complex $W_\infty(\mathbf{w})$ s'il existe une donnée x_i telle que w_1 et w_2 soient les deux prototypes les plus proches de cette donnée. On généralise cette définition aux dimensions supérieures : le k-simplexe $(w_1, ..., w_{k+1})$ appartient à $W_\infty(\mathbf{w})$ s'il existe une donnée telle que $(w_1, ..., w_{k+1})$ sont les $k+1$ prototypes les plus proches.

En réalité on n'utilise que très peu W_∞, mais plutôt $W_1 \supseteq W_\infty$: on applique bien la définition pour les arêtes, puis pour les simplexes de dimension supérieure, on les ajoute à W_1 si toutes les arêtes qui les constituent appartiennent à W_1. Cette définition est plus faible, puisqu'elle ne nécessite pas que le témoin de chacune des arêtes soit forcément le même pour toutes les arêtes d'un simplexe.

Cet algorithme est le principal point de comparaison avec celui qui sera développé dans cette thèse : il répond à la même problématique, et surtout il se fonde non pas directement sur les données, mais sur des prototypes. Pour le choix des prototypes, De Silva et Carlsson proposent de les choisir de manière aléatoire parmi les données, ou en utilisant l'algorithme *maxmin* : un premier prototype est choisi aléatoirement parmi les données, le second est le point le plus éloigné de ce prototype parmi les données, le troisième le point le plus éloigné des deux premiers prototypes parmi les données et ainsi de suite pour les suivants. Ils déconseillent l'utilisation de techniques comme les k-means pour positionner leurs prototypes, ceci pouvant être très cher en temps de calcul dans le cas de grands jeux de données ainsi que pour éviter de suréchantilloner certaines zones de l'espace dense en points. L'algorithme *maxmin* génèrent des prototypes régulièrement répartis dans l'espace, et c'est la principale qualité recherchée par les auteurs. Cette méthode est en revanche assez sensible au bruit, puisqu'un "outlier" isolé, éloigné de tous les autres sera forcément choisi par l'algorithme *maxmin*.

Comme avec le complexe de Vietoris-Rips, les auteurs introduisent un Witness-Complex à ϵ près (une tolérance sur les distances minimum pour qu'un simplexe appartienne au complexe total) afin d'induire une filtration pour la persistance homologique.

2.9 La persistance homologique

La persistance topologique puis homologique par la suite, était dans l'air du temps à la fin des années 90 (Frosini and Landi, 1999; Robins, 1999), avant que le terme et une définition précise n'apparaissent dans les travaux d' Edelsbrunner et al. (2000), puis une généralisation de ces premiers travaux limités à $\frac{\mathbb{Z}}{2\mathbb{Z}}$ à tous les corps commutatifs a été proposée (Zomorodian and Carlsson, 2005). Deux limitations des complexes simpliciaux sont contournées avec cette méthode : la choix du paramètre ϵ (trop petit et l'on a qu'une collection de sommets isolés, trop grand, et l'on n'obtient qu'une composante connexe, un *blob*), et le problème posé par des données bruitées.

Le principe est de faire varier ε. Pour chacune de ses valeurs, on obtient un complexe simplicial soit identique, soit différent mais inclus dans le complexe précédent, et basé sur les mêmes prototypes : $\varepsilon < \varepsilon' \Rightarrow W_\varepsilon \subseteq W_{\varepsilon'}$. Pour une suite $(\varepsilon_1 < ... < \varepsilon_N)$ on obtient alors une famille emboîtée de complexes simpliciaux (W_{ε_n}, que l'on appelle une filtration. On étudie alors l'apparition et la disparition des caractéristiques topologiques mesurées par les nombres de Betti au cours de la variation de ε : les composantes connexes qui se connectent petit à petit, les cycles qui se créent avant d'être "bouchés" par des surfaces ou des volumes.

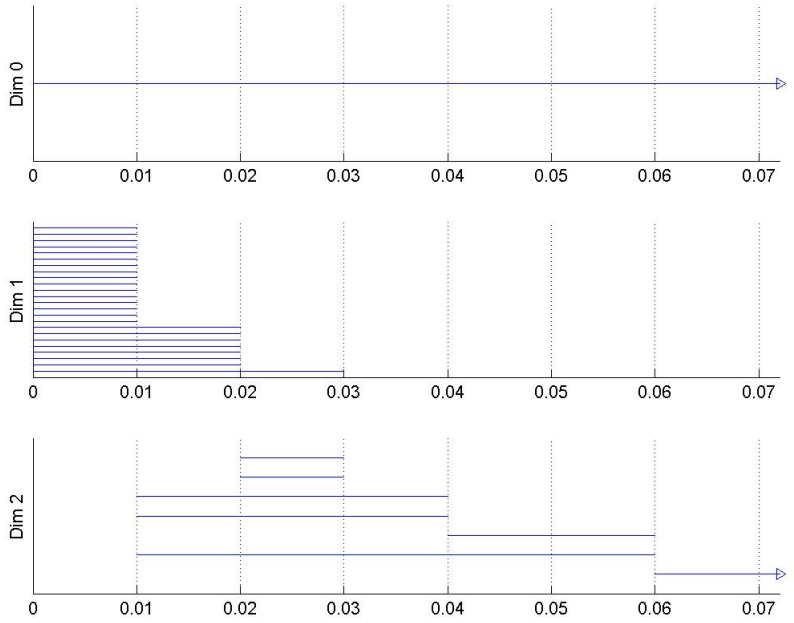


FIGURE 2.2 – Au cours de l'évolution de ε, différentes caractéristiques topologiques naissent puis disparaissent. On peut les représenter sous forme de code-barre. Celui-ci correspond à des données issues d'une sphère. Les flèches symbolisent les caractéristiques qui persistent le plus longtemps. C'est celles qui sont conservées : 1 pour b_0, 0 pour b_1 et 1 pour b_2

La persistance homologique dispose de nombreux résultats théoriques sur la validité de l'approche et sa capacité à retrouver la bonne topologie sous certaines hypothèses (Chazal et al., 2011a, 2012).

Enfin, grâce à l'analogie qui peut être faite entre un groupe de données en statistique et une composante connexe en topologie, il est possible d'utiliser les complexes de Rips et la persistance homologique pour extraire des groupes de données dans un espace métrique. Le β_0 donne directement le nombre de classes. Les résultats sont concluants sur des données bruitées notamment avec un bruit gaussien (Chazal et al., 2013). Ceci montre que même pour des caractéristiques statistiques la topologie peut apporter une contribution. L'interprétation statistique et l'interprétation topologique sont deux points de vue sur une réalité inconnue non absolue et non objective.

	$\mathcal{M}$	$\mathcal{H}$	$O()$
GMM	points	β_0	NM^2
GTM	graphe en grille $2D$		$NM^2 + N^2$
TRN	graphe	β_1	NM^2
GGG	graphe	β_1	NM^2
WitC	complexe simplicial	β_n	$D'NM^2$

TABLE 2.1 – **Récapitulatif de l'état de l'art.** Tableau résumant les propriétés du GMM (modèle de mélange gaussien), GTM (generative topographic mapping), TRN (topology representing network), GGG (graphe génératif gaussien) et WitC (Witness Complex). $\mathcal{M}$ indique le type de variété apprise par l'algorithme, $\mathcal{H}$ le plus grand nombre de Betti qui peut être appris par le modèle, et $O()$ est la complexité au pire cas, où M est le nombre de données, N le nombre de prototypes dans la modèle et D' la dimension intrinsèque du modèle.

2.10 Synthèse de l'état de l'art

Nous synthétisons cet état de l'art par le tableau 2.1 qui reprend les différentes caractéristiques des principaux algorithmes précédemment cités. On retrouve dans ce tableau tout d'abord le type de variétés qu'ils peuvent apprendre, la complexité de l'homologie qu'il est capable d'apprendre (composantes connexes β_0, cycles β_1, etc.) et leur complexité algorithmique.

Partant du constat qu'à partir du *Topology Representing Network*, l'apprentissage topologique a évolué dans deux directions différentes :

- une généralisation aux dimensions supérieures grâce au Witness Complex, assortie d'un critère de sélection de modèle grâce à la persistance homologique,
- une version générative qui permet de sélectionner les modèles avec un critère statistique, le Graphe Génératif Gaussien,

on peut imaginer un modèle qui réunirait ces deux branches, et qui pourrait être vu comme une version générative du Witness Complex, ou une généralisation aux dimensions supérieures du GGG. C'est le modèle que nous allons présenter maintenant.

3

Le complexe simplicial génératif

Dans cette partie, nous posons le problème de l'identification et de la caractérisation de la variété sous-jacente aux données, et nous proposons un modèle capable de retrouver correctement les paramètres d'un jeu de données issu de la même famille que celles du modèle (complexe simplicial, bruit gaussien). Le but n'est pas d'obtenir une projection ou une visualisation des données, mais bien d'extraire leur topologie, et plus exactement la topologie de la variété sous-jacente à ces données, directement dans l'espace initial. La topologie est caractérisée par les nombres de Betti de la variété et la dimension intrinsèque.

Le principal problème est qu'il n'y a pas unicité de la caractérisation d'une variété par ses nombres de Betti par exemple comme vu en 1.4.2. Les approches topologiques classiques sont sensibles au bruit et ne modélisent pas la source des observations. A l'inverse, les approches statistiques de type modèle génératif n'extraient pas d'information topologique pertinente.

Nous proposons donc un modèle qui intégrera à la fois une composante statistique, issue des modèles de mélanges, mais qui comportera des contraintes géométriques et topologiques. Le principe est de combiner un complexe simplicial, donc une discrétisation d'une variété, avec un modèle de bruit gaussien qui prend en compte l'incertitude sur la position des simplexes qui constituent le complexe simplicial.

3.1 L'identification de la variété sous-jacente

3.1.1 Modèle génératif

Nous proposons un modèle génératif, appelé complexe simplicial génératif (CSG) pour l'identification des variétés sous-jacentes $\mathcal{V}$ qui peut être décrit de la façon suivante :

- Les données $\mathbf{z}$ sont tirées des variétés $\mathcal{V}$, selon une densité de probabilité $p_{\mathcal{V}}$.
- Il existe un processus d'observation f, modélisé par une fonction qui va de l'espace original de $\mathcal{V}$ dans l'espace où $\mathcal{V}$ est observée. Ce processus transforme $\mathcal{V}$ en $\tilde{\mathcal{V}}$ les données $\mathbf{z}$ en des données observées $\mathbf{x}$, avec un bruit ε qui suit une loi $\mathcal{E}$, de moyenne nulle. On note alors :

$$x = f(z) + \varepsilon \text{ avec } \varepsilon \sim \mathcal{E}(0, \theta_\varepsilon)$$

Le problème qui nous intéresse est donc de retrouver les caractéristiques topologiques de $\mathcal{V}$, alors que nous n'avons à notre disposition que les données $\mathbf{x}$. En effet :

1. La loi de $\mathcal{E}$ est inconnue, ce qui rend difficile le débruitage des données pour accéder à la variété $\tilde{\mathcal{V}}$.
2. La variété $\tilde{\mathcal{V}}$ est inconnue, alors qu'elle permettrait d'accéder à la variété $\mathcal{V}$ en inversant le processus d'observation f.
3. Le processus d'observation f lui-même est inconnu.
4. Enfin la densité de probabilité $p_{\mathcal{V}}$ avec laquelle la variété sous-jacente est échantillonnée est inconnue.

Pour chacun de ces problèmes, il va falloir faire une hypothèse simplificatrice lui correspondant :

1. On suppose que le bruit ε suit une loi gaussienne homoscédastique de moyenne nulle et de variance σ^2.

$$\varepsilon \sim \mathcal{N}(0, \sigma^2) \tag{3.1}$$

   On peut bien sûr considérer d'autres lois pour le bruit.
2. La variété $\tilde{\mathcal{V}}$ est inconnue, mais on suppose qu'il existe toujours un complexe simplicial homéomorphe à la variété. On suppose donc qu'en positionnant N sommets dans l'espace, pour un nombre N assez grand, et pour un positionnement adéquat relativement aux données, il existera, parmi tous les complexes simpliciaux qui peuvent être définis par ces N sommets, un complexe simplicial de même topologie que la variété sous-jacente.
3. La topologie, comme dit dans l'introduction, est une information invariante aux homéomorphismes, une large famille de transformation qui contient notamment les isomorphismes et les similitudes. On fait donc l'hypothèse que f, le processus d'observation appartient à une des familles de transformations qui ne modifient pas la topologie, et en particulier les nombres de Betti.
4. Enfin, pour la densité $p_{\mathcal{V}}$, on fait l'hypothèse qu'elle est uniforme par morceaux, c'est-à-dire constante sur un domaine qui peut être approximé facilement par un simplexe.

3.1.2 Un modèle pour identifier la variété sous-jacente

3.1.2.1 Le simplexe génératif

Un simplexe génératif est le composant élémentaire du CSG. C'est une densité de probabilité obtenue par la convolution d'un bruit ε avec un simplexe. Soit S_i^d un simplexe de dimension d avec $d+1$ sommets dans $\mathbb{R}^D$, $|S_k^d|$ est son volume, g^0 une distribution gaussienne isovariée de dimension D représentant le bruit ε, $\sigma > 0$ son écart-type, et g_k^d la distribution de probabilité induite par le simplexe gaussien associé à S_i^d :

$$g_k^d(x) = \frac{1}{|S_k^d|} \int_{S_k^d} g^0(x|t, \sigma^2) dt. \tag{3.2}$$

Cela peut être vu comme la convolution d'une loi normale multivariée avec un d-simplexe. L'idée principale du modèle est de pouvoir utiliser un simplexe quelconque au lieu d'un point en tant que composante dans un modèle de mélange.

La figure 3.1 montre un triangle génératif : chaque cercle représente une distribution gaussienne, elles partagent toutes la même variance, et la probabilité résultante est la moyenne des contributions de chaque distribution gaussienne.

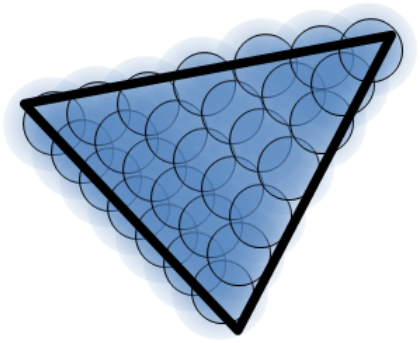


FIGURE 3.1 – Exemple de triangle génératif

3.1.2.2 Le complexe simplicial génératif

Un Complexe Simplicial Génératif est un mélange de simplexes génératifs.

Soit π_k^d la proportion du simplexe S_k^d dans le modèle de mélange, g_k^d sa densité de probabilité, D la dimension maximale d'un simplexe dans le modèle, N_d , le nombre de simplexes de dimension d et σ^2 la variance (la même pour chaque simplexe). Alors le modèle de mélange du CSG p est défini par :

$$p(x) = \sum_{d=0}^{D} \sum_{k=1}^{N_d} \pi_k^d g_k^d(x, \sigma^2) \tag{3.3}$$

$$\forall (k,d), \pi_k^d \geq 0 \tag{3.4}$$

$$\sum_{d=0}^{D} \sum_{k=1}^{N_d} \pi_k^d = 1 \tag{3.5}$$

Le CSG a une double nature : en tant que complexe simplicial, c'est un objet géométrique plongé dans l'espace des données et son homologie simpliciale peut être calculée, en tant que modèle génératif, c'est un objet statistique. En maximisant la vraisemblance (une information statistique) à l'aide de l'algorithme Expectation-Maximization, on peut obtenir l'information topologique à partir de l'objet topologique lié.

3.2 Estimation des paramètres du complexe simplicial génératif

3.2.1 Le principe de l'algorithme Expectation-Maximization

Soit un jeu de données observées $\{x_n \in \mathbb{R}^D\}_{n=1}^N$, un modèle de densité p de paramètres θ, l'algorithme Expectation-Maximization (EM) permet de maximiser la vraisemblance $\mathcal{L}$ du modèle par rapport aux données observées :

$$\mathcal{L}(\theta, \mathbf{x}) = \prod_{n=1}^{N} p(x_n; \theta), \tag{3.6}$$

ou de manière identique la log-vraisemblance :

$$\log \mathcal{L}(\theta, \mathbf{x}) = \sum_{n=1}^{N} \log p(x_n; \theta), \tag{3.7}$$

Si on note Θ l'ensemble des modèles possibles pour θ, le but de EM est alors de trouver l'estimateur du maximum de vraisemblance :

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \mathcal{L}(\theta, \mathbf{x}). \tag{3.8}$$

Introduite par Fisher (1922), la méthode du maximum de vraisemblance a une solution analytique dans certains cas simples : données issues d'une unique distribution exponentielle ou gaussienne. Dans ce cas-là, comme dans tout problème continu et simple de maximisation, une simple dérivation de la vraisemblance permet de trouver son maximum. Pour des modèles de mélanges gaussiens, il n'existe pas de solution analytique (Hasselblad, 1966). Outre les méthodes quasi-bayésiennes (Hamilton, 1991), ou les descentes de gradients (Helmbold et al., 1997), la méthode la plus couramment utilisée est l'algorithme EM (Dempster et al., 1977).

3.2.1.1 L'algorithme EM

On suppose que les données observées $\mathbf{x} \in \mathbb{R}^D$ ne sont qu'une information partielle, l'information totale étant contenue dans les données complétées $\mathbf{x}_c = (\mathbf{x}, \mathbf{z})$, $\mathbf{z} \subset \mathcal{Z} \in \mathbb{R}^N$ étant les données manquantes.

En marginalisant sur les données manquantes supposées continues, on obtient une nouvelle expression de la log-vraisemblance :

$$\mathcal{L}(\theta, \mathbf{x}) = \sum_{n=1}^{N} \log(\sum_{\mathcal{Z}} p(x_n, z_n; \theta)). \tag{3.9}$$

Dans le cas général, on introduit une distribution quelconque r sur les données manquantes :

$$\begin{aligned}
\log \mathcal{L}(\theta, \mathbf{x}) &= \log \int p(\mathbf{x}, \mathbf{z}; \theta) \mathrm{d}\mathbf{z} && (3.10)\\
&= \log \int r(z) \frac{p(\mathbf{x}, \mathbf{z}; \theta)}{r(z)} \mathrm{d}\mathbf{z} && (3.11)\\
&\geq \int r(z) \log \frac{p(\mathbf{x}, \mathbf{z}; \theta)}{r(z)} \mathrm{d}\mathbf{z} \text{ d'après l'inégalité de Jensen} && (3.12)
\end{aligned}$$

L'inégalité de Jensen reposant sur une inégalité de convexité. On pose alors :

$$\int r(z) \log \frac{p(\mathbf{x}, \mathbf{z}; \theta)}{r(z)} \mathrm{d}\mathbf{z} = \mathcal{F}(r(z_1), ..., r(z_M), \theta) \tag{3.13}$$

L'algorithme EM cherche à maximiser $\mathcal{F}$ plutôt que la vraisemblance en elle-même, mais $\mathcal{F}$ est une borne inférieure de la vraisemblance, donc ce faisant, on s'approche du maximum de vraisemblance.

Il se décompose en deux étapes Expectation notée E et Maximization M. Dans la première, on cherche à maximiser $\mathcal{F}$ par rapport à r pour θ fixé, dans la seconde, inversement on cherche à maximiser $\mathcal{F}$ par rapport θ, les $r(\mathbf{z})$ étant fixés. Ces deux étapes peuvent être ensuite répétées plusieurs fois jusqu'à convergence de l'algorithme.

3.2.1.2 Propriétés

La principale propriété de l'algorithme EM est que l'on augmente la vraisemblance à chaque étape. Cependant la vraisemblance n'est généralement pas une fonction convexe, et il peut donc exister une multitude de maxima locaux. Le fait d'être "piégé" dans un maximum local dépend des paramètres initiaux (Boyles, 1983). La première intuition est de lancer plusieurs fois l'algorithme complètement mais avec des paramètres initiaux différents, puis de conserver celui qui renvoie la meilleur valeur de la vraisemblance. Cependant exécuter plusieurs fois EM est très coûteux en temps et d'autres stratégies ont été mises en place : notamment le *short EM* (Biernacki et al., 2003), qui reprend cette idée d'initialisations multiples, mais n'exécute l'algorithme que sur un petit nombre d'itérations, le seul qui sera exécuté jusqu'au bout étant celui ayant la meilleure vraisemblance après la version *short*. Toujours pour éviter les maxima locaux, Celeux and Diebolt (1985), puis Celeux and Govaert (1992) proposent une variante stochastique, qui intercale une étape stochastique entre les étapes E et M, où les classes sont tirées aléatoirement suivant les

distributions conditionnelles calculées à l'étape E. Dans le même article, Celeux et Govaert présentent Classification EM, une version *winner take all* de l'algorithme EM, puisque les données manquantes sont définies comme leur *maximum a posteriori*. Ils proposent d'ailleurs d'appliquer CEM après plusieurs itérations de SEM, puisqu'un des critères classiques d'arrêt pour EM (quand les classes ne changent plus entre deux itérations) ne peut pas être appliqué, puisqu'elles sont tirées aléatoirement à chaque étape.

3.2.2 L'algorithme EM pour le CSG

Soit un jeu de données $\mathbf{x}$, N_0 prototypes $\mathbf{w}$ que l'on supposera *correctement positionnés* par rapport aux données et $CD(\mathbf{w}, E)$, le complexe de Delaunay calculé à partir des $\underline{w}$, la densité de probabilité qui lui est associé est alors :

$$p(x;\theta|CD(\underline{w},E)) = \sum_{d=0}^{D}\sum_{k=1}^{N_d} \pi_k^d g_k^d(x,\sigma^2). \tag{3.14}$$

Les paramètres inconnus θ de ce modèle sont les probabilités de chaque groupe π, dans le modèle et σ, la variance que l'on suppose isotropique et identique pour toutes les composantes. On va chercher à estimer θ en maximisant la vraisemblance de notre modèle avec l'algorithme EM.

3.2.2.1 Etape E

Rappelons la borne minimale obtenue précédemment grâce à l'inégalité de Jensen :

$$\log \mathcal{L}(\theta,\mathbf{x}) \geq \int r(z) \log \frac{p(\mathbf{x},\mathbf{z};\theta}{r(z)} \mathrm{d}\mathbf{z}. \tag{3.15}$$

Dans le cadre discret où nous nous plaçons :

$$\int r(z) \log \frac{p(\mathbf{x},\mathbf{z};\theta}{r(z)} \mathrm{d}\mathbf{z} = \sum_{n=1}^{N}\sum_{d=0}^{D}\sum_{k=1}^{N_d} r(z_{kn}^d) \log\left(\frac{p(x_n, z_{kn}^d;\theta)}{r(z_{kn}^d)}\right) \tag{3.16}$$

en introduisant la donnée manquante $\{z_{kn}^d; d = 0, 1, ..., D; k = 1, ..., N_d\}$ qui correspond au composant ayant généré x_n. La densité r est inconnue, mais plus important encore, l'inégalité est valable pour toute densité proposée pour r. On peut démontrer facilement que cette borne est atteinte en posant $r(z_{kn}^d) = p(z_{kn}^d|x_n;\theta)$.

L'étape E devient donc :

$$\tilde{z}_{kn}^d \leftarrow p(z_{kn}^d|x_n;\theta) = \frac{\pi_k^d g_k^d(x_n;\sigma^2)}{\sum_{l=0}^{D}\sum_{j=1}^{N_l} \pi_j^l g_j^l(x_i;\sigma^2)} \tag{3.17}$$

3.2.2.2 Etape M

En utilisant $p(x_n, z_{kn}^d) = p(x_n|z_{kn}^d)p(z_{kn}^d)$, on peut écrire la borne inférieure :

$$\sum_{n=1}^{N}\sum_{d=0}^{D}\sum_{k=1}^{N_d} r(z_{kn}^d)\log(p(x_n|z_{kn}^d) + \log(p(z_{kn}^d) - \log r(z_{kn}^d)). \tag{3.18}$$

Elle peut se décomposer en trois grandeurs H, G et I avec :

$$H(r) = -\sum_{n=1}^{N}\sum_{d=0}^{D}\sum_{k=1}^{N_d} r(z_{kn}^d)\log(r(z_{kn}^d)), \tag{3.19}$$

$$G(r,\theta) = \sum_{n=1}^{N}\sum_{d=0}^{D}\sum_{k=1}^{N_d} r(z_{kn}^d)\log(p(z_{kn}^d)), \tag{3.20}$$

$$I(r,\theta) = \sum_{n=1}^{N}\sum_{d=0}^{D}\sum_{k=1}^{N_d} r(z_{kn}^d)\log(p(x_n|z_{kn}^d)), \tag{3.21}$$

On cherche à maximiser la somme de ces trois grandeurs uniquement par rapport aux paramètres $\theta = (\pi, \sigma^2)$. Or H est une grandeur indépendante de ces paramètres, donc seul G et I pourront être optimisés.

Etape M pour les proportions Les π doivent vérifier la contrainte :

$$\sum_{d=0}^{D}\sum_{k=1}^{N_d} \pi_k^d = 1 \tag{3.22}$$

On est dans le cadre d'une maximisation sous contrainte. L'outil mathématique adéquat dans ce cas est le multiplicateur de Lagrange. Seul le terme G dépend de $\pi_k^d = p(z_k^d)$, on souhaite donc maximiser :

$$G(\theta) - \lambda(\sum_{d=0}^{D}\sum_{k=1}^{N_d} \pi_k^d - 1) \tag{3.23}$$

En dérivant par rapport à π_k^d, on cherche à résoudre :

$$-\sum_{n=1}^{N}\frac{r(z_{kn}^d)}{\pi_k^d} + \lambda = 0 \tag{3.24}$$

On multiplie alors par π_k^d, puis on somme sur tous les indices k et d :

$$\sum_{n=1}^{N}\sum_{d=0}^{D}\sum_{k=1}^{N_d} r(z_{kn}^d) + \lambda \sum_{d=0}^{D}\sum_{k=1}^{N_d} \pi_{kn}^d = 0 \tag{3.25}$$

Or $\sum_{d=0}^{D}\sum_{k=1}^{N_d} r(z_{kn}^d) = 1$ et $\sum_{d=0}^{D}\sum_{k=1}^{N_d} \pi_{kn}^d = 1$, d'où l'on tire $\lambda = N$, puis :

$$\pi_k^d = \frac{1}{N}\sum_{n=1}^{N} r(z_{kn}^d) = \frac{1}{N}\sum_{n=1}^{N} z_{in}^d \tag{3.26}$$

Etape M pour σ^2 Seul le terme I dépend de σ^2. On peut décomposer I comme une somme de composante I_d :

$$I_d = \sum_{n=1}^{N}\sum_{k=1}^{N_d} r(z_{kn}^d)\log(p(x_n|z_{kn}^d)) \tag{3.27}$$

Pour $d = 0$, on est dans le cas des modèles de mélange gaussien classique :

$$\frac{\partial I_0(r,\theta)}{\partial\sigma} = \frac{-D}{\sigma}\sum_{n=1}^{N}\sum_{k=1}^{N_0} r(z_{kn}^0) + \frac{\|x_n - w_k\|^2}{\sigma^3}\sum_{n=1}^{N}\sum_{k=1}^{N_0} r(z_{kn}^0) \tag{3.28}$$

Pour $d \geq 1$, $p(x_n|z_{kn}^d)$ est définie par une intégrale multiple. Gaillard (2008) introduit une nouvelle donnée manquante q uniformément distribuée le long d'un segment gaussien. Nous procédons de même pour un simplexe gaussien S_k^d, ce qui donne :

$$\begin{aligned} I_d &= \sum_{n=1}^{N}\sum_{k=1}^{N_d} r(z_{kn}^d)\log\Big(\int_{S_k^d} f(q|z_{kn}^d)\frac{p(x_n,q|z_{kn}^d;\theta)}{f(q|z_{kn}^d)}\Big) &(3.29)\\ &\geq \sum_{n=1}^{N}\sum_{k=1}^{N_d} r(z_{kn}^d)\int_{S_k^d} f(q|z_{kn}^d)\log\Big(\frac{p(x_n,q|z_{kn}^d;\theta)}{f(q|z_{kn}^d)}\Big) &(3.30)\end{aligned}$$

On obtient encore cette inégalité grâce à l'inégalité de Jensen. On peut vérifier que cette borne est atteinte pour $f(q|z_{kn}^d) = p(q|x_k, z_{kn}^d;\theta)$.

La dérivée partielle de I_d vaut donc :

$$\frac{\partial I_d(r,\theta)}{\partial\sigma} = \frac{-D}{\sigma}\sum_{n=1}^{N}\sum_{k=1}^{N_d} r(z_{kn}^d) + \frac{\int_{S_k^d} f(q|z_{kn}^d)\,\|x_n - w_k\|^2}{\sigma^3 \mathrm{d}q}\sum_{n=1}^{N}\sum_{k=1}^{N_d} r(z_{kn}^d) \tag{3.31}$$

On pose :

$$V_{kn}^d = \begin{cases} \|x_n - w_k\|^2 \text{ if d} = 0 \\ \int_{S_k^d} f(q|z_{kn}^d) \|x_n - q\|^2 \, \mathrm{d}q \end{cases} \tag{3.32}$$

Finalement, on exprime la dérivée partielle de I :

$$\frac{\partial I(\theta)}{\partial \sigma} = \frac{-D}{\sigma} \sum_{n=1}^{N} \sum_{d=0}^{D} \sum_{k=1}^{N_d} r(z_{kn}^d) + \frac{1}{\sigma^3} \sum_{n=1}^{N} \sum_{d=0}^{D} \sum_{k=1}^{N_d} r(z_{kn}^d) V_{kn}^d \tag{3.33}$$

$$= \frac{-MD}{\sigma} + \frac{1}{\sigma^3} \sum_{n=1}^{N} \sum_{d=0}^{D} \sum_{k=1}^{N_d} r(z_{kn}^d) V_{kn}^d \tag{3.34}$$

On annule la dérivée, et on obtient :

$$\sigma^2 = \frac{1}{ND} \sum_{n=1}^{N} \sum_{d=0}^{D} \sum_{k=1}^{N_d} \tilde{z}_{kn}^d V_{kn}^d. \tag{3.35}$$

La mise à jour de σ est plus coûteuse en temps de calcul que la mise à jour des π_k^d, puisque le calcul de V_{kn}^d nécessite de connaître la densité de probabilité de chaque simplexe génératif. Elle ne sera donc pas réalisée à chaque itération de l'algorithme EM. Arbitrairement, nous décidons de ne faire cette mise à jour qu'une fois toutes les dix itérations de l'algorithme EM.

3.2.3 Le positionnement des sommets

Bien que possible, une mise à jour de la position des sommets du complexe simplicial par algorithme GEM n'a pas été envisagée pour deux raisons liées à la complexité en temps de calcul. Cette mise à jour implique de recalculer toutes les proportions π_k^d, donc de faire une étape d'intégration par Monte-Carlo pour chaque simplexe du modèle. C'est une étape nécessaire aussi pour la variance σ^2. Si c'était la seule contrainte pour la position des sommets $\mathbf{w}$, il n'y aurait pas de perte de temps si elle était faite simultanément à la mise à jour de la variance. En revanche, contrairement à la mise à jour des π_k^d et de σ^2, un nouveau positionnement des sommets implique de calculer un nouveau complexe de Delaunay, car il dépend de la position des $\mathbf{w}$.

Le calcul d'un nouveau complexe de Delaunay pose des problèmes au-delà de la simple question du temps de calcul. Certaines arêtes peuvent apparaître alors qu'elles n'existaient pas avant, et inversement, certaines peuvent disparaître. Et ceci indépendamment de la proportion plus ou moins forte qu'elles pouvaient avoir dans le modèle après plusieurs itérations successives de l'algorithme EM. Il faudrait donc par la suite repartir avec un complexe simplicial ré-initialisé : tous les π_k^d sont égaux.

Pour toutes ces raisons, la mise à jour de la position des sommets n'a pas été étudiée expérimentalement. Néanmoins, nous faisons l'hypothèse qu'un échantillonnage suffisant de la variété, ainsi que des prototypes "bien positionnés", par exemple en utilisant un GMM classique dans une étape d'initialisation, définissent un complexe simplicial qui contient un complexe qui a la même homologie que la variété sous-jacente.

Nous proposons une initialisation par modèle de mélange gaussien classique. Les contraintes apportées au mélange sont : une même variance et une même proportion pour tous les prototypes. En effet, si les proportions π_k^d sont laissées libres, il est possible d'obtenir un GMM dont un composant seul explique une zone dense en données, alors que nous pensons souhaitables que la distribution des $\mathbf{w}$ suive celles des $\mathbf{x}$.

En utilisant cette méthode, on s'assure aussi de placer les prototypes au coeur des données, et non pas dans une région de l'espace où il y aurait des données aberrantes (*outliers*), comme cela est possible par un sous-échantillonnage aléatoire des données ou l'algorithme *maxmin*.

3.2.3.1 Algorithme

Algorithm 2 Principe du CSG

Require: $\mathbf{x}, N_0, D'$
1: $(\mathbf{w}, \sigma_i nit) \leftarrow GMM(\mathbf{x}, N_0)$
2: $CSG \leftarrow Delaunay(\mathbf{w}, D')$
3: $CSG \leftarrow EM(CSG, \sigma_{init})$
4: **for** $d = 1, ..., D'$ **do**
5: **for** $k = 1, ..., N_d$ **do**
6: **if** $\pi_k^d == 0$ **then**
7: $CSG \leftarrow CSG \setminus \{S_k^d\}$
8: **end if**
9: **end for**
10: **end for**
Ensure: CSG

Dans un premier temps, on positionne les prototypes $\mathbf{w}$ qui seront les sommets du GSC. Ensuite on construit le D'-squelette de Delaunay qui est le GSC initial. Chaque simplexe génératif du GSC se voit attribuer une proportion. Ils sont initialisés équiprobables. La variance est initialisée à la variance apprise par GMM. On met à jour les proportions et la variance par plusieurs itérations de l'algorithme EM. Après convergence vers une solution stable, certaines proportions sont nulles ou inférieures au seuil de précision de Matlab (2^{-52}). Les simplexes associés sont retirés du CSG.

Le complexe simplicial obtenu est transformé en complexe simplicial combinatoire au format accepté par l'algorithme *Javaplex*, qui en extrait ensuite les nombres de Betti.

3.3 Validation expérimentale du modèle

Après la construction théorique du modèle, certains choix au niveau algorithmique ont dû être faits. Nous avons décidé de les valider par la pratique, en les testant sur un grand nombre de cas simples que nous maîtrisions. Il en est ainsi de l'estimation de la densité de probabilité générée par les simplexes génératifs, de l'élagage du complexe simplicial, de la sélection du nombres de sommets du complexe de Delaunay initial.

3.3.1 Génération des simplexes pour la validation

Les tests seront faits sur des simplexes de dimensions 2, 3, 4 et 5. Ces simplexes sont générés aléatoirement. Leurs sommets sont tirés de manière uniforme dans un hypercube unité correspondant à leur dimension (carré pour le triangle, cube pour le tétraèdre, etc). Ils sont ensuite redimensionnés par homothétie de manière à ce que leur plus grand côté soit de longueur 1. Le but de ce procédé est de s'assurer de traiter une grande variété de simplexes. Un triangle plat posera plus de problème au CSG qu'un triangle équilatéral, puisqu'il aura plus de chance d'être confondu avec un segment, un tétraèdre plat pourra passer pour un triangle, voire un segment s'il est très allongé. Cela pose un problème au niveau de la dimension intrinsèque des données que l'on peut évaluer avec notre modèle, même si, comme on pourra le voir par la suite, cela ne change pas forcément la topologie des données.

Le fait qu'un simplexe soit "plat" sera quantifié par sa plus petite hauteur.

A partir de chacun des simplexes générés, et d'un écart-type lui aussi tiré aléatoirement de manière uniforme dans le segment $[0; 0, 2]$, deux jeux de données sont tirés : le premier correspond à l'intérieur du simplexe, le second au complexe simplicial constitué par l'enveloppe convexe du simplexe privée de son intérieur de dimension maximale.

3.3.2 Estimation de la densité de probabilité d'un simplexe gaussien par une méthode de Monte-Carlo

L'estimation de la densité de probabilité correspondant à un simplexe génératif nécessite le calcul de l'intégrale d'une gaussienne sur un simplexe. Si l'intégrale simple d'une fonction gaussienne sur un segment peut être exprimée par la fonction erf, fonction présente dans des logiciels tels que Matlab, une expression analytique n'est plus possible à partir d'une intégrale double. Une méthode d'intégration de Monte-Carlo a donc été choisie pour calculer la valeur de cette intégrale.

3.3.2.1 La méthode d'échantillonnage pour l'intégration par Monte-Carlo

On peut illustrer les méthodes de Monte-Carlo par l'intégrale de Riemann, qui est la première approche donnée pour l'intégration. On cherche à intégrer une fonction f continue

sur un segment $[a, b]$. Pour cela, on va l'approximer par une forme discrète appelée somme de Riemann S_n :

$$S_n = \frac{b-a}{n} \sum_{k=1}^{n} f(a + k\frac{b-a}{n}) \tag{3.36}$$

Cela revient à subdiviser le segment $[a, b]$ en n parties égales, et à approximer la fonction sur chaque subdivision par un rectangle. On peut d'ailleurs définir ces subdivisions comme une suite :

$$x_k = a + k\frac{b-a}{n} \text{ avec } 0 \leq k \leq n \tag{3.37}$$

ce qui donne pour S_n :

$$S_n = \sum_{k=1}^{n} (x_k - x_{k-1}) f(x_k). \tag{3.38}$$

Intuitivement, on comprend que plus l'échantillonnage (x_k) est grand, meilleure sera la précision de l'approximation. C'est la définition de l'intégrale de Riemann :

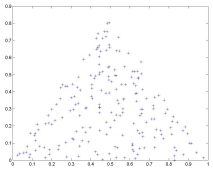
$$\lim_{n \to +\infty} S_n = \int_a^b f(t)\mathrm{d}t \tag{3.39}$$

Ceci n'est qu'un cas particulier de la définition de l'intégrale de Riemann, la condition nécessaire étant que le pas d'intégration tende vers 0 pour que la limite de la somme soit égale à la valeur de l'intégrale. L'idée d'une méthode d'intégration de Monte-Carlo consiste à échantillonner aléatoirement le segment à intégrer, puisqu'il n'y a aucune obligation de régularité dans le pas. Il est aisé de voir que l'on peut généraliser cette méthode à des dimensions supérieures, en échantillonnant uniformément le volume sur lequel doit être intégrée la fonction f.

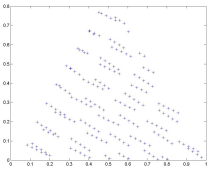
Soit un ensemble $V \subset \mathbb{R}^N$, $|V|$ son volume, une fonction $f : \mathbb{R}^N \longrightarrow \mathbb{R}$. On cherche à calculer $\int_V f(x)\mathrm{d}x$. La méthode de Monte-Carlo consiste à tirer un échantillon $(x_1, ..., x_n)$ uniformément réparti dans V et approximer l'intégrale :

$$\int_V f(x)\mathrm{d}x \approx |V| \sum_{i=1}^{n} f(x_i) \tag{3.40}$$

Cette méthode est particulièrement utile quand on ne connaît pas d'expression analytique de l'intégrale, mais qu'on peut calculer assez facilement les valeurs de f pour les points de l'échantillon. C'est exactement le cas dans lequel se trouve le simplexe gaussien : calculer une gaussienne en un point est possible, en revanche l'intégrale multiple d'une gaussienne sur un volume n'a pas d'expression analytique.



(a) MC



(b) QMC

FIGURE 3.2 – La figure 3.2a représente un échantillonnage aléatoire d'un triangle équilatéral de côté 1. La figure 3.2b représente le même triangle mais échantillonné avec une suite de Sobol. On perçoit un biais dans l'échantillonnage par la méthode Quasi-Monte Carlo.

Outre la méthode *régulière* et la méthode aléatoire, on trouve les méthodes appelées *Quasi-Monte Carlo* (Lemieux, 2009). Partant du constat qu'une séquence de nombres pseudo-aléatoire peut sur-échantillonner une région de l'espace, et en sous-échantillonner une autre, on s'intéresse à une mesure de la qualité de la répartition des points dans un volume : la discrépance. Certaines suites à faible discrépance , comme les suites de Sobol, sont étudiées dans le cadre des méthodes de Monte-Carlo (Zaremba, 1968). Il a été démontré que dans le meilleur des cas, ces méthodes ont une erreur qui décroît en $O(1/N)$, contre une erreur qui décroît en $O(1/\sqrt{N})$ pour la méthode aléatoire (Asmussen and Glynn, 2007). Il a aussi été montré que ces méthodes avaient toujours de meilleures performances que les méthodes aléatoires (Morokoff and Caflisch, 1995). Cependant ces méthodes sont conçues pour des intégrations sur des pavés, des produits de segments, et non pas sur des simplexes. L'intégration par Quasi-Monte-Carlo sur un simplexe est beaucoup moins documentée (Pillards and Cools, 2005).

Pour comparer la méthode régulière (Reg), la méthode Monte-Carlo (MC) et la méthode Quasi-Monte-Carlo (QMC), 100 échantillons aléatoires de taille $M \in \{10, 50, 100\}$ sont générés à partir de simplexes de dimension $d \in \{2, 3, 4\}$ dont les sommets ont aussi été générés aléatoirement suivant la méthode décrite en 3.3.1. On calcule une vraisemblance $\hat{L}$ de référence à partir du modèle qui a généré les données et une méthode de Monte-Carlo aléatoire échantillonnée avec un grand nombre de points ($\binom{d+50}{d}$, ce qui correspond à une subdivision sur chaque dimension du simplexe en 50 parties). Ensuite, avec un nombre plus raisonnable de sommets ($\binom{d+10}{d}$, ce qui correspond à une subdivision de 10 sur chaque dimension du simplexe), on calcule L_{MC} et L_{QMC} qui correspondent respectivement à une vraisemblance calculée avec le modèle qui a généré les données et une méthode de Monte Carlo, et la même chose mais avec une méthode de Quasi-Monte Carlo.

Le tableau 3.1 met en évidence que la meilleure méthode pour évaluer l'intégrale multidimensionnelle sur un simplexe d'une gaussienne est la méthode qui consiste à l'échantillonner régulièrement (Reg). C'est donc celle-ci qui sera retenue dans l'algorithme du Complexe Simplicial Génératif.

	Reg	MC	QMC
d=2	**0.20**	6.19	1.24
d=3	**0.21**	5.33	0.82
d=4	**0.13**	3.67	0.81

TABLE 3.1 – Reg : régulier, MC : Monte-Carlo, QMC : Quasi-Monte-Carlo. On compare les différentes méthodes d'échantillonnage pour l'intégration sur un simplexe. Le tableau représente l'écart quadratique moyen à la vraisemblance de référence, moyenné sur les différentes valeurs de M, dimension par dimension. Plus la valeur est faible, plus l'estimation est proche de la valeur de référence.

3.3.2.2 La taille de l'échantillon pour l'intégration par Monte-Carlo

Un autre paramètre intimement lié à la méthode d'échantillonnage lors du calcul de l'intégrale par Monte-Carlo est le nombre de points N_e à échantillonner à l'intérieur du simplexe. Bien entendu, plus ce nombre sera grand, et meilleure sera la précision obtenue sur la valeur de l'intégrale. Plus grand aussi sera le temps de calcul dont la complexité est en $O(N_e)$, alors que cette étape d'intégration est déjà une des plus coûteuses en temps de calcul. Elle n'est d'ailleurs pas exécutée à chaque passage de l'algorithme EM. Encore une fois, il s'agit d'un compromis entre précision et temps de calcul.

Il y a une deuxième problématique liée à la taille de l'échantillon pour l'intégration par Monte-Carlo : un tétraèdre nécessite plus de points qu'un triangle pour être échantillonné, qui nécessite lui-même plus de points qu'un segment.

Comme on peut le voir sur la figure 3.3, pour qu'un triangle soit échantillonné avec la même précision qu'un segment, le nombre de points tirés dans le triangle doit correspondre au n-ième nombre triangulaire, si n est le nombre de points qui ont été tirés dans le segment. Un raisonnement similaire est appliqué au tétraèdre, vu comme un empilement de triangles de plus en plus petits, et ainsi de suite pour les dimensions supérieures.

Pour la dimension 2, ces nombres portent le nom de nombres triangulaires, nombres tétraèdriques en dimension 3 et pentatopiques en dimension 4. Pour les dimensions suivantes, on les appelle nombres r-topiques, où r est le nombre de sommets du simplexe.

On trouve le n-ième nombre r-topique avec la formule suivante :

$$P_r(n) = \binom{n+r-1}{r} = \frac{(n+r-1)(n+r-2)...(n+1)n}{r!}$$

A titre d'exemple, si un segment est échantillonné par 10 points, il en faut 55 pour un triangle, 220 pour un tétraèdre et 715 pour un pentatope. Comme on le voit, le nombre de points à échantillonner augmente très vite avec la dimension du simplexe. Cependant ce nombre ne dépend que de la dimension du simplexe et de la précision de l'échantillonnage, et pas de la dimension de l'espace ambiant. Même si la dimension ambiante peut être très

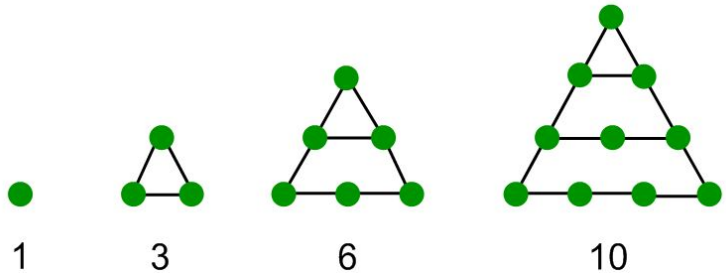


FIGURE 3.3 – Les quatre premiers nombres triangulaires, $n = 1, 2, 3, 4$, $r = 2$

	n=5	n=10	n=15
d=2	2.10	0.21	0.20
d=3	1.87	0.19	0.17
d=4	0.92	0.13	0.13

TABLE 3.2 – On compare les différentes taille d'échantillon pour l'intégration sur un simplexe. Le tableau représente l'écart quadratique moyen à la vraisemblance de référence, dimension par dimension.

grande, il y a de fortes chances pour que la dimension intrinsèque de la variété recherchée reste faible, et dépasse rarement la dimension 3.

La méthodologie de la partie précédente a été suivie, sauf qu'au lieu de comparer les méthodes d'échantillonnage, l'échantillonnage régulier a été choisi, et c'est le nombre d'échantillons que l'on a fait varier. Pour chaque simplexe de dimension $d \in \{2, 3, 4\}$, ont été comparés les nombres d-topiques correspondant à $n \in \{5, 10, 15\}$.

Choisir le 10^e nombre r-topique est le compromis précision-temps de calcul que nous choisissons de faire, puisque le gain en précision est négligeable au-delà.

3.3.3 L'élagage du complexe simplicial

3.3.3.1 Le critère statistique BIC

Le principe du rasoir d'Occam stipule qu'il ne faut pas ajouter de multiplicités non nécessaire dans notre modèle, ce qui nous incite à utiliser un critère de parcimonie comme le critère BIC.

La vraisemblance n'est pas bornée pour les modèles gaussiens (Kiefer and Wolfowitz, 1956; Biernacki and Chrétien, 2003; Ingrassia and Rocci, 2011). En effet, un prototype positionné sur une donnée, et une variance σ^2 qui tendrait vers 0 feraient tendre la vraisemblance vers l'infini. Toutefois en imposant certaines contraintes à notre modèle (comme une variance unique pour tous les composants, ce qui est le cas du CSG), on peut bor-

ner la vraisemblance (Day, 1969; Hathaway, 1985). Cependant, la vraisemblance reste une fonction croissante du nombre de paramètres. Elle n'est donc pas la grandeur adéquate à maximiser si l'on veut satisfaire au critère de parcimonie, et décider du nombre de sommets et de simplexes dans le complexe.

AIC (Akaike Information Criterion) (Akaike, 1974) et BIC (Bayesian Information Criterion) (Schwarz, 1978) sont deux critères d'évaluation de modèles statistiques. Ils sont fondés sur la vraisemblance, mais comportent un terme de pénalisation, noté $\varphi(M, \nu_k)$, où M est le nombre de données et ν_k le nombre de paramètres dans le modèle $\mathcal{M}_k$. Ce terme est aussi une fonction croissante du nombre de paramètres, et de données, mais il est retiré à la vraisemblance, si bien qu'il existe un nombre de paramètres à partir duquel le gain en vraisemblance est inférieur à la perte entraînée par la pénalisation.

$$AIC(\mathcal{M}_k, \mathbf{x}) = \log \mathcal{L}(\hat{\theta}_k; \mathbf{x}) - \nu_k \tag{3.41}$$

$$BIC(\mathcal{M}_k, \mathbf{x}) = \log \mathcal{L}(\hat{\theta}_k; \mathbf{x}) - \frac{\nu_k}{2} \log(M) \tag{3.42}$$

3.3.3.2 Application au cas du CSG

Dans le CSG, le nombre de paramètres est directement lié au nombre de sommets et de simplexes dans le modèle. Contrairement à l'approche Witness Complex par exemple, qui part des arêtes puis construit dimension par dimension les simplexes de dimension supérieure, notre approche contient déjà tous les simplexes initialement. On cherche plutôt à retirer les simplexes superflus. Ce modèle est optimisé avec l'algorithme EM : il est donc optimal du point de vue de la vraisemblance à la fin de l'exécution de l'algorithme (du moins s'il ne tombe pas dans un optimum local). Si on retire un simplexe de ce modèle, on ne peut par conséquent que diminuer la vraisemblance. En revanche, on simplifie aussi le modèle, on améliore donc potentiellement les critères AIC ou BIC.

En toute rigueur, il faudrait tester tous les modèles possibles en retirant ou non chacun des simplexes. Si le modèle compte N_s simplexes en tout, il faudrait tester 2^{N_s} modèles, chacun optimisé par algorithme EM. C'est bien évidemment trop coûteux en temps. En revanche la première exécution de l'algorithme EM sur le complexe total donne une information sur les proportions π_i^d que l'on peut utiliser. On les re-numérote de 1 à N_s de telle façon que :

$$\pi_1 > ... > \pi_{N_s} \tag{3.43}$$

On note alors CSG_k le complexe simplicial composé des k simplexes correspondants à $\pi_1, ..., \pi_k$, CSG_0 est alors $\emptyset$, et CSG_{N_s} le complexe complet. Le tout formant une suite de N_s modèles emboîtés à tester (et non plus 2^{N_s}). Outre le gain en temps de calcul qu'apporte cette heuristique, elle n'est pas dénuée de fondements : en effet, dans le cas où le "bon" modèle appartient à une série de modèles emboîtés, la convergence du critère BIC vers ce modèle est assurée (Burnham and Anderson, 2002). D'ailleurs, bien que le modèle ne soit

pas correctement défini pour les modèles gaussiens, il donne de bons résultats en pratique (Roeder and Wasserman, 1997; Biernacki and Govaert, 1999; Fraley and Raftery, 2002).

Comme vu dans la section 2.5, le complexe de Delaunay d'un ensemble de points ne dépend que de la position de ses points. Une fois que les sommets ont été positionnés par GMM (section 3.2.3), la structure sous-jacente sur laquelle va reposer le CSG est complètement définie, du plus petit simplexe au plus grand. L'apport du CSG par rapport au simple complexe de Delaunay, est justement de permettre de choisir grâce aux proportions π_i^d quel simplexe, parmi tous ceux du complexe de Delaunay, sont pertinents et doivent rester dans la structure finale.

Un CSG est un modèle de mélange (section 3.1.2.2) dont chaque composante élémentaire est un simplexe génératif (section refsimplexegeneratif). Une proportion π_k^d est donc associée à chacun de ces simplexes. Il faut définir un critère pour sélectionner quel simplexe doit rester dans la structure finale, en fonction de son apport à la compréhension du jeu de données. Plus simplement, quel simplexe doit voir sa proportion π_k^d mise à 0.

Puisque le CSG est un modèle de mélange, le choix de l'algorithme Expectation-Maximization était tout indiqué pour optimiser ses paramètres. Cet algorithme brut sera la première version testée. On y ajoute un critère d'élagage qui force à mettre exactement à 0 les proportions trop faibles : à la fin de l'exécution de l'algorithme, si $\pi_k^d < 10^{-12}, \pi_k^d = 0$. Ce seuil a été fixé arbitrairement par rapport à la précision de MATLAB. En effet, même avec une proportion très faible, tout simplexe qui aurait une proportion non nul serait considérée comme appartenant au modèle. Or il est impossible que l'algorithme mette effectivement à 0 une proportion, mais il peut toutefois lui affecter une valeur infinitésimale. L'autre algorithme testé est aussi un algorithme EM complété par une sélection du nombre de composantes à l'aide du critère statistique BIC.

3.3.3.3 Expériences

Il s'agit de vérifier quel critère permet de retrouver la bonne topologie : la topologie dépend uniquement de la présence ou non d'un simplexe dans le modèle, et pas de la proportion plus ou moins forte que ce simplexe peut avoir dans le modèle. On considère comme un succès de l'algorithme le fait de trouver exactement les proportions nulles du modèle qui a généré les données, sans condition d'exactitude sur les valeurs des proportions non nulles. L'algorithme sera testé avec ou sans le critère BIC sur l'intérieur et l'enveloppe convexe de simplexes, toujours générés suivant la méthode décrite en introduction.

Malgré les résultats faibles de l'algorithme EM seul sur les enveloppes de simplexes(figure 3.5a), c'est néanmoins celui-ci que nous allons utiliser par la suite. En effet, retirer un simplexe du modèle est l'action qui modifie le plus la topologie. Il est plus gênant que l'utilisation du critère BIC retire des simplexes quand il ne le faut pas, que de garder des simplexes qui devraient être retirés comme le fait l'algorithme EM. Il était important de vérifier tout de même que l'algorithme EM est capable de retirer un simplexe (c'est un mécanisme dû à la précision de calcul de Matlab mais qui revient à seuiller les proportions en dessous d'une valeur extrêmement faible).

(a) EM seul (b) EM + BIC

FIGURE 3.4 – Succès et échec d'un apprentissage correct de la topologie générée (simplexe plein de dimension 2). En abscisse, la plus petite hauteur du simplexe, en ordonnée l'écart-type avec lequel ont été générées les données. Pour un simplexe plein, l'algorithme EM seul est le plus performant avec 100 % de réussite. En effet pour un simplexe plein, il "suffit" de ne retirer aucun des composants. Le taux de succès du critère BIC est plus faible (< 50 %), ce qui montre que l'enveloppe d'un simplexe peut dominer l'intérieur de ce simplexe, même en utilisant le modèle qui a permis de générer les données.

(a) EM seul (b) EM + BIC

FIGURE 3.5 – Succès et échec d'un apprentissage correct de la topologie (enveloppe du simplexe de dimension 2). En abscisse, la plus petite hauteur du simplexe, en ordonnée l'écart-type avec lequel ont été générées les données. Cette fois-ci le critère BIC obtient de bien meilleurs résultats. On pouvait s'y attendre, puisque l'action de retirer un simplexe est plus difficile à faire pour l'algorithme EM seul. De plus, la forme du simplexe, ainsi que la variance des données peut induire plus facilement l'algorithme en erreur : une enveloppe de triangle "aplati" avec un bruit assez fort est difficilement discernable de l'intérieur de ce même triangle.

3.3.4 La sélection du nombre de sommets

La section précédente consistait à déterminer quel algorithme, de EM et EM+BIC, permet de mieux distinguer les simplexes creux ou les simplexes pleins. Un nombre de sommets différents fournirait un complexe de Delaunay différent, et donc un CSG final différent. Il s'agit maintenant de définir le nombre optimal de sommets dans le complexe de Delaunay.

Alors que dans l'étape précédente l'algorithme EM classique et le critère de vraisemblance fournissaient un bon résultat, cela ne pourra pas être le cas ici : la vraisemblance est une fonction croissante du nombre de sommets, et la meilleure vraisemblance sera donc atteinte pour un nombre de prototypes égal au nombre de données. Il faut pénaliser l'utilisation d'un grand nombre de sommets dans notre modèle. Encore une fois, nous allons étudier le critère BIC, mais aussi le critère AIC.

Ce que nous voulons vérifier est que ce modèle choisi par un critère nous renvoie la bonne topologie. Si l'on veut poser un problème topologique plus complexe que la recherche de composantes connexes qui correspond au β_0, il faut prendre une forme qui a un β_1 différent de 0. La forme la plus simple ayant un β_1 non nul est un cercle, c'est pourquoi on testera la validité du modèle sur cette figure : on vérifie que les bons nombres de Betti $\beta_0 = 1$ et $\beta_1 = 1$ sont obtenus pour le modèle choisi par les critères BIC ou AIC.

Pour cela 100 points sont tirés aléatoirement avec une probabilité uniforme le long du cercle trigonométrique, et sont bruités avec un bruit gaussien d'écart-type $0,1$. On fera varier le nombre de sommets de 8 à 20 dans le modèle.

Les figures 3.6 et 3.7 montrent respectivement les critères BIC et AIC en fonction du nombre de sommets utilisés dans le modèle. Les deux critères atteignent leur minimum pour des nombres de Betti correct, ce qui nous renseigne déjà sur le fait que la plage du nombre de sommets qui donne les bons nombres de Betti est assez grande. En revanche, le critère AIC sélectionne en moyenne 20 sommets, ce qui est beaucoup trop étant donné que l'on a que 100 individus dans l'échantillon. En effet, 20 sommets cela correspond en général (tout dépend de la position initiale des points bien entendu) à plus de 50 composantes dans le modèle initial, une fois pris en compte les segments et les surfaces, et 40 composantes dans le modèle final (un cycle a autant de sommets que d'arêtes). Soit 20×2 paramètres libres pour la position des sommets, 49 proportions libres (puisque la 50-ème est contrainte par le fait que la somme doit faire 1), et un paramètre de variance σ^2, soit 90 paramètres à estimer à partir d'un échantillon comportant 100 données : le nombre de données par rapport au nombre de paramètres ne permet pas de tirer de conclusion de manière significative.

3.3.5 Conclusion

On rappelle ici les différents choix qui ont été fait à partir des résultats des expériences précédentes :

- La méthode régulière (Reg) est retenue pour échantillonner un simplexe gaussien et calculer sa densité de probabilité avec la méthode de Monte-Carlo.
- Le 10^e nombre r-topique est retenu comme taille d'échantillonnage pour un simplexe de dimension r.
- L'élagage du complexe simplicial sera fait par l'algorithme EM seul.
- La sélection du nombre de sommets dans le modèle sera faite à l'aide du critère BIC.

K_1 et K_2 sont les bornes de l'intervalle dans lequel on va choisir le nombre de sommets. L'intervalle peut être grand en première approche : $K_1 = 4$ (qui est le minimum pour décrire l'homologie d'une sphère), par exemple, K_2 est à ajuster en fonction du nombre de données disponible, et du temps de calcul dont on dispose. En pratique, $K_2 = 40$ permettait d'avoir le minimum pour BIC dans l'intervalle étudié. Quitte à affiner ensuite l'intervalle avec ces résultats préliminaires et lancer l'algorithme plusieurs fois dans cet intervalle réduit. D' est la dimension du graphe de Delaunay que l'on construit initialement. En général $D' < D$, et pour de grandes dimensions de l'espace ambiant, on se restreindra à $D' \in \{2, 3, 4, 5\}$. La version finale de l'algorithme qui contient la sélection du nombre de composantes est

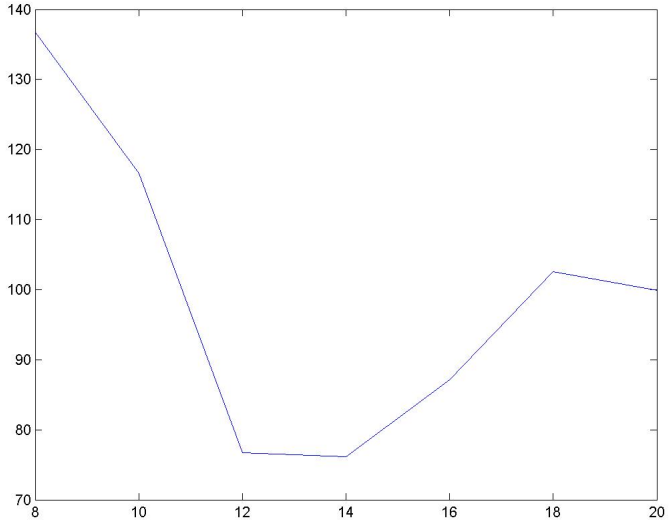


FIGURE 3.6 – Le critère BIC en fonction du nombre de sommets choisis dans le modèle GSC

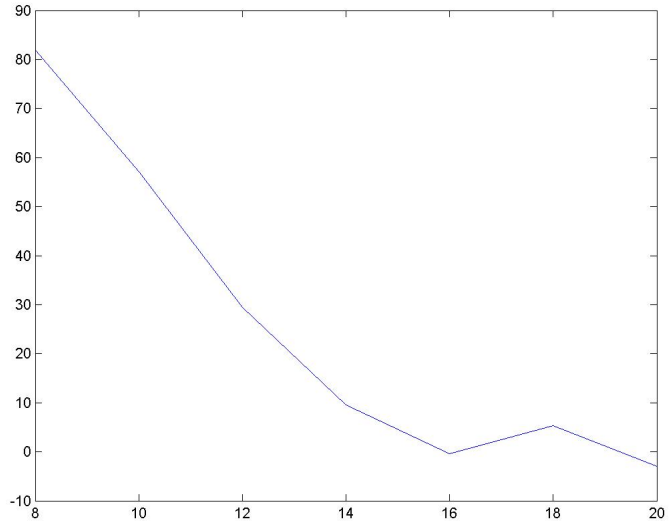


FIGURE 3.7 – Le critère AIC en fonction du nombre de sommets choisis dans le modèle GSC.

alors :

Algorithm 3 Algorithme CSG avec sélection du nombre de composantes

Require: $\mathbf{x}, K_1, K_2, D'$
  for $N_0 = K_1, ..., K_2$ **do**
    $(\underline{w}, \sigma_i nit) \leftarrow GMM(\mathbf{x}, N_0)$
    $CSG_{N_0} \leftarrow Delaunay(\underline{w}, D')$
    $CSG_{N_0} \leftarrow EM(CSG_{N_0}, \sigma_i nit)$
    for $d = 1, ..., D'$ **do**
      for $i = 1, ..., N_d$ **do**
        if $\pi_i^d == 0$ **then**
          $CSG_{N_0} \leftarrow CSG_{N_0} \setminus \{S_i^d\}$
        end if
      end for
    end for
    $bic(N_0) = BIC(CSG_{N_0}, \mathbf{x})$
  end for
  $\hat{N}_0 = \underset{k=K_1,...,K_2}{\operatorname{argmax}} \; bic(k)$
Ensure: $CSG_{\hat{N}_0}$

Nous avons introduit le modèle de Complexe Simplicial ainsi que les algorithmes qui permettent d'en optimiser les paramètres. Certains fondements ont été vérifiés empiriquement, notamment le fait que le critère BIC permet de trouver une topologie correcte.

4

Applications à l'analyse de données

Dans cette partie, on vérifiera d'abord la validité du modèle Complexe Simplicial Génératif sur des données jouets issues d'objets dont les nombres de Betti sont connus. Cette première étape est importante, puisqu'elle permet de valider le modèle sur des données qui ont une vérité terrain. Les données réelles ne possèdent pas forcément cette vérité terrain, surtout en analyse exploratoire. Il faut donc vérifier les bons résultats de l'algorithme sur des données que l'on maîtrise, pour pouvoir faire confiance à l'algorithme sur les données réelles.

4.0.6 Comparaison avec l'état de l'art : le Witness Complex

Nous avons choisi de comparer le CSG avec le Witness Complex (WitC) pour deux raisons :

1. Il répond à la même question que le CSG, en donnant les nombres de Betti d'une variété sous-jacente à un échantillon de données dans $\mathbb{R}^D$.
2. Il ne travaille pas directement sur les données (contrairement au complexe de Vietoris-Rips par exemple), mais sur des prototypes qui les résument.

4.0.6.1 Construction d'un Witness Complex

La première étape de la construction d'un Witness Complex consiste à construire un 1-squelette $W_1(\Delta)$ des prototypes. Appelons $\Delta \in \mathcal{M}_{N\times K_0}(\mathbb{R})$, la matrice qui contient les distances des N données $\mathbf{x}$ aux K_0 prototypes $\mathbf{w}$.

$$\forall a, b \in \mathbf{w}, e = [ab] \in W_1(\Delta) \Leftrightarrow \exists x_n \in \mathbf{x}, a = \underset{w \in \{w_1,...,w_{K_0}\}}{\operatorname{argmin}} \Delta(w, x_n), b = \underset{w \in \{w_1,...,w_{K_0}\} \setminus \{a\}}{\operatorname{argmin}} \Delta(w, x_n) \tag{4.1}$$

Ensuite on ajoute les triangles : $[abc] \in W_1(\Delta)$ si $[ab], [ac], [bc] \in W_1(\Delta)$, $[abcd] \in W_1(\Delta)$ si $[ab], [ac], [ad], [bc], [bd], [cd] \in W_1(\Delta)$ et ainsi de suite pour les dimensions supérieures. Ceci donne un Witness Complex que l'on pourrait appeler "statique". On peut lui donner une structure multi-échelle.

Cette structure dépendra de ν, qui correspond à un rang, et de ϵ, un réel positif. On pose alors :

- Si $\nu = 0$, pour $i = 1, ..., M$ on définit un nombre $m_i = 0$.
- Si $\nu > 0$, pour $i = 1, ..., M$, m_i est la distance au ν-ème plus proche prototype de la donnée x_i.
- Alors une arête $e = [ab]$ appartient à $W(\Delta, \epsilon, \nu)$ s'il existe un témoin x_i de $[ab]$ parmi les données, tel qu'en plus $max(\Delta(a, i), \Delta(b, i)) \leq m_i + \epsilon$
- Un simplexe appartient à $W(\Delta, \epsilon, \nu)$ si toutes ses arêtes appartiennent à $W(\Delta, \epsilon, \nu)$.

Les différentes valeurs de ε permettent de définir une suite de Witness Complex emboîtés formant une filtration pour la persistance homologique.

4.1 Analyse exploratoire de données structurées

4.1.1 Objets topologiques connus

Pour les premières expérimentations, des objets facilement observables en trois dimensions et à la topologie connue ont été choisis : une sphère et un tore. Pour le premier, les nombres de Betti sont $(1, 0, 1, 0....)$ et pour le second $(1, 2, 1, 0....)$. Le Complexe Simplicial Génératif sera comparé au Witness Complex tel qu'implémenté dans la bibliothèque Matlab "Javaplex". Le nombre de prototypes (30 pour la sphère, 40 pour le tore) est le même pour le CSG et le WitC. Le CSG fournit ce nombre grâce aux GMM et au BIC utilisés lors de l'étape d'initialisation. Après l'exécution du Witness Complex, on procède à une filtration (Zomorodian and Carlsson, 2005). Le processus de filtration est essentiel pour extraire la topologie : en faisant croître des boules autour des sommets jusqu'à ce qu'elles s'intersectent, des cycles et cavités apparaissent puis disparaissent. La méthode "infinite-Barcodes" de Javaplex permet de retourner le vecteur de nombres de Betti qui persiste le plus longtemps au cours de la croissance du diamètre des boules.

4.1.1.1 La sphère

Pour la sphère, 1000 points ont été générés suivant une distribution gaussienne isovariée de dimension 3, de moyenne $\mu = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$ et de variance $\sigma^2 = 1$. Chaque point est ensuite

projeté sur la sphère plongée dans $\mathbb{R}^3$, centrée sur l'origine et de rayon 1 en normalisant le vecteur associé :

$$x_i \longleftarrow \frac{x_i}{\|\vec{x_i}\|}. \tag{4.2}$$

On obtient ainsi 1000 points répartis aléatoirement sur la sphère de manière équiprobable. Ensuite, 1000 autres vecteurs ϵ_i sont générés suivant une distribution gaussienne de moyenne nulle et de variance égale à la variance σ^2 retenue. On s'en sert ensuite pour corrompre les points générés sur la sphère :

$$x_i \longleftarrow x_i + \epsilon_i. \tag{4.3}$$

Trois écarts-types différents ont été utilisés pour générer les données : 0.05, 0.1, 0.2.

Le CSG et WitC ont été exécutés 3×100 fois sur un jeu de données vérifiant ces conditions. Une réponse est considérée comme correcte uniquement si les bons nombres de Betti $(1, 0, 1, 0, ...)$ sont obtenus.

On reporte dans le tableau 4.1 les taux de succès du CSG et du WitC pour découvrir les bons nombres de Betti. Pour les variances les plus faibles, WitC domine légèrement le CSG, mais ils sont tous les deux fiables avec plus de 90% de succès. Pour $\sigma_\varepsilon = 0.2$, les performances de CSG diminuent alors que celles de WitC restent stables. Dans ce cas, la filtration avantage grandement WitC : la cavité à l'intérieur de la sphère subsiste très longtemps sans changer le nombre de cycles ou de cavités. Tandis que pour le CSG, si la variance est trop grande, une corde à l'intérieur de la sphère peut persister malgré les différentes étapes d'élagage et peut ajouter un cycle non-voulu dans le modèle.

	WitC	CSG
$\sigma_\varepsilon = 0.05$	100%	95%
$\sigma_\varepsilon = 0.1$	99%	90%
$\sigma_\varepsilon = 0.2$	98%	55%

FIGURE 4.1 – Taux de succès d'extraction des nombres de Betti d'une sphère unité construite avec 1000 points et un bruit gaussien d'écart-type σ

4.1.1.2 Le tore

Pour le tore, 2000 points ont été générés uniformément à la surface d'un tore plongé dans $\mathbb{R}^3$ et de grand rayon $R = 10$ et de petit rayon $r = 3$. Ces points sont ensuite corrompus avec un bruit gaussien. Pour le bruit, trois écarts-types σ_ε différents ont été utilisés : 0.01, 0.05, 0.1. Les algorithmes CSG et WitC ont été exécutés 100 fois sur un jeu de données vérifiant ces conditions. Une réponse est considérée comme correcte uniquement si les bons nombres de Betti $(1, 2, 1, 0, ...)$ sont obtenus.

La topologie d'un tore n'est pas aussi simple que celle d'une sphère : si le bruit est trop grand, l'intérieur du tore se remplit. Dans ce cas une surface peut être ajoutée dans le modèle à l'intérieur du tore, et deux cycles indépendants apparaissent. Ce phénomène peut arriver pour l'anneau formé par la révolution du tore. Ceci explique la baisse de performance que l'on peut observer sur cet objet. Alors que les résultats sont toujours corrects pour le CSG, le pourcentage de bonnes réponses du WitC est très faible. Pour un tore, les deux cycles sont moins à même de persister au travers de la filtration, alors que d'autre cycles incorrects peuvent devenir aussi pertinent que les deux vrais. Ceci explique pourquoi les erreurs du WitC se font en général sur le nombre de cycles, alors que les nombres de composantes connexes et de cavités sont corrects.

	WitC	CSG
σ_ε	5%	63%
σ_ε	8%	60%
σ_ε	9%	57%

FIGURE 4.2 – Taux de succès de l'extraction des nombres de Betti d'un tore fait de 2000 points et corrompu avec une bruit gaussien d'écart-type σ

4.1.1.3 La bouteille de Klein

Bien que ce soit une variété de dimension intrinsèque 2, une bouteille de Klein ne peut être plongée que dans un espace de dimension minimale 4, et elle est plus facilement plongée dans un espace de dimension 5. La deuxième particularité de cette variété est que c'est une surface non-orientable : son intérieur et son extérieur ne sont pas distinct. Le ruban de Moebius par exemple est une autre surface non-orientable connue. Si elle est projetée en trois dimensions, la bouteille de Klein a l'air d'une bouteille dont le goulot traverse la paroi et rejoint le fond. Comme elle est en dimension supérieure et qu'elle est non-orientable, la question de la topologie de la bouteille de Klein est plus complexe à résoudre pour le CSG et le WitC. Nous avons échantilloné 625 points et deux bruits différents ont été choisis : $\sigma_\varepsilon = 0.01$ and 0.05. Pour chaque σ_ε et chaque algorithme, l'expérience a été répétée 100 fois. Les résultats sont des pourcentages de bonne réponse : les bons nombres de Betti sur $\mathbb{Q}$ de la bouteille de Klein sont $(1, 1, 0...)$.

Sur les 100 essais, WitC ne trouve pas une seule fois les bons nombres de Betti. Le CSG produit de bons résultats. Le fait que *Javaplex* ne calcule que les nombres de Betti sur $\mathbb{Q}$ ne nous permet pas de conclure sur la capacité du CSG à extraire la torsion (les termes en $\frac{\mathbb{Z}}{p\mathbb{Z}}$).

	WitC	CSG
$\sigma = 0.01$	0%	80%
$\sigma = 0.05$	0%	73%

FIGURE 4.3 – Taux de succès d'extraction des nombres de Betti pour une bouteille de Klein faite de 650 points avec un bruit σ

4.1.2 Jeu de données réelles : COIL-100

COIL-100 est un corpus d'images disponbile en ligne (Nene et al., 1996). 100 objets différents ont été photographiés en rotation. Chaque photo est prise après que l'objet a été tourné de 5 degrés, ce qui fait 72 photos par objet. Comme les photos sont des objets en très grande dimension (nombre de pixels $\times$ nombres de couleurs), les photos ont été réduite de 128×128 à 64×64 et transformées en nuances de gris sur 256 niveaux. Enfin, comme il n'y a que 72 images par objet, elles ne peuvent générer qu'un espace à 71 dimensions. Nous centrons et réduisons les données puis les projetons par ACP dans un espace de dimension 71 sans perte d'information. Calculer le complexe de Delaunay en très grande dimension coûte cher en temps. Nous travaillons donc avec le d-squelette dimension par dimension : en effet le graphe est assez rapide à obtenir, de même que les triangles. Si pour un d donné, le CSG ne conserve aucun simplexe de dimension d, alors on ne teste pas les dimensions supérieures à d. Dans le cas des données COIL-100, en général $d = 2$, ce qui veut dire que les données reposaient sur une variété de dimension intrinsèque 1.

FIGURE 4.4 – Les 60 objets de la base COIL-100 analysés

Comme ce jeu de données n'est pas un objet connu et maîtrisé, nous faisons l'hypothèse que la topologie recherchée est celle d'un cercle : l'objet est en rotation dans l'espace des pixels, se déplace en ne revenant à son point de départ qu'une fois, après une révolution complète. Il s'agit bien d'une ligne fermée, c'est-à-dire un cycle. On considère donc que les nombres de Betti corrects sont $(1, 1, 0, ...)$. L'algorithme a été lancé 100 fois pour chacun des 60 objets de la base de données. Les nombres de Betti retenus pour chaque objet sont ceux qui ont été obtenus le plus souvent parmi les 100 résultats.

On peut voir les 60 objets en question sur la figure 4.4. On se référera à un objet par son numéro sur cette image : le premier objet en haut à gauche a le numéro 1, et la numérotation augmente de gauche à droite puis de haut en bas jusqu'à 60.

Les résultats obtenus par le CSG et WitC sont donnés dans la figure 4.5 et peuvent être classés dans plusieurs grandes familles :

- $(1, 1, 0...)$ qui correspond au cycle attendu. Pour 17 des 60 objets, c'est le résultat qui a été obtenu le plus souvent par le CSG.
- $(1, 2, 0...)$ qui est le deuxième résultat le plus fréquent pour CSG et le plus fréquent

pour WitC. Une composante connexe et deux cycles, c'est la structure d'un "8".
- $(1, 0, 0...)$, une seule composante connexe homéomorphe à un point.
- $(1, n, 0...)$, on peut voir cette structure comme un trèfle à n feuilles. Les résultats vont de $(1, 3, 0...)$ à $(1, 8, 0...)$.
- $(2, n, 0...)$, deux composantes connexes.

	WitC	CSG
$(1, 1, 0...)$	6	17
$(1, 2, 0...)$	8	15
$(1, n, 0...)$	33	22
$(1, 0, 0...)$	0	1
$(2, n, 0...)$	13	5

FIGURE 4.5 – Nombres d'observation d'une suite de nombre de Betti pour 60 images de la base COIL-100

Comme on peut le voir dans le tableau 4.5, le CSG trouve plus souvent les nombres de Betti attendus a priori que WitC. Dans 17 cas sur 60, CSG trouve une structure de cycle $(1, 1, 0...)$, contre seulement 6 cas sur 60 pour WitC.

La structure $(1, 2, 0...)$ correspond à un cycle qui présenterait un pincement : un objet dont les côtés ou la face avant et arrière se ressemblerait. Par exemple l'objet 22, une bouteille de shampooing : la face avant et la face arrière sont dissemblables, mais les côtés sont très ressemblants, ce qui peut expliquer que deux images soient très proches dans l'espace des pixels et provoquent un pincement du cycle.

La structure $(1, 0, 0...)$ n'apparaît qu'une fois, mais est encore plus simple à expliquer : cela voudrait dire que toutes les images sont quasi-identiques, et le CSG les identifie comme une version bruitée d'une image de base. Elle est à rapprocher de la structure $(1, n, 0...)$ qui apparaît plus souvent : cette fois-ci toutes les images ne sont pas considérées comme proches, mais seulement certaines, qui provoquent un pincement comment dans le cas du $(1, 2, 0...)$. Ces deux cas nous renseignent sur des objets qui présentent certaines symétries par rotation comme les objets 2 ou 4 par exemple : un oignon et une tomate, qui sont presque invariants par rotation.

Enfin la dernière structure correspondant aux nombres de Betti $(2, n, 0...)$ se présente a priori plus comme un échec de l'algorithme : il n'y a clairement qu'une seule composante connexe à identifier par objet. On peut peut-être l'expliquer par des objets dont les côtés sont très dissemblables, créant deux groupes d'images nettement séparés l'un de l'autre dans l'espace des pixels.

Visualisation de la structure avec une ACP

Il n'y a pas de vérité terrain avec ce jeu de données, et la structure de cycle attendue n'est que supposée. Il peut s'avérer intéressant de regarder en détail, et non plus de manière automatique, les images qui ont conduit à des résultats incorrects. Nous allons regarder en détail quelques images qui sont représentatives de leur classe.

Par exemple, pour l'image 25 (figure 4.6), le CSG donne comme résultat les nombres de Betti $(1, n, 0...)$. C'est *a priori* un résultat incorrect : même si l'on ne s'attend pas à un cercle parfait décrit par les données, on attend au moins une certaine continuité. Les objets ne sont tournés que de 5 degrés entre chaque photo. Un humain ne pourrait pas dire qu'un objet est fondamentalement différent parce qu'il l'a vu sous deux angles différents de 5 degrés.

Et pourtant comme on peut le voir sur l'image 4.6, l'objet est quasiment invariant par rotation, ce qui peut expliquer que l'algorithme détecte l'existence de plusieurs cycles, puisque plusieurs angles différents de l'objet sont très ressemblants. D'ailleurs la projection par l'ACP en dimension 2 montre bien cette confusion (image 4.7).

FIGURE 4.6 – Objet 25 de la base COIL-100

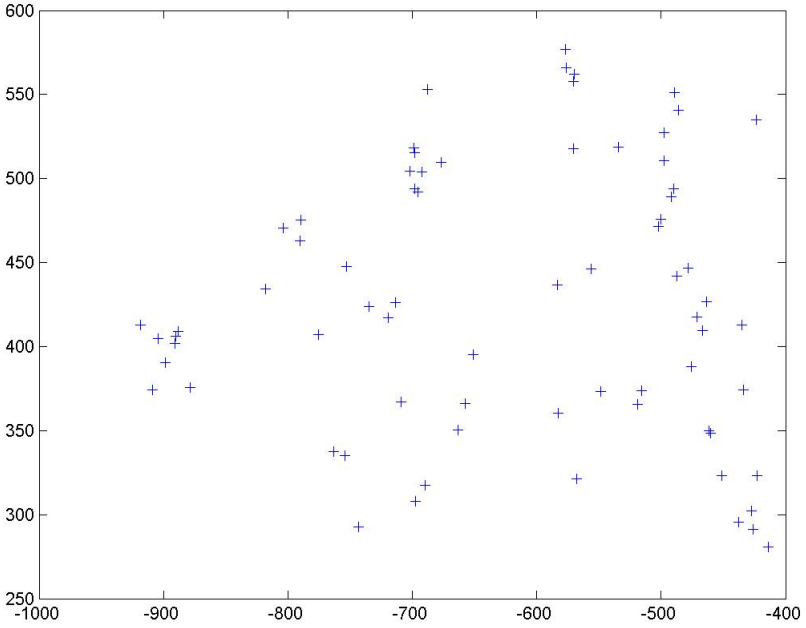
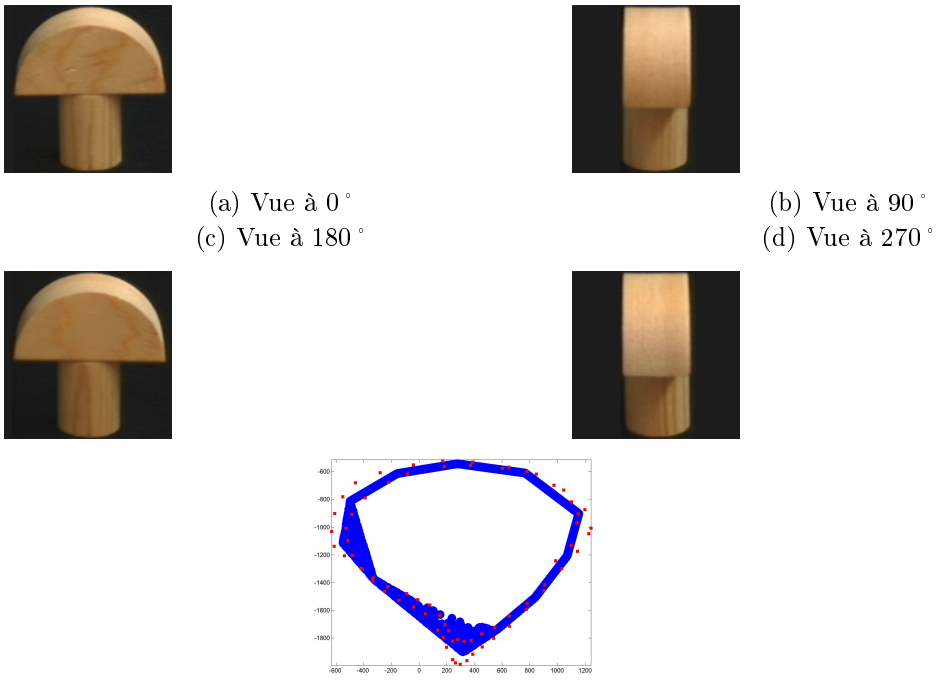


FIGURE 4.7 – Projection de l'image 25 de la base COIL-100

La figure 4.8 donne le résultat attendu $(1, 1, 0, ...)$ pour l'objet 12. Premièrement parce que l'objet 12 n'a pas cette symétrie quasi-parfaite par rotation que possède l'objet 25, et le phénomène de rotation apparaît donc clairement. Mais plus encore, cette rotation est renforcée par une impression de double rotation : l'objet sous les angles 0 et 180 est quasi-

identique, de même que sous les angles 90 et 270. Cette similitude se voit d'ailleurs dans les données projetées par ACP en dimension 2 sur la figure 4.8, où elles ont l'air regroupées par paire. A l'inverse dans ce cas-là, le CSG donne le résultat attendu *a priori*.



(a) Vue à 0 °
(b) Vue à 90 °
(c) Vue à 180 °
(d) Vue à 270 °

FIGURE 4.8 – L'objet 12 de la base de données COIL-100 vu sous 4 angles différents. On peut voir l'invariance de l'objet par une rotation de 180 °. L'ACP en bas montre la projection du modèle CSG. Les deux cycles se superposent (Lee and Verleysen, 2007).

4.1.3 Analyse des méthodes de projection

L'utilisation de l'ACP pour la confirmation du bon fonctionnement de l'algorithme soulève une interrogation : il existe différentes techniques de projection, qui ne sont pas équivalentes, et qui auraient pu produire des résultats différents. Il existe deux grandes familles d'erreurs commises par les méthodes de projection (Aupetit, 2007) :

- les déchirures qui présentent comme éloignés dans l'espace de projection des points qui étaient proches dans l'espace initial
- les recollements qui sont l'inverse, et présentent comme proche dans l'espace de projection, des points qui étaient éloignés dans l'espace initial

Au niveau topologique, une déchirure peut séparer un groupe de points qui auraient du être dans une même composante connexe, en deux composantes connexes différentes. A l'inverse, un recollement peut fusionner deux composantes connexes en une seule.

Les méthodes de projection cherchent à minimiser une erreur totale, comme le stress de Sammon par exemple (Sammon, 1969).

Si on note la distance entre deux individus dans l'espace d'origine

$$d_{ij}^* = \left\| x_i^* - x_j^* \right\| \tag{4.4}$$

et la distance entre deux individus dans l'espace projeté

$$d_{ij} = \| x_i - x_j \| \tag{4.5}$$

alors on note le stress de Sammon :

$$E = \frac{1}{\sum\limits_{i<j} d_{ij}^*} \sum_{i<j} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*}. \tag{4.6}$$

Il ne s'agit là que d'une information géométrique fondée sur les distances, et un algorithme peut très bien avoir une excellente performance de ce point de vue, et tout de même déchirer ou compresser les structures. Ce type d'erreur dans la projection serait reflétée immédiatement par un changement de topologie, et donc des nombres de Betti : disparition ou création d'un cycle, compression d'un volume qui disparaît etc.

Nous proposons par conséquent d'utiliser le CSG pour étudier le comportement des techniques de réductions de dimension. En l'utilisant d'abord dans l'espace d'origine, puis dans l'espace projeté, et de comparer les nombres de Betti obtenus dans les deux cas. Le critère pour évaluer une projection n'est donc plus quantifié par un stress, mais par le fait qu'elle est ou non une fonction qui conserve les propriétés topologiques des objets observés.

La figure 4.9 représente la projection par ACP de l'image 5 de la base de donnéeq. On y voit clairement une structure de cycle, et effectivement le CSG calcule dans l'espace de projection les nombres de Betti $(1, 1, 0...)$. On trouve aussi les mêmes nombres de Betti dans l'espace initial des pixels comme le montre le modèle appris en dimension 71 et projeté avec les données sur la figure 4.10. L'analyste va induire que cette structure sous-jacente très forte qu'est le cycle, n'est pas un artefact dû à la projection, mais bien une réalité présente dans les données.

Comme on peut le voir sur la figure 4.10, le fait que la projection 2D soit correcte n'est pas évident : une projection sur les 3 premiers axes de l'ACP montre une "excroissance" de données qui n'est pas dans le plan principal du cycle. En supprimant encore une dimension, cette excroissance aurait pu par exemple faire apparaître un deuxième cycle dans la structure.

Faute de temps, cet axe de recherche n'a pas pu être développé plus avant, mais nous pensons que c'est une piste de recherche prometteuse.

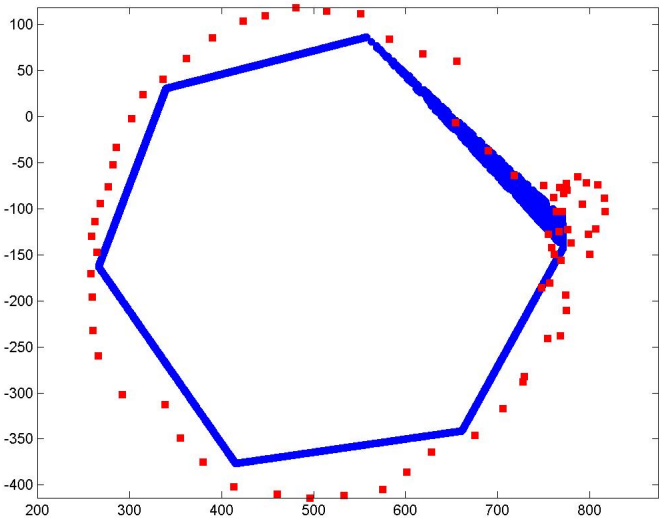


FIGURE 4.9 – Projection de l'image 5 de la base COIL-100 par ACP en dimension 2 et la structure **apprise par le CSG en dimension 2**.

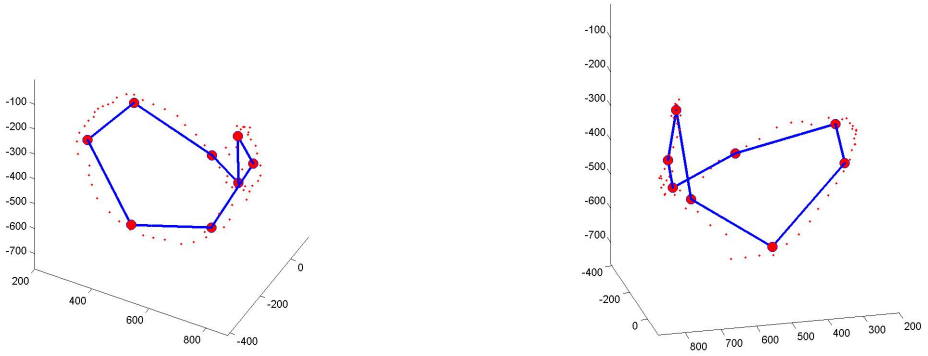


FIGURE 4.10 – Projection de la structure apprise **par le CSG en dimension** 71 et des données correspondant à l'image 5 de la base COIL-100 par ACP en dimension 3 sous deux angles différents. La structure apprise en 71D a les mêmes nombres de Betti que celle apprise en 2D, ce qui signifie que la projection par ACP est correcte et n'a pas induit de distorsion topologique.

5 Conclusion

5.1 Synthèse

5.1.1 Problématique

Soit un jeu de données dans $\mathbb{R}^D$. Les approches statistiques permettent de retrouver plusieurs informations de nature statistique, comme la densité de probabilité qui a pu générer un tel échantillon de données. Ceci permet notamment de réaliser une tâche de classification. On peut toutefois chercher à extraire un autre type d'information. Si les données sont structurées, c'est-à-dire qu'elles sont issues d'une variété de dimension intrinsèque plus faible que la dimension de l'espace d'observation des données, il peut être intéressant de chercher à caractériser cette variété. L'outil mathématique qui permet de caractériser les variétés est l'homologie. Une fois cette information homologique obtenue, elle peut être utilisée pour analyser, visualiser ou classifier des données, ou même être utilisée dans un pré-traitement à des analyses statistiques plus classiques.

5.1.2 Le Complexe Simplicial Génératif

Dans cette thèse, nous nous sommes attachés à décrire un modèle permettant d'apprendre la topologie d'une variété sous-jacente à un échantillon. Il a fallu dans un premier temps formaliser ce problème, puis proposer un modèle capable de le résoudre : le Complexe Simplicial Génératif. En partie géométrique et en partie statistique, il permet de donner un formalisme statistique à une problématique essentiellement géométrique. Le principal intérêt réside dans la possibilité de choisir le modèle adéquat grâce à un critère statistique objectif.

L'idée principale consiste à construire un complexe de Delaunay initial à partir de points "bien" positionnés par rapport aux données. Ensuite, on optimise les paramètres (poids et variance) par maximisation de la vraisemblance grâce à l'algorithme EM et on élague ce complexe, en ne gardant que les composantes pertinentes (celles dont le poids est non nul). Le nombre de sommets utilisés est choisi grâce au critère BIC, qui donne, empiriquement, un modèle qui correspond aux nombres de Betti des variétés sous-jacentes aux données.

Ces travaux peuvent être vus comme une méthode partant du formalisme statistique pour aller vers la résolution de problèmes géométriques. Le pas dans le sens inverse correspond à la persistance homologique, qui permet de diminuer l'impact du bruit des données sur les méthodes géométriques. Le persistance homologique est utilisée par la méthode du Witness Complex, à laquelle nous avons comparé le CSG.

5.1.3 Contributions applicatives

Nous montrons deux applications du Complexe Simplicial Génératif :

- Il permet de retrouver la topologie de variétés sous-jacentes à un jeu de données. Nous montrons sur des données simulées que le CSG donne une topologie correcte pour un critère BIC optimal. Sur des données réelles comme les images COIL, le CSG permet de comprendre le processus qui a généré les données, et de **classifier les objets selon leur invariance par rotation**.
- Les méthodes de projection et de réduction de dimension ont deux problèmes majeurs : les recollements, et les déchirures. Les premiers peuvent créer des cycles dans les données, alors que les seconds vont au contraire les faire disparaître. Le CSG peut permettre de mesurer ces différences de structures entre données dans l'espace initial et données projetées **par une caractérisation topologique**, autrement que par une mesure de distorsion géométrique plus classique.

5.2 Perspectives

5.2.1 Théorie

La validation du modèle du CSG est essentiellement expérimentale, et certaines questions mériteraient d'être abordées d'un point de vue théorique :

- **Optimum du critère BIC et topologie correcte :** Bien que certaines de nos expériences le montrent, il n'y a pas de démonstration que la convergence d'un critère statistique correspond à une estimation correcte des nombres de Betti, même sur des cas simples où les données sont issues d'un CSG.
- **Parcimonie de la dimension intrinsèque du modèle :** Une succession de sommets, ou un triangle aplati peuvent très bien approximer des données issues d'un segment gaussien. Il faudrait prouver théoriquement que le critère BIC (ou un critère de parcimonie), permet de retrouver le bon modèle, c'est-à-dire le segment gaussien.

C'est en tout cas celui qui a le plus petit nombre de paramètres.

- **Établir un lien avec la persistance homologique :** Habituellement dans les modèles génératifs de type GMM on utilise le nombre de sommets comme paramètre d'échelle. Mais cette approche ne permet pas de générer une filtration, qui nécessite un nombre fixe de sommets dont la position a aussi été fixée quelque soit la valeur du paramètre d'échelle.

5.2.2 Le modèle

Il serait possible d'améliorer le modèle ou de le rendre plus complet, en proposant d'optimiser d'autres paramètres, ou d'autres modélisations fondées sur le même concept.

- Il pourrait être intéressant d'utiliser des bruits différents du bruit gaussien dans le modèle, certains résultats ayant été démontrés dans le cadre de la géométrie algorithmique pour différents types de bruits (Balakrishnan et al., 2011).
- Sans aller jusqu'à relâcher toutes les contraintes sur la variance σ^2 dans le modèle, on pourrait au moins utiliser une variance σ_d^2 par dimension dans le modèle, afin de minimiser les conflits entre les simplexes et leurs enveloppes convexes privées de leurs intérieurs.

5.2.2.1 Gestion d'échelles différentes dans différentes composantes connexes

La "bonne" topologie est aussi une question d'échelle : un tore vu "de loin" paraîtra être un anneau, et donc avoir la topologie d'un cercle. C'est seulement de plus près que l'on voit apparaître le vide qui est à l'intérieur du tore. Pour le CSG, deux paramètres liés l'un à l'autre jouent sur l'échelle : le nombre de sommets et la variance. Le nombre de sommets influe sur la variance, premièrement parce que l'apprentissage de la variance se fait à nombre de sommets fixé. Ensuite parce qu'un grand nombre de sommets va faire diminuer la variance du modèle, la distance minimale d'une donnée à un prototype diminuant avec le nombre de prototypes.

Contrairement au Witness Complex qui est un modèle intrinsèquement multi-échelles, puisqu'il consiste justement à trouver les caractéristiques qui persiste à travers différentes échelles, le CSG renvoie uniquement l'échelle qui correspond à la maximisation du critère BIC. En changeant la philosophie de l'algorithme on peut imaginer une méthode se rapprochant de la persistance homologique pour gérer des situations de multi-échelles, comme on peut le voir dans la partie 5.2.2.2. Dans cette partie il est plutôt question de gérer deux échelles différentes qui co-existent dans les données.

Imaginons des données constituées d'un grand et d'un petit cercle. Si l'on se place à l'échelle du grand cercle, le petit a l'air d'un point. S'il faut au minimum trois points pour décrire un cycle, un modèle qui serait calqué sur l'échelle du grand cercle ne comportera qu'un seul prototype pour le petit cercle.

Pour résoudre ce problème, on propose de faire un premier apprentissage sur les données,

pour identifier des composantes connexes. Grâce au *maximum a posteriori*, on étiquette les données correspondant à chacune des composantes. Ensuite, on relance le CSG pour les données correspondants à une composante connexe précise. Ainsi on peut se placer à l'échelle d'une composante connexe, et non pas à l'échelle des données complètes.

5.2.2.2 Vers une forme de persistance pour le CSG

Les prototypes et la variance jouent sur l'échelle à laquelle sont observées les données par le CSG. On pourrait imaginer une autre approche, plus proche des méthodes purement géométriques : on ne se pose pas la question du nombre de prototypes, puisque l'on dispose de résultats disant que la topologie correcte peut être apprise si la variété sous-jacente est suffisamment échantillonnées. On prend donc un nombre de prototypes raisonnablement grand par rapport aux données dont on dispose.

Dans ce cas, seule la variance σ^2 sert de paramètre d'échelle, et on peut la comparer à ϵ dans le cadre du Witness Complex, à un détail près : ϵ serait plutôt homogène à $\frac{1}{\sigma}$.

En effet, quand ϵ est très grand, tous les simplexes sont connectés et l'on obtient une seule composante connexe. Quand σ est très grand, les sommets gaussiens suffisent à expliquer les données, ce qui donne autant de composantes connexes que de sommets dans le modèle. C'est ce qui se produit quand ϵ est petit. A l'inverse, quand σ est petit la distribution des données autour d'un sommet n'est plus "captée" par la variance : les simplexes de dimension 1 puis 2 puis 3 (et ainsi de suite) gagnent du poids dans le modèle.

5.2.3 Pour aller plus loin

On trouvera en annexe certaines pistes qui ont été explorées mais pas exploitées jusqu'au bout faute de temps.

Appendices

A Un critère de sélection arbitraire fondé sur la forme du simplexe

Ce qui est exposé dans cette partie faisait partie des expériences de validation de l'algorithme présentée dans la section 3.3. Comme les conclusions de ces expériences n'ont pas été utilisées pour modifier l'algorithme, elles sont présentées dans cette section.

Il est question ici de pousser la réflexion sur l'influence de la forme du simplexe, et plus exactement de son "aplatissement", que nous mesurons par sa plus petite hauteur h_{min}, sur la qualité des résultats du CSG.

Des simplexes de dimension 2 à 5 ont été générés, avec un nombre de données choisi dans l'ensemble $\{20, 50, 100, 200\}$, et à chaque fois, l'intérieur (figure 2) et l'enveloppe (figure 1) du simplexe ont été générés. On donne au CSG initial les positions exactes des sommets, il doit simplement estimer correctement les poids.

Pour visualiser les résultats, nous avons placés en cas de succès une croix noire d'abscisse h_{min} et d'ordonnée σ. En cas d'échec nous avons mis un point jaune aux mêmes coordonnées.

A.1 La dimension du simplexe

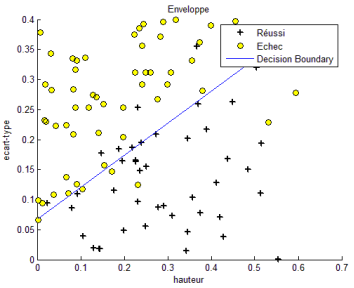
Toutes choses égales par ailleurs, quand la dimension du simplexe augmente, l'algorithme commet plus d'erreurs. C'est tout à fait compréhensible, puisqu'il faudrait plus de données quand la dimension est plus grande pour obtenir la même précision.

On a fait apparaître une séparation linéaire obtenu par régression logistique. Premièrement, on s'aperçoit qu'un séparateur linéaire fonctionne assez bien pour tracer une frontière entre les cas correctement estimés par l'algorithme et ceux qui sont erronés. Ensuite, on voit que la pente est de plus en plus faible quand la dimension augmente.

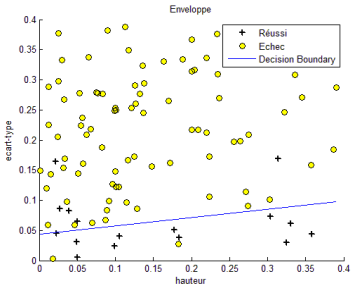
Dans le cas de l'intérieur du simplexe, on constate encore une fois qu'une séparation linéaire modélise assez bien la séparation entre échecs et réussites de l'algorithme. Cependant, on note que l'ordonnée à l'origine est différente de 0. On note aussi que le taux de réussite est meilleur dans ce cas-là.

Comme on pouvait s'y attendre, les résultats positifs sont sous la droite séparatrice : si le bruit est faible devant le facteur d'aplatissement du triangle, le problème est plus simple pour l'algorithme. Bien que l'on ait observé un lien entre la valeur de σ/h_{min} et le taux de succès de l'algorithme, nous n'avons pas poussé cette expérience assez loin pour pouvoir exploiter ce résultat et améliorer l'efficacité du CSG.

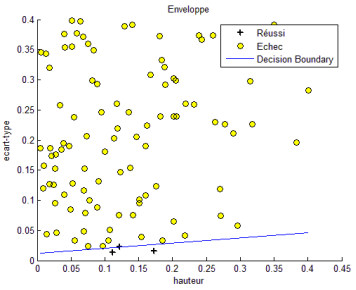
(a) $D = 2, N = 100$



(b) $D = 3, N = 100$



(c) $D = 4, N = 100$



(d) $D = 5, N = 100$

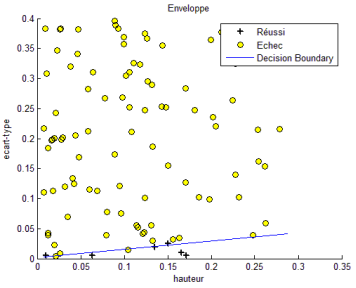


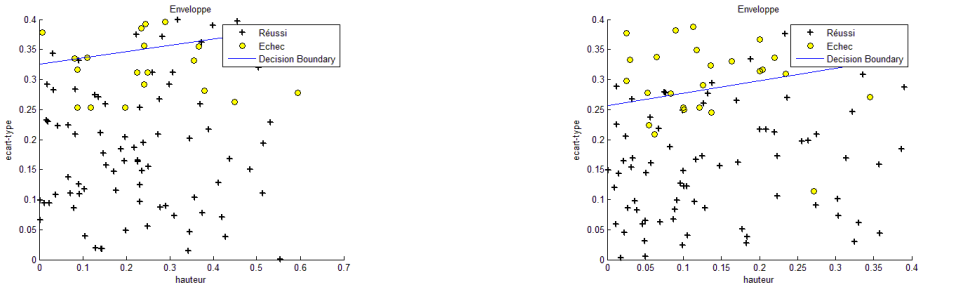
FIGURE 1 – Influence de la dimension du simplexe, pour des données de taille 100, pour retrouver l'enveloppe d'un simplexe avec le CSG dont on a fixé manuellement les sommets sur ceux du simplexe généré creux ou plein.

A.2 Le nombre de données échantillonnées

Toutes choses égales par ailleurs, quand le nombre de données échantillonnées augmente, l'algorithme commet moins d'erreurs. Ce résultat est attendu, il est normal que la précision des résultats augmente avec le nombre de données (figure 3).

On note l'efficacité d'un séparateur linéaire.

(a) $D = 2, M = 100$ (b) $D = 3, M = 100$



(c) $D = 4, M = 100$ (d) $D = 5, M = 100$

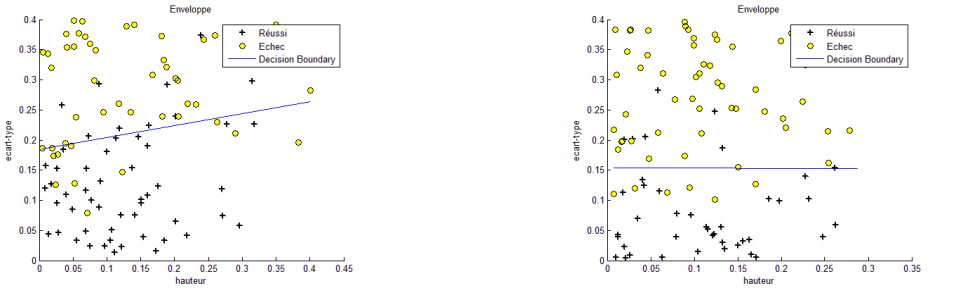


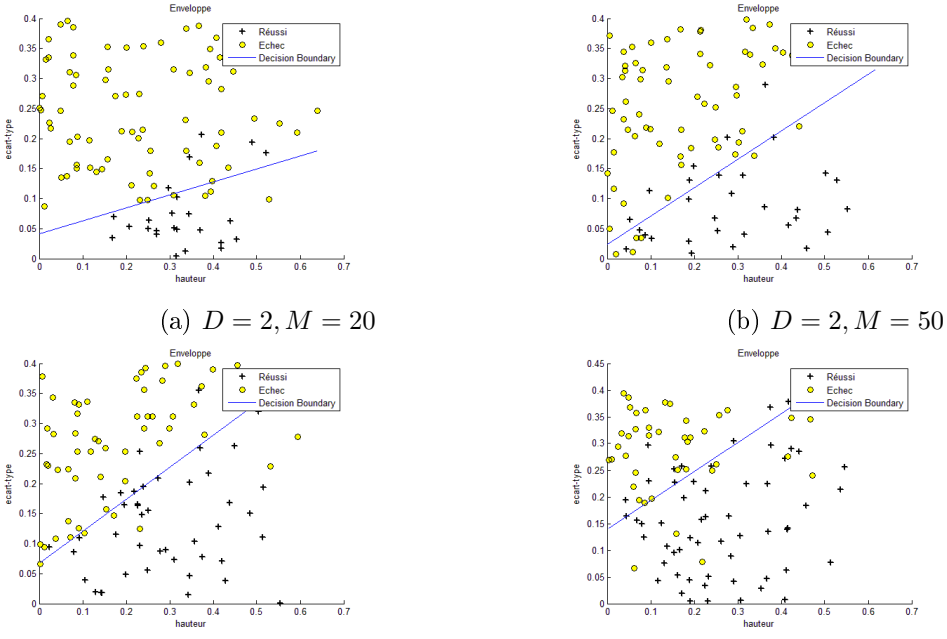
FIGURE 2 – Influence de la dimension du simplexe, pour des données de taille 100, pour retrouver l'intérieur d'un simplexe

A.3 Le problème de l'enveloppe

L'algorithme est bien plus performant pour estimer un simplexe plein qu'un simplexe creux (seulement l'enveloppe). C'est un parti pris qui a été expliqué en section 3.3.3.3, puisque le fait de retirer un simplexe du complexe initial a plus de chance de modifier la topologie, alors que laisser dans le complexe un simplexe "inutile" peut ne pas changer la topologie, il est normal de rendre cette opération plus difficile à réaliser pour l'algorithme. Ces expériences montrent que l'erreur (pour l'enveloppe comme pour l'intérieur) a plus de risque d'arriver quand la hauteur est faible devant le bruit des données. Or c'est aussi ce cas de figure que la triangulation de Delaunay cherche à minimiser.

A.4 Pistes d'améliorations issues de ces observations

En plus d'obtenir des informations qualitatives, on peut imaginer obtenir des informations plus quantitatives. En effet, on a vu que la fiabilité de l'algorithme dépendait du nombre de données N, de la dimension des simplexes d et la forme des simplexes, caractérisée par h_{min} et de l'écart-type σ. On peut supposer l'existence d'une fonction f de fiabilité, telle que la fiabilité $F = f(N, d, h_{min}, \sigma)$ qui permettrait d'ajuster le nombre de sommets dans le modèle initial *a priori*, et donc le nombre de simplexes dans la triangulation de Delaunay, en fonction du nombre de données disponibles, au lieu de le faire *a posteriori* avec le critère BIC.



(a) $D = 2, M = 20$ (b) $D = 2, M = 50$

(c) $D = 2, M = 100$ (d) $D = 2, M = 200$

FIGURE 3 – Influence du nombre de données échantillonnées, pour un simplexe de dimension 2, pour retrouver l'enveloppe d'un simplexe

On pourrait donc, localement, grâce à la valeur de σ^2 et l'aplatissement d'un simplexe, ajuster certains méta-paramètres du CSG, comme le nombre d'échantillonnage de Monte-Carlo

Échantillonnage optimal du simplexe L'intégration avec une méthode de Monte-Carlo pour calculer la densité d'un simplexe génératif est une étape très gourmande en ressource temporelle, et qui est répétée un très grand nombre de fois. Améliorer cette étape est donc primordial si on veut diminuer les temps de calcul du CSG.

Nous avons déjà cherché à trouver la meilleure méthode à taille d'échantillon fixée. Il s'agit maintenant de baisser la taille de l'échantillon sans perdre en précision au niveau de l'estimation. Les expériences ci-dessus pourraient fournir un critère pour ajuster cette taille.

Influence de la forme du simplexe Un triangle plat est en fait un segment, et devrait donc être échantillonné comme tel. On peut donc imaginer une mesure du degré d'aplatissement d'un triangle, la valeur maximum étant atteinte par le triangle équilatéral, et le minimum pour le triangle plat. Le triangle équilatéral est échantillonné avec 55 points, et le plat avec 10 (comme un segment). Le degré d'aplatissement renverrait une valeur comprise entre ces deux nombres pour un échantillonnage correct.

Le même raisonnement peut être appliqué avec un tétraèdre plat qui serait proche du

triangle.

B Présentation de soutenance

On trouvera ci-après la présentation qui a servi de support à la soutenance de cette thèse. L'introduction du sujet est soutenue par plus de schéma, et un parallèle avec les graphes y est développé, ce qui peut favoriser la compréhension de ceux qui sont déjà familiers avec ce domaine.

Apprentissage topologique appliqué à l'analyse exploratoire de données

Directeur : Gérard Govaert
UTC, Compiègne
Codirecteur : Michaël Aupetit
CEA LIST - QCRI, Doha

Maxime Maillot
CEA LIST, UTC
Soutenance le 31 mars 2015

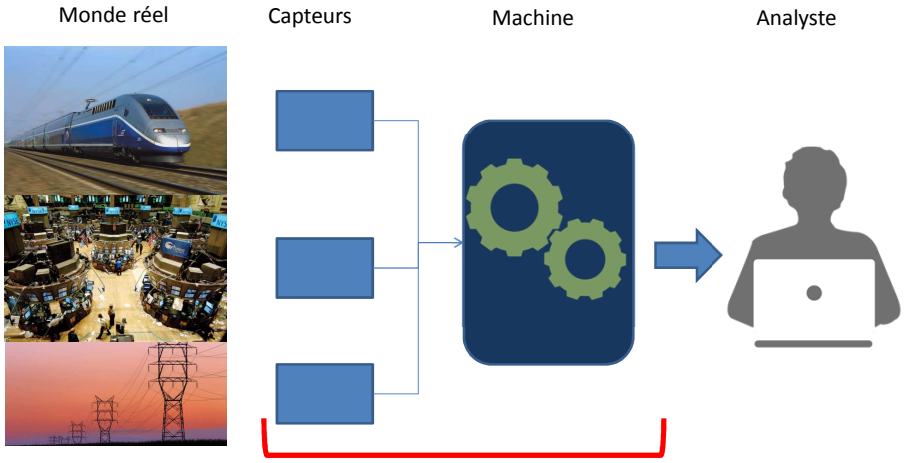
1

Plan

1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
5. Expériences
6. Conclusion

2

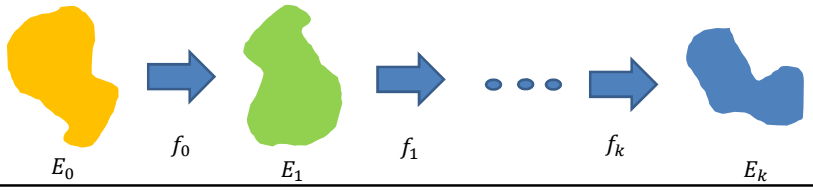
Contexte



3

L'information à travers la chaîne de traitement

- Le monde réel est analogique alors que les capteurs sont numériques
  - Il y a donc un échantillonnage et possiblement une perte d'information
- L'échantillonnage est bruité
- La machine a des contraintes (mémoire, puissance de calcul)
  - On peut être amené à transformer les données pour obtenir un résultat plus rapidement (compression, projection …)
- Il faut que le résultat soit compréhensible par l'utilisateur final
  - Projection en 2D ou 3D, résultat numérique, intervalle de confiance



4

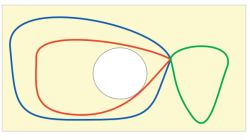
Plan

1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
5. Expériences
6. Conclusion

5

L'homologie

- On cherche une information susceptible de subsister tout au long de la chaîne de traitement
- Les transformations usuelles sont :
  - Isométries (rotation, translation …), Similitudes
  - Applications linéaires inversibles
  - Toute transformation qui ne rompt pas la continuité
- On regroupe ces transformations dans une large famille : les équivalences homotopiques
- Les groupes d'homologie sont conservés par équivalence homotopique
- Ils caractérisent une variété



6

Définitions

- **Homéomorphisme :** application bijective continue dont la réciproque est aussi continue. On dit que deux objets sont homéomorphes s'il existe un homéomorphisme qui transforme l'un en l'autre.
- **Variété topologique :** On dit que V est une variété topologique si pour tout point x de V, il existe un voisinage ouvert U de x, et U' un ouvert de $\mathbb{R}^n$ tel que U et U' sont homéomorphes.

Variété topologique de dimension 1 (homéomorphe à $\mathbb{R}$)

Variété topologique de dimension 2 (homéomorphe à $\mathbb{R}^2$)

7

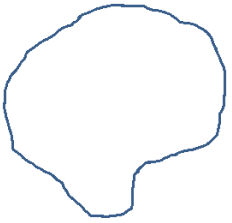
Hypothèses sur les données

- Les données peuvent être vues comme des points dans un espace euclidien de grande dimension
- En réalité, le phénomène est plus simple, et les données se situent au voisinage d'un sous-espace de dimension plus faible
- Ce sous-espace est une variété

8

Exemple d'un objet en rotation

- Une base de données de 100 objets pris en photo (128x128 pixels) en rotation sous 72 angles différents (de 0 à 360°, 5 ° par 5°)



9

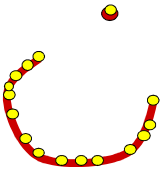
Les structures sous-jacentes

Des variétés génératrices inconnues

10

Les structures sous-jacentes

Des variétés génératrices inconnues

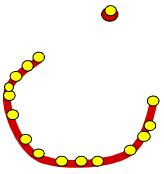


Desquelles on tire un échantillon aléatoire selon une loi inconnue

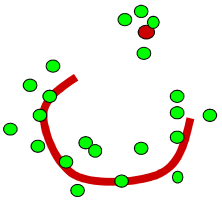
11

Les structures sous-jacentes

Des variétés génératrices inconnues



Desquelles on tire un échantillon aléatoire selon une loi inconnue



Corrompues par un bruit inconnu

12

Les structures sous-jacentes

Le phénomène réel

Les données observées

Proposer un modèle qui permet de retrouver et caractériser la variété sous-jacente à un échantillon de données

13

Plan

1. Contexte
2. Introduction
3. Décomposition du problème
   1. Une structure de données pour représenter les variétés
   2. Caractériser mathématiquement les variétés
   3. Construire une variété à partir d'un échantillon
4. Notre modèle
5. Expériences
6. Conclusion

14

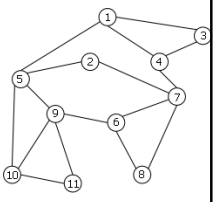
Représenter les variétés dans une structure de données

- On cherche à résoudre le problème de manière informatique
- Les variétés sont des objets continus
- Il faut une structure de données, qui permet de représenter les variétés

15

Du graphe au complexe simplicial

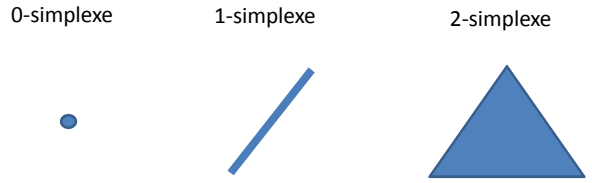
- Un graphe permettrait de représenter une variété de dimension intrinsèque 1
- Un graphe a une double nature
  - Objet combinatoire (V,E), avec V l'ensemble des sommets, et E l'ensemble des arêtes, qui sont des paires d'éléments de V
  - On peut le plonger dans le plan
- Arête d'un graphe = segment = unité « linéique » élémentaire
- Unité surfacique élémentaire = le triangle
  - Peut être obtenu par la donnée de trois sommets
- Ainsi de suite pour les dimensions supérieures



16

Le simplexe

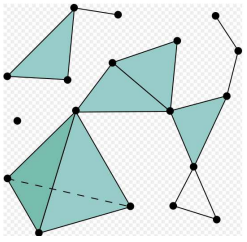
- Objet combinatoire : un d-simplexe est un ensemble de d+1 éléments
- On peut faire correspondre à chacun de ces éléments un point w_k de $\mathbb{R}^n$
- On nomme aussi simplexe l'enveloppe convexe dans $\mathbb{R}^n$ des d+1 sommets : plongement dans $\mathbb{R}^n$ du simplexe abstrait



17

Le complexe simplicial

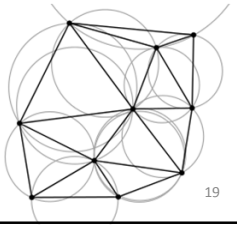
- On note $X = (V, \Sigma)$, un complexe simplicial abstrait, avec
  - V, un ensemble fini d'élément
  - Σ une famille de sous-ensembles de V
- Chaque sous-ensemble σ de Σ est un simplexe, et si $\tau \subset \sigma, \text{alors } \tau \in \Sigma$
- On peut associer au complexe simplicial abstrait, un espace topologique qui est l'union des enveloppes convexes de chacun des simplexes du complexe
- Les calculs d'homologie se font sur la structure combinatoire



18

Le complexe de Delaunay

- Un cas particulier de complexe simplicial
- Soient E un ensemble de N points dans $\mathbb{R}^D$
- On prend un sous-ensemble de D+1 sommets
  - Le simplexe formé par ces sommets appartient au complexe de Delaunay si et seulement si la sphère circonscrite à ce simplexe ne contient comme seul points de E que les sommets de ce simplexe
- Maximise l'angle minimum des triangles
- Construit des simplexes de la dimension de l'espace ambiant



19

Plan

1. Contexte
2. Introduction
3. Décomposition du problème
   1. Une structure de données pour représenter les variétés
   2. Caractériser mathématiquement les variétés
   3. Construire une variété à partir d'un échantillon
4. Notre modèle
5. Expériences
6. Conclusion

20

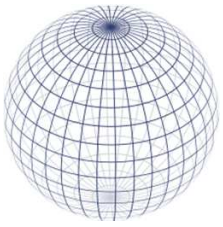
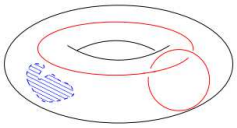
Comment caractériser ces variétés

- Comment caractériser les complexes simpliciaux
- Les variétés peuvent être caractérisées par leur homologie
- Les complexes simpliciaux peuvent être caractérisés par leur homologie simpliciale
- Sous certaines conditions (différentiabilité de la variété, par exemple [Whitehead, 1940 et Munkres 1966]), il existe un complexe simplicial de même homologie que la variété

21

Extraction des caractéristiques d'une variété

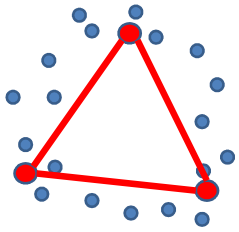
- Une manière de résumer l'homologie simpliciale est d'utiliser les nombres de Betti
  - b_0 : nombre de composantes connexes
  - b_1 : nombre de cycles ou de tunnels
  - b_2 : le nombre de cavités
  - Il existe des b_n pour tout $n \in \mathbb{N}$, et pour $n > D, b_n = 0$

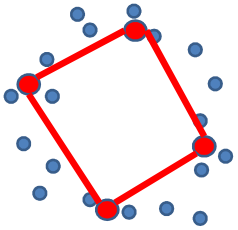




22

La robustesse de l'information homologique

- Même structure de cycle, même jeu de données, mais des complexes simpliciaux différents





- Il faut utiliser le bon outil mathématique qui permet de dire que ces deux objets décrivent la même variété : l'homologie simpliciale

23

Plan

1. Contexte
2. Introduction
3. Décomposition du problème
   1. Une structure de données pour représenter les variétés
   2. Caractériser mathématiquement les variétés
   3. Construire une variété à partir d'un échantillon
4. Notre modèle
5. Expériences
6. Conclusion

24

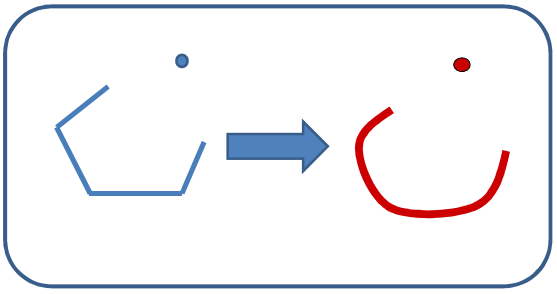
Construire une variété à partir d'un échantillon

Le phénomène réel

Les données observées

25

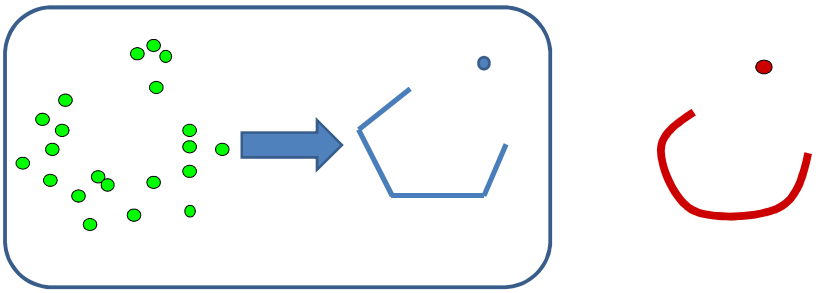
Construire un complexe simplicial à partir d'un échantillon



Outils topologiques

26

Construire un complexe simplicial à partir d'un échantillon



27

Plan

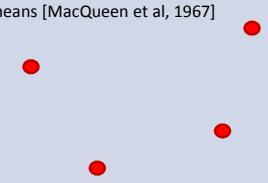
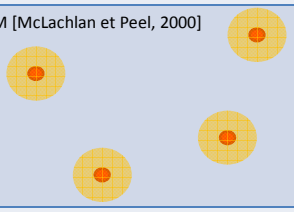
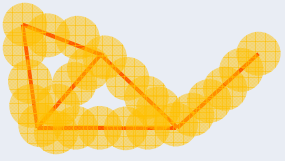
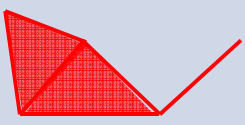
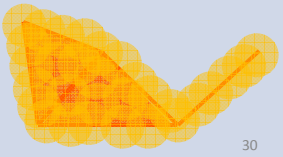
1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
5. Expériences
6. Conclusion

28

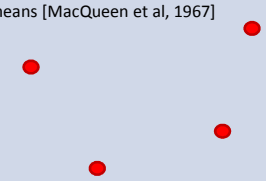
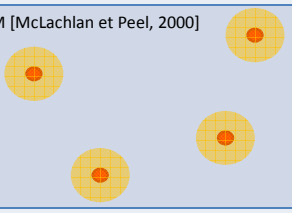
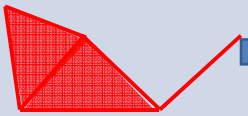
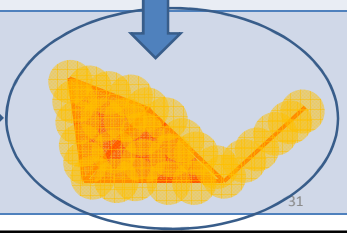
Plan

1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
   1. Formalisation du problème
   2. Le simplexe génératif
   3. L'algorithme d'optimisation
   4. Réglages de certains méta-paramètres
   5. Sélection du modèle
5. Expériences
6. Conclusion

29

	Modèles géométriques	**Modèles génératifs**
Dim = 0	K-means [MacQueen et al, 1967] 	GMM [McLachlan et Peel, 2000] 
Dim = 1	TRN [Martinetz et Schulten, 1994] 	GGG [Aupetit, 2005 ; Gaillard 2008] 
Dim = n	Witness Complex [De Silva et Carlsson, 2004] 	

30

	Modèles géométriques	Modèles génératifs
Dim = 0	K-means [MacQueen et al, 1967] 	GMM [McLachlan et Peel, 2000] 
Dim = 1	TRN [Martinetz et Schulten, 1994] 	GGG [Aupetit, 2005 ; Gaillard 2008] 
Dim = n	Witness Complex [De Silva et Carlsson, 2004] 	

31

Plan

1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
   1. Formalisation du problème
   2. Le simplexe génératif
   3. L'algorithme d'optimisation
   4. Réglages de certains méta-paramètres
   5. Sélection du modèle
5. Expériences
6. Conclusion

32

Formalisation du problème

- Les données **Z** sont tirées d'une variété sous-jacente M selon une densité p_M
- Il existe un processus d'observation f, qui va de l'espace initial à l'espace d'observation. Il transforme M en M' et les données **z** en données observées **X**. Auquel s'ajoute un bruit ε additif centré

$$x = f(z) + \varepsilon \text{ avec } \varepsilon \sim \mathcal{E}(0, \theta_\varepsilon)$$

- On suppose :
  - La transformation f conserve l'homologie de M (ie est une équivalence d'homotopie)
  - Il existe un complexe simplicial de même homologie que M
  - Le bruit suit une loi gaussienne
  - La variété est échantillonnée suivant une loi uniforme par morceaux, ie constante sur un domaine qui peut être approximé par un simplexe

33

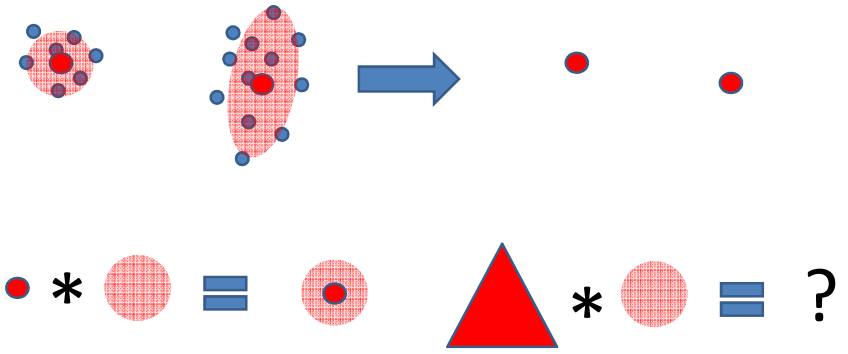
Les caractéristiques du modèle

- Nous choisissons un modèle génératif qui permet de définir ces hypothèses explicitement et permet un ajustement des paramètres à partir de l'échantillon
- Le modèle a une structure sous-jacente de complexe simplicial qui permet d'obtenir des caractéristiques homologiques

34

Du mélange gaussien au simplexe génératif

- La connexité, la caractéristique la plus simple, peut être apprise grâce à un modèle de mélange classique



35

Plan

1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
   1. Formalisation du problème
   2. Le simplexe génératif
   3. L'algorithme d'optimisation
   4. Réglages de certains méta-paramètres
   5. Sélection du modèle
5. Expériences
6. Conclusion

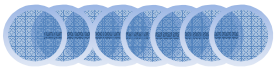
36

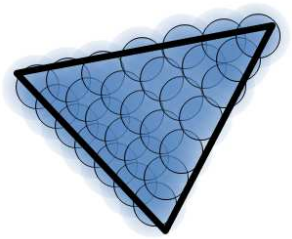
Le simplexe génératif

- C'est la brique élémentaire du modèle
  - Un complexe simplicial est constitué de simplexes, un complexe simplicial génératif est constitué de simplexes génératifs
- La convolution d'un simplexe et d'une gaussienne
- La densité de probabilité associée est :

$$g_k^d(x) = \frac{1}{|S_k^d|} \int_{S_k^d} g^0(x|t, \sigma^2) dt.|$$

- Avec
  - $|S_k^d|$ le volume du simplexe S_k^d
  - g^0 une distribution gaussienne isovariée





37

Le complexe simplicial génératif

- C'est un modèle de mélange de simplexes génératifs

$$p(x) = \sum_{d=0}^{D} \sum_{k=1}^{N_d} \pi_k^d g_k^d(x, \sigma^2)$$
$$\forall (k,d), \pi_k^d \geq 0$$
$$\sum_{d=0}^{D} \sum_{k=1}^{N_d} \pi_k^d = 1$$

- On ajuste ses paramètres par maximisation de la vraisemblance en utilisant l'algorithme EM (Expectation-Maximization)
- Etape E

$$\tilde{z}_{kn}^d \leftarrow p(z_{kn}^d | x_n; \theta) = \frac{\pi_k^d g_k^d(x_n; \sigma^2)}{\sum_{l=0}^{D} \sum_{j=1}^{N_l} \pi_j^l g_j^l(x_i; \sigma^2)}$$

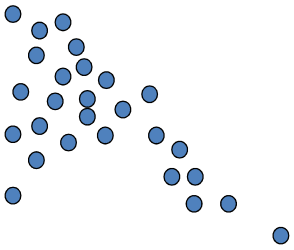
38

Plan

1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
   1. Formalisation du problème
   2. Le simplexe génératif
   3. L'algorithme d'optimisation
   4. Réglages de certains méta-paramètres
   5. Sélection du modèle
5. Expériences
6. Conclusion

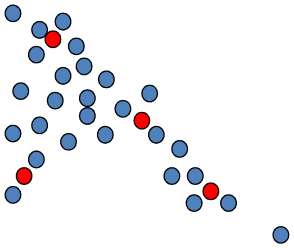
39

Des données au complexe simplicial



40

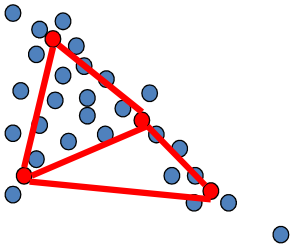
Des données au complexe simplicial



La position des sommets en rouge est initialisée à l'aide d'un modèle de mélange gaussien dont les paramètres sont optimisés par EM

41

Des données au complexe simplicial

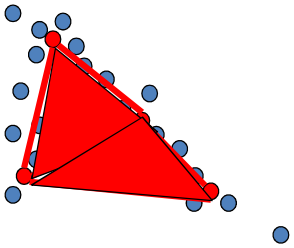


Un complexe de Delaunay est construit avec ces sommets

1. Les arêtes

42

Des données au complexe simplicial

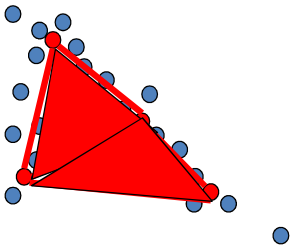


Un complexe de Delaunay est construit avec ces sommets

1. Les arêtes
2. Les surfaces

43

Des données au complexe simplicial

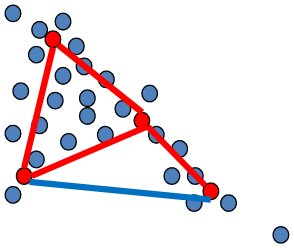


Un complexe de Delaunay est construit avec ces sommets

1. Les arêtes
2. Les surfaces
3. …

44

Des données au complexe simplicial



Le poids des arêtes est optimisé par EM

Etape M :

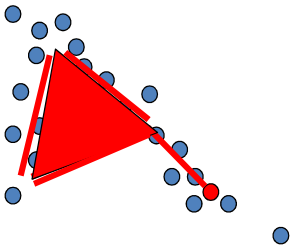
$$\pi_k^d = \frac{1}{N}\sum_{n=1}^{N} r(z_{kn}^d) = \frac{1}{N}\sum_{n=1}^{N} z_{in}^d$$

$$\sigma^2 = \frac{1}{ND}\sum_{n=1}^{N}\sum_{d=0}^{D}\sum_{k=1}^{N_d} z_{kn}^d V_{kn}^d.$$

$$V_{kn}^d = \begin{cases} \|x_n - w_k\|^2 \text{ if d} = 0 \\ \int_{S_k^d} f(q|z_{kn}^d)\,\|x_n - q\|^2\,\mathrm{d}q \end{cases}$$

45

Des données au complexe simplicial



Etape suivante : idem avec le poids des surfaces et ainsi de suite

Avec une contrainte : pour qu'un simplexe soit ajouté à l'étape n+1, il faut que ses sous-facettes aient survécu à l'étape n

46

Plan

1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
   1. Formalisation du problème
   2. Le simplexe génératif
   3. L'algorithme d'optimisation
   4. Réglages de certains méta-paramètres
   5. Sélection du modèle
5. Expériences
6. Conclusion

47

Estimation de la densité de probabilité

$$g_k^d(x) = \frac{1}{|S_k^d|} \int_{S_k^d} g^0(x|t, \sigma^2) dt.$$

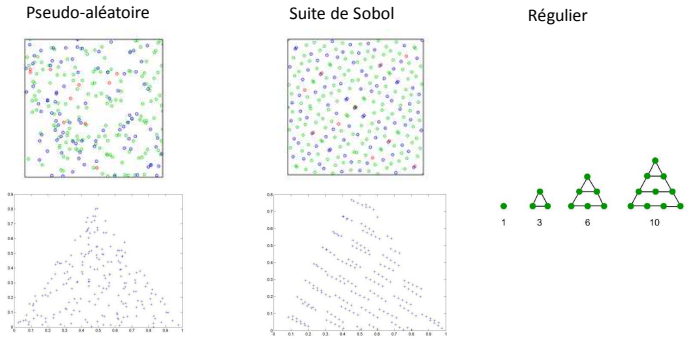
- La densité de probabilité d'un simplexe génératif nécessite l'intégrale d'une gaussienne sur un volume
- Il n'existe pas d'expression analytique de cette intégrale
- Il faut donc passer par une méthode numérique
  - Nous avons choisi l'intégration par une méthode de Monte-Carlo

compromis entre précision et temps de calcul !

48

La technique d'échantillonnage du simplexe (1)

- Les techniques d'échantillonnage ont été créées pour des pavés



- Elles ne sont pas adaptées pour les simplexes

49

La technique d'échantillonnage du simplexe (2)

- 100 échantillons aléatoires de taille 10, 50 et 100, et sur un simplexe de dimension 2, 3 et 4, soit 900 échantillons au total
- Une vraisemblance « référence » est calculée avec un échantillon régulier de grande taille
- On compare l'écart quadratique moyen à la valeur référence de chacune des méthodes

	Reg	MC	QMC
d=2	**0.20**	6.19	1.24
d=3	**0.21**	5.33	0.82
d=4	**0.13**	3.67	0.81

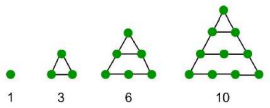
- On choisit donc l'échantillonnage régulier

50

La taille de l'échantillon pour l'intégration par Monte-Carlo

- Une expérience similaire est réalisée pour définir la taille de l'échantillon
- La taille de l'échantillon dépend aussi de la dimension du simplexe
  - Il faut plus de points pour échantillonner un triangle qu'un segment, un tétraèdre qu'un triangle
  - On utilise pour cela les nombres r-topiques

	n=5	n=10	n=15
d=2	2.10	0.21	0.20
d=3	1.87	0.19	0.17
d=4	0.92	0.13	0.13



- Utiliser 10 subdivisions par dimension est un bon compromis

51

Plan

1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
   1. Formalisation du problème
   2. Le simplexe génératif
   3. L'algorithme d'optimisation
   4. Réglages de certains méta-paramètres
   5. Sélection du modèle
5. Expériences
6. Conclusion

52

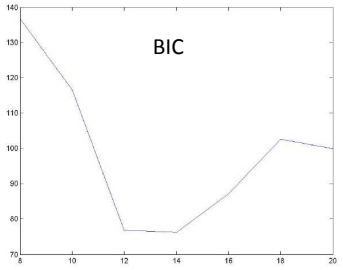
Complexité du modèle (1)

- Il suffit de 3 sommets pour retrouver la topologie d'un cercle
- Ajouter des sommets au modèle ne peut qu'en améliorer la vraisemblance
- Il faut utiliser un critère statistique qui permet de choisir un nombre optimal de sommets par rapport aux données disponibles
- Critère utilisé : BIC
- A tester sur le cercle (forme la plus simple avec un b_1 non nul)
- On fait varier N_0, on optimise le modèle pour chacune des valeurs, puis on trouve le N_0 optimal selon le critère, et on vérifie que la topologie est la bonne

53

Complexité du modèle (2)

- Entre 8 et 20 sommets dans le modèle, la topologie retournée est bien celle d'un cercle
  - Robustesse de l'information topologique par rapport au nombre de prototypes dans le modèle



Nombre de sommets

54

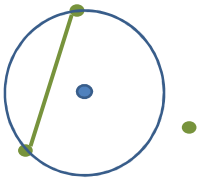
Plan

1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
5. Expériences
   1. Données jouets (sphère, tore, bouteille de Klein)
   2. Analyse exploratoire d'un ensemble d'images
6. Conclusion

55

Le Witness Complex

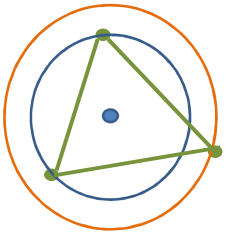
- Complexe simplicial construit sur des prototypes et non pas sur toutes les données
- Sous-ensemble du graphe de Delaunay des prototypes
- On ajoute une arête si elle a un témoin dans les données
  - ie les sommets de l'arête sont les deux plus proches sommets du témoin dans l'ensemble des prototypes
- Permet de construire une filtration
  - Calcul de persistance homologique



56

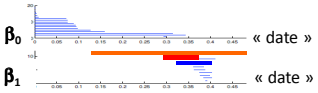
Le Witness Complex

- Complexe simplicial construit sur des prototypes et non pas sur toutes les données
- Sous-ensemble du graphe de Delaunay des prototypes
- On ajoute une arête si elle a un témoin dans les données
  - ie les sommets de l'arête sont les deux plus proches sommets du témoin dans l'ensemble des prototypes



- Permet de construire une filtration
  - Calcul de persistance homologique

Représentation « code barre »



57

Données jouets : la sphère

- Une sphère de rayon 1 est échantillonnée aléatoirement avec 1000 points. On constitue 100 échantillons de ce type.
- La position de chaque point est perturbée par ajout d'un bruit gaussien de moyenne nulle et d'écart-type dans $\{0.05, 0.1, 0.2\}$
- Nombres de Betti de la sphère (1,0,1)

	WitC	CSG
$\sigma_\varepsilon = 0.05$	100%	95%
$\sigma_\varepsilon = 0.1$	99%	90%
$\sigma_\varepsilon = 0.2$	98%	55%

- La cavité intérieure de la sphère va survivre longtemps dans le code barre du Witness Complex, ce qui l'avantage sur la sphère

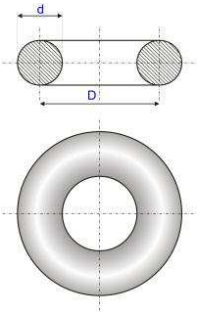
58

Données jouets : le tore

- De la même manière, un tore de rayon 10 et de rayon intérieur 3 est échantillonné avec 2000 points, et la position des points est bruitée grâce à un bruit gaussien de moyenne nulle et d'écart-type {0.01, 0.05, 0.1}.
- Nombres de Betti du tore (1,2,1)

	WitC	CSG
σ_ε	5%	63%
σ_ε	8%	60%
σ_ε	9%	57%

Beaucoup d'erreur du WitC sur le nombre de cycles : deux échelles différentes dans le tore

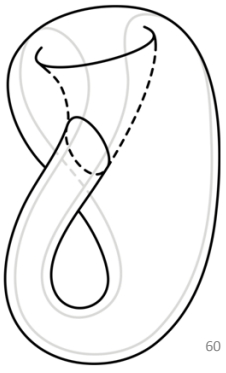


59

Données jouets : la bouteille de Klein

- Objet de dimension intrinsèque 2 mais représentable sans auto-intersection uniquement en dimension 4 ou supérieure
- Echantillonnage de 625 points
- Surface non orientable
- Nombres de Betti de la bouteille de Klein (1,1)

	WitC	CSG
$\sigma = 0.01$	0%	80%
$\sigma = 0.05$	0%	73%



60

Plan

1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
5. Expériences
   1. Données jouets (sphère, tore, bouteille de Klein)
   2. Analyse exploratoire d'un ensemble d'images
6. Conclusion

61

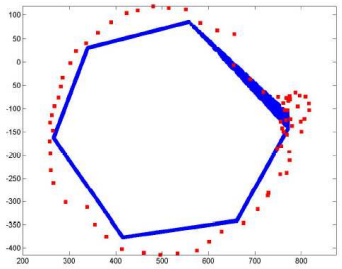
COIL-100

- Une base de données de 100 objets pris en photo en rotation sous 72 angles différents (de 0 à 360°, 5 ° par 5°) (128x128)
- On peut représenter chaque image comme un point dans l'espace des pixels de dimension 128x128 en niveau de gris
- On a étudié les 72 photos de chaque objet indépendamment
- Projetées par ACP en dimension 71 (sans perte d'information topologique)

62

Résultat attendu

- Chaque point est différent, puisque chaque photo est prise sous un angle différent
- Jusqu'à ce qu'on arrive à 360° → on obtient donc une ligne continue fermée qui a la structure topologique d'un cercle



63

Expériences

- Witness Complex et CSG ont été appliqués sur les 60 premiers objets de la base

	WitC	CSG
$(1, 1, 0...)$	6	17
$(1, 2, 0...)$	8	15
$(1, n, 0...)$	33	22
$(1, 0, 0...)$	0	1
$(2, n, 0...)$	13	5

- Chaque objet est classé selon le type de résultat qu'il a renvoyé
- Certains objets sont invariants par rotation
- Certains objets sont identiques sous plusieurs angles bien définis

64

Interprétation des résultats (1)

- (1,1,0 …)
  - Structure de cycle bien retrouvée

- (1,2,0)
  - Objet qui peut avoir deux faces qui se ressemblent

65

Interprétation des résultats (2)

- (1,n,0) et (1,0,0)
  - Des objets ayant une grande symétrie par rotation

- (2,n,0)
  - Rupture de la continuité -> Echec de l'algorithme

66

Plan

1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
5. Expériences
   1. Données jouets (sphère, tore, bouteille de Klein)
   2. Analyse exploratoire d'un ensemble d'images
6. Conclusion

67

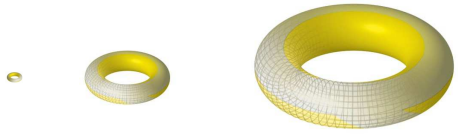
Conclusion

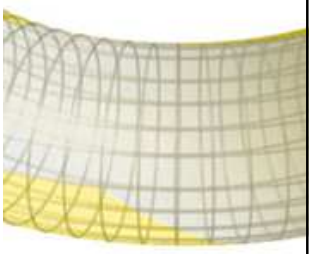
- Nous avons introduit un modèle, le Complexe Simplicial Génératif
- Dans le cas où des données seraient issues d'une variété sous-jacente de dimension inférieure à l'espace des données, il permet d'identifier la variété sous-jacente et d'en extraire les caractéristiques homologiques
- Sur un jeu de données réelles comme COIL-100, il permet de comprendre le processus qui a généré les images
- Caractérisation d'un objet par ses invariances par rapport à la rotation

68

Perspectives

- Passer d'une validation expérimentale à une validation théorique
  - Optimum du critère BIC coïncide avec la bonne topologie
  - Parcimonie de la dimension intrinsèque du modèle
- Utiliser des bruits différents (et pas uniquement un bruit gaussien)
- Gestion d'échelles différentes et classification hierarchique

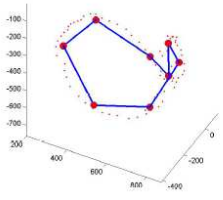


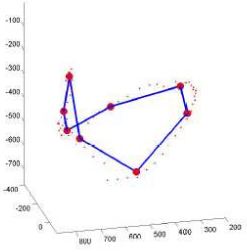


69

Conservation de l'homologie après réduction de dimension

- Les données sont projetées en dimension 3 par Analyse en Composantes Principales





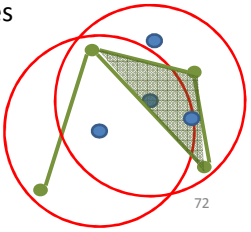
- On compare l'homologie dans l'espace initial des données et dans l'espace de projection
- On peut donner une indication à l'utilisateur final s'il y a eu des déchirures ou des recollements au cours de la projection

70

Merci de votre attention

Le Witness Complex

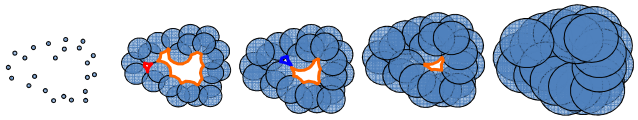
- Soit **x** un ensemble de points dans $\mathbb{R}^D$, et V un ensemble de prototypes dans ce même espace.
- Version forte : On ajoute au Witness Complex le simplexe A = $[a_0 a_1 \dots a_p]$ si et seulement s'il existe un "témoin" x du simplexe tel que :
  $|x - a| < |x - b|$ pour $a \in A$ et $b \in V \backslash A$
- Version faible : si toutes les arêtes sont dans le squelette de dimension 1, on ajoute automatiquement le triangle correspondant, et ainsi de suite pour les tétraèdres



[de Silva & Carlsson, 2004]

Filtration et persistance homologique

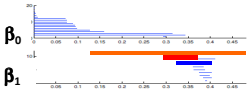
- On fait grossir des boules autour des données ou des prototypes
- On ajoute un segment entre deux sommets quand deux boules s'intersectent, un triangle quand trois boules s'intersectent etc.

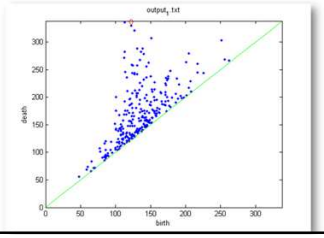


(n,0,…) (1,2,…) (1,2,…) (1,1,…) (1,0,…)

- On obtient un complexe simplicial différent (et donc une homologie différente) pour chaque rayon

Représentation « code barre »





[Zomorodian, 2001]

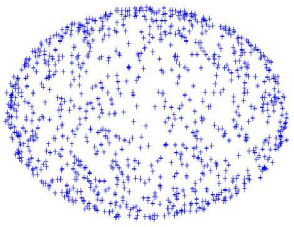
73

Contexte

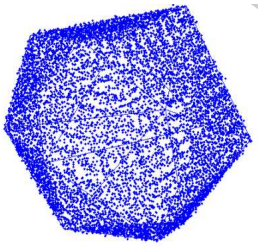
- L'analyse topologique des données (TDA)
  - A partir de données échantillonnées, reconstituer la variété sous-jacente à ces données
- Au début des années 2000 des mathématiciens spécialistes de la topologie algébrique ont construit des outils informatiques : la topologie algorithmique [Edelsbrunner et al, 2000 ; de Silva 2003]
- Papier fondateur : *Topology and Data* [Carlsson, 2009]
  - La topologie permet d'analyser les données de type nuage de points
  - Robustesse par rapport aux changements de coordonnées
  - Equivalence classe (apprentissage) – composante connexes (topologie)

74

Les données générées à partir du modèle appris



Les données initiales



Les données générées par notre modèle

75

L'élagage du complexe simplicial (1)

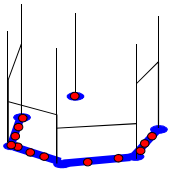
- Tous les simplexes existent « a priori » dans notre modèle
- L'homologie d'un complexe simplicial dépend uniquement de la présence ou de l'absence d'un simplexe dans le complexe final
  - On ne s'intéresse pas à la valeur de π_i^d mais uniquement au fait qu'il est nul ou non
    - Si $\pi_i^d = 0$, alors le simplexe n'appartient pas à $\hat{X}$
    - Si $\pi_i^d > 0$, alors le simplexe appartient à $\hat{X}$
  - Initialement, tous les poids sont non nuls
    - On peut seuiller les poids faibles et les mettre arbitrairement à zéro
    - On peut définir plusieurs modèles différents qui contiennent ou non les simplexes et utiliser un critère statistique pour choisir le meilleur modèle

76

Plan

1. Contexte
2. Introduction
3. Décomposition du problème
4. Notre modèle
5. Expériences
6. Conclusion

77



78

Bibliographie

Adcock, A., Rubin, D., and Carlsson, G. (2012). Classification of hepatic lesions using the matching metric. *arXiv preprint arXiv :1210.0866*.

Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6) :716–723.

Allili, M., Mischaikow, K., and Tannenbaum, A. (2001). Cubical homology and the topological classification of 2d and 3d imagery. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 2, pages 173–176. IEEE.

Asmussen, S. and Glynn, P. W. (2007). *Stochastic simulation : Algorithms and analysis*, volume 57. Springer.

Aupetit, M. (2005). Learning topology with the generative gaussian graph and the em algorithm. In *Advances in neural information processing systems*, pages 83–90.

Aupetit, M. (2007). Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 70(7) :1304–1330.

Bajaj, C. L., Bernardini, F., and Xu, G. (1995). Automatic reconstruction of surfaces and scalar fields from 3d scans. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 109–118. ACM.

Balakrishnan, S., Rinaldo, A., Sheehy, D., Singh, A., and Wasserman, L. (2011). Minimax rates for homology inference. *arXiv preprint arXiv :1112.5627*.

Banfield, J. D. and Raftery, A. E. (1992). Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. *Journal of the American Statistical Association*, 87(417) :7–16.

Bernardini, F., Bajaj, C. L., Chen, J., and Schikore, D. R. (1999). Automatic reconstruction of 3d cad models from digital scans. *International Journal of Computational Geometry & Applications*, 9(04n05) :327–369.

Biernacki, C., Celeux, G., and Govaert, G. (2003). Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3) :561–575.

Biernacki, C., Celeux, G., Govaert, G., and Langrognet, F. (2006). Model-based cluster and discriminant analysis with the mixmod software. *Computational Statistics & Data Analysis*, 51(2) :587–600.

Biernacki, C. and Chrétien, S. (2003). Degeneracy in the maximum likelihood estimation of univariate gaussian mixtures with em. *Statistics & probability letters*, 61(4) :373–382.

Biernacki, C. and Govaert, G. (1999). Choosing models in model-based clustering and discriminant analysis. *Journal of Statistical Computation and Simulation*, 64(1) :49–71.

Bishop, C. M., Svensén, M., and Williams, C. K. (1998a). Gtm : The generative topographic mapping. *Neural computation*, 10(1) :215–234.

Bishop, C. M., Svensén, M., and Williams, C. K. I. (1998b). Gtm : The generative topographic mapping. *Neural Computation*, 10(1) :215–234.

Boissonnat, J.-D. and Geiger, B. (1993). Three-dimensional reconstruction of complex shapes based on the delaunay triangulation. In *IS&T/SPIE's Symposium on Electronic Imaging : Science and Technology*, pages 964–975. International Society for Optics and Photonics.

Boissonnat, J.-D., Yvinec, M., and Brönnimann, H. (1998). *Algorithmic geometry*, volume 5. Cambridge university press Cambridge.

Boyles, R. A. (1983). On the convergence of the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 47–50.

Burnham, K. P. and Anderson, D. R. (2002). *Model selection and multi-model inference : a practical information-theoretic approach*. Springer.

Camastra, F. and Vinciarelli, A. (2002). Estimating the intrinsic dimension of data with a fractal-based method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(10) :1404–1407.

Carlin, B. P. and Louis, T. A. (1997). Bayes and empirical bayes methods for data analysis. *Statistics and Computing*, 7(2) :153–154.

Carlsson, G. (2008). Topology and data. Technical report, Department of Mathematics, Stanford University.

Celeux, G. and Diebolt, J. (1985). The sem algorithm : a probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Computational statistics quarterly*, 2(1) :73–82.

Celeux, G. and Govaert, G. (1992). A classification em algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, 14(3) :315–332.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern recognition*, 28(5) :781–793.

Chazal, F., Cohen-Steiner, D., and Mérigot, Q. (2011a). Geometric inference for probability measures. *Foundations of Computational Mathematics*, 11(6) :733–751.

Chazal, F., de Silva, V., Glisse, M., and Oudot, S. (2012). The structure and stability of persistence modules. *arXiv preprint arXiv :1207.3674*.

Chazal, F., Guibas, L. J., Oudot, S. Y., and Skraba, P. (2011b). Persistence-based clustering in riemannian manifolds. In *Proceedings of the 27th annual ACM symposium on Computational Geometry*, pages 97–106. ACM.

Chazal, F., Guibas, L. J., Oudot, S. Y., and Skraba, P. (2013). Persistence-based clustering in riemannian manifolds. *Journal of the ACM (JACM)*, 60(6) :41.

Day, N. E. (1969). Estimating the components of a mixture of normal distributions. *Biometrika*, 56(3) :463–474.

De Silva, V. (2003). A weak definition of delaunay triangulation. *arXiv preprint cs/0310031*.

De Silva, V. and Carlsson, G. (2004). Topological estimation using witness complexes. In *Proceedings of the First Eurographics conference on Point-Based Graphics*, pages 157–166. Eurographics Association.

de Silva, V. and Ghrist, R. (2007). Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7(339-358) :24.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *J. Royal Statistical Society, Series B*, 39(1) :1–38.

Dumas, J.-G., Heckenbach, F., Saunders, D., and Welker, V. (2003). Computing simplicial homology based on efficient smith normal form algorithms. In *Algebra, Geometry and Software Systems*, pages 177–206. Springer.

Edelsbrunner, H., Letscher, D., and Zomorodian, A. (2000). Topological persistence and simplification. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 454–463. IEEE.

Erwin, E., Obermayer, K., and Schulten, K. (1992). Self-organizing maps : ordering, convergence properties and energy functions. *Biological cybernetics*, 67(1) :47–55.

Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222 :309–368.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458) :611–631.

Fritzke, B. et al. (1995). A growing neural gas network learns topologies. *Advances in neural information processing systems*, 7 :625–632.

Frosini, P. and Landi, C. (1999). Size theory as a topological tool for computer vision. *Pattern Recognition and Image Analysis*, 9(4) :596–603.

Fukunaga, K. and Olsen, D. R. (1971). An algorithm for finding intrinsic dimensionality of data. *Computers, IEEE Transactions on*, 100(2) :176–183.

Gaillard, P. (2008). Apprentissage statistique de la connexité d'un nuage de point par un modèle génératif. application à l'analyse exploratoire et la classification semi-supervisée.

Gromov, M. (1987). *Hyperbolic groups*. Springer.

Hamilton, J. D. (1991). A quasi-bayesian approach to estimating parameters for mixtures of normal distributions. *Journal of Business & Economic Statistics*, 9(1) :27–39.

Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley & Sons, Inc.

Hasselblad, V. (1966). Estimation of parameters for a mixture of normal distributions. *Technometrics*, 8(3) :431–444.

Hastie, T. and Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, 84(406) :502–516.

Hathaway, R. J. (1985). A constrained formulation of maximum-likelihood estimation for normal mixture distributions. *The Annals of Statistics*, 13(2) :795–800.

Hausmann, J.-C. et al. (1995). On the vietoris-rips complexes and a cohomology theory for metric spaces. *Ann. Math. Studies*, 138 :175–188.

Helmbold, D. P., Schapire, R. E., Singer, Y., and Warmuth, M. K. (1997). A comparison of new and old algorithms for a mixture estimation problem. *Machine Learning*, 27(1) :97–119.

Hertz, J. A., Krogh, A. S., and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computacion*, volume 1. Basic Books.

Huang, Y., Englehart, K. B., Hudgins, B., and Chan, A. D. (2005). A gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses. *Biomedical Engineering, IEEE Transactions on*, 52(11) :1801–1811.

Ingrassia, S. and Rocci, R. (2011). Degeneracy of the {EM} algorithm for the {MLE} of multivariate gaussian mixtures and dynamic constraints. *Computational Statistics & Data Analysis*, 55(4) :1715 – 1725.

Kiefer, J. and Wolfowitz, J. (1956). Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters. *The Annals of Mathematical Statistics*, pages 887–906.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9) :1464–1480.

Lee, J. A. and Verleysen, M. (2007). *Nonlinear dimensionality reduction*. Springer Science & Business Media.

Lemieux, C. (2009). *Monte Carlo and Quasi-Monte-Carlo Sampling*. Springer.

Levina, E. and Bickel, P. J. (2004). Maximum likelihood estimation of intrinsic dimension. In *Advances in neural information processing systems*, pages 777–784.

Lindsay, B. G. (1983). The geometry of mixture likelihoods : a general theory. *The annals of statistics*, 11(1) :86–94.

Lo, Z.-P. and Bavarian, B. (1991). On the rate of convergence in topology preserving neural networks. *Biological Cybernetics*, 65(1) :55–63.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA.

Maillot, M., Aupetit, M., and Govaert, G. (2012a). A generative model that learns betti numbers from a data set. In *ESANN2012, 20th European Symposium on Artificial Neural Networks*, pages 537–542.

Maillot, M., Aupetit, M., and Govaert, G. (2012b). The generative simplicial complex to extract betti numbers from unlabeled data. In *NIPS 2012 Workshop on Algebraic Topology and Machine Learning*, pages 1–5.

Maillot, M., Aupetit, M., and Govaert, G. (2013). Extraction des nombres de betti avec un modèle génératif. *EGC2013*, pages 1–6.

Martinetz, T. and Schulten, K. (1994). Topology representing networks. *Neural Networks*, 7(3) :507–522.

Martinetz, T. M., Berkovich, S. G., and Schulten, K. J. (1993). Neural-gas' network for vector quantization and its application to time-series prediction. *Neural Networks, IEEE Transactions on*, 4(4) :558–569.

Martis, R. J., Chakraborty, C., and Ray, A. K. (2009). A two-stage mechanism for registration and classification of ecg using gaussian mixture model. *Pattern Recognition*, 42(11) :2979–2988.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley series in probability and statistics : Applied probability and statistics. John Wiley & Sons.

Méndez, J. and Lorenzo, J. (2012). Efficient computation of voronoi neighbors based on polytope search in pattern recognition. In Carmona, P. L., Sánchez, J. S., and Fred, A. L. N., editors, *ICPRAM (2)*, pages 357–364. SciTePress.

Mobasher, B., Cooley, R., and Srivastava, J. (2000). Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8) :142–151.

Morokoff, W. J. and Caflisch, R. E. (1995). Quasi-monte carlo integration. *Journal of computational physics*, 122(2) :218–230.

Munkres, J. R. (1966). *Elementary differential topology*. Number 54. Princeton University Press.

Nene, S. A., Nayar, S. K., and Murase, H. (1996). Columbia Object Image Library (COIL-100). Technical report.

Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3) :1065–1076.

Perea, J. and Harer, J. (2013). Sliding windows and persistence : An application of topological methods to signal analysis. *arXiv preprint arXiv :1307.6188*.

Pillards, T. and Cools, R. (2005). Transforming low-discrepancy sequences from a cube to a simplex. *Journal of computational and applied mathematics*, 174(1) :29–42.

Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. (2000). Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1) :19–41.

Robins, V. (1999). Towards computing homology from finite approximations. In *Topology Proceedings*, volume 24, pages 503–532.

Roeder, K. and Wasserman, L. (1997). Practical bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, 92(439) :894–902.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pages 832–837.

Rousseau, Y. (2014). Plus d'un milliard de smartphones écoulés en 2013.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on computers*, 18(5) :401–409.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2) :461–464.

Stanford, D. C. and Raftery, A. E. (2000). Finding curvilinear features in spatial point patterns : principal curve clustering with noise. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(6) :601–609.

Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE.

Tibshirani, R. (1992). Principal curves revisited. *Statistics and Computing*, 2(4) :183–190.

Vejdemo-Johansson, M., Skraba, P., and de Silva, V. (2012). Topological analysis of recurrent systems.

Vellido, A., El-Deredy, W., and Lisboa, P. J. (2003). Selective smoothing of the generative topographic mapping. *Neural Networks, IEEE Transactions on*, 14(4) :847–852.

Verbeek, J., Vlassis, N., Krose, B., et al. (2003). Self-organization by optimizing free-energy. In *11th European Symposium on Artificial Neural Networks (ESANN'03)*, pages 125–130.

Vietoris, L. (1927). Über den höheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen. *Mathematische Annalen*, 97(1) :454–472.

Vlassis, N. A., Dimopoulos, A., and Papakonstantinou, G. (1997). The probabilistic growing cell structures algorithm. In *Artificial Neural Networks—ICANN'97*, pages 649–654. Springer.

Whitehead, J. (1940). On c 1-complexes. *The Annals of Mathematics*, 41(4) :809–824.

Zaremba, S. (1968). The mathematical basis of monte carlo and quasi-monte carlo methods. *SIAM review*, 10(3) :303–314.

Zomorodian, A. (2010). Fast construction of the vietoris-rips complex. *Computers & Graphics*, 34(3) :263–271.

Zomorodian, A. and Carlsson, G. (2005). Computing persistent homology. *Discrete & Computational Geometry*, 33(2) :249–274.