Caractérisation topologique d'un jeu de données images avec les nombres de Betti et un modèle génératif

Maxime Maillot[1,2], Michaël Aupetit[1], et Gérard Govaert[2]

[1]CEA LIST
[2]UT Compiègne

1er juin 2013

Résumé

Dans cet article, nous proposons un modèle génératif qui permet d'extraire les nombres de Betti d'un ensemble de variétés de $\mathbb{R}^D$ à partir d'un échantillon. Ce modèle est basé sur le Complexe Simplicial Génératif, un modèle de mélange dont les composantes sont des simplexes géométriques convolués à une distribution gaussienne multivariée. La nouveauté de la méthode consiste à optimiser BIC, un critère statistique de vraisemblance pénalisée, pour obtenir une estimation des invariants topologiques des variétés génératrices des données, les nombres de Betti. Les résultats d'une telle méthode montrent une plus grande robustesse au bruit que ceux d'une méthode fondée sur une approche purement géométrique. Le Complexe Simplicial Génératif est comparé au Witness Complex (WitC) et la persistance homologique sur des données jouet (sphère, tore, bouteille de Klein) et un jeu de données réelles, COIL-100, une base d'images d'objets en rotation.

Mots-clef : Nombres de Betti ; Complexe simplicial ; modèle de mélange ; BIC.

1 Introduction

La croissance exponentielle du nombre de capteurs et de la taille des bases de données engendre une masse de données multivariées toujours plus importante à analyser. L'analyse consiste à extraire des données des caractéristiques relatives à la population dont elles forment un échantillon. Les techniques habituelles recherchent des groupes de variables liées entre elles, des groupes de données similaires ou des données atypiques [MP00] . Si l'on considère la population génératrice des données comme un ensemble de variétés support d'une fonction densité de probabilité, l'analyse statistique exploratoire s'attache à estimer cette fonction de densité de probabilité, là où l'analyse topologique cherche à extraire les caractéristiques topologiques de ces variétés. Parmi ces caractéristiques, la dimension intrinsèque ou les composantes connexes sont deux invariants topologiques bien connus. La connexité fait partie d'une famille d'invariants plus large dont les nombres de Betti sont des indicateurs numériques. L'intérêt de l'analyse topologique est qu'elle fournit une analyse qualitative là où la géométrie et les probabilités fournissent une analyse plus quantitative de la population [Car08]. Cette analyse qualitative s'affranchit de nombreuses distortions que la chaîne de mesure fait subir au signal issu du phénomène observé, par exemple certains invariants topologiques comme les nombres de Betti ne dépendent pas du système de coordonnées et sont robustes aux homotopies, tranformations qui incluent les isomorphismes, les similarités, et les homéomorphismes. Ainsi dans l'expérience que nous menons, il est possible de détecter dans un ensemble d'images quelles sont celles d'un même objet prises sous différents angles de rotation formant un cycle complet. Cette approche topologique fournit de nouveaux moyens complémentaires de ceux de l'analyse exploratoire géométrique ou statistique, pour caractériser et classer les phénomènes à partir d'un ensemble d'observations. Par exemple, des travaux récents utilisent l'analyse topologique combinée à un classifieur automatique pour la modélisation du cerveau [LKC+12] ou avec une méthode à noyaux pour retrouver la forme d'un circuit de course à partir de la position des voitures durant la course [PEKK12].

Les modèles de carte auto-organisée [Koh89] et

leur pendant génératif comme le Generative Topographic Map [BSW98] ou le Generative Principal Manifold [Tib92] ont été proposés depuis longtemps dans le domaine de l'apprentissage automatique, mais ces modèles imposent la topologie a priori en supposant une composante connexe de dimension intrinsèque homogène de valeur 1 ou 2 en général. Le Graphe Génératif Gaussien [Aup05, GAG08] peut être vu comme la version générative du Topology Representing Network (TRN) [MS94], pour apprendre la connexité d'une population dans le cadre de l'apprentissage statistique. Il définit un sous-graphe de Delaunay pondéré dont les sommets sont des vecteurs prototypes positionnés sur le nuage de données par un Modèle de Mélange Gaussien. Ce graphe est convolué avec une fonction de densité gaussienne. Les sommets et arêtes du graphe sont les composantes du mélange, et les proportions sont optimisées pour maximiser la vraisemblance. Les arêtes avec les poids les plus faibles sont retirées du graphe, et seules les arêtes qui expliquent le mieux les données sont conservées dans le modèle. Cependant ce modèle ne gère pas les dimensions supérieures à 1, il ne modélise que la connexité simple, et ne peut détecter par exemple la présence d'un ou plusieurs trous dans une variété de dimension 2 (une surface usuelle). Certaines méthodes purement géométriques s'appuient directement sur les données et non sur des prototypes représentant ces données, en construisant un complexe simplicial de Vietoris-Rips [Vie26, Zom10] des données. Ces méthodes utilisent un méta-paramètre, une distance seuil qui permet de décider si deux points doivent être connectés ou non dans le complexe. Le moyen classique pour choisir la meilleure valeur de ce méta-paramètre est la persistance homologique [ZC]. Il s'agit d'une analyse multi-échelle, les invariants topologiques sont extraits du complexe simplicial pour différentes valeurs de la distance seuil, les invariants forment une signature topologique, la signature qui persiste pour le plus grand intervalle de valeurs de la distance seuil est considéré comme représentative de la topologie de la structure sous-jacente au nuage de données. Une autre technique appelée Witness Complex (WitC) utilise des prototypes comme sommets d'un complexe simplicial [dSC04]. Les sommets sont un sous-ensemble des données, placés de telle sorte que chaque sommet soit le plus éloigné possible des autres sommets. Deux sommets sont reliés s'ils sont les premier et deuxième plus proches voisins euclidien d'au moins une donnée. Ce critère est adapté pour relier k sommets et former ainsi un (k-1)-simplexe du Witness Complex. La persistance homologique peut aussi être utilisée avec le Witness Complex pour améliorer la qualité de l'estimation des invariants topologiques.

Dans cet article, nous introduisons un modèle génératif qui étend le Graphe Génératif Gaussien aux variétés de dimension intrinsèque supérieure à 1 ce qui permet d'extraire de nouveaux invariants topologiques auparavant inaccessibles avec une approche générative. Nous remplaçons le graphe de Delaunay par le complexe simplicial de Delaunay comme ensemble de variétés de base, et nous utilisons le critère BIC pour sélectionner le sous-complexe simplicial le plus vraisemblable pour expliquer les données. De celui-ci nous extrayons les nombres de Betti grâce à la bibliothèque *javaplex* [TVJA11]. Nous comparons notre approche avec celle du Witness Complex qui se base elle aussi sur des prototypes représentant les données.

Comme dit précédemment, notre méthode, le Complexe Simplicial génératif (CSG) utilise des prototypes que nous positionnons avec un Modèle de Mélange Gaussien [MP00], mais contrairement au Witness Complex qui est essentiellement géométrique, le CSG est statistique. Le CSG repose sur un objet élémentaire, appelé Simplexe Génératif : c'est à la fois un objet géométrique et une densité de probabilité. Il peut être vu comme la convolution d'une loi normale multivariée avec un simplexe. De cette façon, la loi normale multivariée classique est un 0-simplexe (un point) convolué avec un bruit gaussien. Une fois que nous avons défini cet objet de base, nous pouvons définir un Complexe Simplicial Génératif : c'est un ensemble de Simplexes Génératifs, de la même façon qu'un complexe simplicial est un ensemble de simplexes. Chacun des simplexes peut être vu comme une composante d'un modèle de mélange. En utilisant l'algorithme EM [DLR77], les poids sont optimisés dans le modèle de mélange pour maximiser la vraisemblance, et le critère BIC (Bayesian Inference Criterion) est utilisé pour retirer les composantes les moins probables du modèle [Sch78], et la méthode fournit en sortie un CSG qui explique le mieux les données. Les nombres de Betti du complexe simplicial sous-jacent sont calculés grâce à la bibliothèque javaplex. Le CSG est le premier modèle génératif qui permet d'extraire tous les nombres de Betti.

2 Background

2.1 Les nombres de Betti

Les nombres de Betti sont des invariants topologiques qui permettent de qualifier une variété. Ils sont à valeurs dans $\mathbb{N}$. b_k correspond au rang du k-ième groupe d'homologie de l'espace topologique

considéré. Pour une variété de dimension d, on a forcément pour tout $k > d$, $b_k = 0$ [Car08].

De manière moins formelle, le k-ième nombre de Betti correspond au nombres de représentations indépendantes de S_k, la sphère unité de dimension k, que l'on peut trouver dans la variété :

1. b_0 compte donc le nombre de composantes connexes de la variété.
2. b_1 compte le nombre de cycles indépendants de la variété : un "O" n'a qu'un seul cycle, un "8" a deux cycles, de même qu'un tore a deux cycles 1.
3. b_2 compte donc le nombre de volumes emprisonnés. Une sphère par exemple, emprisonne un volume, un tore aussi emprisonne un volume.

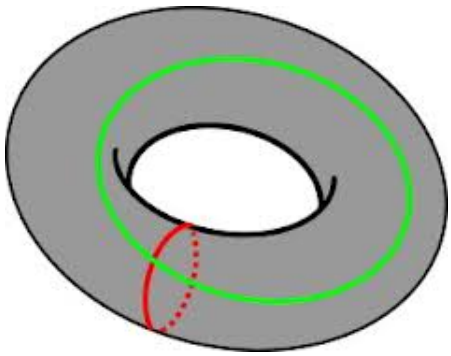

FIGURE 1 – Mise en évidence des deux cycles d'un tore. Il n'existe pas d'homéomorphisme transformant un des cycles en l'autre.

2.2 Le complexe simplicial

Un k-simplexe est l'objet géométrique le plus simple défini par $k+1$ points en position générale dans un espace de dimension k. "En position générale" signifie que les sommets sont linéairement indépendants, par exemple trois points ne peuvent pas être alignés, ou quatre points ne peuvent pas être coplanaires. Un 0-simplex est un point, un 1-simplexe est un segment, un 2-simplexe un triangle et ainsi de suite. Un k-simplexe σ est constitué de $k-1$-simplexes, eux-mêmes constitués de $k-2$-simplexes et récursivement. Par exemple un tétraèdre est fait de triangles qui sont fait de segments, ces derniers étant constitués de points. Tous les simplexes qui appartiennent à σ sont appelés ses facettes, et nous appelons F_σ un tel ensemble.

Un complexe simplicial S est un ensemble de simplexes qui vérifie deux conditions [**?**] :

1. si un simplexe σ appartient à S, alors tout $\alpha \in F_\sigma$ appartient aussi à S.
2. si deux simplexes $\sigma_1, \sigma_2 \in S$, alors $\sigma_1 \cap \sigma_2 \in S$.

2.3 Le complexe de Delaunay

En deux dimensions, le complexe de Delaunay est aussi appelé triangulation de Delaunay. Pour qu'un triangle appartienne à la triangulation, son cercle circonscrit ne doit pas contenir de points autres que les sommets du triangle en question. Pour les dimensions supérieures, la notion de triangle est généralisée par le simplexe. Pour qu'un simplexe appartiennent au complexe de Delaunay, sa sphère circonscrite ne doit contenir que les sommets du simplexe. Contrairement au complexe de Vietoris-Rips, le complexe de Delaunay vérifie toujours la condition 2 de la définition d'un complexe simplicial (il n'y a pas de facettes de simplexes qui se croisent dans un complexe de Delaunay, seulement des simplexes reliés par une facette commune).

La fonction MATLAB *delaunayn* fournit un complexe de Delaunay complet, c'est-à-dire tous les simplexes de dimension D, où D est la dimension de l'espace ambiant. Pour des dimensions faibles, cette fonction est très utile, mais pour $D \geq 5$, la complexité est telle qu'il devient trop long de l'exécuter. Le défaut de cette méthode est de ne renvoyer que le complexe qui correspond à la dimension de l'espace ambiant, alors que bien souvent, un premier travail sur le graphe (donc uniquement les 1-simplexes) ou les triangles (le complexe formé par les 1 et 2-simplexes) permet déjà d'élaguer un grand nombre de simplexes qui ne sont pas pertinents dans le modèle. C'est pourquoi nous utilisons l'algorithme *delaunay* décrit dans [ML13]. Pour un ensemble de sommets $\underline{w}$ dans $\mathbb{R}^D$, $delaunay(\underline{w}, d)$ renvoie l'ensemble des simplexes de dimension $d \leq D$ qui appartiennent au complexe de Delaunay de $\underline{w}$. Cela permet de travailler dimension par dimension, et de faire croître d pas à pas, jusqu'à ce que $d = D$. On peut même définir un critère d'arrêt pour s'arrêter pour une valeur de d plus petite.

2.4 Les modèles de mélange

Un modèle de mélange est une distribution de probabilité. Les mélanges sont utilisés à chaque fois qu'un échantillon ne peut pas être décrit simplement à partir d'un modèle simple de distribution de probabilité classique, mais plutôt comme une union de sous-populations, ces sous-populations pouvant elles, en revanche, être décrites par un modèle simple et

classique. Le modèle souvent utilisé pour ces sous-populations et que nous utiliserons dans ce papier est le modèle gaussien g. Chaque sous-population a sa propre moyenne et peut avoir sa propre variance selon les hypothèses, et bien sûr son propre poids dans le modèle. Donc chaque population P_k a une moyenne μ_k, une variance σ_k^2 et un poids π_k. Alors la distribution p associée à ce mélange de populations est 2 :

$$p(x) = \sum_{k=1}^{N} \pi_k g(x|\mu_k, \sigma_k) \quad (1)$$

$$\forall k, \pi_k \geq 0 \quad (2)$$

$$\sum_{k=1}^{N} \pi_k = 1 \quad (3)$$

La double contrainte sur les π_k assure que p est une distribution de probabilité : elle est positive et son intégrale sur tout le domaine fait.

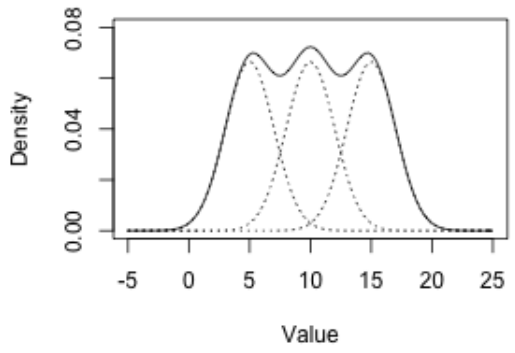

FIGURE 2 – Exemple de mélange de gaussienne

3 Le Complexe Simplicial Génératif

Dans cet article, nous supposons que les données sont des vecteurs de $\mathbb{R}^D$, échantillonnés d'après une collection de variétés, perturbés par un bruit centré gaussien, dont la variance σ^2 est inconnue. Nous supposons que cette collection de variétés peut être approchée par un complexe simplicial de Delaunay dont les sommets $\underline{w}$ appartiennent à $\mathbb{R}^D$. Nous définissons le Complexe Simplicial Génératif (CSG) en tant que modèle, et utilisons l'algorithme EM [DLR77] pour optimiser ses paramètres et maximiser sa vraisemblance.

3.1 Le Simplexe Génératif

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

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

Cela peut être vu comme la convolution d'une loi normale multivariée avec un 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.

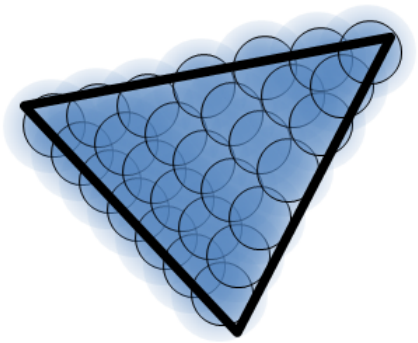

FIGURE 3 – Exemple de triangle génératif

La figure 3 montre un triangle génératif : chaque cercle représente une distribution gaussienne, ils partagent tous la même variance, et la probabilité résultante est la moyenne des contributions de chaque distribution gaussienne.

3.2 Du Simplexe Génératif au Complexe Simplicial Génératif (CSG)

Un Complexe Simplicial Génératif est un mélange de simplexes génératifs.

Soit π_i^d la proportion du simplexe S_i^d dans le modèle de mélange, g_i^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 CSG p est défini par :

$$p(x) = \sum_{d=0}^{D} \sum_{i=1}^{N_d} \pi_i^d g_i^d(x, \sigma^2) \quad (4)$$

$$\forall (i,d), \pi_i^d \geq 0 \quad (5)$$

$$\sum_{d=0}^{D} \sum_{i=1}^{N_d} \pi_i^d = 1 \quad (6)$$

Le CSG a une double nature : en tant que complexe simplicial, c'est un objet géométrique, en tant que modèle génératif, c'est un objet statistique. Le but est de maximiser la vraisemblance (une information statistique) à l'aide de l'algorithme Expectation-Maximization pour obtenir l'information topologique à partir de l'objet géométrique.

3.3 La sélection des composantes

Un complexe de Delaunay dépend uniquement de la position des sommets $\underline{w}$. Même s'ils ne sont pas calculés, tous les simplexes du complexe de Delaunay sont fixés une fois que la position des sommets a été définie. On peut donc considérer que le CSG élague un complexe de Delaunay pré-établi en retirant les simplexes qui correspondent le moins aux données.

Comme dit précédemment, le but est d'optimiser un critère statistique pour obtenir la topologie. Si ce critère était simplement la vraisemblance, le modèle sélectionnerait beaucoup trop de composantes, puisque la vraisemblance est une fonction croissante du nombre de composantes. Il y a N_s composantes au départ dans le modèle, il faut en garder seulement $\hat{N} \leq N_s$, pour que $\hat{N}$ soit optimal pour un critère qui tient compte de la vraisemblance, mais aussi du nombre de composantes. Le critère BIC (Bayesian Inference Criterion) [RW97] donne une bonne réponse pour les modèles de mélange gaussien, puisqu'il est un estimateur consistent. Comme le CSG dérive d'un mélange gaussien, nous supposons que le critère BIC fournira aussi une bonne réponse. La fonction Π suivante retourne les K plus π_i^d d'un modèle CSG :

$$\begin{array}{rccc} \Pi & : & [\![1; N_s]\!] & \rightarrow & [0,1]^{\mathbb{N}} \\ & & k & \mapsto & \{\text{k greatest } \pi_i^d\} \end{array}$$

Ce qui nous permet de définir $M_k = \{\underline{w}, S_i^d | \pi_i^d \in \Pi(k)\}$ un complexe simplicial qui ne contient que les simplexes dont le poids π_i^d est le plus fort. Pour $0 \leq k \leq N_s$, le BIC est calculé pour chaque M_k. BIC est relié à la vraisemblance par la formule suivante :

$$log\mathcal{L}(\underline{x}; \sigma; M_k) = \sum_{i=1}^{M} log(p(x_i; M_k, \sigma))$$

$$BIC(M_k) = log\mathcal{L}(\underline{x}; \sigma; M_k) - \frac{\nu_k}{2} log(M)$$

où M est le nombre de données, ν_k est le nombre de paramètres libres dans le CSG associé à M_k. Les paramètres libres sont les coordonnées des sommets $k \times D$, les poids π_i^d non nuls (qui correspondent aux simplexes gardés dans le modèle), moins un puisque la somme des poids est fixée à 1, et le paramètre de variance. On définit alors :

$$\hat{N} = \arg\max_k BIC(M_k)$$

3.4 L'algorithme associé au CSG

Algorithm 1 Algorithme général

Input : data X
Initialization $prototypes = GMM + BIC(data)$
$dimension = 0$
$CSG = Delaunay(prototypes, dimension)$
$continue = TRUE.$
while *continue* **do**
  $dimension++$
  $CSG = \{CSG, Delaunay(prototypes, dimension)\}$

  $CSG = EM(CSG)$
  $CSG = selectionBIC(CSG)$
  if $CSG(dimension).pi == 0$ **then**
    $continue = FALSE$
  end if
end while
return : $Betti(GSC)$

3.4.1 GMM+BIC

La toute première étape de l'algorithme consiste à initialiser la position des sommets. Un modèle de mélange gaussien est utilisé pour optimiser ces positions. Le nombre de sommets choisis se fait par maximisation du critère BIC.

3.4.2 EM

La fonction EM se réfère à l'algorithme Expectation-Maximization. Il est utilisé pour maximiser la vraisemblance du modèle et renvoie des valeurs optimales pour les π_i^d et la variance σ^2 grâce aux formules suivantes :

– Initialisation :

$$\pi_i^d = \frac{1}{N_s}$$

où

$$N_s = \sum_{d=0}^{D'} N_d$$

Étape Expectation :

$$z_{mi}^d = \frac{\pi_i^d g_i^d(x_m;\sigma^2)}{\sum_{l=0}^{D'}\sum_{j=1}^{N_l}\pi_j^l g_j^l(x_m;\sigma^2)} \quad (7)$$

Étape Maximisation :

$$\pi_i^d \leftarrow \frac{1}{M}\sum_{m=1}^{M} z_{mi}^d \quad (8)$$

$$\sigma^2 \leftarrow \frac{1}{DM}\sum_{m=1}^{M}\sum_{d=1}^{D'}\sum_{i=1}^{N_d} z_{mi}^d V_{mi}^d \quad (9)$$

où

$$V_{mi}^d = \begin{cases} (x_m - S_i^d)^2 \text{ if d} = 0 \\ \dfrac{1}{g_i^d(x_m;\sigma^2)|S_i^d|}\displaystyle\int_{S_i^d} g^0(x_m|t;\sigma)(x_m-t)^2 dt \end{cases}$$

Ce sont les définitions analytiques de ces grandeurs, mais g_i^d étant une intégrale multiple d'une gaussienne, il n'existe pas de forme analytique pour la calculer rapidement. L'approximation numérique est donc faite avec une méthode de Quasi-Monte Carlo et une séquence de Halton [MC95] : pour un simplexe S_i^d, une suite pseudo-aléatoire $\underline{s} = \{s_k \in S_i^d\}$ de points tirés dans le simplexe sont générés. Bien sûr pour un sommet, un seul point suffit à l'approcher, et un 1-simplexe a besoin de moins de points qu'un 2 ou 3-simplexe pour avoir une approximation précise. Mais le nombre de points générés dans le simplexe doit être fonction de la dimension : si 10 points sont utilisés pour approximer un segment, il en faut 100 pour approximer un carré avec la même précision et 1000 pour un cube. Pour un simplexe de dimension d, nous avons donc choisi de générer une séquence de $K = \binom{d+9}{d}$ points. Ce qui donne l'approximation numérique suivante :

$$g^d(x|S_i^d;\sigma^2) \approx \frac{1}{K}\sum_{k=1}^{K} g^0(x|s_k,\sigma^2) \quad (10)$$

$$V_{mi}^d \approx \frac{1}{g_i^d(x_m;\sigma)K}\sum_{k=1}^{K} g^0(x_m|s_k;\sigma^2)(x_m-s_k)^2 \quad (11)$$

3.4.3 selectionBIC

L'entrée de la fonction *selectionBIC* est un GSC. Le nombre optimal de composantes $\hat{N}$ est choisi avec la formule introduite dans la section 3.3. Puis *selectionBIC* renvoie le CSG élagué qui correspond au modèle $M_{\hat{N}}$.

3.4.4 Betti

La fonction *Betti* est fournie par un un outil créé par des spécialistes de la topologie algorithmique : Plex. Il s'agit d'une librairie pour MATLAB qui prend un complexe simplicial en entrée et retourne ses nombres de Betti en sortie.

4 Expériences

4.1 Objets topologiques connus

Pour les premières expérimentations, des objets facilement obervables 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 CSG sera comparé au Witness Complex tel qu'implémenté dans le plugin pour 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'éxecution du Witness Complex, on procède à une filtration. 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 "infiniteBarcodes" de Javaplex permet de retourner la topologie qui persiste le plus longtemps au cours de la croissance du diamètre des boules.

4.1.1 La sphère

Pour la sphère, 1000 points ont été tirés d'une sphère de dimension 3 et de rayon 1. Ces points sont ensuite corrompus avec un bruit gaussien. Trois écarts-types différents ont été utilisés : 0.05, 0.1, 0.2. Le CSG et WitC ont été éxecutés 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.

4.1.2 Le tore

Pour le tore, 2000 points ont été tirés d'un tore de dimension 3 et de grand rayon $R = 10$ et de petit rayon $r = 3$. Ces points sont ensuite corrompus avec un bruit gaussien. Trois écarts-types différents ont été utilisés : 0.01, 0.05, 0.1. Le CSG et WitC ont été éxecutés 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 2 1 0 ....) sont obtenus.

4.2 La bouteille de Klein

Bien que ce soit une variété de dimension intrinsèque 2, une bouteille de Klein n'existe que dans un espace de dimension au moins 4, et elle est plus facilement représentée 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 : $\sigma = 0.01$ and 0.05. Pour chaque σ et chaque algorithme, l'expérience a été répétée 100 fois. Les résultats sont des pourcentages de bonne réponse : les bons nombres de Betti de la bouteille de Klein sont (1 1 0 0 0 ...).

4.3 Jeu de données réelles : COIL-100

COIL-100 est un corpus d'images disponbile en ligne [**?**]. 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 noir et blanc. 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 dimension par dimension : en effet le graphe est assez rapide à obtenir, de même que les triangles. Le critère d'arrêt termine très souvent l'algorithme après la première ou deuxième étape (dimension 2).

FIGURE 4 – Les 60 objets de la base COIL-100 analysés

Comme ce jeu de données n'est pas un objet connu et maîtrise, 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, à la fin. Il s'agit bien d'une ligne fermée, ie un cycle. On considère donc que les nombres de Betti corrects sont $(1, 1, 0, ...)$. L'algorithme a été lancé 100 fois sur 60 objets de la base de données. Les nombres de Betti retenus pour chaque objet sont ceux qui sont sortis le plus souvent parmi les 100 résultats.

On peut voir les 60 objets en question sur la figure 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 et de jaut en bas ensuite jusqu'à 60.

4.4 Résultats

4.4.1 La sphère

On peut trouver les résultats de la sphère dans le tableau 5. Pour les variances les plus faibles, WitC domine légèrement le CSG, mais ils sont tous les deux fiables. Pour $\sigma = 0.2$, 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 ajouter un cycle non-voulu dans le modèle

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

FIGURE 5 – 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

4.4.2 Le tore

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 n'est pas creux. Dans ce cas une surface peut être ajoutée dans le modèle à l'intérieur du tore, et ou deux cycles indépendants apparaissent. La même chose 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 n'est pas du tout significatif. 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 le nombre de composantes connexes et de cavités sont corrects.

	WitC	GSC
$\sigma = 0.01$	5%	63%
$\sigma = 0.05$	8%	60%
$\sigma = 0.1$	9%	57%

FIGURE 6 – Success rate of extracting the Betti numbers of a torus made of 2000 points corrupted with a noise σ

4.4.3 La bouteille de Klein

Comme dit précédemment, retrouver la topologie de la bouteille de Klein est un challenge. Sur les 100 essais, WitC ne trouve pas une seule fois les bons nombres de Betti. Pendant ce temps le CSG produit quand même des résultats raisonnables. Le CSG peut être considéré comme un algorithme robuste vis-à-vis de certains changements de topologie. Que ce soit l'addition de cycles indépendants ou le fait que la surface soit non-orientable n'affecte significativement ses résultats.

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

FIGURE 7 – Taux de succès d'extraction des nombres de Betti pour une bouteille de Klein faite de 650 points avec un bruit σ

4.4.4 COIL-100

Les résultats obtenus par le CSG et WitC pour ces 60 images peuvent être classés dans plusieurs grandes familles :

- $(1, 1, 0...)$ qui correspond au cycle attendu, c'est le résultat le plus fréquent pour le CSG.
- $(1, 2, 0...)$ qui est le deuxième résultat le plus fréquent pour GSC et le plus fréquent pour WitC. Une composante connexe et deux cycles, c'est une structure proche de celle d'un "8".
- $(1, 0, 0...)$, une seule composante connexe homéomorphe à un point.
- $(1, n, 0...)$, on peut voit 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	GSC
$(1, 1, 0...)$	6	17
$(1, 2, 0...)$	8	15
$(1, n, 0...)$	33	22
$(1, 0, 0...)$	0	1
$(2, n, 0...)$	13	5

FIGURE 8 – 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 8, le GSC trouve plus souvent les nombres de Betti attendus a priori que WitC. Dans 17 cas sur 60, GSC 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 provoquer 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 les iden-

tifient 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.

5 Conclusion et perspectives

Dans cet article, le Complexe Simplicial Génératif a été comparé à l'algorithme de l'état de l'art Witness Complex. Le Witness Complex a donné de meilleurs résultats sur un objet topologique simple et très régulier comme la sphère, mais a enregistré une baisse significative de performance sur d'autres objets, alors que celles du CSG ont légèrement diminué quand la complexité de l'objet a augmenté, comme le tore ou la bouteille de Klein. Le CSG permet donc d'extraire correctement les nombres de Betti d'une variété échantillonnée et bruitée, il peut être considéré comme robuste et fiable. Bien que le procédé de filtration est très puissant si certaines caractéristiques persistent longtemps comme la cavité à l'intérieur d'une sphère. Dans le CSG, le paramètre qui correspond le plus à ce procédé de filtration est la variance. Il est plus fiable dans le cas du tore ou de la bouteille de Klein : l'algorithme est moins susceptible de mal interpréter une caractéristique topologique qui n'est en fait qu'un artefact.

Pour ce qui est des images, il n'y avait pas de bonne réponse connue a priori, même si l'on pouvait en supposer une. Cette réponse a été trouvée le plus souvent par le GSC, qui a donc de meilleurs résultats que WitC sur des données réelles. Pour les données réelles, l'intérêt des nombres de Betti ne réside pas forcément dans le résultat en lui-même, car comme on l'a vu l'interprétation n'en est pas forcément aisée. En revanche ils peuvent servir de données supplémentaires pour caractériser un jeu de données, comme entrée dans un algorithme de classification par exemple.

Il faut tout de même noter que l'exécution du Witness Complex est bien plus rapide que celle du CSG. Et certaines améliorations peuvent être proposées pour obtenir de meilleurs résultats sur le tore et la bouteille. L'algorithme Witness Complex est aussi capable de traiter des grandes dimensions, ce qui n'est pas encore le cas du GCS.

Dans des travaux futurs, nous souhaitons travailler sur le problème de la différenciation des surfaces orientables et non-orientables : la bouteille de Klein présentée dans cet article a les mêmes nombres de Betti qu'un cylindre classique, alors que ces deux variétés sont différentes. Une autre caractéristique topologique, appelée *torsion* permet de faire cette distinction. Pour finir, tous les objets étudiés dans cet article n'ont qu'une composante connexe. Que se passe-t-il si l'on trouve plusieurs composantes ? Les nombres de Betti ne décrivent plus une unique variété, et il faut donc adapter notre algorithme à ce cas-là.

Références

[Aup05] Michaël Aupetit. Learning topology with the generative gaussian graph and the em algorithm. In *NIPS*, 2005.

[BSW98] Christopher M. Bishop, Markus Svensén, and Christopher K. I. Williams. Gtm : The generative topographic mapping. *Neural Computation*, 10(1) :215–234, 1998.

[Car08] Gunnar Carlsson. Topology and data. Technical report, Department of Mathematics, Stanford University, 2008.

[DLR77] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Royal Statistical Society, Series B*, 39(1) :1–38, 1977.

[dSC04] V. de Silva and G. Carlsson. Topological estimation using witness complexes. *IEEE Symposium on Point-based Graphic*, pages 157–166, 2004.

[GAG08] Pierre Gaillard, Michaël Aupetit, and Gérard Govaert. Learning topology of a labeled data set with the supervised generative gaussian graph. *Neurocomputing*, 71(7-9) :1283–1299, 2008.

[Koh89] T. Kohonen. *Self-organization and associative memory : 3rd edition*. Springer-Verlag New York, Inc., New York, NY, USA, 1989.

[LKC+12] Hyekyoung Lee, Hyejin Kang, Moo K. Chung, Bung-Nyun Kim, and Dong Soo Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE*

Trans. Med. Imaging, 31(12) :2267–2277, 2012.

[MC95] William J. Morokoff and Russel E. Caflisch. Quasi-monte carlo integration. *JOURNAL OF COMPUTATIONAL PHYSICS*, 122 :218–230, 1995.

[ML13] Juan Mendez and Javier Lorenzo. Computing voronoi adjacencies in high dimensional spaces by using linear programming. In *Mathematical Methodologies in Pattern Recognition and Machine Learning*, pages 33–49. Springer, 2013.

[MP00] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley series in probability and statistics : Applied probability and statistics. John Wiley & Sons, 2000.

[MS94] Thomas Martinetz and Klaus Schulten. Topology representing networks. *Neural Netw.*, 7(3) :507–522, March 1994.

[PEKK12] Florian T. Pokorny, Carl Henrik Ek, Hedvig Kjellström, and Danica Kragic. Persistent homology for learning densities with bounded support. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1826–1834. 2012.

[RW97] Kathryn Roeder and Larry Wasserman. Practical bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, 92(439) :894–902, 1997.

[Sch78] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2) :461–464, 1978.

[Tib92] Robert Tibshirani. Principal curves revisited. *Statistics and Computing*, 2 :183–190, 1992.

[TVJA11] Andrew Tausz, Mikael Vejdemo-Johansson, and Henry Adams. Javaplex : A research software package for persistent (co)homology. Software, 2011.

[Vie26] L. Vietoris. Über den höheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen. *Mathematische Annalen*, 1926.

[ZC] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete Comput. Geom*, 33 :249–274.

[Zom10] Afra Zomorodian. Fast construction of the vietoris-rips complex. *Computers & Graphics*, 34(3) :263–271, 2010.