Introduction à un domaine de recherche : Analyse topologique des données et réduction de dimension

Antoine Commaret

2 Octobre 2020

Table des matières

1 Intoduction

Avec l'apparition d'appareils informatiques toujours plus puissants, on peut collecter d'énormes jeux de données (tendance surnommée *Big Data* dans les médias). Pour les traiter, on utilise des algorithmes issus du monde de l'apprentissage automatique (*Machine Learning*, raccourci en *ML*). L'expérience montre toutefois les limites de ces méthodes, même les plus simples, lorsque le jeu de données se trouve dans un espace de grande dimension $\mathbb{R}^m$ où m est très grand. Il n'est pas question, par exemple, de créer un algorithme de reconnaissance d'animaux sur un jeu de données de 10^4 images composées de plus de 10^6 pixels. En effet, ces images ne sont que des points dans $\mathbb{R}^{10^6}$, et sans informations supplémentaires sur celles-ci, il y a bien trop de degrés de liberté pour être sûr d'obtenir des résultats corrects.

D'autres difficultés apparaissent : effectuer une descente de gradient automatique en haute dimension est d'une grande complexité, étant donné qu'il faut effectuer m approximations de dérivées à chaque étape. Les comportements singuliers apparaissant en grande dimension sont appelés malédiction de la dimension.

Pour résoudre ces difficultés, il y a plusieurs approches :

— Tout simplement augmenter la taille du jeu de données. La puissance actuelle des ordinateurs (plus précisément, des GPU dédiés aux méthodes d'apprentissage profond) rend possible l'utilisation des algorithmes au prix de journées de calculs. Le défi restant consiste à disposer d'une base de données assez importante pour compenser le fléau de la dimension, pas toujours de façon légale...
— Supposer que les données ne sont pas simplement réparties "au hasard" dans $\mathbb{R}^m$. On peut pour cela construire des modèles mathématiques et formuler des *a priori* menant à de l'inférence bayésienne.
— Déterminer les dimensions intrinsèques du jeu de donnée, c'est-à-dire établir sur quelle sous-variété de $\mathbb{R}^m$ le jeu de données se trouve. On parle d'apprentissage de métriques puisqu'il s'agit d'appliquer les méthodes classiques en considérant les distances géodésiques plutôt que la distance euclidienne sur $\mathbb{R}^m$.
— Tenter de réduire la complexité du jeu de données en le faisant vivre dans $\mathbb{R}^d$, où $d \ll m$. La perte d'information est inévitable, le défi étant alors de réussir à en conserver un maximum. L'objectif est parfois de rendre compte de la sous-variété latente, comme dans la technique précédente. Le traitement sur des données de $\mathbb{R}^d$ est cependant beaucoup plus simple qu'un traitement géodésique. Dans l'exemple ci-dessous, dit "Rouleau Suisse" (*Swiss Roll*), on associe à chaque point du jeu initial dans $\mathbb{R}^3$ un point dans $\mathbb{R}^2$, en conservant un maximum les propriétés.

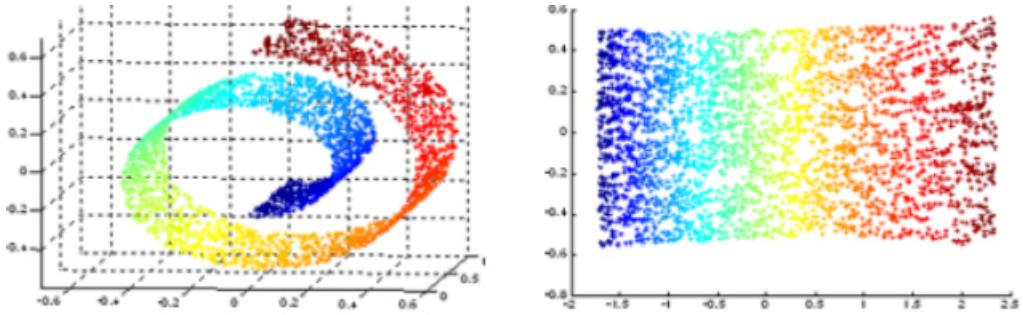

FIGURE 1 – Rouleau Suisse après application de l'algorithme IsoMAP

D'autre part, bien que principalement motivées par le traitement de grands jeux de données, les techniques de réduction de dimension peuvent aussi être utilisées pour visualiser de grands jeux de données. Dans l'exemple ci-dessous, une étude de la distance syntaxique entre des articles de philosophie contemporaine permet de visualiser rapidement les relations entre différents courants, déjà connus.

Si les procédés de visualisation sont souvent impossibles à rigoureusement interpréter, ils permettent de rapidement mieux comprendre un grand jeu de données. Dans ce cadre, ils sont notamment utilisés en génomique [3], [8], [12], parfois pour visualiser l'effet d'algorithmes de d'apprentissage profond [9]...

2 Outils mathématiques

Les méthodes de réduction de dimension font appel à des idées variées et utilisent des outils bien différents suivant les buts qu'elles se donnent. Ils sont souvent élémentaires, mais deux sortent du lot.

2.1 Théorie géométrique de la mesure et statistiques

Une hypothèse courante est celle d'un tirage à densité μ sur une sous-variété $\mathcal{V}$ de $\mathbb{R}^m$ dont le jeu de données M n'est qu'un simple échantillon. Dans ce cadre, un algorithme de réduction de

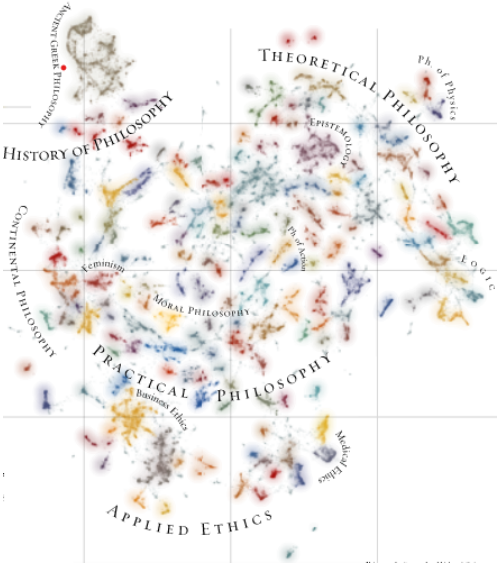

FIGURE 2 – Carte de la philosophie contemportaine après application de l'algorithme UMAP

dimension cherchera à vérifier que son fonctionnement est asymptotiquement correct lorsqu'on tire un grand nombre de points.

Pour ce faire, un outil d'analyse *locale* du jeu de donnée est simplement d'étudier le voisinage de chaque point, soit dans un rayon prescrit, soit en prenant en compte les plus proches voisins. À cette fin, on peut considérer la fonction de répartition multi-dimensionnelle de tirage autour d'un point $x \in \mathcal{V}$:

$$v_x(t) = \int_{\mathcal{V} \cap B_x(t)} \mathrm{d}\mu$$

La monographie de Biau et Devroye [2] décrit en détail les propriétés qu'on obtient avec grande probabilité lorsqu'on regarde les k plus proches voisins d'un tirage de n points selon μ. En particulier, si l'on note $U_{i,n}$ la variable aléatoire du i-ème plus petit tirage obtenu après avoir tiré uniformément n points dans $[0,1]$, on sait que les variables aléatoires conjointes $U_1, \dots, U_n$ sont de même loi qu'un tirage uniforme dans le simplexe

$$A_n = \{(x_1, \dots x_n) \in \mathbb{R}^n : 0 \leq x_1 \leq \cdots \leq x_n \leq 1\}$$

Si l'on note d_i^x la distance du i-ème plus proche voisins de x, il vient les lois conjointes

$$(d_1^x, \dots, d_n^x) \sim v_x^{-1}(U_1), \dots, v_x^{-1}(U_n)$$

Ces considérations permettent, à terme, de montrer qu'on obtient un estimateur de la densité de μ autour de ce point lorsque $\log(n) = O(k), k = o(n)$ et que $n \to +\infty$. On connaît de même la fonction de répartition des d_i^x grâce à des binomiales :

$$\mathbb{P}(d_i^x > t) = \sum_{j=0}^{i-1} \int_0^t \binom{n}{j} (1 - v_x(t))^{n-j} v_x(t)^j \, \mathrm{d}t$$

On a pu calculer la moyenne des tirages dans un cadre simple :

Proposition 2.1. *Dans le cadre d'un tirage d'une boule de dimension d de volume $\frac{1}{a}$, l'espérance se calcule :*

$$\mathbb{E}(d_{k+1} - d_1) = \frac{(an)^{-1/d}}{d} \cdot \exp(\gamma_d - \gamma/d) \cdot \frac{P_k(1/d) - 1}{1/d} \cdot (1 + o_{d,n}(1))$$

où $\gamma_d = \sum_{j \geq 1} \frac{1}{dj} - \ln(1 + \frac{1}{dj})$, $o_{d,n}(1)$ ne dépend pas de k et où $P_i(X) = \prod_{j=1}^{i} (1 + \frac{X}{j})$

Sur une sous-variété de dimension d, on a $v_x(t) = at^d + o(1)$, où a est à une constante multiplicative près la densité de μ en x. Si l'on veut être plus précis, on utilise la théorie géométrique de la mesure, initiée par Herbert Federer dans les années soixante. En particulier, la thèse d'Eddie Aamari [1] fournit des estimations concrètes. Il en ressort en particulier que

$$\left|v_x(t) - at^d\right| = O(t^{d+2})$$

lorsque la densité est localement constante et que la variété est au moins C^2.

Cela permet d'obtenir un encadrement quantitatif, bien que compliqué :

Proposition 2.2. *Supposons qu'on ait une borne inférieure relative :*

$$\mathbb{E}(d_k^\mu - d_{k-1}^\mu)(1 - H(z)) \leq \int_0^z \psi(v_x(t))\,\mathrm{d}t \leq \mathbb{E}(d_k^\mu - d_{k-1}^\mu)$$

Alors il existe deux fonctions h, h^* décroissant plus vite que tout polynôme en ∞ telles que pour tout $z \in \mathbb{R}$:

$$\frac{1 - h^*(z)}{1 + \max_{t\leq z} |Q(t)|} \leq \frac{\mathbb{E}(d_k^\mu - d_{k-1}^\mu)}{a^{-1/d}\mathbb{E}(d_k - d_{k-1})} \leq \frac{(1 - H(z))^{-1}}{1 - \max_{t\leq z} |Q(t)|}$$

où $Q(t)$ est fonction polynomiale de l'erreur relative de $v_x(t)$ par rapport à la distribution dans une boule, telle que $Q = 0$ si et seulement si $v_x(t)$ est la fonction de répartition multidimensionnelle d'une boule de dimension d, où d_i est le i-ème tirage plus proche de zéro dans une boule de volume 1, et a tel que $v_x(t) \sim at^d$.

Ce résultat provient de la grande concentration de la masse de $v_x(t)^{n-i}(1 - v_x(t))^i$ dans les petites valeurs de t ; il s'agit ensuite d'estimer l'erreur d'approximation dans le calcul de 2.1 pour estimer une fonction h qui convient. Enfin, on obtient le résultat final en reparamérant les intégrales considérées pour mieux comparer v_x à son étalon.

2.2 Homologie Persistante

L'Homologie persistante est un outil classique de l'analyse topologique des données. Elle permet d'étudier la topologie d'un nuage de points. Reprenons un nuage de point M dans un espace euclidien $\mathbb{R}^m$. L'Analyse Topologique des Données (dont le livre [7] de 1987 est une bonne introduction) utilise le théorème du nerf pour calculer les nombres de Betti de l'ensemble

$$M_\varepsilon = \bigcup_{x\in M} B_x(\varepsilon)$$

et en observer l'évolution lorsqu'on fait varier ε dans $[0, +\infty[$. Pour se faire, on utilise le très pratique formalisme des complexes simpliciaux. La suite croissante des complexes simpliciaux associés au nuage de points M est appelée *Filtration de Čech*. Le simplexe $[x_1, \dots x_i]$ appartient au complexe associé à M_ε si et seulement si $\cap_{i=1}^n B_{x_i}(\varepsilon) \neq \emptyset$. On peut alors noter l'apparition et la disparition de composantes connexes/ de cycle via des *barcodes*, comme ci-dessous :

On transforme les *barcodes* en *diagramme de persistance* en notant simplement les points par le rayon de naissance puis de mort - ce sont les termes consacrés - de l'invariant topologique dans la filtration de Čech.

La théorie de la persistance s'est développée depuis les années 2000, dans ses définitions via des considérations algébriques, et en développant des méthodes de comparaison. En particulier, on peut comparer deux modules de persistance via la distance d'appareillage (bottleneck en anglais) des points des diagrammes entre eux, en ajoutant la diagonale.

De nombreux résultats, comme [4] montrent que le diagramme de persistance associés à un nuage de points est robuste au bruit. En transformant des données topologiques - commes les nombres de Betti - en des diagrammes de persistance, qu'on peut quantitativement comparer

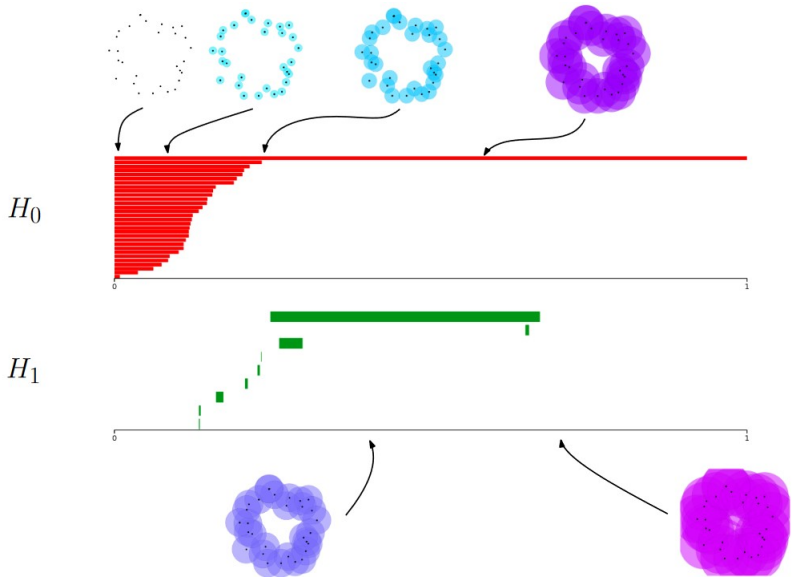

FIGURE 3 – Exemple de barcode, en dimension 0 et 1 emprunté à Raphaël Tinarrage

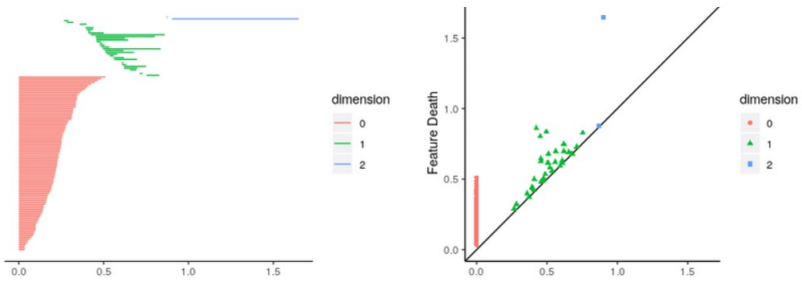

FIGURE 4 – Diagramme de persistance à partir d'un barcode

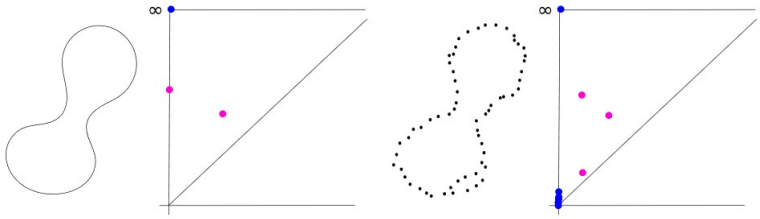

FIGURE 5 – Diagrammes de persistance semblables

3 Algorithmes existants

Il existe de très nombreux algorithmes de réduction de dimension, chacun ayant ses forces. Pour mieux comprendre les différents types de fonctionnement, voici une liste des principaux algorithmes suivant leurs propriétés. Outre les algorithmes On dira que le fonctionnement de l'algorithme est *parfait* lorsqu'il n'y a aucune perte d'information.

3.1 ACP

L'algorithme *d'Analyse en Composante Principale* est le plus ancien (théorisé dans la première moitié du XXème siècle par des statisticiens) des algorithmes de réductions de dimension. On lui fournit les coordonnées des points de M ainsi qu'une dimension d'arrivée et il renvoie des points de $\mathbb{R}^d$ placés de sorte à ce qu'ils représentent le sous-espace de dimension d qui préserve le plus les distances L^2 entre les points de M.

En particulier, le fonctionnement d'ACP est parfait si les points de M se trouvent appartenir à un même sous-espace affine de dimension d.

3.2 MDS

L'algorithme MDS, pour *Mulidimensionnal Scaling*, consiste à tenter de préserver un maximum le graphe $D = (d_{i,j})$ des distances en haute dimension et les points $y_1, \dots, y_n$ qui leur correspondent en dimension d. Pour ce faire, il cherche à minimiser la quantité

$$\left(\sum_{i<j} (d_{i,j} - ||y_i - y_j||)^2 \right)^{1/2}$$

En particulier, cet algorithme a un fonctionnement parfait lorsque la configuration de distance existe en dimension d ; si les distances sont euclidiennes MDS a le même fonctionnement parfait qu'ACP. Pourtant, ces deux algorithmes ont des sorties bien différentes dès qu'on sort de ce cadre.

3.3 IsoMAP

IsoMAP [13] suppose un tirage uniforme sur une sous-variété de $\mathbb{R}^m$. Il calcule les chemins les plus courts dans les voisinages de k-plus proches voisins autour de chaque point, permettant un calcul de la distance géodésique entre chaque point - en supposant que celle-ci est une ligne droite pour éviter une complexité démentielle. Il applique ensuite l'algorithme MDS pour retrouver la sous-variété "dépliée", c'est-à-dire avant d'être plongée en grande dimension. Plus rigoureusement, son fonctionnement parfait n'est garanti que lors d'un plongement depuis un ouvert convexe de $\mathbb{R}^d$ et s'écrit de la sorte :

Théorème 3.1. *Supposons que $\mathbb{V} = \psi(\Theta)$ où Θ est un ouvert convexe de $\mathbb{R}^d$, et ψ est une fonction localement isométrique de Θ dans $\mathbb{R}^n$. Pour tout $\varepsilon > 0$ si la densité des points sur mathcal$V$, est suffisamment grande, il existe un nombre de voisins k assez grand pour que le calcul des distances géodésiques soit correct à une erreur relative plus petite que ε avec grande probabilité, et p.s quand ce nombre points tend vers l'infini.*

Une analyse quantitative précise issue de la théorie géométrique de la mesure et de la géométrie algorithmique permet aux auteurs d'IsoMAP de donner un résultat puissant :

Corollaire 3.2. *Sous les hypothèses du théorème précédent, on peut retrouver les coordonnées isométriques de $\Theta \subset \mathbb{R}^d$ à une isométrie près.*

3.4 Hessian Eigenmaps

Hessian Eigenmaps [5] est un algorithme de 2003 ayant pour objectif de compléter IsoMAP en garantissant de revenir à Θ même si celui-ci n'est pas convexe. Pour cela, les auteurs définissent une forme quadratique sur toute variété $\mathcal{V}$ munie d'une mesure μ à densité strictement positive :

$$\mathcal{H}(f) = \int_{\mathcal{V}} ||H(f)||_2^2 \, \mathrm{d}\mu$$

où $H(f)$ est la Hessienne de f ; on peut en effet prouver que $||H(f)||$ est invariante aux isométries. On peut ensuite regarder le noyau de $\mathcal{H}$:

Théorème 3.3. *Supposons que $\mathbb{V} = \psi(\Theta)$ où Θ est un ouvert connexe de $\mathbb{R}^d$, et ψ est une fonction localement isométrique de Θ dans $\mathbb{R}^n$. Alors $H(f)$ a un noyau de dimension $d+1$ consitué des fonctions constantes et de l'espace engendré par les fonctions par les fonctions coordonnées dans $\mathbb{R}^d$.*

L'estimation des coordonnées locales mène au corollaire suivant :

Corollaire 3.4. *Sous les hypothèses du théorème précédent, on peut retrouver les coordonnées isométriques de $\Theta \subset \mathbb{R}^d$ à une isométrie près.*

Pour autant, Hessian Eigenmaps n'est pas la panacée de la réduction de dimension. Son utilisation nécessite un bon échantillonnage de la variété, sans quoi les estimations de la Hessienne sont très instables numériquement. On a besoin d'énormément de données à cause de la malédiction de la dimension. Les temps de calculs sont alors rédhibitoires.

3.5 TopoMAP

Cet algorithme [6] itératif très récent (2020) ne cherche pas à maintenir les voisinages. Il conserve le diagramme de persistence en dimension 0, c'est-à-dire qu'il maintient une identification entre la haute et la basse dimension sur l'évolution des composantes connexes le long de la filtration de Rips 3 lorsqu'on fait grossir M_ε. Il projette uniquement en dimension 2.

Les résultats ne sont pas satisfaisant, mais il ouvre la porte à de prochains algorithmes exploitant l'analyse topologique des données...

4 UMAP : l'État de l'Art de la réduction de dimension non-linéaire

4.1 Description

UMAP [11] est un algorithme de 2018 développé par Leland McInnes. Il est considéré (en 2020) comme l'état de l'art des algorithmes de réduction de dimension non-linéaires, remplaçant à ce titre t-SNE [10], datant de 2008, notamment grâce à une complexité moindre et des résultats mieux interprétables. Son auteur décrit de nombreuses heuristiques pour justifier qu'UMAP sait retrouver la géométrie sous-jacente aux données, même si elles connaissent une densité variable.

Cet algorithme souffre néanmoins d'un grand défaut pour un mathématicien, contrairement aux algorithmes décrits précédemment : il n'a *jamais* de fonctionnement parfait. C'est celui que j'ai étudié dans le cadre de mon stage de fin de M2.

UMAP construit un graphe pondéré à partir du nuage de points en haute dimension. Deux points x, y sont reliés si et seulement si parmi les points du nuage y fait partie des `n_neighbors` plus proches voisins de x, ou vice versa. `n_neighbors` est un paramètre global qu'on écrira souvent k une fois qu'il est fixé.

L'attribution des poids se fait d'abord dans le cadre d'un graphe orienté. En effet, si l'on étudie le voisinage du point x et que y fait partie des k plus proches voisins de x, on rajoute à $[x, y]$ à l'ensemble A des arêtes qu'étudie UMAP. Le poids $p(\overrightarrow{[x, y]})$ est aussi fonction des autres voisins de x : si l'on note $y_1, \dots, y_k$ les k plus proches voisins de x, rangés suivant leur distance $d_i = d(x, y_i)$ à x, on pose

$$p(\overrightarrow{[x, y_i]}) = \exp(-\frac{d_i - d_1}{\sigma_x})$$

où σ_x est choisi de sorte à ce que

$$\sum_{1 \leq i \leq k} \exp(-\frac{d_i - d_1}{\sigma_x}) = \log_2(k)$$

On effectue ensuite une opération de symétrisation sym entre les deux orientations pour obtenir un graphe symétrique pondéré de haute dimension. La version originale d'UMAP symétrise par exemple suivant l'union probabiliste :

$$\mathrm{sym}(x, y) = x + y - xy$$

mais on peut aussi prendre

$$\mathrm{sym}(x, y) = \sup(x, y).$$

Ensuite, UMAP initialise un même nombre de points en dimension d. Il construit un graphe pondéré en basse dimension, dont les arêtes sont les mêmes que celles de haute dimension, via la correpondance de points. Les poids des arêtes sont attribués en fonction de la distance et décroissant dès que la distance dépasse le paramètre global `min_dist` de la façon suivante :

$$\rho_{[x,y]} = \begin{cases} 1 & \text{si } ||x-y|| \leq \texttt{min\_dist} \\ e^{-(||x-y||_{\mathbb{R}^d} - \texttt{min\_dist})} & \text{sinon.} \end{cases}$$

UMAP compare le poids $\mu(a)$ en haute dimension de celui $\nu(a)$ en basse, pour $a \in A$, via la fonction d'*entropie croisée* :

$$\mathrm{entrop}(\mu(a), \nu(a)) = \mu(a) \ln\left(\frac{\mu(a)}{\nu(a)}\right) + (1-\mu(a)) \ln\left(\frac{1-\mu(a)}{1-\nu(a)}\right)$$

Cette quantité est positive et s'annule uniquement lorsque $\mu(a) = \nu(a)$.

Il effectue ensuite une descente de gradient sur le coût total

$$C(A, \mu, \nu) = \sum_{a \in A} \mathrm{entrop}(\mu(a), \nu(a))$$

pour actualiser la position des points en basse dimension.

4.2 Modèle

Voici un modèle ayant pour but de comprendre le fonctionnement de l'algorithme. On espérerait atteindre les mêmes résultats que Hessian Eigenmaps. Pour aider à la compréhension générale plutôt qu'à un fonctionnement parfait, on inverse le sens des flèches.

On suppose que M est un nuage de points inclus dans $\mathcal{U}$ une partie de $\mathbb{R}^m$ de mesure de Lebesgue finie. $\mathcal{O}$ est une partie de $\mathbb{R}^d$ de mesure de Lebesgue finie $f : \mathcal{U} \to \mathcal{O}$ est une fonction continue qu'on qualifiera, vue la perte de dimension, de projection. L'objectif est de donner une estimation de l'entropie croisée entre M et $f(M)$ fournie par UMAP. Idéalement, on voudrait que les points de $f(M)$ se comportent comme issus d'un tirage uniforme sur $\mathcal{O}$, dont on connaît rapidement le comportement via 2.1.

UMAP s'occupe uniquement des voisinages de chaque point. Pour se concentrer sur les voisinages, on peut considérer un recouvrement de $\mathcal{U}$ de p ouverts :

$$\mathcal{U} = \bigcup_{i=1}^{p} \mathcal{U}_i$$

Pour bien maîtriser les k plus proches voisins de chaque point, on impose quelques conditions sur le recouvrement :

Définition 4.1 (Point sympathique et k-recouvrement)**.** Étant donné un échantillon fini M de points d'une partie mesurable $\mathcal{U}$ de $\mathbb{R}^d$, et un recouvrement fini de $\mathcal{U} = \bigcup_{i=1}^{p} \mathcal{U}_i$, on dit qu'un point $x \in M$ est **k-sympathique** lorsqu'il existe un i tel que $x \in \mathcal{U}_i$ et tel que les k plus proches voisins de x soient aussi dans $\mathcal{U}_i$. On dit que ce recouvrement est un **k-recouvrement** lorsque tous les points de M sont k-sympathiques.

À la manière d'IsoMAP, on cherche un cadre de locale isométricité. La normalisation décrite précédemment n'est pas sensible aux multiples d'isométrie. On cherche à mesurer ce défaut d'isométricité :

Hypothèse 4.2 (Hypothèse de quasi-isométricité)**.** On admet qu'il existe des isométries f_i et des $\sigma_i > 0$ tels que

$$\left|\left|f_i - \sigma_i f_{|\mathcal{U}_i}\right|\right|_\infty \leq \varepsilon_i \sigma_i$$

On appelle ε_i les *erreurs isométriques* et σ_i les *facteurs de gonflements.*

Exemple 4.3. L'idée du formalisme précédent, c'est que les ouverts soient assez gros pour former un k-recouvrement, tout en étant assez petit pour se rapprocher d'une quasi-isométrie.

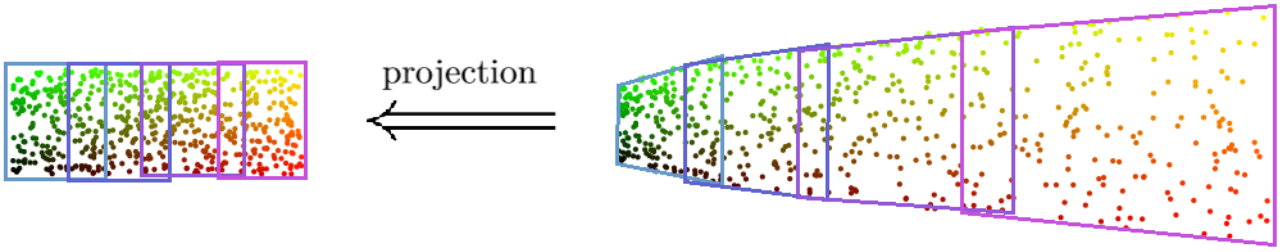

FIGURE 6 – Recouvrement constitué de 4 ouverts

On passe de l'échantillon de gauche, de densité uniforme, à celui de droite en effectuant une dilatation suivant l'abscisse des points. Ce découpage selon quatre blocs permet d'approximer la fonction de projection par la dilatation moyenne sur un bloc, plutôt que sur tout l'échantillon.

Si la fonction de symétrisation est sup, on sait que le poids de $a = [x, y] \in A$ en haute dimension est celui de l'arête orientée qui pèse le plus entre $[\overrightarrow{x, y}]$ et $[\overrightarrow{y, x}]$. Supposons sans perte de généralité que c'est le premier qui l'emporte. On peut centrer le raisonnement sur x et sur son image $z = f(x)$. Si $y = y_j$ est le j-ième plus proche voisin de x, on note $d_j = d(x, y_i)$. On note $z_j = f(y_j)$ sa projection, et σ_x la constante de normalisation.

Posons $c = \ln(\frac{\mu(a)}{\nu(a)})$. On cherche à minimiser c pour miniser l'entropie croisée de a. Compte tenu des notations précédentes, on peut l'écrire, si $\mu(a), \nu(a) < 1$:

$$c = \frac{d_j - d_1}{\sigma_x} - ||z_j - z|| + \texttt{min\_dist}$$

qu'on réarrange

$$c = \left[\frac{d_j}{\sigma_x} - ||z_j - z||\right] + \left[\texttt{min\_dist} - \frac{d_1}{\sigma_x}\right]$$

Le terme σ_x^{-1} varie selon x, alors qu'une projection quasi-isométrique comme f modifie les distances suivant un simple facteur de gonflement indépendemment du point.

D'autre part, $\frac{d_1}{\sigma_i}$ se compare facilement à la distance post-projection $||z_1 - z||$.

En introduisant ces nouveaux termes, il vient :

Proposition 4.4. *En gardant les notations précédentes, pour toute arête $a \in A$ telle que $\mu(a), \nu(a) < 1$, on a grâce à la quasi-isométricité 4.2*

$$c = \underbrace{\left[\frac{d_j}{\sigma_i} - ||z_j - z||\right]}_{\leq \varepsilon_i} + \underbrace{\left[||z_1 - z|| - \frac{d_1}{\sigma_i}\right]}_{\leq \varepsilon_i} + \underbrace{[\texttt{min\_dist} - ||z_1 - z||]}_{\text{écart à la moyenne en dimension } d} + \underbrace{\left[\left(\frac{1}{\sigma_i} - \frac{1}{\sigma_x}\right)(d_k - d_1)\right]}_{\text{écart à la moyenne en dimension } m}$$

Ce travail en tête, il est aisé d'estimer l'autre configuration :

Proposition 4.5. *Pour toute arête $a \in A$ telle que $\mu(a) = 1, \nu(a) < 1$, on a*

$$c = \underbrace{\texttt{min\_dist} - ||z_1 - z||}_{\text{écart à la moyenne en dimension } d} + \underbrace{||z_k - z|| - ||z_1 - z||}_{\text{Estimation réalisée en 2.1}}$$

4.3 Interprétation des estimations

À la lumière de ces équations, comme nous cherchons à minimiser c le paramètre `min_dist` optimal pourrait s'interprêter comme la valeur moyenne des $||z_1 - z||$. C'est corroboré par nos expériences, où l'augmentation de ce paramètre mène à un gonflement des regroupements, c'est-à-dire une plus petite densité de points.

En outre, l'efficacité de ce modèle ne dépend pas uniquement de la quasi-isométricité du plongement, mais aussi de la bonne répartition des points. Si les voisinages de points voisins sont trop différents, le modèle est mis à mal : c'est le phénomène **d'émiettement** :

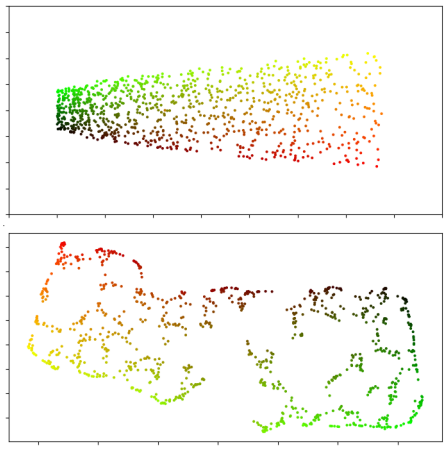

FIGURE 7 – Phénomène d'émiettement : la réduction de dimension 2 -> 2 ne se solde pas vraiment par un rectangle

5 Perspectives

Dans le monde de la réduction de dimension, il reste de nombreux problèmes à creuser

— Comment prendre en compte la topologie quand on compare des données ? Comment définir un coût "topologique" à la computationnellement raisonnable ? Par exemple, en calculant le diagramme de persistence en dimension 0 ou 1 (c'est-à-dire l'évolution des composantes connexes et des cycles suivant la filtration de Čech) des points en grandes dimensions M, trouver des points $N = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ qui minimisent la distance entre son diagramme de persistance et l'original ? Il faudrait pour cela que l'application $x_1, \dots x_n \mapsto d_{0,1}(M, N)$ [1] deviennent assez compréhensibles pour qu'on puisse la traiter avec des méthodes classiques.

— Comment expliquer rigoureusement les algorithmes récents (comme UMAP) ? Et à travers ces explications, pourrait-on optimiser chacune de leurs étapes plutôt que de suivre de simples heuristiques ?

1. $d_{0,1}(M, N)$ dénote la distance entre les diagrammes de persistance en dimension 0 (composantes connexes) et dimension 1 (cycles) de M et N.

Références

[1] E. Aamari, *Rate convergence for Geometric Inference*, PhD thesis, Université Paris-Saclay, 2017.

[2] D. Biau, *Lectures on the nearest neighbors method*, 2015.

[3] Q. e. a. Cao, Spielmann, *The single-cell transcriptional landscape of mammalian organogenesis*, Nature, 566 (2019), pp. 496–502.

[4] F. Chazal, V. de Silva, M. Glisse, and S. Oudot, *The structure and stability of persistence modules*, 2013.

[5] D. L. Donoho and C. Grimes, *Hessian eigenmaps : Locally linear embedding techniques for high-dimensional data*, Proceedings of the National Academy of Sciences, 100 (2003), pp. 5591–5596.

[6] H. Doraiswamy, J. Tierny, P. J. S. Silva, L. G. Nonato, and C. Silva, *Topomap : A 0-dimensional homology preserving projection of high-dimensional data*, 2020.

[7] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, 1987.

[8] J. H. C.-A. D. F. G. E. W. N. Etienne Becht, Leland McInnes, *Dimensionality reduction for visualizing single-cell data using umap*, Nature biotechnology, 37 (2019), pp. 38–44.

[9] P. S. E. P. H. D. D. S. K. T. C. T. J. K. R. H. W. J. I. M. Jonathan S. Packer Qin Zhu, Chau Huynh, *A lineage-resolved molecular atlas of c. elegans embryogenesis at single-cell resolution*, Science, 365 (2019).

[10] G. H. Laurens van der Maaten, *Visualizing high-dimensional data using t-sne*, Journal of Machine Learning Research, (2008), pp. 2579–2605.

[11] L. McInnes, J. Healy, and J. Melville, *UMAP : Uniform Manifold Approximation and Projection for Dimension Reduction*, ArXiv e-prints, (2018).

[12] X. X. . M. W. Mohammed Ali, Mark W. Jones, *Timecluster : dimension reduction applied to temporal data for visual analytics*, The Visual Computer, 35 (2019), pp. 1013–1026.

[13] L. Tenenbaum, De Silva, *Isomap*, (2000).