Rapport de stage

Analyse Mathématique de l'algorithme de réduction de dimension UMAP

Antoine Commaret
Année Scolaire 2019-2020

Co-tuteurs : M. Frédéric CHAZAL
M. Bertrand MICHEL

13 Avril 2020 — 14 Août 2020

Table des matières

1 Description du stage **2**

2 UMAP : Késako ? **3**
2.1 Qu'est-ce qu'un algorithme de réduction de dimension ? . . . 3
2.2 Historiques des algorithmes de réduction de dimension . . . 4
2.3 2018 : un algorithme parmi tant d'autres ? . . . 4
2.4 Description du fonctionnement d'UMAP . . . 4

3 Étude du processus de normalisation d'UMAP **6**
3.1 Fonctionnement de la normalisation dans une distribution uniforme . . . 6
3.2 Variation autour du fonctionnement typique . . . 10

4 Relier la haute à la basse dimension **12**
4.1 Entropie croisée . . . 12
4.2 UMAP en basse dimension . . . 15
4.3 Construction du graphe en haute dimension . . . 16

5 Constats empiriques **18**
5.1 Paramètres à maîtriser . . . 18
5.2 Quelle initialisation ? . . . 21
5.3 Convergence de la descente de Gradient . . . 22

6 Cas particuliers **24**
6.1 Tirage dans une boule . . . 24
6.2 Tirage gaussien . . . 25
6.3 Tirage équiréparti d'une sous-variété . . . 26
6.4 Tirage uniforme d'une sous-variété . . . 27

7 Modèle de fonctionnement optimal **28**
7.1 État de l'art . . . 28
7.2 Reflexion sur l'efficacité d'une réduction de dimension . . . 28
7.3 Où se situe UMAP : mi-chemin entre réduction parfaite et partitionnement . . . 28

8 Modèle **31**
8.1 Description . . . 31
8.2 Interprétation des estimations . . . 33

9 Limite intrinsèque à notre modèle : l'émiettement **34**

10 Conclusion **36**

A Étude de la fonction Bêta incomplète **37**

B Fonction de répartition d'un tirage sur une variété **39**

1 Description du stage

J'ai effectué ce stage sous la cotutelle de MM. Frédéric Chazal et Bertrand Michel. Compte tenu du contexte sanitaire, il a été effectué dans son intégralité en télétravail. Je tiens à remercier mes tuteurs pour leur encadrement malgré ce contexte difficile ; il consistait essentiellement en deux appels hebdomadaires pour se tenir au courant d'éventuelles avancées ou de résultats expérimentaux. J'ai intégré l'équipe DataShape, dont fait partie M. Chazal et ai pu suivre les séminaires à distance.

Il s'agissait d'un stage de recherche autour d'un algorithme de réduction de dimension, UMAP, qui passe pour l'état de l'art dans le domaine de réduction de dimension non linéaire. Ma mission était de chercher à percer le secret de l'article [6] de Leland McInnes en justifiant le fonctionnement. Après quelques jours d'études, on s'est aperçu que la justification théorique de McInnes était surtout une heuristique cachée derrière des arguments faisant intervenir la théorie des catégories. Passées ces considérations algébriques, l'algorithme est assez proche de son principal concurrent, t-SNE avec plus de flexibilité et d'efficacité à l'utilisation.

Le travail était grosso modo découpé en deux parties :

— une partie *expérimentale* de tests d'UMAP sur des jeux de données de notre composition, pour en faire apparaître les lois,

— une partie de *modélisation* qui cherche à rendre compte quantiativement des faits observés dans la première partie.

On présente dans ce rapport les modélisations qui nous ont semblé les plus convaincantes.

La comparaison haute et basse dimension d'UMAP semble assez arbitraire. J'ai très légèrement modifié le code original dans mes expériences pour ajouter deux nouveaux types de fonctionnement : sUpMAP et UMAP Moyenne, dont les différences seront décrites en partie 4.3.

2 UMAP : Késako ?

2.1 Qu'est-ce qu'un algorithme de réduction de dimension ?

La réduction de dimension est un procédé de traitement des données qui consiste à simplifier l'étude de points en grande dimension m. On fournit à l'algorithme n points chacun décrit par m entrées (par exemple, des points de $\mathbb{R}^m$), et une dimension d'arrivée $d < m$. L'algorithme va alors retourner n points chacun décrit par d entrées, en correspondance avec les n points en basse dimension. Il cherche, malgré la perte d'information, à distinguer les caractéristiques principales du nuage de points et à les restituer en basse dimension.

Prenons un exemple : supposons que l'on possède un grand jeu de données sur des villes qu'on ne sait pas placer sur une carte, comme la température moyenne en août, en juillet ; les précipitations de tous les jours de l'année, l'ensoleillement hebdomadaire, etc.

Ces données sont énormes et dès que le nombre de villes est un peu élevé, il devient difficile de les analyser. Pourtant, il est simple de faire la différence entre les villes du nord et du sud de la France, en regardant l'ensoleillement, entre les plaines et le climat de montagne suivant les températures. Derrière ces tendances, on imagine des regroupements. Une réduction en dimension 2 a pour objectif la visualisation de ces différents climats. S'établit alors une sorte de carte qui résume les groupements et les liens les plus importants entre les groupes.

À l'inverse, si l'on connaît déjà les groupes, on peut aussi utiliser la réduction de dimension pour visualiser les relations entre groupes. L'exemple ci-dessous est issu de la comparaison d'articles de philosophie contemporaines :

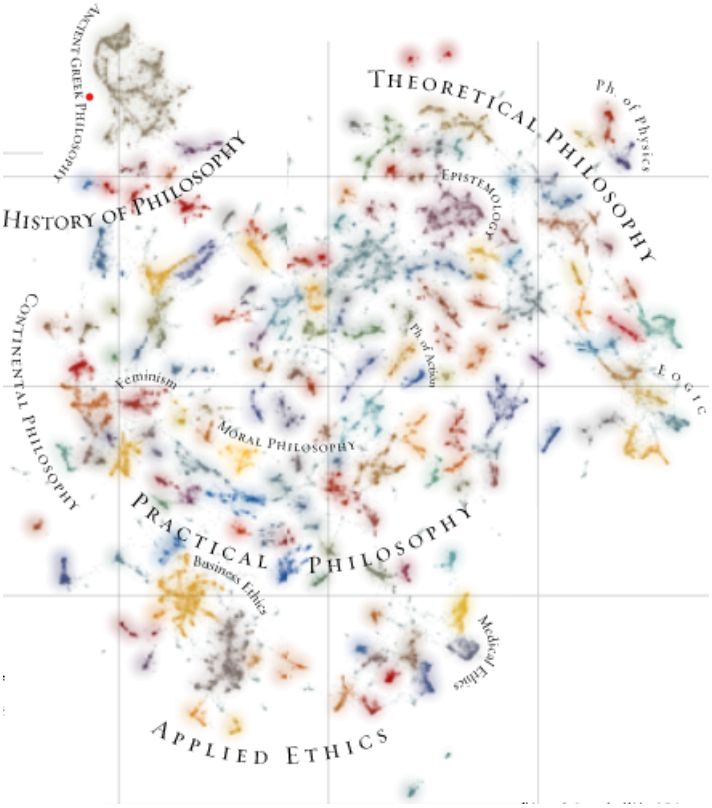


Figure 1 – Carte de philosophie contemporaine en utilisant UMAP, par Maximilian Noichl

D'autre part, ces données se recoupent souvent. On n'a pas besoin de mesures journalières pour différencier les types de climat ; simplement de repérer des tendances plus faciles à analyser. On peut demander arbitrairement à un algorithme de réduction de dimension de passer d'une très grande dimension $m = 1000$ à $d = 50$ pour ensuite les traiter plus rapidement.

2.2 Historiques des algorithmes de réduction de dimension

Avant l'apparition d'UMAP, c'est t-SNE qui était l'algorithme de réduction de dimension non-linéaire le plus utilisé. Voici une liste non-exhaustive des algorithmes les plus communs :

- ACP, pour *Analyse en Composante Principale*. Il s'agit d'un algorithme de réduction de dimension dit *linéaire* car une réduction de dimension m vers d calcule le sous-espace de dimension d qui colle le mieux aux données.
- MDS, pour *Multi-dimensional scaling*. Il cherche à conserver les distances entre points dans le nuage de haute dimension.
- LLE [8], pour *Locally Linear Embedding*. Sommairement, il applique MDS sur les voisinages de chaque point plutôt qu'en prenant toutes les distances en compte. 2000.
- IsoMAP [9]. Cherche à repérer les géodésiques en analysant les distances des voisins de chaque point. Cet algorithme a un fonctionnement garanti lorsque qu'on tire un très grand nombre de points uniformément sur une sous-variété convexe. 2000.
- Laplacian Eigenmaps [1] Il suppose que les données sont réparties le long d'une sous-variété de l'espace ambiant, effectue une approximation de l'opérateur Laplace-Beltrami pour en calculer les fonctions propres de plus petite énergie. C'est la méthode d'initialisation qu'UMAP utilise pour des petites données. 2002.
- Hessian Eigenmaps [3]. Il n'a pas besoin de l'hypothèse convexe d'IsoMAP, mais fonctionne en approximant les dérivées secondes ; grande instabilité numérique malgré un fonctionnement garanti. 2003.
- t-SNE [4], principal concurrent d'UMAP. Il a un fonctionnement semblable : construction de graphes en haute et basse dimension qu'il compare à l'aide de la divergence de Kullback-Leiber. 2008.

2.3 2018 : un algorithme parmi tant d'autres ?

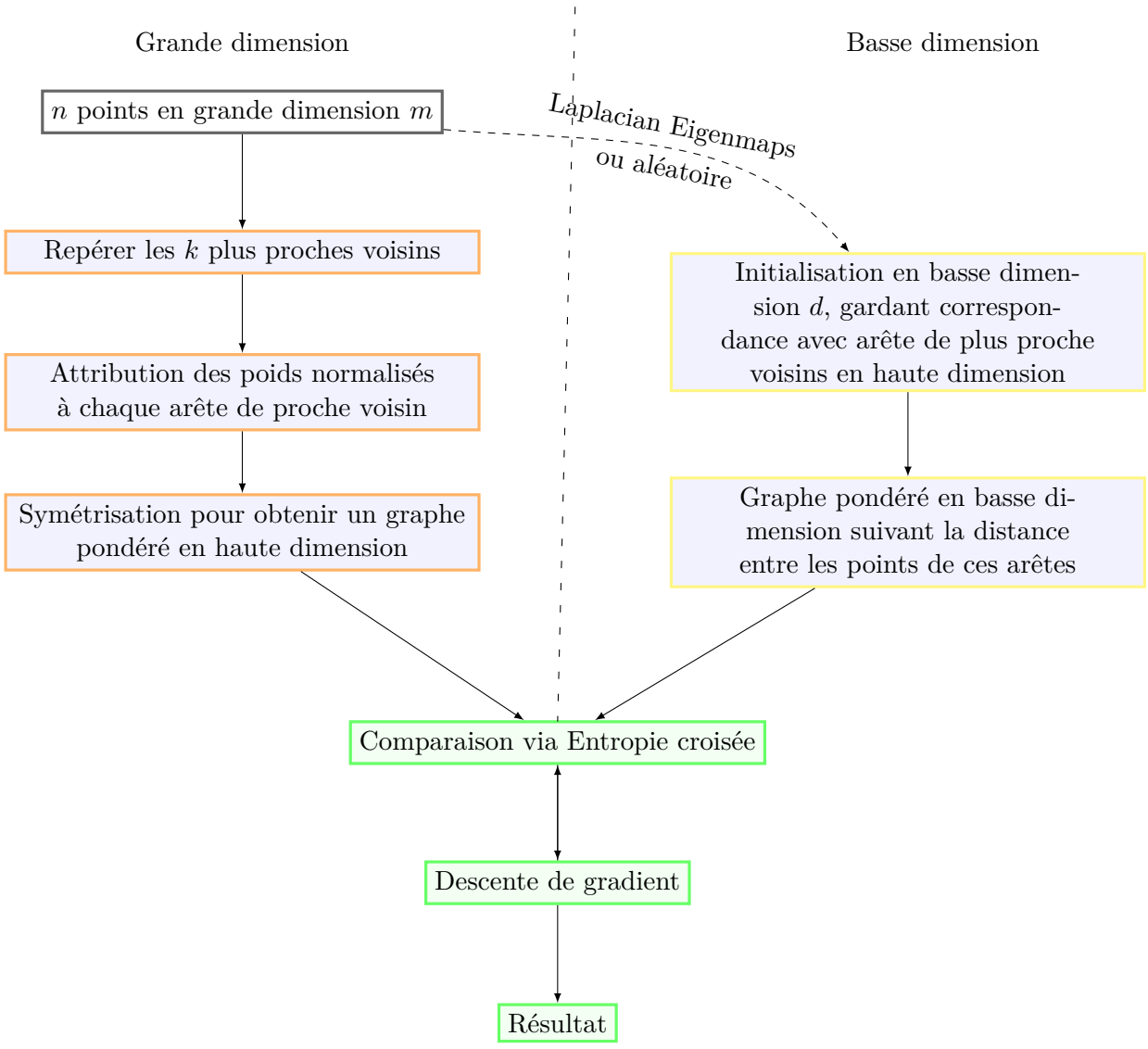
UMAP est un outil de réduction de dimension développé [5] et théorisé [6] par Leland McInnes en 2018. Il forme, avec son concurrent direct t-SNE [4], l'état de l'art de la réduction de dimension non linéaire. Il permet une projection en basse dimension d quelconque là où t-SNE échoue au delà de la dimension 3. On l'utilise principalement en génomique dans l'analyse de données dites *single-cell* pour visualiser des données autrement très complexes. Il n'est pas limité à ce domaine comme en témoigne cette sélection d'articles scientifiques.

2.4 Description du fonctionnement d'UMAP

Pour effectuer la réduction de dimension, 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, suivant l'étape de normalisation qu'on étudiera en détail dans la section 3. On effectue ensuite une opération de symétrisation entre les deux orientations pour obtenir un graphe symétrique pondéré de haute dimension.

Ensuite, UMAP initialise un même nombre de points en dimension d soit de façon aléatoire, soit suivant une autre technique de réduction de dimension, [1]. Il construit alors 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 `min_dist`, qui est un paramètre global.

UMAP effectue ensuite une descente de gradient stochastique pour minimiser l'entropie croisée de ces deux graphes. Il s'agit simplement de la somme des entropie croisée comparant les poids de chaque arête en haute et basse dimension.


Grande dimension
Basse dimension
n points en grande dimension m
Laplacian Eigenmaps
ou aléatoire
Repérer les k plus proches voisins
Attribution des poids normalisés à chaque arête de proche voisin
Symétrisation pour obtenir un graphe pondéré en haute dimension
Initialisation en basse dimension d, gardant correspondance avec arête de plus proche voisins en haute dimension
Graphe pondéré en basse dimension suivant la distance entre les points de ces arêtes
Comparaison via Entropie croisée
Descente de gradient
Résultat

3 Étude du processus de normalisation d'UMAP

Pour chaque point x de l'échantillon, UMAP repère les k plus proches voisins $y_1, \dots, y_k$ et associe à chacun un poids entre 0 et 1, que l'auteur interprète comme étant la probabilité qu'a l'arête $[x, y]$ de représenter une liaison directe entre les points x et y.

Il fait l'hypothèse que l'échantillonnage a lieu sur une sous variété de $\mathbb{R}^m$, assez régulière pour que les k plus proches voisins dans l'espace ambiant soient situés dans un voisinage géodésique de la sous-variété. Sont ainsi exclues les variétés assez tordues pour que des points de lointaine distance géodésique soient proches dans l'espace ambiant. UMAP considère que le point y_1 le plus proche de x est nécessairement lié à x (lui attribuant un poids de 1) et attribue ensuite des poids décroissants suivant $d_i = d(x, y_i)$:

$$p(x, i) = \exp(\frac{d_1 - d_i}{\sigma_x})$$

normalisé par σ_x de sorte à ce que

$$\sum_{i=1}^{k} p(x, i) = \log_2(k)$$

Remarquons que cette normalisation est une invariance par multiple d'isométrie. En effet,

— appliquer une isométrie à un nuage de points conserve les distances, menant aux mêmes $p(x, i)$ qu'avant la normalisation,

— appliquer une homothétie aux points multiplie les distances par une même constante, menant après normalisation à la même suite $p(x, i)_i$ qu'avant.

3.1 Fonctionnement de la normalisation dans une distribution uniforme

Après quelques expériences sur ce processus de normalisation, on remarque la suite de poids $p(x, i)$ a tendance à suivre la fonction inverse en moyenne. Le nombre de voisins k considérés, la dimension d dans laquelle on effectue les simulations, ou encore le nombre de points n de l'échantillon semblent avoir un impact limité. Par exemple, tirons n points $x_1, \dots, x_n$ sur une sphère de dimension d. Ensuite, calculons les poids moyens après normalisation d'UMAP avec `n_neighbors` $= k$:

$$\bar{p}(i) = \frac{1}{n} \sum_{1 \leq j \leq n} p(x_j, i)$$

et traçons en la courbe, qu'on compare à la fonction inverse et à la situation pré-normalisation (pour $\sigma = 1$) :

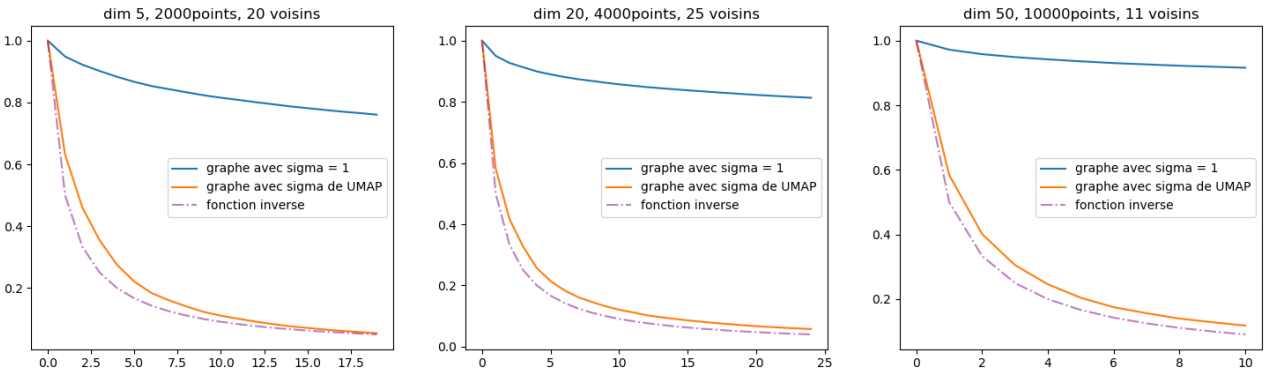


FIGURE 2 – Les poids moyens semblent suivre la fonction inverse quels que soient les paramètres

Essayons de constater un comportement moyen à cette normalisation en considérant qu'on effectue des tirages aléatoires autour d'un point x sur la variété. Notons

$$v_x = \left\{ \begin{array}{rcl} \mathbb{R}^+ & \to & [0,1] \\ t & \mapsto & v_x(t) = \int_{\mathcal{M}\cap B_x(t)} \mathrm{d}\mu \end{array} \right.$$

fonction de répartition généralisée représenant la probabilité qu'un point tiré au hasard sur une distribution μ dont le support est la variété soit à une distance euclidienne inférieure ou égale à t. On supposera que cette fonction admet une réciproque au moins $\mathcal{C}^1$ presque partout v_x^{-1}.

Supposons que la variété soit un disque plat et la densité constante autour de x à une valeur a. On a alors

$$v_x(t) = \left\{ \begin{array}{ll} at^d & \text{si } at^d \leq 1 \\ 1 & \text{sinon.} \end{array} \right.$$

Comme les variétés sont proches de leur espace tangent et qu'il est naturel de supposer une densité régulière, on s'attend à ce qu'on ait

$$v_x(t) = at^d(1 + r(t))$$

où $r(t) = o(1)$ au voisinage de 0, justifiant qu'on commence par une étude de ce cas particulier.

Étude de la distance du k-ème plus proche voisin

Commençons par une étude sommaire de d_k la variable aléatoire de la distance du k-ième plus proche voisin à X.

Suivant Biau et Devroye [2], en posant $U_{(i)}$ la variable aléatoire du i-ème plus petit tirage parmi n point dans [0,1] (dont on connait la loi), on a

$$d_1, \ldots, d_k \sim v_x^{-1}(U_{(1)}), \ldots, v_x^{-1}(U_{(k)})$$

sachant que dans ce cas du disque plat , $v_x^{-1}(u) = (\frac{u}{a})^{1/d}$.

On connait la fonction de répartition de d_k suivant la répartition multidimensionnelle, qui consiste à vérifier qu'il y a moins de k points tirés à une distance plus petite que u :

$$\begin{aligned} \mathbb{P}(d_k > u) =& \mathbb{P}(\mathcal{B}(n, v_x(u)) < k) \\ =& \sum_{i=0}^{k-1} \binom{n}{i} v_x(u)^i (1 - v_x(u))^{n-i}. \end{aligned}$$

On peut en calculer l'espérance, via la fonction béta :

$$B(x,y) = B(y,x) = \int_0^1 (1-u)^{x-1} u^{y-1} \,\mathrm{d}u = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}.$$

On a en effet

$$\begin{aligned} \mathbb{E}(d_k) &= \int_0^\infty \mathbb{P}(d_k > t) \,\mathrm{d}t \\ &= \sum_{i=0}^{k-1} \binom{n}{i} \int_0^\infty (1 - v_x(u))^{n-i} v_x(u)^i \,\mathrm{d}u \\ &= \sum_{i=0}^{k-1} \binom{n}{i} a^{-1/d} \int_0^1 (1 - u^d)^{n-i} u^{di} \,\mathrm{d}u \\ &= \sum_{i=0}^{k-1} \binom{n}{i} a^{-1/d} B(n - i + 1, i + 1/d). \end{aligned}$$

Comme $\Gamma(z+1) = z\Gamma(z)$, si $i \geq 1$:

$$\begin{aligned}\binom{n}{i} B(n-i+1, i+1/d) =& \frac{\Gamma(n+1)\Gamma(i+1/d)}{\Gamma(n+1+1/d)\Gamma(i+1)} \\ =& \frac{1}{i}\prod_{j=i}^{n}(1+\frac{1}{dj})^{-1}\end{aligned}$$

Une étude élémentaire du produit $\prod_{j\leq n}(1+\frac{1}{dj})$ montre qu'on a

$$\binom{n}{i} B(n-i+1, i+1/d) = \frac{1}{i}\prod_{j=1}^{i-1}(1+\frac{1}{dj}) \cdot \exp(\gamma_d - \gamma/d) \cdot n^{-1/d} \cdot (1+o_{d,n}(1))$$

où

$$\gamma_d = \sum_{j\geq 1}\frac{1}{dj} - \ln(1+\frac{1}{dj}).$$

De la même façon, si $i = 0$, on a

$$\binom{n}{i} B(n-i+1, i+1/d) = d \cdot \exp(\gamma_d - \gamma/d) \cdot n^{-1/d} \cdot (1+o_{d,n}(1)).$$

Finalement, en omettant le terme d'erreur, il vient :

$$\mathbb{E}(d_k) = \frac{1}{(an)^{1/d}} \cdot \exp(\gamma_d - \gamma/d) \cdot \left[1 + \sum_{i=1}^{k-1}\frac{1}{di}\prod_{j=1}^{i-1}(1+\frac{1}{dj})\right]$$

et donc

$$\mathbb{E}(d_k - d_1) = \frac{1}{(an)^{1/d}} \cdot \exp(\gamma_d - \gamma/d) \cdot \sum_{i=1}^{k-1}\frac{1}{di}\prod_{j=1}^{i-1}(1+\frac{1}{dj})$$

On peut légèrement simplifier cette expression en posant

$$P_i(X) = \begin{cases} 1 & \text{si } i = 0 \\ \prod_{j=1}^{i}(1+\frac{X}{j}) & \text{sinon.}\end{cases}$$

On obtient alors, via la relation de récurrence $P_i(X) = (1+\frac{X}{i})P_{i-1}$, la proposition suivante :

Proposition 3.1. *En gardant les notations précédentes, on a*

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

Fonctionnement moyen de la normalisation typique

Pour traiter la normalisation d'UMAP, on a besoin d'information sur les variables aléatoires du type

$$\exp(-(d_i - d_1))$$

Celles-ci sont hélas bien plus difficiles à traiter que $d_i - d_1$. Les méthodes précédentes ne permettent même pas de calculer l'espérance après le passage à l'exponentielle. Pour contourner ce problème, on garde en tête l'inégalité de Jensen

$$\mathbb{E}(\exp(-(d_i - d_1)) \geq \exp(-\mathbb{E}(d_i - d_1))$$

qu'on traitera dans les calculs comme une égalité. Cette inégalité est d'autant plus forte que la fonction appliquée est convexe. La fonction $x \mapsto e^{-x}$ l'est surtout aux alentours de 0. On revient donc aux estimations des $\mathbb{E}(d_i - d_1)$.

Nous connaissons les deux premiers termes du polynôme

$$H_k(X) = \frac{P_k(X) - 1}{X} = H_k + \sum_{i=1}^{k} \frac{H_{i-1}}{i} X + \dots$$

Et finalement, si l'on ne garde que les deux premiers termes, on a le développement asymptotique suivant :

$$H_k(1/d) = \ln(k) \left[1 + \frac{\gamma}{d} + \frac{\ln(k)}{2d}\right] + \gamma + \frac{\gamma^2}{d} + o(1)$$

Comme $H_k(1/d)$ est le seul facteur variant selon k, trouver le σ de la normalisation d'UMAP 3 revient à une constante multiplicative près à trouver par dichotomie le σ tel que

$$1 + \sum_{i=2}^{k} \exp(-\frac{H_k(1/d)}{\sigma}) = \log_2(k).$$

On vérifie expérimentalement l'efficacité de ce calcul en prenant

$$H_i(1/d) = \ln(k) \left[1 + \frac{\gamma}{d} + \frac{\ln(k)}{2d}\right] + \gamma$$

pour $i > 2$. On fait le choix empirique de $H_1(X) = 0.8$ pour corriger l'erreur de convexité qui est la plus grande lorsque $i = 1$. L'expérience consiste à tirer 200 fois n points suivant une distribution centrée autour de 0. On applique la normalisation d'UMAP sur chacun de ces tirages, puis on effectue la moyenne des poids comme dans 2. Sur ces données on effectue des statistiques.

— La moyenne des normalisations en orange
— La fonction de moyenne estimée en cyan.
— La médiane en pointillés gris
— Les quantiles 5 % → 25 % → 75%x → 95% sur fond respectivement jaune, orange et jaune.

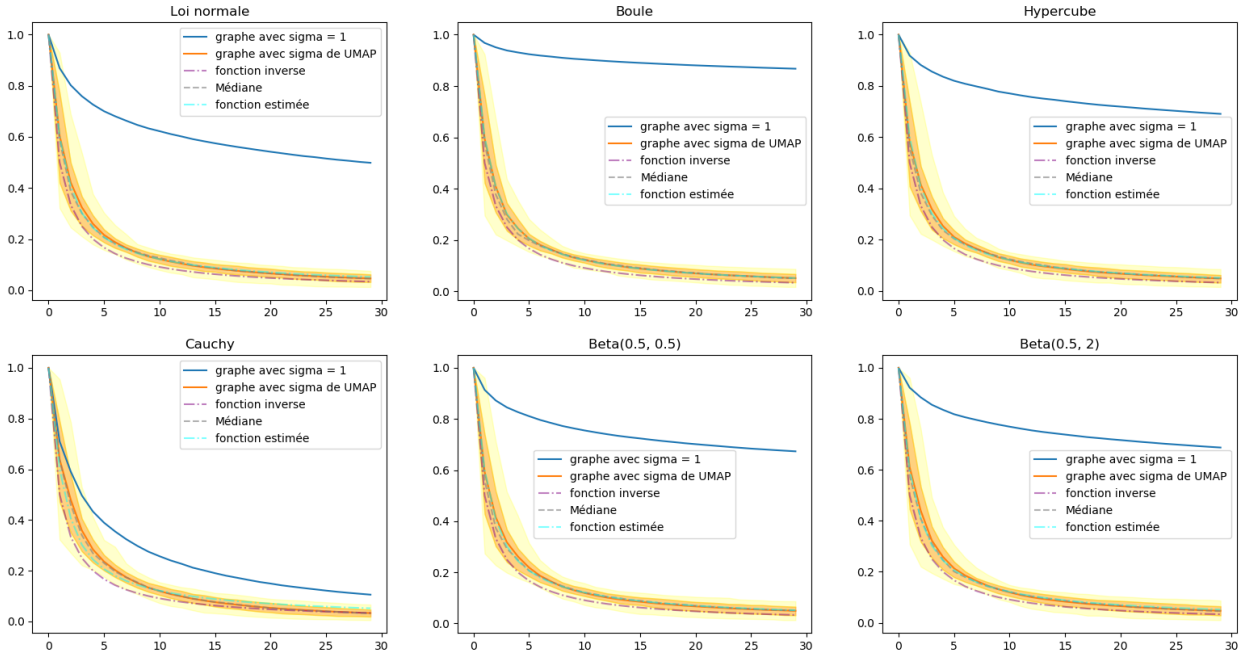


FIGURE 3 – $k = 30$, $d = 20$ et tirage de $n = 2000$ points

Pour se donner un ordre d'idée, on garde en bleu la moyenne pré-normalisation (c'est-à-dire la moyenne des $\exp(d_1 - d_k)$), et le tracé de la fonction inverse en longs pointillés.

On a effectué 6 types de simulations, dans l'ordre de lecture :
— Tirage suivant une loi normale centrée de matrice de covariance identité en dimension $d = 20$,
— Tirage uniforme dans une boule de rayon 1 en dimension d,
— Uniforme dans l'hypercube $[-1, 1]^d$,
— Tirage où chacune des d composantes suit une loi de Cauchy de paramètres $(0, 1)$,
— Chacune des d composantes suit une loi Beta$(\frac{1}{2}, \frac{1}{2})$ recentrée sur $[-1, 1]$,
— Chacune des d composantes suit une loi Beta$(\frac{1}{2}, 2)$ recentrée sur $[-1, 1]$.

Les statistiques post-normalisation sont semblables, si ce n'est peut-être celle suivant une loi de Cauchy. Cela laisse penser qu'à quelques erreurs près, c'est la densité du point central autour duquel auquel on effectue le tirage qui détermine la normalisation. Enfin, comme $H_i(1/d) \simeq H_i$, on comprend pourquoi le tracé moyen est semblable à la fonction inverse.

3.2 Variation autour du fonctionnement typique

Cherchons à généraliser le cadre précédent. Après normalisation, ces différents tirages mènent aux mêmes courbes. C'est assez naturel : si l'on regarde les k plus proches voisins de x lors d'un tirage de n point autour de x, si la densité est régulière et que $k \ll n$, on se retrouve à étudier un tirage très local, où la densité est presque constante.

Quantitativement, on va maintenant chercher à estimer les variations moyennes lorsque la distribution se fait le long de d dimensions :

$$v_x(u) \simeq au^d$$

Remarquons que comme le montre le calcul de l'appendice B menant au théorème B.1, le tirage localement uniforme sur une sous-variété gagne même un terme de plus sur l'approximation polynomiale :

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

Il est difficile d'établir un résultat concis. Supposons que l'on dispose d'une approximation polynomiale de $v'_x(t)$:

$$v'_x(t) = dat^d + \sum (d+i)a_i t^{d+i-1} + r(t) = dat^d \left(1 + \sum \frac{a_i}{a}\frac{d+i}{d}t^i\right)$$

découlant naturellement de l'approximation polynomiale de $v_x(t)$:

$$v_x(t) = at^d + \sum a_i t^{d+i} + R(t) = at^d \left(1 + \frac{a_i}{a}t^i + \frac{R(t)}{at^d}\right)$$

Nous cherchons à comparer la moyenne de la différence du k-ème plus proche voisin et du $k-1$-ème sur un tirage de distribution μ

$$\int_0^\infty \psi(v_x(t))\,\mathrm{d}t = \binom{n}{k}^{-1} \mathbb{E}(d_k^\mu - d_{k-1}^\mu)$$

à un tirage de densité constante $a^{-1/d}$. On garde d_k pour le k-ième plus proche voisin lors d'un tirage dans une boule de rayon 1.

$$\int_0^\infty \psi(g(u))\,\mathrm{d}u = a^{-1/d}\binom{n}{k}^{-1} \mathbb{E}(d_k - d_{k-1})$$

où $\psi(x) = (1-x)^{n-k}x^k$ et

$$g_x(u) = \begin{cases} au^d & \text{si } u \leq a^{-1/d} \\ 1 & \text{sinon.} \end{cases}$$

Pour mieux comparer les deux, effectuons un changement de variable de sorte à ce que $v_x(t) = g_x(u)$. On a alors

$$\int_0^z \psi(g(u))\,\mathrm{d}u = \int_0^{v_x^{-1}(g(z))} \psi(v(t))v'(t)(g^{-1})'(v(t))\,\mathrm{d}t$$

Heureusement, nous savons que $(g^{-1})'(x) = \frac{a^{-1/d}}{d}x^{1/d-1}$, menant à terme à

$$\int_0^{g^{-1}(v(z))} \psi(g(u))\,\mathrm{d}u = \int_0^{v^{-1}(g(z))} \psi(v_x(t))\left[1 + \sum \frac{a_i}{a}\frac{d+i}{d}t^i + \frac{r(t)}{dat^{d-1}}\right]\left[1 + \sum \frac{a_i}{a}t^i + \frac{R(t)}{at^d}\right]^{1/d-1}\mathrm{d}t$$

qu'on peut réécrire plus sobrement en posant Q convenablement

$$\int_0^{g^{-1}(v(z))} \psi(g(u))\,\mathrm{d}u = \int_0^{v^{-1}(g(z))} \psi(v_x(t))\,[1+Q(t)]\,\mathrm{d}t$$

Or on a, d'après l'étude de la fonction beta incomplète A.1 :

$$a^{-1/d}\mathbb{E}(d_k - d_{k-1})\,(1-h^*(z)) \leq \int_0^{g^{-1}(v(z))} \psi(g(u))\,\mathrm{d}u \leq a^{-1/d}\mathbb{E}(d_k - d_{k-1})$$

où $h^*(z) = h(g^{-1}(v(z)))$.

Finalement, en gardant les notations précédentes, on a :

Proposition 3.2. *Pour tout $z \in \mathbb{R}$, on a :*

$$\frac{1-h^*(z)}{1+\max_{t\leq z}|Q(t)|} \leq \frac{\int_0^z \psi(v_x(t))\,\mathrm{d}t}{a^{-1/d}\mathbb{E}(d_k - d_{k-1})} \leq \frac{1}{1-\max_{t\leq z}|Q(t)|}$$

En rajoutant un encadrement, on obtient immédiatement la proposition suivante :

Proposition 3.3. *Supposons qu'on ait à la manière de la fonction Bêta incomplète une borne inférieure relative :*

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

On a alors, pour tout $z \in \mathbb{R}$:

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

Derrière cet encadrement un peu barbare, quelques remarques :

— S'il est difficile d'obtenir une forme explicite, la composition $g^{-1}\circ v$ est typiquement proche de l'identité. A.1 donne une formule explicite pour h. De la même façon, comme v est proche de g, on a typiquement $H(z) \simeq h(z)$. Ces termes décroissent vite passée la valeur critique $\frac{b}{b+a}$.

— Les termes fonctions de $\max_{t\leq z}|Q(t)|$ découlent simplement de l'inégalité pour toute fonction $f \geq 0$ mesurable :

$$\int_0^z f(t)Q(t)\,\mathrm{d}t \leq \max_{t\leq z}|Q(t)| \int_0^z f(t)\,\mathrm{d}t$$

et sont assez grossiers. Ils permettent cependant de vérifier quantitativement qu'une distribution proche de celle d'un disque (en effet, $Q = 0$ équivaut à une distribution en distances équivalent à celle d'un tirage sur un disque de dimension d) mène à une bonne approximation de l'espérance.

4 Relier la haute à la basse dimension

Pour comparer la structure des données en haute et en basse dimension, UMAP développe son propre critère. Résumons cette structure en le comparant avec le fonctionnement du plus ancien t-SNE [4].

t-SNE crée pour chaque point une probabilité conditionnelle sur les autres points du jeu de données. Il effectue ensuite une symétrisation menant à une distribution liant chaque point aux autres (surtout ses plus proches voisins) à la manière d'une chaîne de Markov sur un graphe.
t-SNE associe à chacun des points en haute dimension un point initialisé au hasard en basse. À chacun des points en basse dimension est associé une distribution sur les autres points, suivant un noyau d'une distribution de Student.
t-SNE choisit ensuite de comparer les distributions via la divergence de Kullback-Leibler.

UMAP s'occupe d'arêtes plutôt que de points. Autour d'un point x, on ne garde que les arêtes formées par les k plus proches voisins, en attribuant à chacune de ces arêtes un poids probabiliste dans $[0,1]$, comme étudié précédemment. On forme de ce fait un graphe orienté pondéré A'. Il est symétrisé via une opération d'union probabiliste, le graphe final A étant défini par

$$A = A' + {}^tA' - A' \cdot {}^tA'$$

où $\cdot$ est l'opération de produit terme à terme. Tout comme t-SNE, UMAP associe à chaque point en haute dimension un point en basse. Il construit un graphe pondéré en basse dimension, aux mêmes arêtes dont le poids est simplement fonction de la distance euclidienne entre les points du haut, suivant le paramètre `min_dist`. Enfin, UMAP compare les poids de chaque arête et les somme pour obtenir son critère.

4.1 Entropie croisée

UMAP construit ainsi deux graphes, l'un représentant les points en haute dimension et l'autre en basse dimension. À chaque arête sélectionnée par la normalisation, on associe un poids en haute dimension (après normalisation) et un autre en basse dimension, fonction de la distance entre les deux points. Comme les poids sont dans $[0,1]$, UMAP les compare si c'étaient des proba via la fonction coût d'entropie croisée entrop. On somme la contribution de chaque arête pour avoir le coût global :

$$C(\mu,\nu,A) = \sum_{a\in A} \text{entrop}(\mu(a),\nu(a)) = \sum_{a\in A} \mu(a)\ln\left(\frac{\mu(a)}{\nu(a)}\right) + (1-\mu(a))\ln\left(\frac{1-\mu(a)}{1-\nu(a)}\right)$$

La fonction d'entropie croisée

$$\text{entrop} : \left\{ \begin{array}{ccc}]0,1[^2 & \to & \mathbb{R}^+ \\ (x,y) & \mapsto & x\ln\left(\frac{x}{y}\right) + (1-x)\ln\left(\frac{1-x}{1-y}\right) \end{array} \right.$$

est positive sur son ensemble de définition, par concavité du logarithme. Elle est nulle seulement lorsque $x = y$. Cette diagonale joue le rôle d'une cuvette, comme le montre la représentation graphique ci-dessous.

D'un point de vue computationnel, entrop a une dérivée en y aisément calculable. Dans les calculs explicites de descente de gradient [6] rajoute un ε correcteur pour éviter les problèmes apparaissant lorsque $y \to 0$. La cuvette est très plate autour de la diagonale, et prend des grandes valeurs uniquement lorsqu'un des termes s'approche de 1.

Cette fonction coût permet d'assurer la préservation du voisin le plus proche auquel un poids 1 est attribué en grande dimension. Si on a exactement $x = 1$, c'est facile à voir puisque la fonction devient $\text{entrop}(x,y) = -\ln(y)$ et minimiser le coût consiste à avoir $y \to 1$.

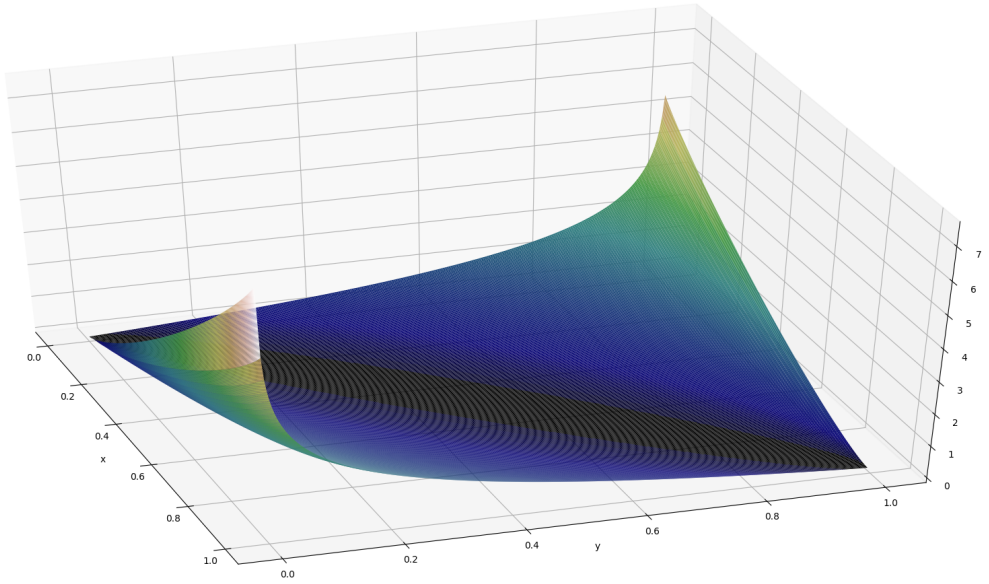


FIGURE 4 – Fonction entropie croisée

Dans les autres configurations, les voisins plus proches, aux plus gros x, ont un coûts plus important. On s'en assure quantitativement via le lemme suivant, simplifiant l'expression lorsque x et y sont semblables :

Lemme 4.1. *En posant* $\frac{x}{y} = \exp(c)$*, l'entropie croisée au voisinage de la diagonale* $x = y$ *a pour premier terme*

$$\text{entrop}(x, y) = \text{entrop}(x, xe^{-c}) = g(x)c^2 + o(c^2)$$

où $g : x \mapsto \frac{x}{2}\left[1 + \frac{x}{1-x}\right]$ *est une fonction croissante qui tend vers* $+\infty$ *lorsque* $x \to 1$*.*

Démonstration. Notons $\frac{x}{y} = \exp(c)$. Alors l'entropie croisée $\text{entrop}(x, y) = x\ln(\frac{x}{y}) + (1-x)\ln(\frac{1-x}{1-y})$ devient

$$\begin{aligned}
\text{entrop}(x, y) =& xc + (x-1)\ln(1 + \frac{x}{1-x}(1 - \exp(-c))) \\
=& xc + (x-1)\ln(1 + \frac{x}{1-x}(1 + c - c^2/2 + o(c^2))) \\
=& g(x)c^2 + o(c^2)
\end{aligned}$$

□

Il se trouve que cette approximation est moins correcte quand $c \to \ln(x)$. Dans ce coin, on repère une erreur relative plus élevée, comme illustrée sur cette image :

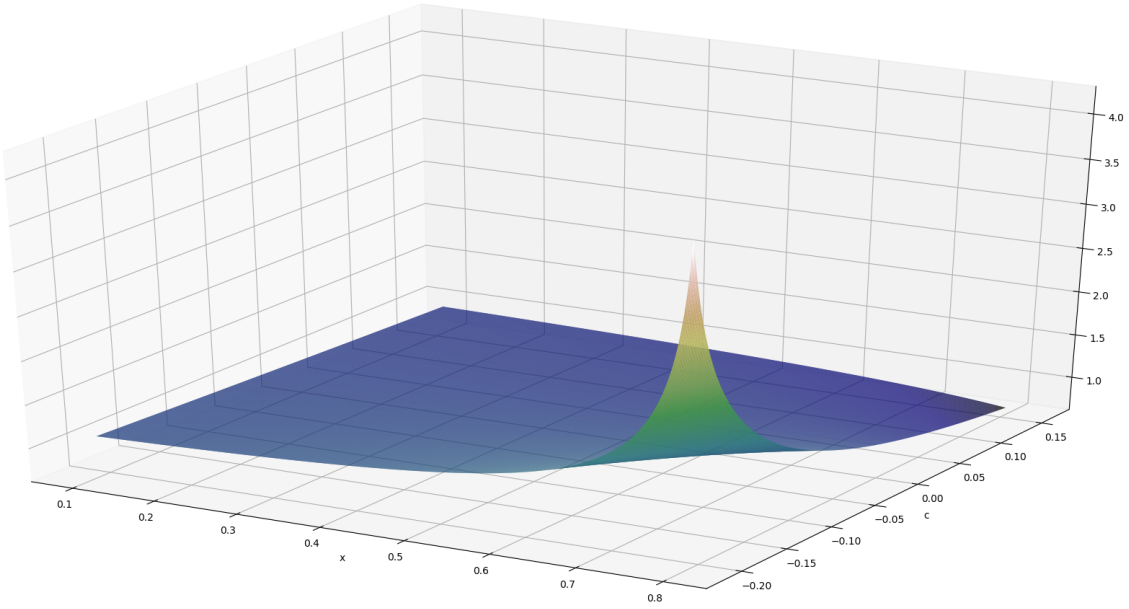


FIGURE 5 – Erreur relative : $(x, c) \mapsto \frac{\texttt{entrop}(x, xe^{-c})}{g(x)c^2}$ sur $[0.1, 0.8] \times [-0.2, 0.2]$

4.2 UMAP en basse dimension

L'attribution des poids en basse dimension s'effectue plus simplement, et surtout plus uniformément que celle en haute dimension. Le paramètre `min_dist` devient très important.

Rappelons qu'à chaque point de haute dimension, on associe un point en basse dimension, dont les positions sont soit initialisées aléatoirement, soit grâce à l'algorithme [1] Laplacian Eigenmaps. Si deux points sont liés par une arête a en haute dimension, on associe à leurs point x, y en basse dimension un poids fonction de la distance euclidienne :

$$\nu(a) = \phi_{\texttt{min\_dist}}(||x-y||) = \begin{cases} 1 & \text{si } ||x-y|| \leq \texttt{min\_dist} \\ e^{-(||x-y||-\texttt{min\_dist})} & \text{sinon.} \end{cases}$$

Si `min_dist` est strictement positif, l'entropie croisée entrop$(\mu(a), \nu(a))$ n'est même pas définie lorsque $\nu(a) = 1$: en effet, un poids en haute dimension $\mu(a) = 1$ n'est pas gênant grâce au facteur $1 - \mu(a)$. McInnes a comme idée de remplacer l'attribution du poids via ϕ par un proxy rationnel $\psi_{\texttt{min\_dist}}$:

$$\nu(a) = \psi_{\texttt{min\_dist}}(||x-y||) = \frac{1}{1 + a\,||x-y||^{2b}}$$

où a et b sont choisis de sorte à minimiser l'erreur de moindres carrés non-linéaires entre ψ et ϕ sur des mesures dont McInnes ne précise pas la nature.

Comparons par exemple les deux fonctions lorsque `min_dist` $= 0$:

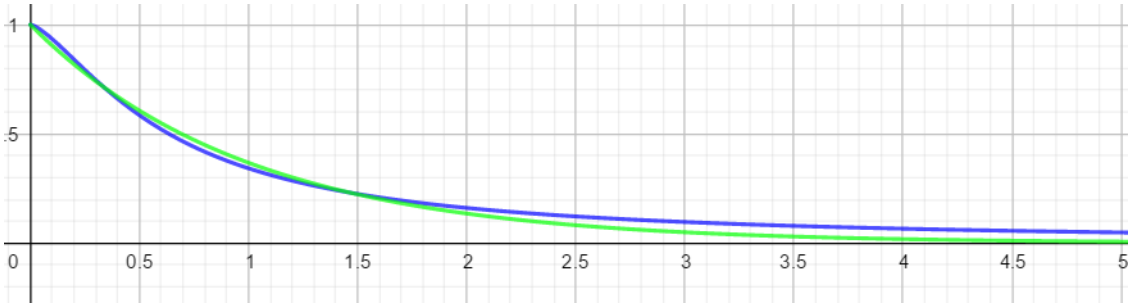


FIGURE 6 – ϕ_0 en vert, ψ_0 en bleu, avec $a = 1,929, b = 0,7915$

Pour compenser l'inévitable erreur apparaissant pour de grandes valeurs, la courbe bleue s'affaisse un peu trop et commence par passer sous la verte. Dans les conditions optimales, c'est cette approximation qui empêche à notre modèle 8 d'être exact.

4.3 Construction du graphe en haute dimension

L'opération de symétrisation de graphe semble bien mystérieuse. L'auteur choisit d'effectuer une opération d'union probabiliste : une arête aux poids orientés ρ_1, ρ_2 finit par avoir le poids $\rho_1 + \rho_2 - \rho_1\rho_2$.

Si l'on suit l'heuristique selon laquelle la normalisation accorde une appartenance en probabilité, cette opération d'union probabiliste paraît naturelle. Cependant, la comparaison des graphes de haute et de basse dimensions en sort compliquée. Plaçons nous dans un cadre généreux d'un tirage en densité uniforme. Prenons une arête a consituée de deux points x et y. Les rayons locaux σ_x et σ_y sont semblables, ainsi que la distance du plus proche voisin - particulièrement en grande dimension.

On a ainsi le plus souvent $\rho_1 \simeq \rho_2 = \rho$ et donc un poids $p_{[x,y]} = p = \exp(-\frac{||x-y||-\rho}{\sigma})$ menant après symétrisation au poids

$$\mu_{[x,y]} = 2p - p^2$$

En basse dimension, les points x', y' correspondant à x et y connaissent un poids

$$\nu_{[x,y]} = \exp(-\,||x' - y'|| - \rho)$$

On pourrait directement nommer ce poids $p'_{[x,y]}$ car sa construction est en tout point semblable à celle de p, il semblerait qu'il faille donc comparer l'entropie croisée de p à celle de p'.

Finalement, cela revient à vouloir une symétrisation suivant une fonction

$$\text{sym} : \left\{ \begin{array}{ccc} [0,1]^2 & \to & [0,1] \\ (x,y) & \mapsto & \text{sym}(x,y) \end{array} \right.$$

avec

1. $\text{sym}(x,y) = \text{sym}(y,x)$
2. $\text{sym}(x,x) = x$

On peut ainsi prendre la fonction moyenne

$$x, y \mapsto \frac{x+y}{2}$$

La moyenne perd cependant une propriété heuristiquement bien pratique de l'union probabiliste : dans la version originale d'UMAP, il suffit que y appartienne presque sûrement au voisinage immédiat de x pour que l'arête symétrisée soit de poids 1. Ce n'est pas le cas en prenant la fonction moyenne.

On rajoute donc la condition :

3. $\text{sym}(1,y) = \text{sym}(x,1) = 1$ pour tout x, y dans [0,1].

La fonction la plus simple qui valide les 3 conditions est la fonction sup :

$$x, y \mapsto \sup(x,y)$$

Cette fonction a le désavantage de ne pas prendre en compte la moitié de l'information qui lui est donnée, même lorsque les poids sont faibles (ce qui est le cas de la majorité des arêtes).

On a légèrement modifié UMAP pour qu'il puisse gérer ces différentes versions. En pratique, les résultats sont très semblables, la version moyenne étant parfois en deça, car ne gardant pas vraiment les plus proches voisins, comme expliqué ci-après. On peut par exemple constater le peu de différence entre les trois versions UMAP Moyenne, Originale et sUpMAP sur le traitement de la base de données classique `digits` 8.

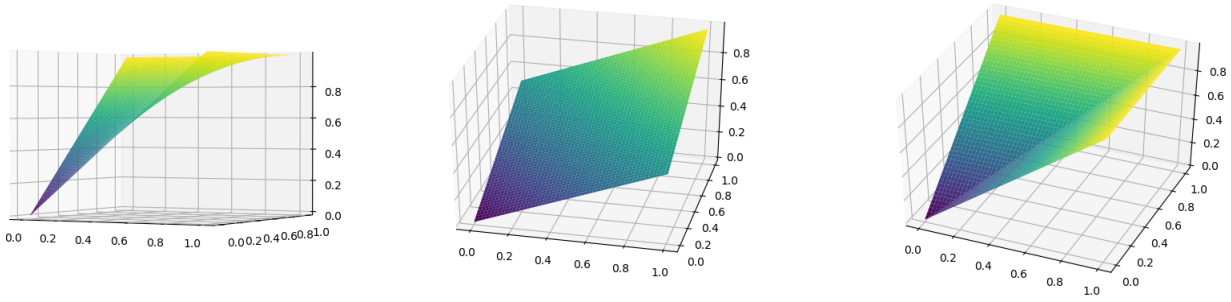


FIGURE 7 – De gauche à droite, sur $[0,1]^2$: Fonctions Union, Moyenne, Sup

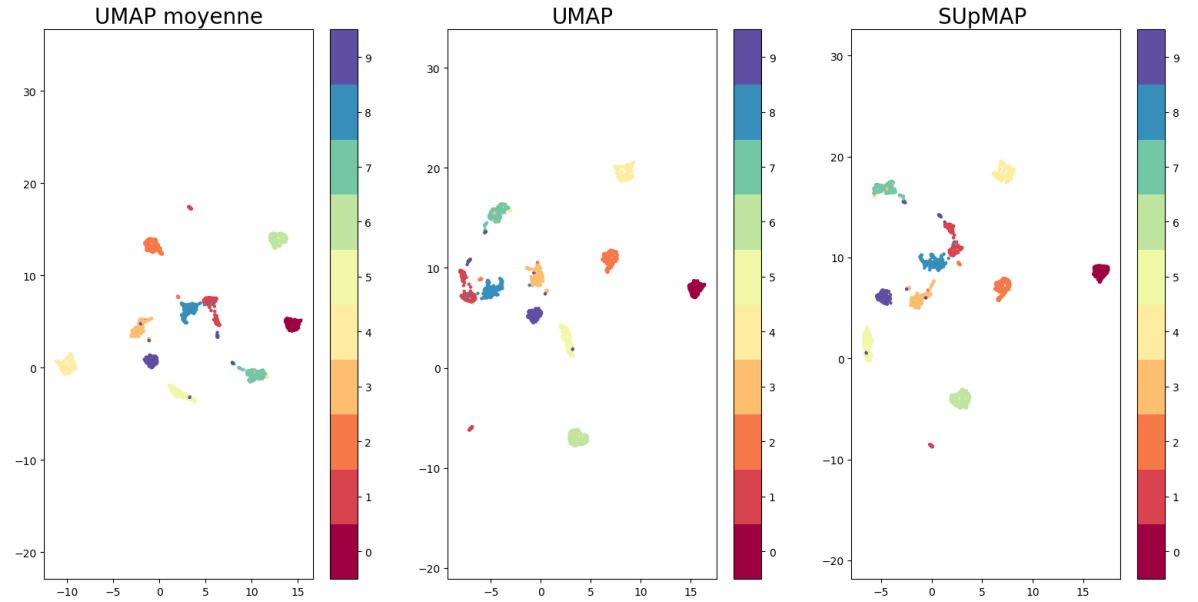


FIGURE 8 – 3 types de fonctionnement avec paramètres classiques

5 Constats empiriques

5.1 Paramètres à maîtriser

UMAP comprend une myriade de paramètres, portant tantôt sur la descente de gradient, tantôt sur la métrique de comparaison des points, tantôt sur le fonctionnement même de l'algorithme. Concernant ce dernier thème, deux paramètres prennent une importance particulière :

— `n_neighbors` est le nombre de plus proches voisins qu'on considère autour de chaque point dans l'algorithme. Plus il est grand, plus les voisinages pris en compte sont grands.

— `min_dist` est un paramètre de regroupement des points étudié en détail dans 4.2. Plus il est grand, plus les points auront tendance à être regroupés. .

Voici une illustration de ces phénomènes, suivant deux tirages initiaux, en réduction triviale $m = d$:

— Tirage de 4 régions denses dans une région plus diffuse

— Tirage bruité de 1600 points sur une sphère.

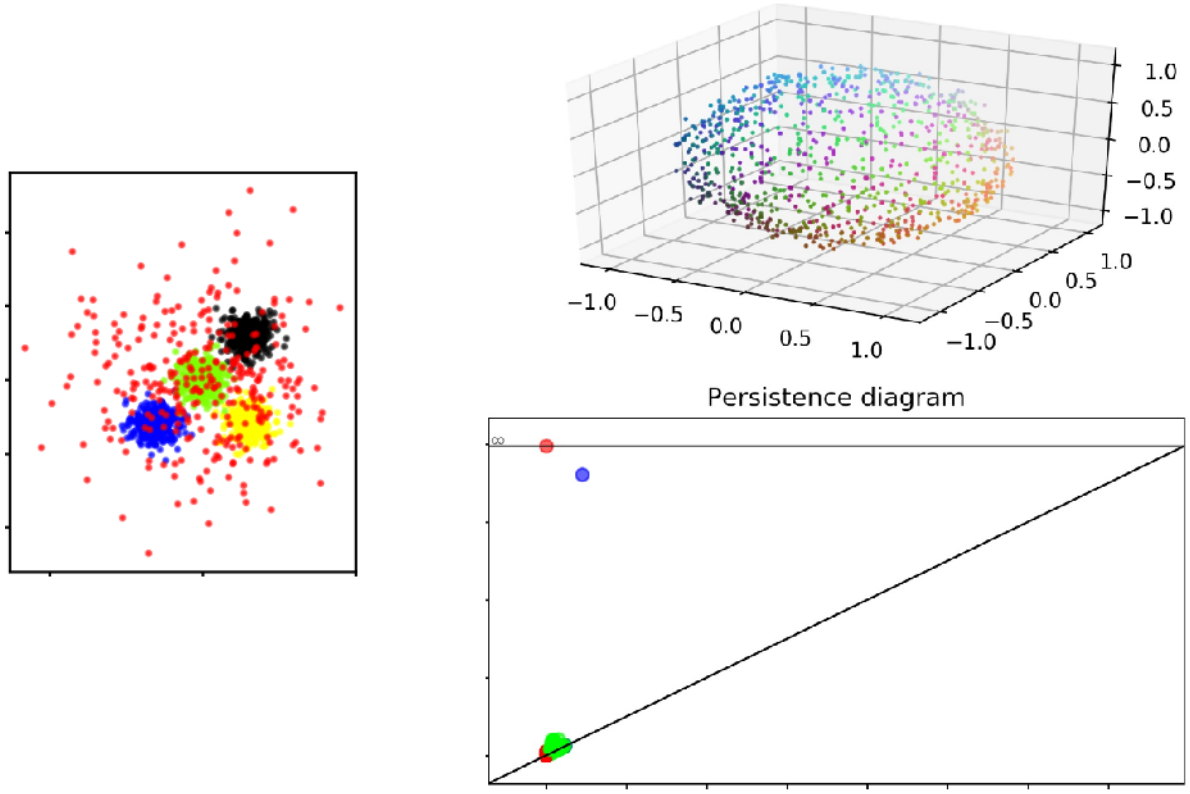


FIGURE 9 – Tirages originaux

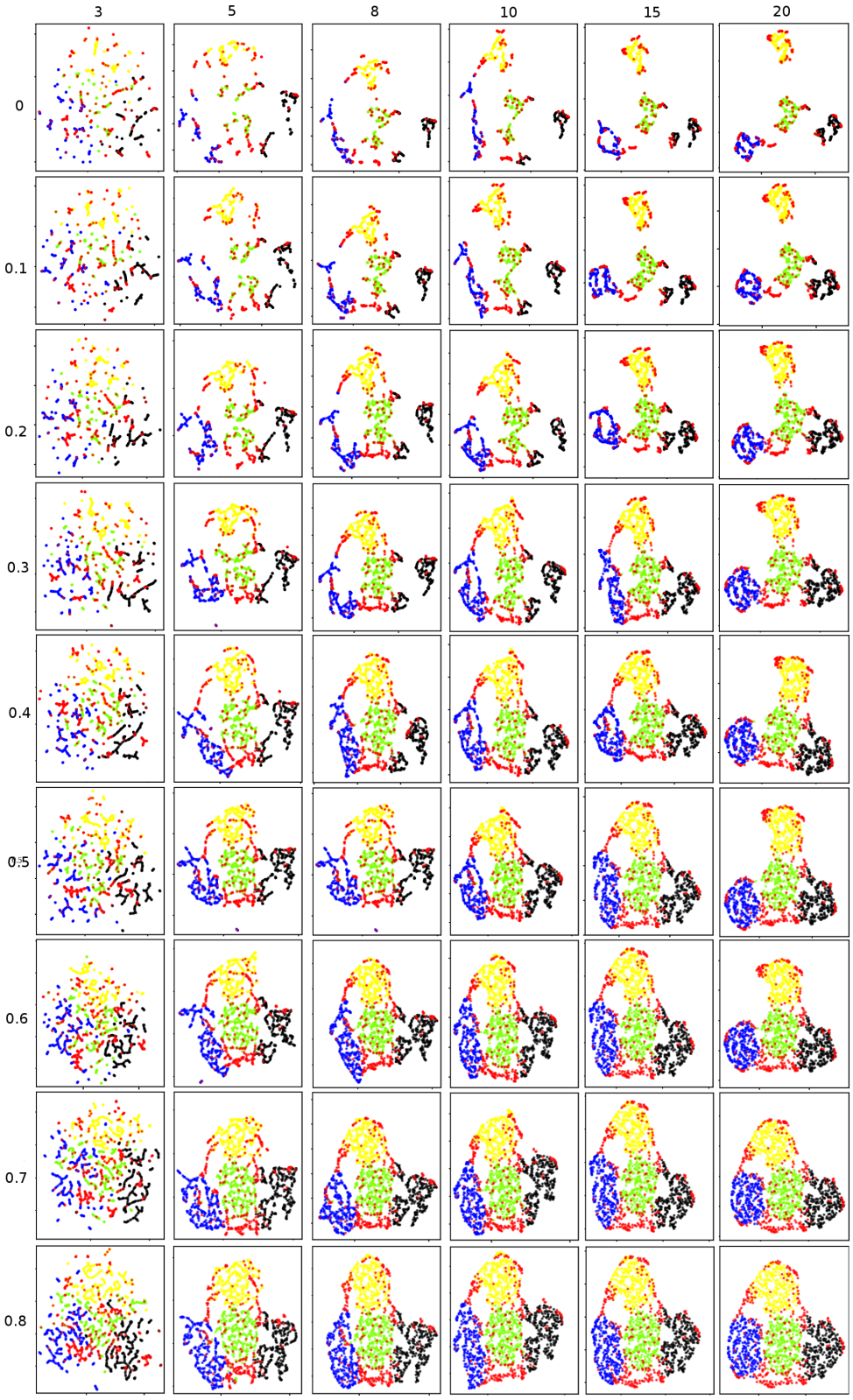


FIGURE 10 – Influence des paramètres `min_dist` (en abscisse) et `n_neighbors` (en ordonnée) sur des groupements

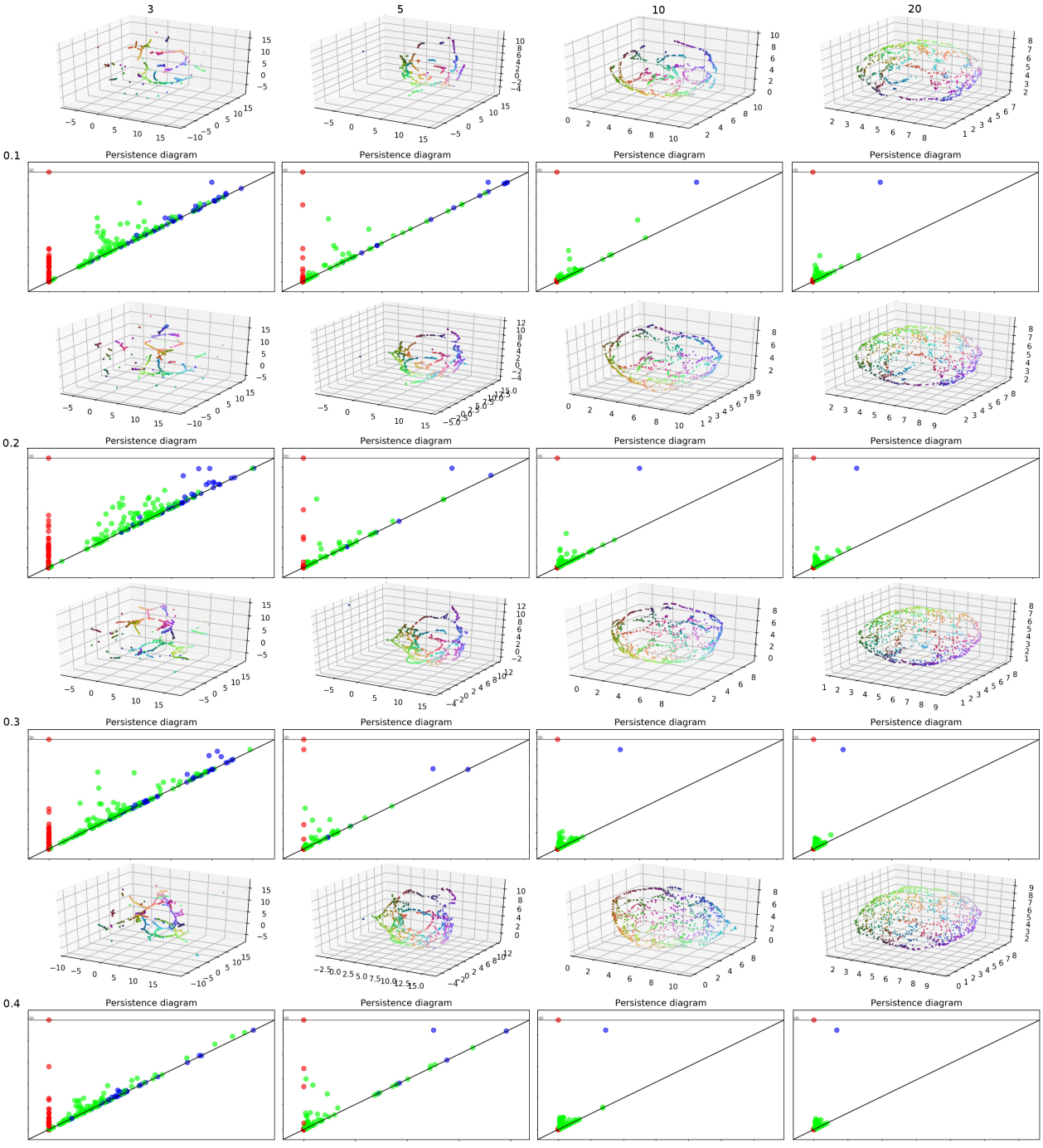


FIGURE 11 – Influence des paramètres `min_dist` (en abscisse) et `n_neighbors` (en ordonnée) sur une sphère

De ces simulations, on tire quelques enseignements empiriques :

— Quand `min_dist` est petit, les regroupements sont extrêmement denses. Quand ce paramètre augmente, UMAP rend compte du voisinage plus lointain en créant des liens. De plus, les regroupements gonflent et perdent en densité. On reviendra sur ce trait dans le modèle 8. Toutefois, augmenter ce paramètre diminue l'efficacité d'UMAP à partitionner les données.

— Quand `n_neighbors` est petit, UMAP tend à renvoyer des lignes. Quand ce paramètre augmente, on la structure globale est mieux prise en compte et les regroupements sont moins fibreux. Cela semble logique : pour prendre en compte un voisinage complet en d dimension, il faut couvrir toutes les directions, faisant varier k exponentiellement avec d.

— D'un point de vue topologique, c'est l'augmentation conjuguée des paramètres qui permet des résultats satisfaisants.

5.2 Quelle initialisation ?

Il est vital pour l'algorithme de fonctionner avec une initialisation spectrale (via Laplacian Eigenmaps). Une initialisation aléatoire mène à des minima locaux farfelus, souvent dûs à des plis, Dans l'exemple ci-dessous, on tente de réduire un tirage aléatoire de 2000 points dans $[0,3]\times[-1,1]$ en dimension 2, avec un paramètre `n_neighbors` $= 30$ et `min_dist` $= 0.1$. Les résultats sont les suivants :

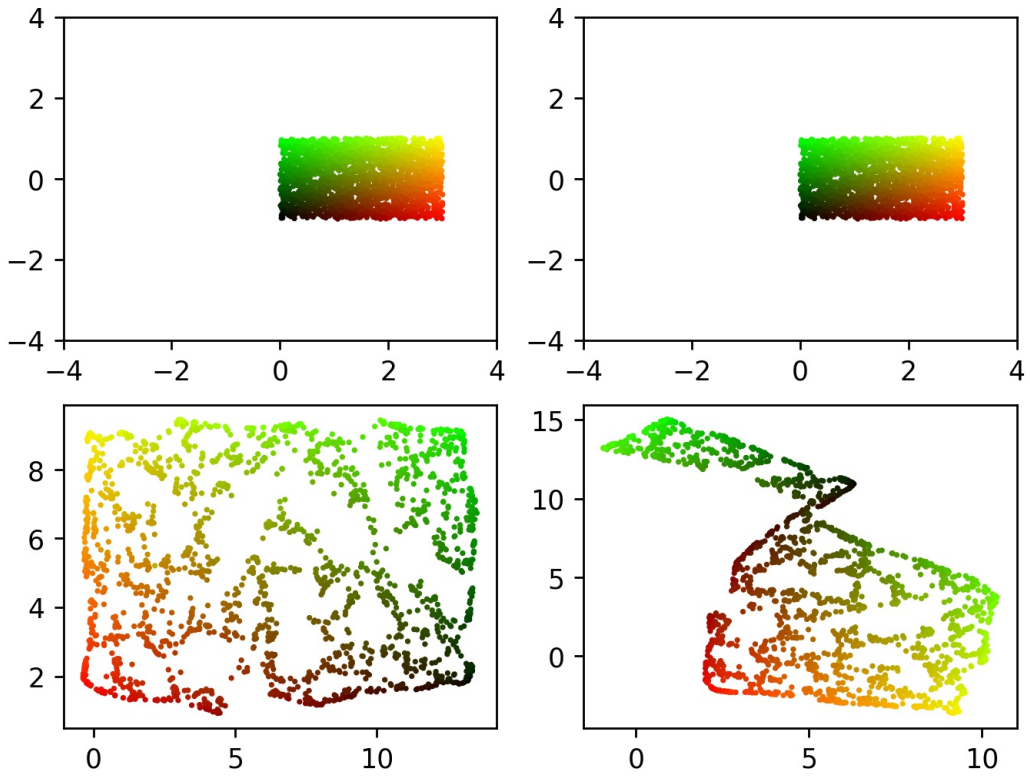


FIGURE 12 – Bas, gauche : Initialisation spectrale. Bas, droite : initialisation aléatoire

Même dans le cas d'une simple non-réduction ($m = d$), l'initialisation aléatoire semble poser problème. Il faut se méfier de l'interprétation de données initialisées aléatoirement. Ce phénomène de pliage est courant dans nos expériences.

5.3 Convergence de la descente de Gradient

La descente s'effectue en 500 étapes sur de petites données ($< 10^4$ points) et en 200 étapes sur de grandes données. Pour autant, ce n'est pas suffisant pour obtenir la convergence de l'algorithme. Lors d'une batterie de tests où on transformait un tirage uniforme de 2000 points d'un rectangle (en haut gauche) en un tirage de moins en en moins dense vers la droite, (en haut à droite), on a parfois remarqué une différence entre 500 étapes de descente de gradient (en bas à gauche) et 2000 étapes (en bas à droite).

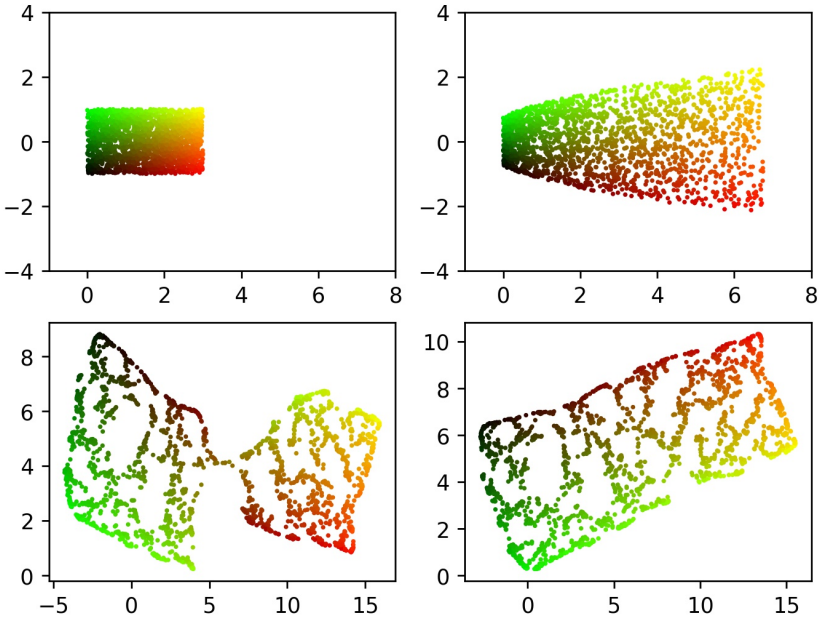


FIGURE 13 – Dépliage (`min_dist` : 0.1, `n_neighbors` : 33)

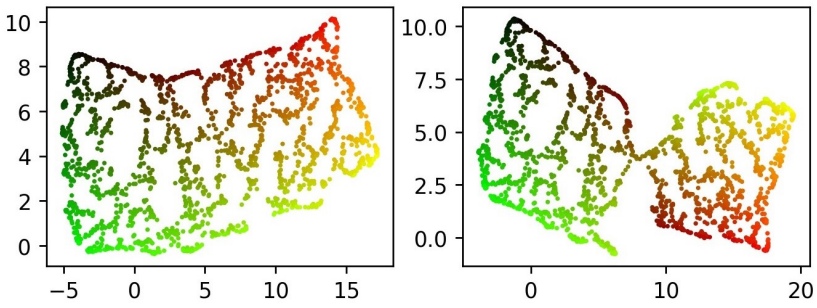


FIGURE 14 – Pliage (`min_dist` : 0,2, `n_neighbors` : 17)

L'incidence de ces phénomènes est toutefois assez basse. Dans cette batterie de 250 tests, on trouve une dizaine d'améliorations et cinq pliages. Rien qu'avec cet exemple simple une moitié (!) des simulations se solde par un pliage. L'optimisation est donc loin d'être efficace.

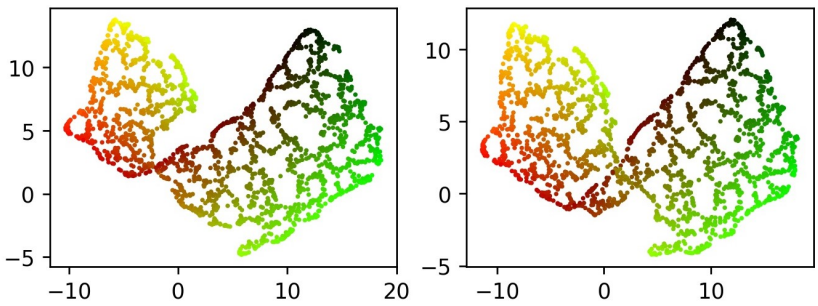


FIGURE 15 – Début de dépliage (`min_dist` : 0.4, `n_neighbors` : 10)

6 Cas particuliers

6.1 Tirage dans une boule

Lorsqu'on effectue un tirage uniforme dans une boule en dimension quelconque, UMAP (ainsi que ses versions modifiées) renvoie une boule dans la dimension plus petite.

Grâce à la connaissance de la fonction de répartition de la norme, on sait recadrer sa distribution en une distribution uniforme sur $[0, 1]$ - ou toute autre échelle linéaire. En particulier, cela permet d'associer à chaque point en haute dimension une couleur, de sorte que la distribution de couleur soit uniforme (ici, entre bleu et noir). On a besoin de ce recadrage car en haute dimension, l'immense majorité du volume d'une boule se trouve sur le bord.

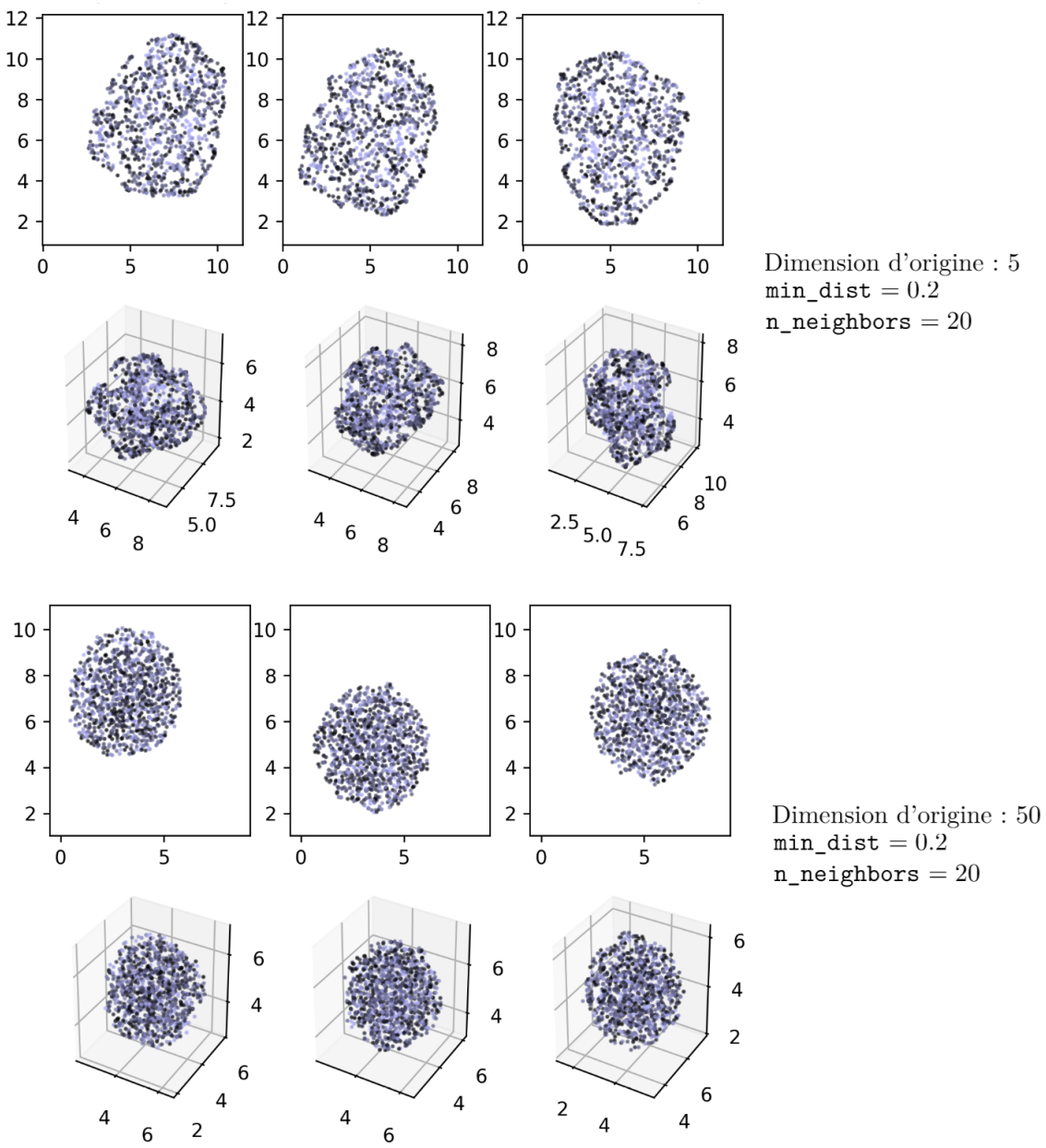
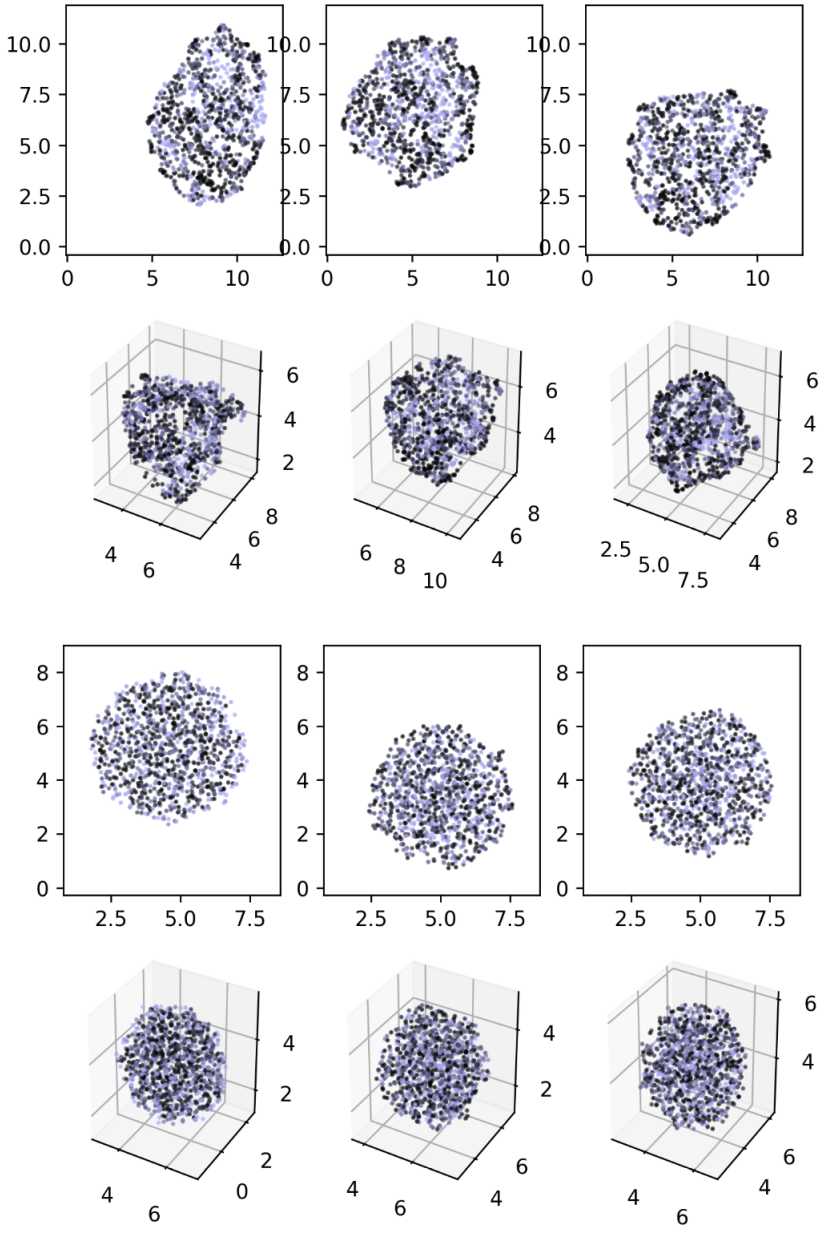


FIGURE 16 – Réduction en dim 2 et 3. De gauche à droite : version Moyenne, Originale et sUpMAP

6.2 Tirage gaussien

Un tirage gaussien isotrope en dimension quelconque renvoie une boule de densité uniforme (sans les problèmes de bord). Par exemple, si `n_neighbors` = 30, `min_dist` = 0.3 et qu'on tire suivant une gaussienne en dimension 5, on obtient grossièrement une boule.



Dimension d'origine : 5
`min_dist` = 0.2
`n_neighbors` = 20

Dimension d'origine : 50
`min_dist` = 0.2
`n_neighbors` = 20

FIGURE 17 – Réduction en dim 2 et 3. De gauche à droite : version Moyenne, Originale et supMAP

6.3 Tirage équiréparti d'une sous-variété

Observons la réduction triviale ($m = d = 3$) d'un tore bruité échantillonné de façon régulière, constitué de 1600 points :

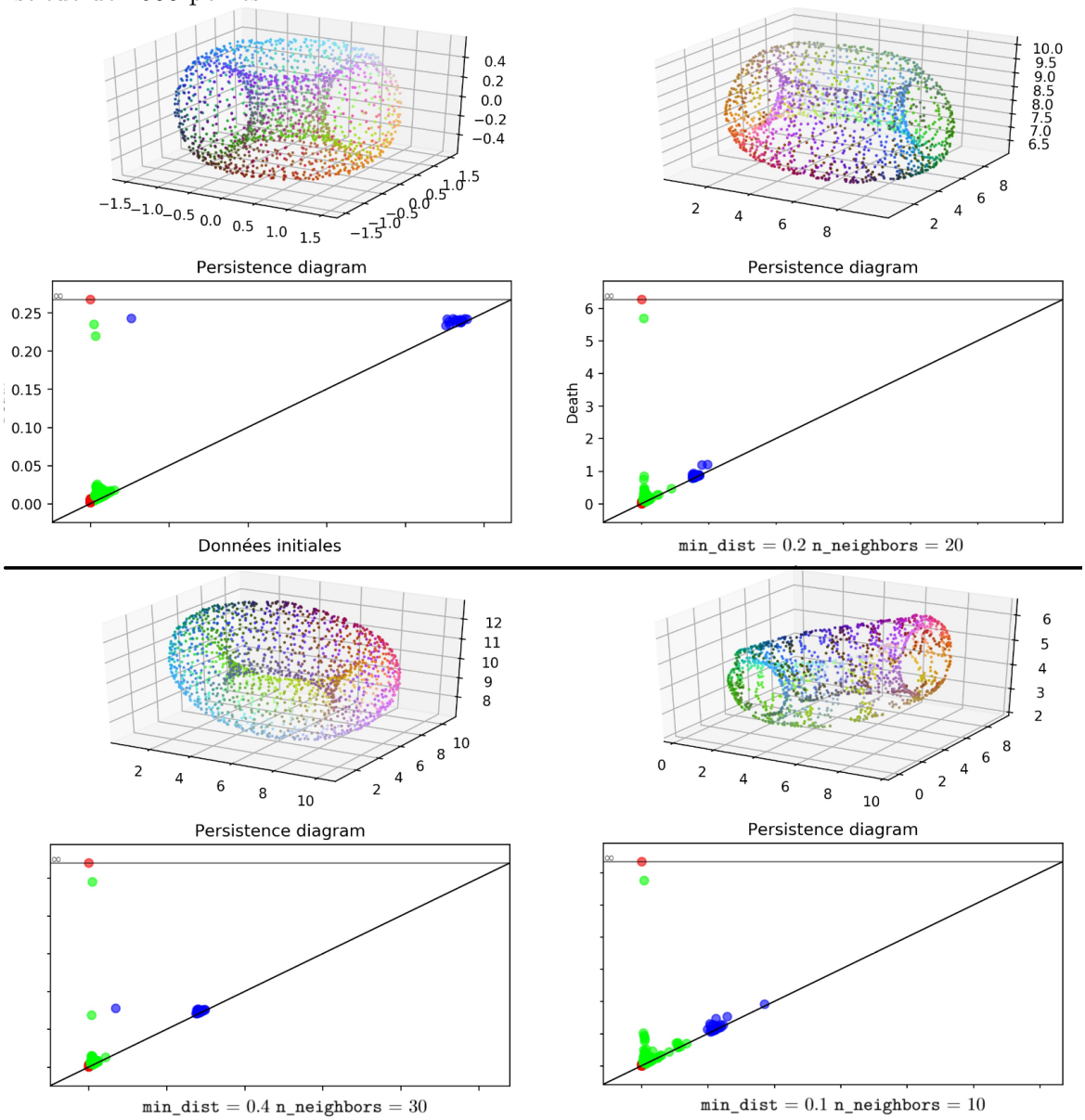


FIGURE 18 – Tirage initial en haut à gauche, puis réduction suivant 3 jeux de paramètres

La réduction n'est jamais parfaite, surtout du point de vue des diagramme de persistence. Il est toutefois remarquable de noter que la réduction du tore bruité se situe sur une surface : UMAP a remarqué que la dimension sous-jacente était 2, et ce dès de petits paramètres comme `min_dist` $= 0.1$ et `n_neighbors` $= 10$.

6.4 Tirage uniforme d'une sous-variété

Si l'échantillonage est uniforme (c'est-à-dire l'hypothèse faite par McInnes dans [6]) lorsque $m = d = 3$ pour $S^2 \subset \mathbb{R}^3$, et qu'on ajoute un peu de bruit, la réduction est nettement moins bonne à l'oeil que lors de l'exemple précédent. La sphère initiale est consituée de 1600 points.

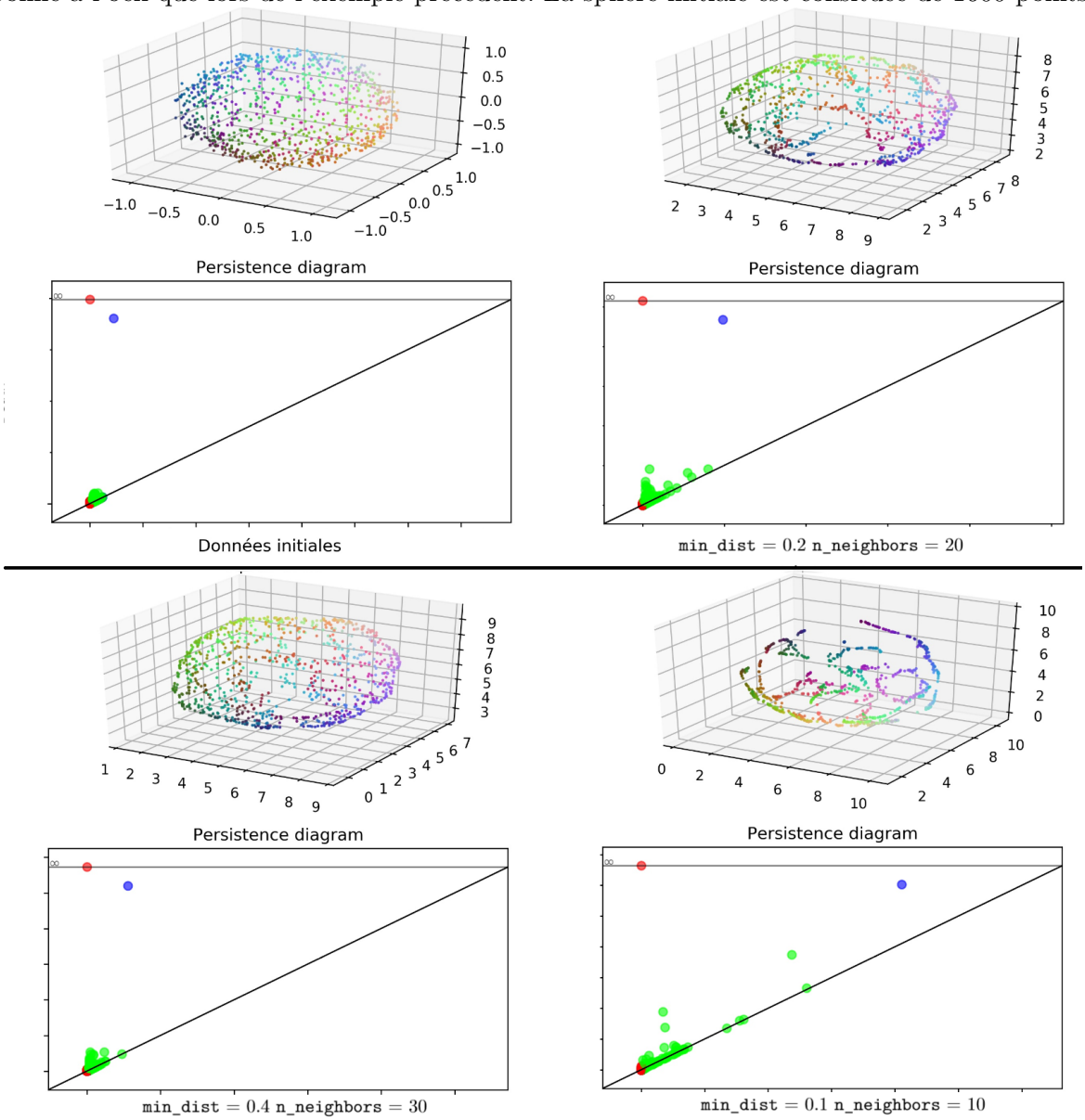


FIGURE 19 – Mêmes paramètres

7 Modèle de fonctionnement optimal

7.1 État de l'art

Il est difficile d'établir un critère pertinent pour évaluer l'efficacité d'une réduction de dimension.

On peut s'intéresser à l'existence de réductions parfaites sous des hypothèses fortes. C'est ce que font IsoMAP [9], qui recouvre des données provenant isométriquement d'un ouvert convexe en basse dimension, ou d'Hessian Eigenmaps [3] , qui recouvre des données provenant isométriquement d'un ouvert en basse dimension au prix d'une forte complexité et d'instabilité numérique. Le fonctionnement parfait sous ces hypothèses fortes laisse envisager un fonctionnement correct dans des situations semblables.

D'autres, comme Laplacian Eigenmaps [1] ou l'algorithme classique d'Analyse en Composante Principale (ACP) cherchent à déterminer les d meilleures contributions des points selon un critère précis. Pour l'ACP, il s'agit de déterminer les d meilleures directions orthogonales pour conserver la norme des points. Pour Laplacien Eigenmaps, comme toute fonction sur la variété se décompose en somme de fonctions propres de son Laplacien, il s'agit de déterminer les d contributions qui en minimisent l'énergie.

UMAP cherche un compromis entre les deux. L'heuristique développée par McInnes dans [6] veut que tout échantillonnage de sous-variété soit réduit vers un tirage uniforme de celle-ci ; comme pour un fonctionnement idéal. Pourtant, UMAP cherche à réduire le critère d'entropie croisée. Celui-ci est difficilement interprétable, contrairement aux deux exemples du paragraphe précédent...

7.2 Reflexion sur l'efficacité d'une réduction de dimension

La visualisation de haute dimension en basse dimension ne saurait être parfaite. Par exemple, en dimension n, on peut construire une figure à $n+1$ points équidistants deux-à-deux, alors que l'inégalité triangulaire rend cette même construction impossible en dimension $n' < n$. Il est impossible de visualiser correctement un tétraèdre régulier en dimension 2.
La réduction parfaite de dimension se résume de la sorte :

Réduction parfaite de dimension 7.1. *Étant donnés n points en haute dimension m, si la configuration des points existe en dimension d, alors renvoyer cette configuration.*

Posons nous la question de l'efficacité d'une réduction de points répartis selon une structure de dimension $> d$ qu'on tente de réduire en dimension d. C'est dans ce cadre qu'UMAP est utilisé dans les sciences expérimentales.

On pourrait espérer pouvoir remonter les géodésiques entre deux points en regardant les plus proches voisins, mais on subit grosso modo les mêmes difficultés qui sont qu'on ne couvre qu'un petit nombre de directions différentes. C'est une interprétation à ne surtout pas faire ! Par exemple, quand UMAP réduit des tirages uniformes dans une boule euclidienne de dimension 100 vers la dimension 2 ou 3, on trouve respectivement les images ci-après et il est clair que les points du bord (de couleurs foncées) ne sont pas réduits sur le bord ; les voisins de ces points sont complètement dispersés. Il n'y a guère que les réductions où $m = d$ qui peuvent réussir un tel tour de passe-passe.

7.3 Où se situe UMAP : mi-chemin entre réduction parfaite et partitionnement

UMAP n'a pas de fonctionnement optimal. La différence de traitement entre les haute et basse dimensions empêche d'obtenir un coût nul dans le cadre d'une possible réduction optimale de dimension 7.1. Le traitement n'en est pas pour autant catastrophique, et ce cadre mène à des

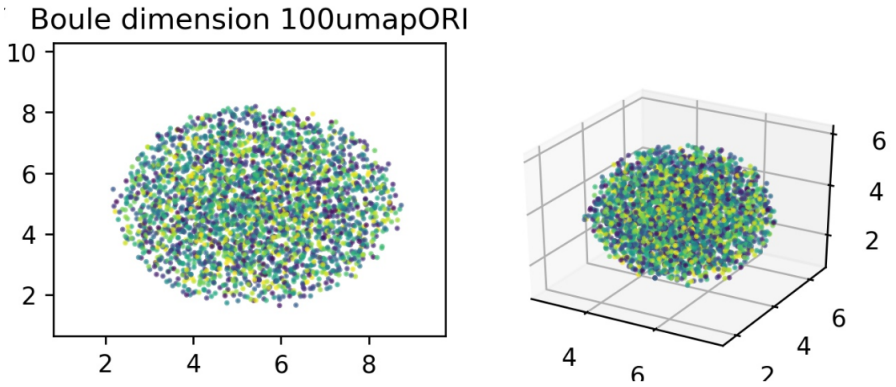


FIGURE 20 – Les couleurs sont fonctions de la norme et ont été recadrées de sorte à ce qu'elles soient tirées uniformément sur une échelle linéaire

résultats corrects pour l'oeil humain. Il n'est cependant pas possible de s'en satisfaire de façon générale.

En vérité, il semble que c'est la robustesse d'UMAP dans les cadres de réduction difficile qui fait sa force, comme dans la base de donnée *digits* 8. En effet, UMAP est robuste pour effectur du clustering (*Partitionnement de données*, en français).

Conserver les composantes connexes est la seule garantie qu'on peut demander lorsque $d \ll m$. UMAP ne réfléchit cependant pas en terme de composante connexe classique ; mais il permet de distinguer les groupes dits "sympathiques", les parties d'échantillons dont tous les k plus proche voisins appartiennent à la même partie. Si l'on considère que les points sont tirés sur une sous-variété, cette distinction est généralement moins fine que celle de la composante connexe classique ; rejoignant plutôt celle de connexité d'un graphe lorsque $k = 1$ et servant plus ou moins d'entre deux lorsque $k > 1$. De fait, si des k-composantes connexes distinctes existent, UMAP le remarquera et éloignera les groupes via une des astuces computationnelles de descente de gradient de [6] : l'échantillonnage négatif.

Plus k est grand, plus les k-composantes sont grosses. On ne peut cependant pas donner de sens à la position des clusters par rapport aux autres tant qu'ils ne sont pas reliés par un "pont", et si c'est le cas, les clusters ne sont proches que dans le sens où une partie de leurs éléments - ceux qui sont directement reliés - le sont. Si l'on reprend le test sur *digits* 8, l'interprétation rigoureuse qu'on peut faire des trois résultats est la même.

Cependant, il suffit que quelques liens relient quelques unes des quasi k-composantes pour que la séparation des blocs soit incomplète. comme l'illustre l'exemple suivant.

On a tiré 300 points suivant 5 gaussiennes

— L'une centrée en 0, de grande variance *(points rouges)*
— L'une centrée en 0, de petite variance *(points verts)*
— L'une centrée plus bas à gauche, de petite variance *(points bleus)*
— L'une centrée plus bas à droite, de petite variance *(points jaunes)*
— L'une centrée plus haut à droite, de petite variance *(points noirs)*

On peut à travers cette figure observer comment fonctionne un clustering imparfait à travers UMAP. Les boules denses restent des boules (en témoignent les blocs vert, jaune, bleu et noir). La banlieue de ces boules denses vient s'y coller. Les quelques points qui se trouvaient entre les deux forment des petits liens.

En effet, UMAP et sUpMAP favorisent le clustering même dans le cadre de quasi k-composantes. Lorsque le poids en grande dimension vaut 1, le poids associé en basse dimension va se rapprocher de grandes valeurs pour éviter un coût prohibitif. Toutefois, il n'y pas d'assurance théorique, car c'est la somme des entropies croisées qu'on compare. Ces termes sont simplement plus gros que les autres.

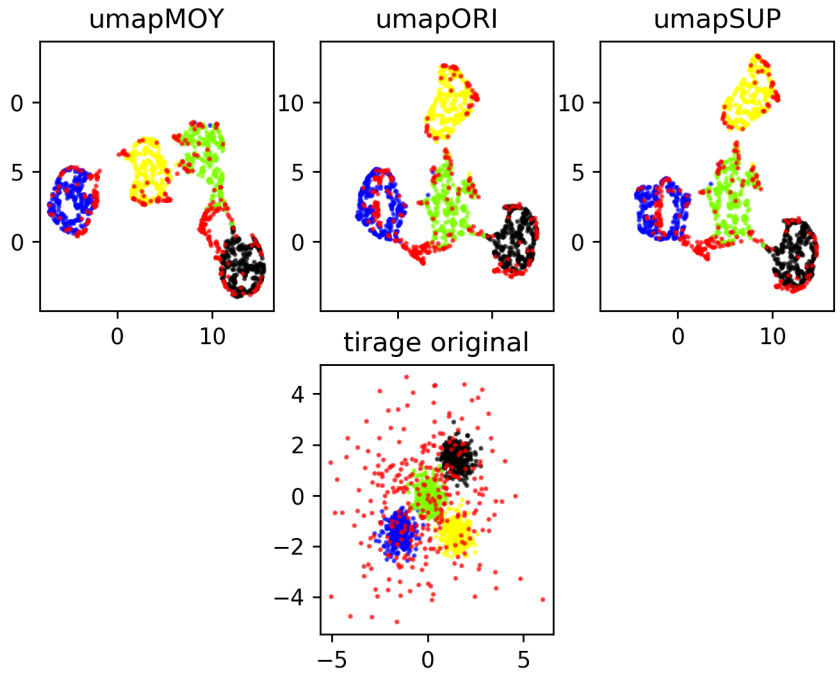


FIGURE 21 – Partitionnement de données avec `n_neighbors` = 20, `min_dist` = 0.2

8 Modèle

UMAP ne compare pas localement les distances en basse dimension comme il le fait en haute dimension, car il calcule un rayon local en haute dimension mais pas en basse. Cela laisse supposer qu'il existe un rayon local par défaut en basse dimension, laissant penser qu'UMAP cherche à reproduire un tirage de densité uniforme. [1]

Cela nous pousse à chercher à montrer que dans des bonnes conditions, il existe une "projection" $f : \mathcal{U} \subset \mathbb{R}^m \to \mathbb{R}^d$ telle que les configurations de points M et $f(M)$ connaissent une entropie croisée faible. Il faut de *bonnes conditions*, car il est impossible de retrouver les données initiales si elles sont tournées dans tous les sens.

L'objectif de notre modèle est de montrer que sous certaines hypothèses de quasi-isométricité et de bonne répartition des points, il existe une configuration à bas coût. Pour cela, on s'efforcera de donner une majoration de l'entropie croisée engendrée par ces hypothèses.

8.1 Description

Voici le cadre et les notation utilisés dans cette section :

- On utilise la version modifiée sUpMAP
- M est un nuage de points inclus dans $\mathcal{U}$ une partie de $\mathbb{R}^m$ de mesure de Lebesgue finie. Dans des considérations probabilistes, on considère que M est un échantillonage de n points d'une variable aléatoire Z à support dans $\mathcal{U}$.
- $\mathcal{O}$ 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, vu la perte de dimension, de projection. Pour les considérations probabilistes, on veut que $f(Z) \sim Y$, où Y est un tirage uniforme sur O.
- A est l'ensemble des arêtes prises en compte par l'algorithme avec $k = \texttt{n\_neighbors}$.
- Les estimations n'utilisent le proxy rationnel $\psi_{\texttt{min\_dist}}$, c'est-à-dire qu'en basse dimension le poids de l'arête $a = [f(x), f(y)]$ s'écrit :

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

  Un problème de définition réapparaît : le cas où $\nu(a) = 1$. On effectuera nos estimations lorsque cette égalité est fausse. D'autre part, nos estimations ne sont pas exactement conformes à celles d'UMAP.

L'objectif est de donner une estimation de l'entropie croisée entre M et $f(M)$ fournie par UMAP.

UMAP n'est pas sensible aux applications de multiples d'isométrie, et s'occupe uniquement des voisinages de chaque point. Demander à ce que f ressemble localement à un multiple d'isométrie est une hypothèse beaucoup moins forte que celle d'une réduction parfaite 7.1 qui demande à f de ressembler à une isométrie. Considérons maintenant un recouvrement de $\mathcal{U}$ de p ouverts :

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

Hypothèse 8.1 (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*.

1. On peut considérer qu'une réduction de dimension devrait respecter les différences de densité en haute dimension. C'est ce qui a été étudié dans DensMAP [7] en rajoutant dans le coût à optimiser un terme de perte pour mieux respecter la densité : malheureusement, rajouter ce terme empêche toute l'analyse mathématique de cette section.

Définition 8.2 (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.

Exemple 8.3. 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.

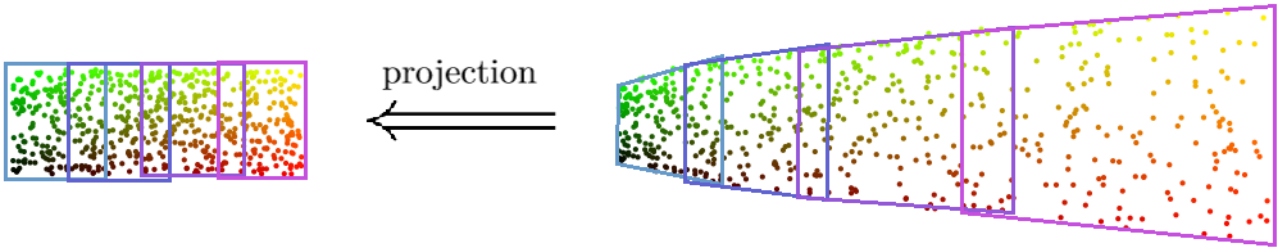


FIGURE 22 – Recouvrement constitué de 4 ouverts

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.

Définition 8.4 (Arête de poids fort)**.** On dit que $a = [x, y] \in A$ est une **arête de poids fort** lorsque $\mu(a) = 1$, c'est-à-dire lorsque x est le plus proche voisin de y, ou vice-versa. L'ensemble des arêtes de poids fort est noté A^1. Dans cet ensemble, on a $\forall a \in A^1, \text{entrop}(\mu(a), \nu(a)) = -\ln(\nu(a))$ En particulier, si on a de même $\nu(a) = 1$, la contribution de cette arête est nulle. De telles arêtes forment l'ensemble $A^{1,+}$ et on pose $A^{1,-} = A^1 \backslash A^{1,+}$. Par opposition, on pose l'ensemble des arêtes de poids faibles $A^0 = A \backslash A^1$. Par analogie, les $a \in A^0$ tels que $\nu(a) = 1$ forment l'ensemble $A^{0,+}$.

Concentrons nous sur la majorité des arêtes, formant l'ensemble $A^{0,-} = A^0 \backslash A^{0,+}$. L'étude suivante porte sur une estimation de leur coût.

Fixons $a[x, y] \in A^{0,-}$ et posons $c = \ln\left(\frac{\mu(a)}{\nu(a)}\right)$.

Supposons que y est le j-ème plus proche voisin de x, et réciproquement x le j'-ème plus proche voisin de y. On suppose que $[x, y]$ est une arête vivant dans un des ouverts $\mathcal{U}_i$ du recouvrement. On note d_q^x la distance entre x et son q-ème plus proche voisin.
Avant l'étape de symétrisation, on a deux poids :

— $\exp(-\frac{d_i^x - d_1^x}{\sigma_x})$ centré en x

— $\exp(-\frac{d_{i'}^y - d_1^y}{\sigma_y})$ centré en y

Supposons que $\mu(a) = \exp(-\frac{d_k - d_1}{\sigma_x})$, c'est-à-dire que le poids centré en x est le plus grand des deux. On substitue la notation d_k à d_k^x.

D'autre part, on note $z = f(x)$ l'image de x en basse dimension, et on note $z_i = f(y)$. De la même façon, z_1 est l'image du plus proche voisin de x.

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

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'on plongement 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 pré-plongement $||z_1 - z||$.

En introduisant ces nouveaux termes, il vient :

Proposition 8.5. *En gardant les notations précédentes, pour toute arête $a \in A^{0,-}$, on a grâce à la quasi-isométricité 8.1*

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

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

Proposition 8.6. *Pour toute arête $a \in A^{1,-}$, on a*

$$c = \underbrace{\texttt{min\_dist} - ||z_1 - z||}_{\text{écart à la moyenne en dimension } d} + \underbrace{||z_k - z|| - ||z_1 - z||}_{\text{Estimation réalisée en 3}}$$

8.2 Interprétation des estimations

À la lumière de ces équations, comme nous cherchons à minimiser c le paramètre `min_dist` optimal pourrait s'interpréter comme la valeur moyenne des $||z_1 - z||$. Ces estimations sont toutefois réalisées sur les ensembles où `min_dist` $< ||z_k - z||$, biaisant la lecture de ce paramètre. C'est corroboré par nos expériences 10, 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** qu'on va décrire dans la partie suivante.

9 Limite intrinsèque à notre modèle : l'émiettement

Considérer qu'on tire uniformément des points sur une sous-variétés, et construire le complexe simplicial dont les arêtes sont formées par l'union des k plus proches voisins s'avère être un très mauvais choix pour approximer la topologie de la sous-variété : quel que soit le nombre de points, les tirages sont trop mal répartis pour qu'on puisse distinguer la nature de la sous variété. Ci-dessous, exemple avec un tirage uniforme sur la sphère, avec un petit nombre de points pour une meilleure lisibilité.

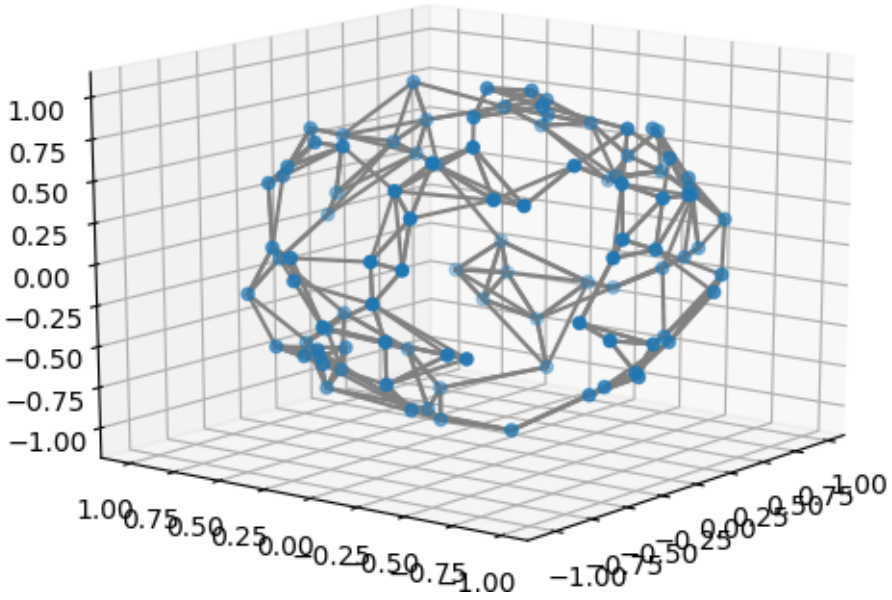


FIGURE 23 – Tirage uniforme bruité de 100 points sur la sphère. On relie chaque point à ses 4 plus proches voisins.

Un tirage uniforme n'est jamais vraiment bien réparti. Dans l'exemple ci-dessus, on repère des sortes de cluster, fruits de l'aléatoire. Après un passage d'UMAP, ils seront séparés des autres. C'est ce genre **d'émiettement** qu'on observe dans tous les exemples d'utilisation d'UMAP de ce rapport. On aura beau raisonner sur des comportements *en moyenne*, un tirage uniforme s'en écartera toujours.

Tirer plus de points sur la variété ne change malheureusement rien à ces problèmes ; les clusters aléatoires sont juste de plus petite taille. Augmenter le nombre de voisins permet d'éviter quelque peu cet effet, permettant aux points d'un cluster de découvrir des points hors de celui-ci, diminuant la variance des rayons locaux σ_x. Pour autant la normalisation fait que les poids importants sont uniquement attribués dans le cluster et on n'empêche pas le fond du problème.

Mathématiquement, on pourrait résumer ça de la façon suivante :

Conjecture 9.1. *Soit un tirage uniforme iid $X_1, \dots, X_n$ sur une sous-variété $\mathcal{M} \subset \mathbb{R}^m$ de dimension d et k un nombre de plus proches voisins.*

Posons comme 1-simplexes les segments reliant les points à leurs k-plus proches voisins. Considérons le plus grand complexe simplicial de dimension d M_n dont ces segments forment le 1-squelette.

Alors il n'y a aucun phénomène de convergence probabiliste de $Pers(M_n)$ vers $Pers(M)$, où $Pers$ est le diagramme de persistence d'un nuage de points, et sa limite en densité dans le cadre d'une variété.

Exemple 9.2. Reprenons le rectangle uniforme dilaté suivant l'abscisse. Suivant un bon découpage, d'après le modèle 8, UMAP devrait ramener le tirage vers un rectangle uniforme. Pourtant, ce n'est pas ce qu'on observe :

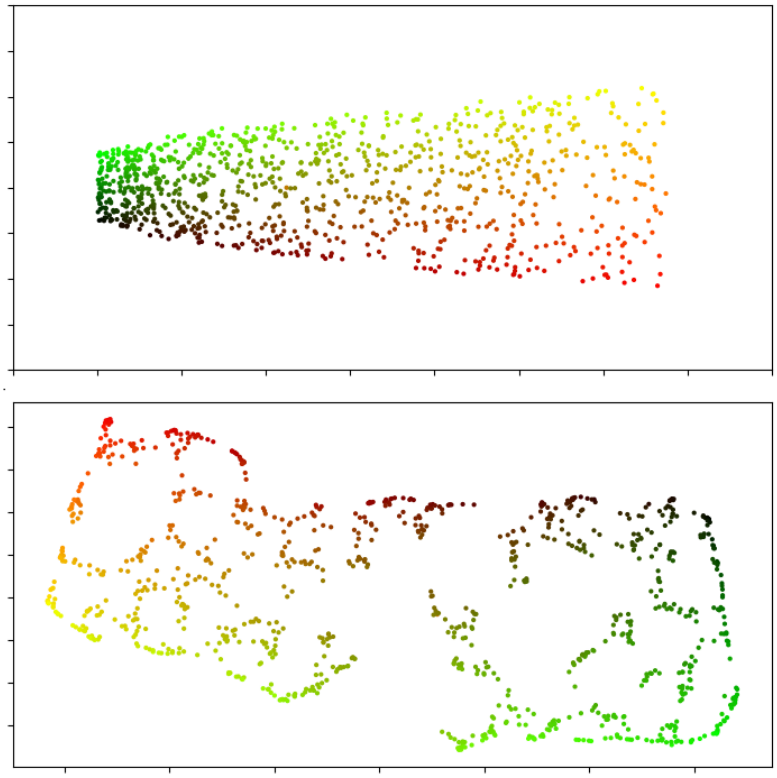


FIGURE 24 – Phénomène d'émiettement avec `min_dist` = 0.1 et `n_neighbors = 20`. En haut, échantillon initial. En bas, après UMAP.

À la place, on découvre des petites structures denses et de grands espaces vides au milieu du rectangle.

Augmenter `min_dist` permet de quelque peu réduire le phénomène d'émiettement :

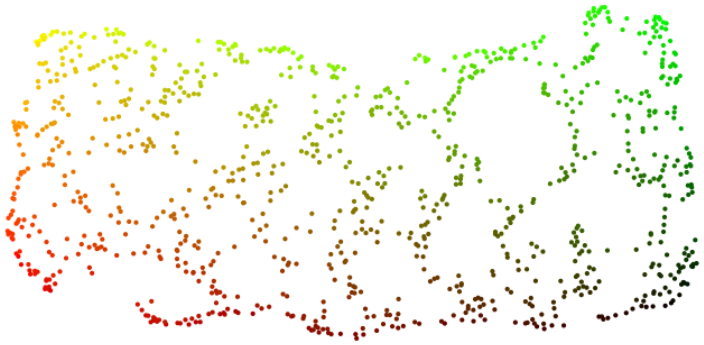


FIGURE 25 – Après UMAP, `min_dist` = 0.3 et `n_neighbors` = 20

Finalement, le modèle semble marcher principalement lorsque les points ne subissent pas un tirage uniforme mais un tirage bien réparti. En pratique, ce n'est malheureusement jamais le cas. Il faut donc bien avoir le phénomène d'émiettement en tête.

10 Conclusion

Finalement, le stage aura permis d'éclaircir certaines étapes de l'algorithme, comme le fonctionnement moyen de la normalisation 3, dans un cadre idéal puis dans le cadre d'un tirage de points sur une sous-variété de $\mathbb{R}^m$. La comparaison entre la haute et basse dimension a permis l'ajout de deux nouveaux types de fonctionnements, UMAP Moyenne et sUpMAP. En pratique, ceux-ci ne changent pas vraiment l'effet d'UMAP ; pour autant sous sUpMAP on a pu poser le modèle 8 qui explique certains comportements d'UMAP. En particulier, il justifie la proposition de McInnes qui était qu'UMAP réduisait des parties de densités différentes en des blocs de densité uniforme.

A Étude de la fonction Bêta incomplète

Il s'agit de comprendre où l'essentiel de la masse de l'espérance se trouve. Commençons par l'étude de la fonction Bêta incomplete en généralisant un peu le cadre.

On veut évaluer une intégrale $I_{a,b}$:

$$I_{a,b} = \int_0^1 (1-u)^a u^b \,\mathrm{d}u$$

Dans notre cas, c'est l'étude de $I_{a,b}$ lorsque $b \ll a$ qui nous intéresse. Il semble raisonnable de s'attendre à ce que toute la masse de l'intégrale soit concentrée aux petites valeurs de u, car l'élévation à la puissance a va considérablement affaiblir la contribution du reste. L'objectif de cette section est d'obtenir une estimation dans le genre de

$$\int_0^z (1-u)^a u^b \,\mathrm{d}u = I_{a,b}^z = I_{a,b}(1 - O(h(z)))$$

où $h(z) \to 0$ en décroissant lorsque z dépasse une certaine valeur.

Le candidat naturel pour cette valeur est l'argmax de $u \mapsto (1-u)^a u^b$, dont l'étude des variations montre que le maximum est atteint en $u = \frac{b}{a+b}$. On note $M_{a,b}$ ce maximum. Pour mettre en valeur ce maximum, on utilise une technique classique de reparamétrisation en posant $u = \frac{b}{a+b}(1+x)$ On s'intéresse dorénavant à

$$I_{a,b} = \int_{-1}^{a/b} \frac{b}{b+a} M_{a,b} f_{a,b}(x) \,\mathrm{d}x$$

où on a posé $f_{a,b}(x) = (1 - \frac{bx}{a})^a (1+x)^b$ le quotient de $(1-u)^a u^b$ et de $M_{a,b}$ après changement de variable. En particulier, $f_{a,b}(x) \leq 1$ avec égalité seulement lorsque $x = 0$.

Il convient d'étudier la décroissance de $f_{a,b}$. C'est plus simple en posant $g_{a,b}(x) = \ln(f_{a,b}(x)) = a\ln(1 - \frac{bx}{a}) + b\ln(1+x)$. l'inégalité classique $\ln(1-x) \leq -x - \frac{x^2}{2}$ montre que

$$g_{a,b}(x) \leq -bx\left(1 - \frac{\ln(1+x)}{x}\right) - \frac{(bx)^2}{2a}$$

pour $x > 0$.

Ainsi,

$$\begin{aligned}
\int_y^{a/b} f_{a,b}(x) &\leq \int_y^{a/b} \exp\left(-bx\left(1 - \frac{\ln(1+x)}{x}\right) - \frac{(bx)^2}{2a}\right) \mathrm{d}x \\
&\leq \int_y^{a/b} \exp\left(-bx\left(1 - \frac{\ln(1+y)}{y}\right) - \frac{(bx)^2}{2a}\right) \mathrm{d}x \\
&\leq \frac{\exp(as(y)^2/2)}{b} \int_{by}^{a} \exp(-\frac{(u+as(y))^2)}{2a}) \,\mathrm{d}u \\
&\leq \frac{\exp(as(y)^2/2)}{b} \int_{by+as(y)}^{a(1+s(y))} \exp(-u^2/2a) \,\mathrm{d}u \\
&\leq \frac{\sqrt{a}\exp(as(y)^2/2)}{b} \int_{\frac{by+as(y)}{\sqrt{a}}}^{\sqrt{a}(1+s(y))} \exp(-u^2/2) \,\mathrm{d}u \\
&\leq \frac{a}{b(by+as(y))} \exp\left(-(by+as(y))^2/2a + as(y)^2/2\right) \\
&\leq \frac{1}{b\left[s(y)+by/a\right]} \exp(-\frac{(by)^2}{2a} - \frac{as(y)^2}{2})
\end{aligned}$$

où on a posé $s(y) = 1 - \frac{\ln(1+y)}{y}$ et utilisé $\int_x^\infty e^{-t^2/2}\,\mathrm{d}t \leq \frac{1}{x}e^{-x^2/2}$. Le passage de la première à la seconde ligne provient de la croissance de s.

Cette majoration est agréable, mais c'est plutôt le rapport de l'intégrale tronquée avec l'intégrale complète qui nous intéresse. On s'attelle à majorer $\int f_{a,b}$ par une forme close plus simple. On a

$$\begin{aligned}\int_{-1}^{a/b} f_{a,b} =& \frac{a+b}{b} M_{a,b} B(a+1, b+1) \\ =& \sqrt{2\pi}\frac{a+b}{a+b+2}\sqrt{\frac{a}{b(a+b)}}\frac{l(a)l(b)}{l(a+b)}\end{aligned}$$

où $l(z)$ est le rapport entre la fonction Gamma et son approximation de Stirling $\sqrt{2\pi(z+1/6)}(\frac{z}{e})^z$. Comme $l(z)$ tend très vite vers 1, on peut écrire, pour $b \ll a$:

$$\int_{-1}^{a/b} f_{a,b} = \sqrt{\frac{2\pi}{b+1/6}}$$

en particulier,

$$\int_y^{a/b} f_{a,b}(x)\,\mathrm{d}x \leq \frac{\exp(-\frac{(by)^2}{2a} - \frac{as(y)^2}{2})}{\sqrt{2\pi b}s(y)} \int_{-1}^{a/b} f_{a,b}(x)\,\mathrm{d}x$$

Cela permet de conclure.

Théorème A.1. *Pour $0 \leq z \leq \frac{a}{b}$, on a :*

$$\int_0^{(z+1)\frac{b}{b+a}} (1-u)^a u^b = I_{a,b}(1 - O^*(\frac{\exp(-\frac{(bz)^2}{2a} - \frac{as(z)^2}{2})}{\sqrt{2\pi b}\,[s(z) + bz/a]}))$$

où $O^*(x)$ note un réel plus petit que x en valeur absolue.

B Fonction de répartition d'un tirage sur une variété

Si la densité est constante, on a à une constante multiplicative près

$$\begin{aligned} v_x(t) &= \int_{\mathcal{M}\cap B^m(x,t)} 1 \\ &= \int_{\phi^{-1}(B^m(x,t))} ||\partial_1\varphi \wedge \cdots \wedge \partial_d\varphi||\, \mathrm{d}y_1 \ldots \mathrm{d}y_d \end{aligned}$$

Remarquons que $||\partial_1\varphi \wedge \cdots \wedge \partial_d\varphi||^2$ est le déterminant de la matrice de Gram $A \in \mathbb{R}^{d\times d}$ définie par les produits scalaires dans l'espace ambiant :

$$A_{i,j} = \langle \partial_i\varphi, \partial_j\varphi \rangle$$

L'expression sous le signe intégral est liée à la seconde du plongement. Plus précisément, s'il a la forme d'un graphe, on a, en notant $\mathrm{pr}_1 : \mathbb{R}^{d+(m-d)} \to \mathbb{R}^d$ la projection sur les d-premières coordonnées :

$$\mathrm{pr}_1 \circ \varphi = \mathrm{Id}_{\mathbb{R}^d}$$

Notons alors $\varphi = (\varphi^1, \ldots, \varphi^m)$ les composantes. La définition d'espace tangent impose, pour tout $1 \leq j \leq d$, $d < i \leq m$

$$\partial_j\varphi^i(0) = 0$$

Au voisinage de zéro, on a donc l'approximation linéaire des dérivées :

$$\partial_j\varphi(y) = e_j + y_j \sum_{d<i\leq m} [\partial_j\varphi^i + o(1)]e_i$$

et finalement, on a l'approximation quadratique

$$A_{j,j} = 1 + y_j^2 \sum_{d<i\leq m} [(\partial_j\varphi^i)^2 + o(1)]e_i$$

qui mène à l'approximation suivante de l'élément de surface (car les autre composantes du produit menant au déterminant sont d'ordre 3 ou plus si $d \geq 3$) :

$$||\partial_1\varphi \wedge \cdots \wedge \partial_d\varphi||^2 = 1 + \sum_{i,j} (\partial_j\varphi^i)^2 y_j^2 + o(||y||^2)$$

Il reste à déterminer la variation de volume. On intègre sur

$$B^\varphi(t) = \varphi^{-1}(B_x(t)) = \{y |\, ||\varphi(y)|| \leq t\}$$

Comme φ est proche de l'identité, on s'attend à ce que $B^\varphi(t) \simeq B(t)$.
Écrivons le DL quadratique de φ au voisinage de 0 :

$$\varphi(y) = \sum_{j=1}^{d} y_j e_j + \frac{1}{2} \sum_{d<k\leq m} [{}^t y H_k y + o(||y||^2)]e_k$$

où H_k est la hessienne de φ^k en zéro. Il vient naturellement :

$$||\varphi(y)||^2 = ||y||^2 + \frac{1}{4} \sum_{d<k\leq m} ({}^t y H_k y)^2 + o(||y||^4)$$

Comme l'application

$$y \mapsto \frac{1}{4} \sum_{d<k\leq m} ({}^t y H_k y)^2$$

est un tenseur d'ordre 4 pris en y sur ces 4 entrées, on peut au moins dire par continuité qu'il existe une constante C telle que

$$\frac{1}{4}\sum_{d<k\leq m}({}^t yH_k y)^2 \leq C\,||y||^4$$

un développement à l'ordre 2 pour résoudre l'inéquation $||y||^2 + C\,||y||^4 \leq t^2$ donne

$$B(t(\sqrt{1-Ct^2}) \subset B^\phi(t) \subset B(t)$$

Même si le développement ne donne pas de reste exact, on en conclut que quand un voisinage de 0, lorsque la densité est constante à $f(x)$, on a

$$\begin{aligned} v_x(t) - f(x)V_d t^d \leq & \frac{f(x)V_d}{d} t^{d+2} \\ v_x(t) - f(x)V_d t^d \geq & f(x)V_d t^d[(1-Ct^2)^{\frac{d}{2}} - 1] \end{aligned}$$

Plus concisement, on a le théorème suivant :

Théorème B.1. *Si v_x est la fonction de répartition centrée en x d'un tirage sur une sous-variété de classe au moins C^2 de densité constante autour de x f, alors*

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

Références

[1] M. Belkin and P. Niyogi, *Laplacian eigenmaps for dimensionality reduction and data representation*, Neural Comput., 15 (2003), pp. 1373–1396.

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

[3] 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.

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

[5] N. S. Leland Mcinnes, John Healy and L. Grossberger, *Umap : Uniform manifold approximation and projection*, The Journal of Open Source Software, 3 (2018), p. 861.

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

[7] A. Narayan, B. Berger, and H. Cho, *Density-preserving data visualization unveils dynamic patterns of single-cell transcriptomic variability*, bioRxiv, (2020).

[8] L. S. Sam Roweis, *Nonlinear dimensionality reduction by locally linear embedding*, Science, (2000), pp. 2323–2326.

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