

State of the art

How to evaluate the faithfulness of visual data projection ?

December 17, 2021

Students :

Cyril	GARDENAT	cyril.gardenat9@etu.univ-lorraine.fr
Nathan	METZGER	nathan.metzger6@etu.univ-lorraine.fr
Muhamed	SILIC	muhamed.silic3@etu.univ-lorraine.fr

Tutors :

Lydia BOUDJELOUD-ASSALA

Keywords: projection; méthodes; critères; évaluation; qualité; réduction de dimension

Abstract :

Dans cet état de l'art, nous nous sommes concentrés sur une approche de visualisation de données : la projection. Ainsi, différentes méthodes de projection (linéaires et non linéaires, supervisées et non supervisées) ont été présentées de façon non exhaustive. Une perte de donnée étant inévitable lors de la projection de grands jeux de données, la problématique de la qualité de cette dernière se pose. De ce fait, nous avons également cité les biais qui peuvent découler d'une projection ainsi que les critères qui permettent de juger de sa qualité.

Contents

Introduction	1
1 Les méthodes de projection	5
1.1 Analyse en Composantes Principales(ACP)	6
1.2 Linear Multidimensional Scaling (MDS)	6
1.3 ClassNerv	7
1.4 S-Isomap	8
1.5 t-SNE	9
1.6 Autres méthodes de projection non linéaires	10
1.7 Conclusion partie	10
2 Les critères d'évaluation des projections	11
2.1 Visualisation du stress	11
2.2 Les critères de mesure	11
2.3 Conclusion de la partie :	16
Conclusion	17

Introduction

La représentation visuelle des grands jeux de données est un enjeu de plus en plus important dans de nombreux domaines scientifiques. En effet, une immense partie des analyses de données se fait dans des domaines de recherches dont les acteurs ne sont pas des “data analyst”. Cela rend l’analyse compliquée et peut engendrer certains biais lors de l’interprétation des données [1]. C’est pour cela, que le domaine de la visualisation des données se doit d’être accessible à ce genre de profil.

La visualisation permet aux utilisateurs d’avoir un aperçu global des données et de choisir en conséquence quel type de traitement et quel paramétrage utiliser dans leurs analyses. Elle a donc pour rôle d’être une interface interactive pour représenter et naviguer à travers les données extraites, et ce dans le but de les rendre compréhensibles et donc exploitables[2]. C’est exactement ce vers quoi tend ce domaine : “*Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets*” [3]. Différentes techniques de visualisation existent [4], mais c’est sur la projection que nous allons nous concentrer.

Dans une projection, toutes les données sont projetées selon une projection linéaire ou non des axes dans un plan en 2D. Cela permet de représenter les données sous la forme d’un nuage de points (on parle de “scattered representation”) en essayant de respecter au mieux la structure des données [4].

Le principe de base d’une lecture projection est le suivant : les points qui sont proches les uns des autres sont censés être “similaires” tandis que ceux qui sont éloignés sont censés être “différents”. Les projections permettent également d’inférer des propriétés sur les données et de les vérifier. Par exemple, il est possible de vérifier si les clusters sont séparables linéairement (notamment à l’aide d’une projection linéaire) [4] .

D’une façon générale dans la visualisation, le plus compliqué n’est pas tant d’acquérir ou de traiter directement des données, au contraire ces deux étapes sont de moins en moins compliquées et coûteuses. Le défi est celui de la représenter fidèlement des jeux de données qui se font de plus en plus grands; jusqu’à atteindre des tailles allant jusqu’à des dizaines milliers de dimensions. C’est pour cela que pour projeter en 2D ou en 3D un jeu de données de n dimensions, il faut passer par un processus de réduction de dimension. Cependant, ce processus a un coût, étant donné qu’il est lié à une perte d’information causée par l’étirement ou la compression de la plupart des distances entre paires, car la dimension de l’espace couvert par les données d’origine est généralement supérieure à la dimension de l’espace de projection. ce qui cause l’apparition d’artefacts dans la projection. Ces artefacts compromettent ainsi grandement la fiabilité et la bonne interprétation de ces graphiques[5]. Dans une projection, les axes ne jouent pas forcément un rôle : c’est principalement la distance entre les points qui donne un sens et une interprétabilité à la projection[6]. Néanmoins, représenter fidèlement une distance euclidienne après réduction et projection est tout bonnement impossible.

Par conséquent, la mesure et la visualisation de ces distorsions appelées “artefacts” sont cruciales pour l’analyse. Elles se doivent d’être analysées afin de détecter quelles distances entre paires ont été préservées, et donc d’évaluer si les caractéristiques observées dans l’espace de projection sont des images fidèles de certaines caractéristiques de l’espace originel, ou simplement des artefacts de la projection.

Les artefacts sont définis comme des points “mal placés”(en comparaison aux points d’origine). Ils sont dus au processus de réduction qui peine à respecter fidèlement les distances. Il existe deux types d’artefacts : les artefacts géométriques et les artefacts topologiques [1].

- Les artefacts géométriques sont causés par de légères distorsions des distances, dans ce cas précis les distances sont fausses, mais le voisinage des points est conservé, l’interprétation n’est donc pas (ou très peu) perturbée.
- Les artefacts topologiques sont causés par des distorsions trop importantes des distances, c’est-

à-dire que dans ce cas ni les distances ni le voisinage des points ne sont conservés. Cela peut par exemple mener à des situations qui nuisent grandement à la qualité de la visualisation. Il est possible de retrouver deux clusters différents dans la même zone (on parle dans ce cas de faux voisinages), ou bien que des clusters soient séparés en plusieurs parties (on parle dans ce cas d'artefacts de déchirure [4]). Par conséquent, cela engendre forcément des problèmes d'interprétation.

Parmi ces types d'artefacts, nous pouvons distinguer 4 sous-types d'artefacts : [7] :

- Les compressions : Les distances par paire de projections sont inférieures aux distances par paire d'origine correspondante, mais la topologie du voisinage d'origine est préservée.
- Les étirements : Les distances par paires projetées sont plus grandes que les distances par paires d'origine correspondantes, mais la topologie du voisinage d'origine est préservée.
- Les collages : Une compression particulière dans laquelle les points très éloignés dans l'espace d'origine deviennent des voisins proches dans l'espace de projection, changeant ainsi radicalement la topologie du voisinage.
- Les déchirements : Un étirement particulier où des points proches dans l'espace d'origine sont projetés loin les uns des autres, changeant drastiquement la topologie du voisinage.

Il y aurait deux causes principales à l'origine de ces artefacts [7] :

- Les causes structurelles(ou intrinsèques) : Les structures géométriques et topologiques des collecteurs d'origine et celles de l'espace de projection ne sont pas compatibles, de sorte que la projection ne peut se faire sans modifier les distances par paires entre les données.
- Causes techniques(ou extrinsèques) : Si la technique de projection est non linéaire, elle peut créer des artefacts qui n'existent pas à l'optimum global et donc modifier la forme initiale.

Les artefacts ayant des causes techniques peuvent être annulés alors que ceux ayant des causes structurelles ne le peuvent pas. Cependant, dans la pratique, il n'est guère possible de distinguer les deux causes d'artefacts dans le cas des projections non-linéaires.

Pour tenter de pallier aux difficultés induites par les artefacts, des techniques ont été mises en place pour assister l'utilisateur dans la visualisation. Parmi ces techniques, nous pouvons noter l'utilisation d'échelles colorimétriques[8], qui permettent de mesurer les différents artefacts, mais ne permettent pas de mettre en valeur les clusters cachés ni de donner une idée claire ou de préciser de la qualité de la projection. Dans ce cas, il est compliqué d'exploiter une projection de n dimensions de façon fiable sans prendre en compte les artefacts qu'elle présente[7]. Dans l'article "Visualizing Dimensionality Reduction Artifacts : An evaluation"(Heulot) une technique colorimétrique est utilisée pour voir s'il est possible de surmonter les difficultés dues aux artefacts dans d'interprétation d'échelles multidimensionnelles. Ils ont donc mis en place une méthode s'appelant "ProxiViz". Grâce à celle-ci, sur un jeu de test , les informations locales d'un item sont beaucoup mieux représentées. Cette méthode permettrait de détecter plus facilement les outlier et les clusters. Elle pourrait donc, en partie, aider les non "spécialistes des data" à mieux distinguer certaines informations. Même si encore une fois cela est limité par la qualité de l'écran, le contraste et la vision de l'humain qui peut peiner à distinguer les nuances entre les couleurs[1] [9] [10]. L'idée a notamment été reprise par Heulot[4], il se base sur le concept de la "visualisation interactive des proximités" afin de mettre en évidence les artefacts. Cela permet de visualiser sur la projection les proximités d'origines des différents points en fonction d'une référence choisie par l'utilisateur.

Importance du facteur humain

Même si nous allons fortement nous limiter aux facteurs numériques qui interviennent dans la projection, le facteur humain est également à prendre en compte, il est d'autant plus compliqué à

contrôler à cause de sa subjectivité et son irrégularité. En effet, l'humain peut commettre des erreurs de compréhension et d'interprétation à plusieurs étapes de l'analyse. D'une part dans la sélection des métriques il est susceptible de commettre des erreurs, alors que c'est une partie clé de l'analyse [11]. De plus, le rapport "humain-matériel" peut également être limitant dans la visualisation de données. Par exemple, la taille de l'écran et sa résolution jouent grandement dans la lecture d'une visualisation. Si on couple ce facteur à la vision humaine, qui est également limitée, l'apparition de biais dans l'appréhension de certaines représentations telle que les distances ou la nuance de couleur semble inévitable. Cela ajoute donc de l'incertitude et de la variabilité, ce qui finit par rendre l'interprétation finale biaisée.

Ces biais peuvent se manifester dans différents cas d'analyse de données. Un des cas proches de notre sujet est celui de l'analyse de données progressive ou Progressive Visual Analytics(PVA) qui consiste en l'analyse des données intermédiaires. Elle est utilisée dans de nombreux domaines scientifiques comme la biologie. Dans un cas d'une projection elle peut par exemple être appliquée à des méthode l'ACP, ce qui donne la PrACP (Progressiv ACP)[12]. Le fait que l'analyse soit "en cours de traitement" est encore plus susceptible de générer des biais chez ceux qui analysent. C'est pour cela que dans un contexte analyse de visualisation de données il faut mettre l'accent sur la compréhension du facteur humain. Une des approches possibles est de catégoriser les différents types d'utilisateurs des projections et de la PVA. C'est ce qui a été fait par Micallef dans son article "*The Human user In Progressive Visual Analytic*"[13], dans lequel il distingue trois principaux types d'utilisateurs de l'analyse progressive :

- Le "progressiv searcher" : Son but est de trouver rapidement des informations précises pour répondre à une question concrète dans un temps limité. Il va utiliser l'analyse progressive pour déterminer si les données de certaines recherches sont suffisantes et tenter de prédire au mieux les résultats finaux(avant le compte rendu final). Il pourra également, en fonction des résultats qui sont affichés, affiner sa zone de recherche. La PVA lui permet donc d'étudier de plusieurs façons les modèles qu'il étudie : d'une part de mettre en valeur des fluctuations, des singularités (pour pointer certains problèmes) et d'étudier en parallèle plusieurs modèles. D'une autre, d'avoir un aperçu de la qualité des données au cours de l'analyse.
- Le "progressiv observer" : Il n'a pas de connaissance particulière sur la PVA et son mécanisme. Dans son cas, la compréhension de l'algorithme sous-jacent et son fonctionnement n'est pas primordiale, bien que conseillée. Son intérêt va principalement se porter sur l'output, c'est-à-dire qu'il veut juger la qualité et la quantité des données obtenues et raisonner à partir des graphiques. Dans son cas, la PVA permettrait de vulgariser certaines notions et de réduire la complexité visuelle des processus. Pour s'aider, il peut également utiliser des ancrs entre les périodes d'avancement et réduire la vitesse des calculs trop rapides.
- Le "progressiv explorer" : Son but est de comprendre globalement l'analyse des données et les processus sous-jacents. Il est intéressé par la répercussion de ses manipulations et des changements de paramètres sur les résultats, il veut ainsi être capable de simuler différents scénarios. Dans son cas, la PVA lui permet de rendre les paramètres plus ajustables, de lui donner une idée plus claire des outputs dans le but de lui permettre de tester différents scénarios alternatifs, on parle dans ce cas d'"Interactiv Simulation".

Suivant cette idée de distinctions entre types d'utilisateurs, on peut différencier trois types de contextes d'utilisation des projections de données [4] .

- Le contexte "exploratoire" : Ce contexte d'utilisation n'est pas lié à un modèle scientifique en particulier, le but sera d'identifier visuellement les outliers et les clusters . Cela pourra permettre d'une part d'identifier de potentielles anomalies (les outliers) et d'y remédier, et également de classer les clusters. La pratique exploratoire peut se montrer très fastidieuse

en fonction du type de données, la projection peut donc être utilisée comme un outil pour organiser ce travail d’exploration et guider au mieux l’utilisateur.

- Le “contexte confirmatoire” : Ce contexte d’utilisation dépend d’un modèle. En fonction des clusters et de la classification qui s’ensuit, le modèle pourra être validé ou non. Le but est d’étudier la proximité entre les classes et leur potentiel chevauchement avec notamment la détection d’outlier de classe [4] [14].
- Le contexte “présentation des données” : L’objectif est de communiquer des observations d’une analyse de la façon la plus claire et concise possible.

Même si plusieurs profils ont été définis, et qu’il est possible de les utiliser pour adapter au mieux l’interface et les processus, cela n’empêchera pas les erreurs qui peuvent se manifester de plusieurs manières [15]. Par exemple, l’utilisateur peut tout simplement se tromper : il peut trouver et interpréter un résultat qui n’existe pas et/ou mal juger les résultats intermédiaires. Il peut également y avoir des “biais de confirmation”, c’est-à-dire que les utilisateurs confirment leurs hypothèses favorites alors que les résultats sont incomplets[13]. On peut également se retrouver dans des situations dans lesquelles l’utilisateur attend trop longtemps des résultats qui ne sont pas pertinents pour répondre à sa problématique. La réduction dimensionnelle est souvent pensée comme un processus de fond, peu intuitif et qui est géré en totalité par le programme et les algorithmes. Ainsi, plusieurs études ont tenté de rendre celui-ci beaucoup plus intuitif et d’incorporer le facteur humain dans sa réalisation[13][16]. C’est dans ce concept “réduction interactive” que Sara et Jimmy Johansson ont mis en place une méthode dans laquelle les paramètres et les filtres de réductions sont sélectionnés par l’utilisateur lui-même[5]. La démarche d’utilisation est la suivante : l’utilisateur commence par sélectionner les "quality-metric" et le programme va par la suite analyser la qualité de celles-ci et donner un feedback à l’utilisateur concernant le taux de perte d’information. À partir de cela, l’utilisateur pourra choisir le poids des variables ainsi que leur classement. Ensuite à partir de cela le programme se charge de la réduction dimensionnelle, l’utilisateur sélectionne ensuite les variables à garder à supprimer. Par la suite, la réduction finale sera affichée en fonction des paramètres choisis par l’utilisateur. Il pourra également revenir sur sa décision concernant les paramètres et changer le poids des variables ainsi que leur hiérarchie. Ce système a pour objectif de rendre le processus de projection plus interactif. Cela permet à l’utilisateur de cibler ce qu’il recherche et à connaître le taux de perte dans de la représentation qu’il va analyser. Tout cela va rendre le processus de projection plus compréhensible pour l’utilisateur. Néanmoins, en diminuant le poids du facteur machine on augmente le poids du facteur humain, et ce dernier peut se montrer encore plus imprévisible de par sa subjectivité.

Dans cette introduction, nous avons pu voir les enjeux et les problématiques liés à la projection de données.

Dans ce papier, nous allons nous limiter au facteur numérique en tentant de déterminer comment évaluer fidélité des projections de données. Ainsi, dans les prochaines parties nous parlerons de différentes méthodes de projections en expliquant leur principe et leurs avantages, en les illustrant à l’aide du jeu de données Globe[17]. Puis nous aborderons également les différents critères existants qui permettent d’évaluer la qualité d’une projection.

1 Les méthodes de projection

Lorsque l'on souhaite représenter en 2D ou en 3D de grands jeux de données multidimensionnels (au-delà de vingt dimensions [18] [4]), il faut passer par une réduction de dimension. Cette étape intervient dans le processus de projection et est indispensable dans ce cas précis où il y a un nombre important de dimensions (plus d'une vingtaine [4]). Néanmoins, une réduction implique inexorablement une perte d'information et donc une perte de précision, ce qui peut biaiser le résultat de la projection ainsi que son interprétation. Il est donc important, malgré des pertes inévitables, d'essayer de conserver au maximum les structures et les données utiles. Le facteur numérique est donc un facteur clé, qui se doit d'être compris et maîtrisé.

Dans l'espace des données, il est possible d'avoir recours à différents types de mesures (différents critères) pour déterminer à quel point deux individus de la projection sont similaires. Cela dépend grandement de la sémantique sous-jacente à la notion de similarité, celle-ci étant liée au domaine scientifique d'où proviennent les données et à leur nature [4]. Comme mentionné plus haut, la projection se base sur un encodage des similarités par le biais de la variable de position: plus les objets se ressemblent, plus ils seront proches sur la projection, et inversement. Dans une projection, une dissimilarité métrique dans l'espace des données correspond à la définition mathématique d'une distance [19] [4]

$$d : \mathbb{R}^m * \mathbb{R}^m \rightarrow \mathbb{R}$$

, qui pour

$$\forall (u, v, w) \in \mathbb{R}^3 * m$$

satisfasse les conditions suivantes [4] :

- La définition : $d(u, v) = 0 \Leftrightarrow u = v$
- La positivité : $d(u, v) \geq 0$
- La symétrie : $d(u, v) = d(v, u)$
- L'inégalité triangulaire : $d(u, v) \leq d(u, w) + d(w, v)$

La distance de Manhattan et la distance euclidienne peuvent subir la "domination" d'une dimension qui aurait des valeurs réparties sur un spectre plus large que les autres [4]. Pour corriger cela, on normalise ou on pondère les dimensions (il faut cependant faire attention à bien prendre en compte les poids dans l'interprétation de la similarité).

Il est possible de séparer les méthodes de projections en plusieurs catégories. Elles peuvent être non supervisées ou supervisées : c'est-à-dire qu'elles nécessitent l'étiquetage des données et/ou l'intervention de l'utilisateur pour positionner les points sur la projection [20] [4]. Les algorithmes de projections peuvent également être séparés en deux catégories : les algorithmes de projections linéaires et non linéaires. Dans le cas des algorithmes de projection linéaire, l'espace de plus faible dimension (celui de la projection) est obtenu par la combinaison linéaire des dimensions de l'espace de départ.

Dans le cas des algorithmes de projection non linéaires partent du principe qu'il existe une variété non linéaire de plus faibles dimensions que l'on peut projeter sans trop de déformations dans un espace 2D (ou 3D). Les données sont ainsi projetées en conservant les propriétés de leur variété. Pour cela, elles utilisent des mesures de stress métriques (selon les distances) ou Non-Métriques (selon le rang des distances). Ensuite plusieurs méthodes d'optimisation sont utilisées telles que la descente de gradient, les réseaux de neurones [21] ou bien le placement par force (pour les jeux de données les moins gros) [22]. Cela permet soit de trouver un optimum global (qui préserve la structure globale) ou un optimum local (dans le cas où les méthodes préservent les voisinages locaux).

1.1 Analyse en Composantes Principales(ACP)

Ce type de projection linéaire vise à préserver de la façon la plus optimale la variance dans les données en transformant des variables corrélées en variables non corrélées(également appelées "composantes principales") qui expliquent la variabilité dans les données. L'ACP mesure l'inertie des données pour réduire au maximum le nombre de variables. Géométriquement parlant, elle a recours à une projection orthogonale des données dans un sous-espace principal obtenu par combinaison des dimensions dans le jeu de données. Le sous-espace est calculé de sorte que le nuage de points obtenu à partir de la projection maximise la variance des données [4].

Cette méthode est beaucoup utilisée dans de nombreux domaines d'application. Elle est réversible et fournit une pondération des composantes principales (ce qui permet de déterminer les directions de dispersions des données les plus importantes). La projection résultante est directement interprétable, car les vecteurs propres décrivent la contribution de chaque variable [4]. Néanmoins, elle ne permet pas de séparer des relations non linéaires et est plus susceptible d'introduire du faux voisinage entre les points.

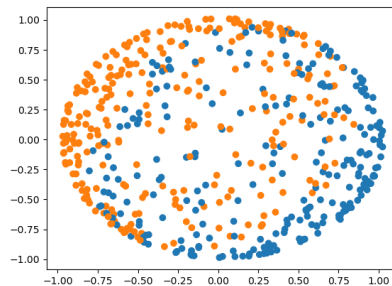


Figure 1: Réduction du dataset Globe

Cette méthode est très performante en termes de temps de calcul et respecte fortement la distance au voisinage de chaque point, cependant elle ne représente pas correctement les classes. La méthode est implémentée dans la librairie scikit-learn.

1.2 Linear Multidimensional Scaling (MDS)

Le principe de fonctionnement de cette méthode linéaire est de projeter les données dans un sous-espace linéaire[23]. L'objectif de cette approche est de conserver de la meilleure façon possible les paires de distances au carré provenant de l'espace des données. Pour y parvenir, l'algorithme cherche pour chaque métrique de l'espace des données, une combinaison linéaire optimale. Globalement cette technique a les mêmes défauts que l'ACP quand il s'agit de projeter des données linéairement séparables. Néanmoins, dû au fait que cette technique prend une matrice de similarité en entrée, les axes de projection de cette technique ne sont pas directement interprétables[4].

Cette méthode est performante en termes de temps de calcul et de mémoire. Dans le cas du jeu de données utilisé (GLOBE), les classes sont préservées, néanmoins une d'entre elles est projetée à l'extérieur.

La méthode est implémentée dans la librairie scikit-learn. <https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html>

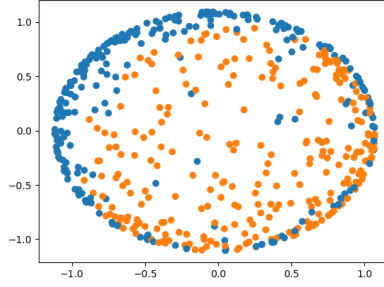


Figure 2: Réduction du dataset Globe

1.3 ClassNerv

Cette méthode définie dans l'article [17] et présentée lors de la prestigieuse conférence NeurIPS de 2020.

C'est une méthode de projection qui permet de réduire la dimension de l'espace de décision, et de réduire la dimension des données en conservant les classes et leur voisinage. Elle présente les avantages des méthodes supervisées et non supervisées, c'est à dire la combinaison du traitement par classe et par distance au voisinage. On peut y faire varier les paramètres

$$\tau^* = \frac{\tau^\infty + \tau^\notin}{2}$$

$$\varepsilon = \frac{\tau^\infty - \tau^\notin}{2}$$

$\tau^* \in [0, 1]$ avec 0 pour un minimum de faux voisin et 1 pour un minimum de voisin raté.

$\varepsilon \in [0, 0.5]$ 0 pour réduire la supervision, et 0.5 pour l'augmenter.

Par exemple sur le Dataset Globe (deux classes séparées en deux hémisphères, représentation d'origine en 3D), on obtient les résultats suivants :

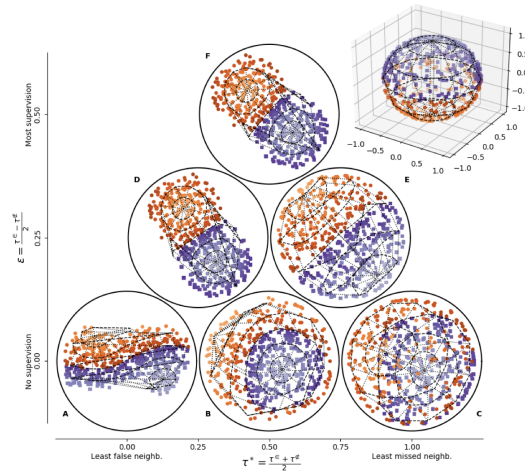


Figure 3: Réduction du dataset Globe

On peut remarquer que l'on obtient un meilleur résultat avec un ε de 0.25 et un τ^* de 0.75, c'est-à-dire une supervision moyenne et en privilégiant les faux voisins plutôt que les voisins manqués.

Implementation

Il est également important de noter que cette méthode est très coûteuse en temps de calcul.

1.4 S-Isomap

Cette méthode est définie dans l'article [24] qui est une surcouche de Isomap(Isomap signifie "cartographie isométrique").

Bien que moins performante que la méthode ClassNerv, elle n'en reste pas moins très utilisée. Isomap est une méthode de réduction de dimensionnalité non linéaire basée sur la théorie spectrale qui essaie de préserver les distances géodésiques dans la dimension inférieure.

Cela se déroule deux phases : Tout d'abord, Isomap commence par créer un réseau de voisinage, puis avec les labels des classes, on peut atténuer le bruit sur les données se basant sur ces quelques propriétés :

1. Les points de la même classe (label) sont plus proches les uns des autres et inversement.
2. La fonction $D(x_i, x_y)$ définie dans l'article permet d'identifier le bruit.
3. Chaque fonction de dissimilarité est croissante monotone par rapport à la distance euclidienne.
4. À l'instar de classNeRV on peut contrôler la similarité interclasse et intra classe avec le paramètre α .

En projetant le dataset Globe par le biais de cette méthode nous obtenons les résultats suivants :

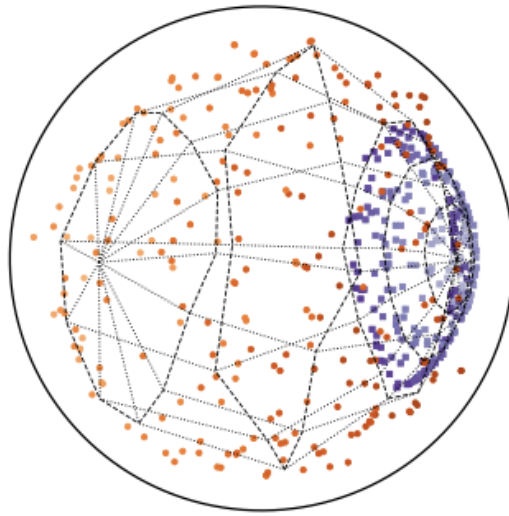


Figure 4: Réduction du dataset Globe

Nous pouvons remarquer que les classes sont préservées mais que la distance au voisinage est moins bien conservée que sur ClassNeRV.

Il est également utile de noter que cette méthode est beaucoup plus performante en termes de temps de calcul.

Ce lien donne une implémentation possible de S-Isomap.

1.5 t-SNE

Cette méthode non linéaire se base sur un algorithme de projection: le SNE [25] celui-ci permet de bonnes visualisations, mais a une fonction de coût qui est difficile à optimiser. C'est notamment de ce souci dont le t-SNE s'affranchit en utilisant une fonction coût "symétrique" et une t-distribution plutôt qu'une distribution Gaussienne. Ces deux changements lui permettent d'enlever les soucis d'optimisation du SNE. Cette méthode est donc plus simple et plus rapide à exécuter. De plus, les maps produites seraient même de meilleure qualité[26]. Avec le t-SNE, il est possible de restituer fidèlement la structure d'un grand jeu de données au niveau local et au niveau global. Cette méthode permet en effet de repérer la présence de clusters à différentes échelles. Il est par ailleurs possible de retrouver les clusters "séparés", et de faire une approximation du spectral clustering avec un certain paramétrage. [27] [28] [29].

La démarche de cette méthode est la suivante : elle commence par calculer la probabilité P que deux instances puissent être voisines. Ainsi, P aura une haute probabilité pour des voisins proches et très faibles pour des voisins éloignés.

La variance de la distribution Gaussienne est différente à chaque fois, ce qui permet de capturer la densité pour différents voisinages multidimensionnels. Le calcul va s'effectuer de façon itérative jusqu'à ce que la "perplexité" (au préalable définie par l'utilisateur) soit atteinte.

Une fois, toutes les probabilités calculées le but est de trouver une distribution de probabilité Q qui représente fidèlement le jeu dans un espace de plus faible dimension. Cette fois-ci, au lieu d'utiliser une distribution Gaussienne, c'est une distribution du t de Student qui est utilisée, avec un degré de liberté de 1.

Contrairement à P , Q n'est pas paramétré avec une variable de densité de voisinage, par conséquent des voisinages avec des densités très différentes dans l'espace original peuvent être "transformés" en voisinages de tailles équivalents dans la représentation à plus faible dimension. La recherche de Q qui représente fidèlement P se fait en optimisant la fonction coût suivant la divergence de Kull-Leibler des deux distributions. Puis, c'est également effectué de façon itérative, en faisant une descente de gradient pour un nombre d'itérations déterminée par l'utilisateur [26].

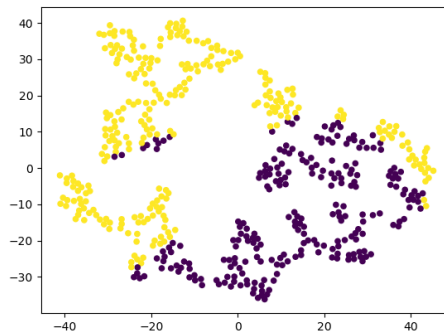


Figure 5: Réduction du dataset Globe

Cette méthode est très performante en temps de calcul, mais cela se fait au détriment des résultats qui sont beaucoup moins précis. Les classes sont relativement bien séparées, mais la distance au voisinage n'est absolument pas respectée.

Celle-ci est implémentée dans des projets github par Laurens Van Der Maaten dans une grande diversité de langages (Matlab, Python/Cuda, R, ...).

1.6 Autres méthodes de projection non linéaires

La liste des méthodes de projections non linéaires ne se limite évidemment pas à celles citées. Nous pouvons également retrouver le Sammon mapping[30], le curvilinear components analysis(CCA)[31], le Maximum Variance Unfolding(MVU)[32] [33], ou le Laplacian Eigenmaps[34]. Ces techniques sont performantes, mais peinent à être efficaces quand il s'agit de visualiser des jeux de données avec beaucoup de dimensions. Il nous était également compliqué de restituer correctement les structures locales ET globales dans une seule projection[35] [26]. C'est en partie pour ces raisons et le manque d'implémentation disponibles qu'elles n'ont pas été retenues pour notre projet.

1.7 Conclusion partie

Comme nous avons pu le voir, il existe de nombreux algorithmes de projections qui se basent tous sur différentes hypothèses mathématiques, différents critères et différents paramétrages. Ainsi, avec un même jeu de données, il est possible d'obtenir plusieurs projections différentes en fonction des algorithmes choisis(le nuage de points ne sera pas le même) . Le choix du type d'algorithme de projection à utiliser dépend de la nature des données que l'on veut étudier, du type de résultat que l'on souhaite observer et de nombreuses caractéristiques propres aux données qui peuvent être très complexes à appréhender sans de bonnes connaissances en projection de données.

Par exemple, nous avons vu pour L'ACP et Isomap, qu'elles ne sont pas stables face aux variations des données. La raison est que l'ajout ou la suppression de quelques instances peut modifier de manière significative les vecteurs et les valeurs propres, affectant ainsi considérablement la cartographie. Même de petites perturbations dans les données peuvent avoir un impact conséquent sur la cartographie finale en raison de l'inversion des vecteurs propres. Ce qui entraîne des configurations très différentes avant et après la perturbation [36]. Le t-SNE quant à lui est basé sur des procédures d'optimisation dont la solution optimale dépend des conditions initiales. En général, des initialisations différentes conduisent à des solutions différentes. La stabilité n'est pas garantie même en fixant la condition initiale, de sorte qu'une modification du nombre d'éléments peut conduire à des agencements sensiblement différents [37]. Quant au MDS, c'est une technique qui repose sur des points de repère ou de contrôle et qui a tendance à être stable tant que les points de contrôle et de repère restent fixes. Comme la position de chaque instance ne dépend que des points de contrôle (points de repère), si ces points ne changent pas, le mappage ne change pas non plus [37].

Nous pouvons donc affirmer qu'aucune méthode n'est parfaite : peu importe la réduction de dimension appliquée une perte de données est inévitable ce qui cause forcément l'apparition d'artefacts. Ce qui soulève la question suivante : Comment s'assurer de la qualité d'une projection ? Pour appréhender cela, nous passerons en revue plusieurs critères et méthodes d'évaluation de la qualité d'une projection dans le chapitre suivant.

2 Les critères d'évaluation des projections

Comme nous l'avons vu dans le chapitre précédent, la projection d'un jeu de données multidimensionnel passe par un processus de réduction de dimension. Or, celui-ci a un coût : il implique une perte de données qui va se répercuter lors de la représentation finale de celles-ci. La réduction induit une erreur : le "stress" qui a un impact sur l'approximation des distances au sein de la représentation.

Par définition, une "bonne projection" est une projection qui représente le plus fidèlement possible la structure sous-jacente du jeu de données. Par conséquent, une bonne configuration de projection est une configuration qui minimise le stress de la projection. En d'autres termes: plus les distances entre le jeu de données et la projection seront proches, moins il y aura d'erreurs. Ainsi, le but dans une projection est de minimiser le stress.

Mathématiquement parlant, le stress est défini comme la somme quadratique des écarts de distances :

$$\epsilon_{\psi} = \sum_{i,j}^n [(d_m(x_i, x_j) - d_p(y_i, y_j))]^2$$

2.1 Visualisation du stress

Le stress va produire des artefacts lors de la projection de données. Ceux-ci vont avoir un impact sur l'approximation et l'interprétation de la représentation graphique. Il y a plusieurs façons de le quantifier :

Les mesures peuvent être locales, c'est-à-dire qu'elles sont visualisées en chaque point et aident à expliquer la projection. C'est possible en utilisant un diagramme de Shepard [38] [4]. D'autres méthodes utilisent un "jitter disc" autour de chaque point. Cela permet de visualiser le stress, mais ne donne pas des informations sur les origines de celui-ci [1].

Il est également possible de le visualiser en utilisant un encodage colorimétrique, qui en fonction d'une couleur et de son intensité indiquera les points les moins fiables de la projection [8]. On peut aussi avoir recours à différents types d'interpolation de la couleur entre les points, selon différents paramètres, ce qui forme une carte de précision [39].

Une méthode pour mettre en avant les artefacts topologiques (faux voisinages et déchirures) a été mise en place par Lespinats et Aupetit [8] : l'idée est de visualiser deux mesures de stress simultanément à l'aide d'une échelle de couleur 2D uniforme. Cela permet d'utiliser la projection malgré les erreurs de positionnement des différents points: cependant la mesure de qualité ne garantit pas qu'on puisse exploiter la projection (surtout pour les tâches de clustering visuel) .

Il y a également, des méthodes interactives avec différentes vues possibles (stress) globaux, faux voisinages, déchirures ... selon les différents niveaux de granularité (point seul ou groupes de points) [40] [4]. De plus, en fonction du type d'artefacts, différentes mesures d'erreurs sont proposées : coloration interpolée ou edge bundling (arc compact) pour, par exemple relier les points et faire le parallèle pour montrer les déchirures sur la projection [41] [42] [4]. Les arcs sont aussi colorés de façon différente en fonction de l'intensité des erreurs.

Ce système est interactif, car l'utilisateur peut choisir manuellement un groupe de point à étudier afin de visualiser seulement les artefacts qui lui sont relatifs. Il y a aussi beaucoup de vues statiques qui peuvent l'aider à filtrer.

2.2 Les critères de mesure

Pour évaluer la qualité d'une projection, plusieurs approches se basant sur des critères existent. Ces critères peuvent être relatifs à la structure des données projetées avec le taux d'oublier de la projection (outlying), le taux de cluster et à leurs qualités (visual clustering, class consistency, cluster separator, class density), et à la forme de la projection (clumpy, skinny). Nous allons les détailler au cours de ce chapitre.

Clustering

Le clustering d'une projection sert à partitionner les données en groupe homogènes. En d'autres termes, cela consiste à évaluer si la projection contient des "groupes"(ou clusters) ainsi que leur qualité (c'est-à-dire s' ils sont séparés et facilement identifiables). Pour cela, il existe plusieurs critères.

- **Class consistency :** Tout d'abord, il y a la mesure de la "class consistency"[43]. Dans un jeu de données, chaque classe est composée d'un sous-ensemble du data-space qui lui est assigné, ainsi chaque point est assigné à une classe. L'ensemble des classes du data-space est appelé "class structure". L'objectif est donc de vérifier si la séparation des classes est correcte en fonction des dimensions étudiées, sachant qu'une bonne représentation visuelle d'une "Class structure" doit être fidèle à cette même structure de classe. Pour déterminer cela, nous pouvons le calculer selon la règle de la "distance to centroid"(distance consistency) : c'est-à-dire que chaque point d'une classe doit être à une distance inférieure de son centroïde(i.e le centre de sa classe) qu'à celui d'une autre classe. Or c'est une des propriétés qui est très souvent perdue lors de la projection. Par conséquent, si la "class consistency" est bonne, les classes seront situées à des régions visuellement séparées dans la projection graphique, les clusters seront donc plus facilement discriminables et la projection potentiellement de meilleure qualité.
- **Class density :** Pour justifier de la qualité du clustering d'une projection, nous pouvons également mesurer la densité de classe (ou "class density"). Le principe est de retenir les graphes avec le moins de chevauchement [44]. Ainsi, lors d'une projection, les points appartenant à un cluster forment une image, c'est -à -dire que chaque classe forme une image. L'algorithme se fait en fonction de la densité basée sur le voisinage : pour chaque pixel, la distance de son voisin le plus proche de la même classe est enregistrée. Puis, la densité locale est calculée dans une sphère de rayon de la "distance maximum possible". Le chevauchement global des classes est ensuite estimé en calculant la somme de la différence de chaque paire de pixels à la valeur absolue). Puis la visualisation avec le chevauchement le plus faible sera retenue.

$$CDM = \sum_{k=1}^{m-1} \sum_{l=k+1}^m \sum_{i=1}^P \|P_k^i - P_l^i\|$$

- **Histogram density :** Il y a aussi la mesure de l'"histogram density" [4] [45] qui est une mesure de l'entropie de la projection (i.e l'information moyenne de différentes portions de la projection découpée en grille). La projection est séparée en plusieurs "bin" (à l'image des histogrammes). Dans chacun de ces bins on comptabilise le nombre de points et leur étiquette de classe. L'entropie de chaque bin se calcule de cette façon.

$$H(p) = - \sum_c * \frac{P_c}{\sum_c * P_c} * \log_2 * \frac{P_c}{\sum_c * P_c}$$

Ainsi, si tous les points d'un bin sont de la même classe, alors l'entropie est égale à zéro. Par conséquent le clustering sera de bonne qualité.

Techniques de clustering

Deux des techniques de clustering les plus efficaces sont **Clique** et **DBscan** [40]. Leurs fonctionnements ne sont pas similaires et c'est pourquoi nous allons les développer ici.

Clique

Il s'agit d'un algorithme de clustering de sous-espaces qui est "density-based" et "grid-based" .

"Grid - based", car il traduit l'espace de données sous la forme d'une grille et estime la densité en comptant le nombre de points dans une cellule de grille.

"Density-based", car il définit un cluster comme étant un ensemble maximal d'unités denses connectées dans un sous-espace.

Un regroupement de sous-espaces est un ensemble de cellules denses voisines dans un sous-espace arbitraire.

Dans un grand jeu de données, l'algorithme Clique identifie automatiquement les sous-espaces qui permettent un meilleur regroupement de l'espace d'origine.

Il se déroule en plusieurs étapes. Premièrement, il identifie les sous-espaces qui contiennent des clusters :

pour ce faire, il répartit l'espace de données et trouve le nombre de points qui se trouvent à l'intérieur de chaque cellule de la répartition. Ensuite, il identifie les sous-espaces qui contiennent des clusters par raisonnement *a priori* : si une collection de points S est un cluster dans un espace à k dimensions, alors S fait également partie d'un cluster dans toute projection à $(k - 1)$ dimensions de cet espace).

Ensuite, il identifie les clusters :

Il va déterminer les unités denses dans tous les sous-espaces d'intérêt. Par la suite, il va déterminer les unités denses connectées dans tous les sous-espaces d'intérêt.

Enfin, il génère des descriptions minimales pour les clusters:

Pour cela, il détermine les régions maximales qui couvrent un cluster d'unités denses connectées pour chaque cluster, et détermine la couverture minimale pour chaque cluster.

Cette technique est insensible à l'ordre des enregistrements en entrée et ne présume pas d'une distribution canonique des données. De plus, il s'échelonne linéairement avec la taille de l'entrée et possède une bonne adaptabilité lorsque le nombre de dimensions dans les données augmente. C'est également une méthode simple avec des résultats faciles à interpréter[46].

Néanmoins, comme dans toutes les approches de clustering "grid bases", la qualité des résultats dépend du choix du nombre et de la largeur des cellules de la grille [46]

DBscan

Pour déterminer les clusters il est possible d'utiliser le DbScan(Density Based Spatial Clustering of Applications with noise), un algorithme non supervisé et non paramétriques [40] . Il recherche les voisins par densité(Minimum Points) dans une sphère de n -dimension et de rayon ϵ . L'idée principale est que pour qu'un point appartienne à un cluster il doit se trouver en présence d'autre point appartenant à ce même cluster. Spécifier le nombre cluster au préalable n'est pas obligatoire. Il fonctionne aussi bien avec les clusters de taille arbitraire. Par ailleurs, DBscan est robuste et détecte les valeurs aberrantes, il peut donc être aussi bien utilisé pour détecter les clusters que les outliers dans des distributions [47];

Cette méthode sépare les points d'une projection en 3 catégories :

- Les "cores points", qui sont des points dont le nombre de voisins est supérieur ou égal au paramètre "Minimum Point" qui est défini par l'utilisateur. Et qui correspond au nombre de voisins minimum qu'un point doit avoir pour être considéré comme "Core point".
- Les "Border point", qui sont des points dont le nombre de voisins est inférieur à "Minimum point" mais sont toujours "density reachable" par les points à l'intérieur du cluster. C'est-à-dire qu'il existe un ensemble de "core point" reliant deux points entre eux. .
- Les "outliers", qui sont des points qui ne sont dans aucun cluster : Ceux qui ne sont pas density reachable par les points dans le cluster.

Pour les catégoriser , il fonctionne suivant ce principe : Un point Q est atteignable, en partant du point P s'il y a un chemin $p_1 \dots p_n$ tel que $p_1 = P$ et $p_n = Q$. Dans lequel p_{i+1} est directement atteignable en partant de p_i . Ainsi, tous les points du chemin peuvent être des "core points" avec

comme seule exception de Q . Deux points ($petq$) sont “density connected” s’il y a un point O qui est “density reachable” par p et q (il y a une symétrie). L’algorithme se déroule en 3 phases : Dans un premier temps, il détermine ϵ et $minPts$. Par la suite, il choisit aléatoirement un point de départ et détermine son voisinage à l’aide du rayon (ϵ). Pour finir, il choisit à nouveau aléatoirement un point parmi les points qui n’ont pas été visités lors des étapes précédentes et applique la même procédure en boucle. Quand tous les points sont visités, le processus prend fin.

Il est possible d’utiliser dbScan avec Sci-Kit Learn[48]. Il y a deux paramètres importants dans l’implémentation :

- Les unités spatiales : si l’espace est de 2 ou 3 dimensions la distance euclidienne sera préférée, dans le cas d’un espace à n -dimension (avec $n > 3$) on préférera utiliser la distance de Manhattan.
- Epsilon (ϵ) : qui correspond à la distance spécifiant les voisinages. Lorsque la distance qui sépare deux points est inférieure ou égale à ϵ , ces deux points sont considérés comme voisins. Il faut prêter une attention toute particulière à ce paramètre, car si sa valeur est trop haute il y aura trop peu de clusters, à l’inverse, si elle est trop basse il y aura trop de clusters sans sens. En général on la fixe entre 0.25 et 0.75

Enfin, il y a d’autres approches qui visent à mettre le clustering en évidence telle que la Local Affine MDS (LAMP) [49] ou l’ACP sous contraintes. L’objectif est d’adapter la projection en fonction de contraintes de positions choisies par l’utilisateur ou définie sur la matrice de similarité. Ces approches ne permettent toujours pas de mettre de côté les artefacts et ne sont pas forcément optimales dans le contrôle de la position.

Outlying

Le taux d’outliers est important à définir pour justifier de la qualité d’une projection. Un “outlier” est une donnée aberrante. Pour déterminer si un point est un outlier, il y a plusieurs méthodes.

Critères utilisant les graphes :

- Une des façons de raisonner en graphe et d’utiliser le concept de Minimum Spanning Tree(MST) [50], c’est-à-dire l’arbre le moins long que l’on peut créer à partir du nuage de points de la projection. En prenant comme référentiel le MST, les outliers sont soit des points qui sont localisés aux extrémités de l’arbre ou en intérieur dans des régions relativement vides. Ainsi, on peut les définir comme des nœuds de degrés 1, et dont la somme des poids des arêtes adjacentes est supérieure à ‘w’ selon la formule suivante

$$\omega = q_{75} + 1.5(q_{75} - q_{25})$$

. Puis, nous pouvons calculer le taux d’outliers dans la projection (outlying) en divisant la longueur totale de tous les outliers par la longueur totale de l’arbre selon la formule suivante :

$$c_{outlying} = length(T_{outliers})/length(T)$$

- Il y a également la méthode de “Isolation Forest”. Dans celle-ci, l’algorithme choisit une feature du jeu donnée et une valeur “split” comprise entre le maximum et le minimum des valeurs. Cette étape est effectuée pour toutes les observations, par la suite la moyenne des tous les arbres est faite pour construire la forêt. L’entraînement de l’algorithme consiste à comparer les observations avec la splitting value dans un nœud. Ce nœud comprendra également deux

“sous-noeuds” dans lesquels on fera aussi la même comparaison. Ainsi le nombre de “splitting” est égal à la longueur du chemin. Toutes les comparaisons auront un score allant de 0 à 1. 0 signifiant "normalité", et 1 signifiant qu'il y a un grand nombre d'outlier[51]. Elle est implémentable en python avec scikit Learn.

Puis, nous pouvons calculer le taux d'outliers dans la projection (outlying) en divisant la longueur totale de tous les outliers par la longueur totale de l'arbre selon la formule suivante.

$$c_{outlying} = length(T_{outliers})/length(T)$$

Le Local Outlier Factor(LOF): Pour ce critère, on considère un outlier selon son voisinage local[52]. Le LRD(Local Reachability Density) de chaque point est comparé avec le LRD de ses voisins. Puis on calcule le ratio de la moyenne des LRD des voisins du point sur le LRD du point lui-même. Si le Local Outlier Factor est supérieur à 1 alors le point est un outlier. Cette mesure est très utile pour détecter des outliers dans les clusters extrêmement denses. Néanmoins cette valeur étant un ratio et non un seuil, elle est parfois compliquée à interpréter en fonction de la problématique de recherche.

Critères de formes

Les informations sur la forme de la projection sont également très importantes.[50] La forme est souvent définie par le critère “skinny”. Ce critère représente grossièrement la finesse de la courbe. Celui-ci est défini en calculant le ratio air/périmètre d'une projection (avec normalisation)

$$C_{skinny} = 1 - \sqrt{4\pi area(A)/perimeter(A)}$$

Dans le cas où le chemin le plus court d'un Minimum Spanning Tree est quasiment aussi long que le somme des arêtes de l'arbre d'origine, la courbe peut être qualifiée de “stringy”

$$C_{stringy} = diameter(T)/length(T)$$

Critère de densité

Dans une configuration éparpillée (scattered) il y a également le critère de densité qui importe. C'est la distribution des arêtes du Minimum Spanning Tree qui va donner des informations sur la densité relative de points. Le calcul de cette densité se base sur la mesure des quantiles de la longueur des arêtes :

$$c_{skew} = (q_{90} - q_{50})/(q_{90} - q_{10})$$

Ainsi, si la dimension d'un Minimum Spanning Tree est extrêmement asymétrique, les clusters y seront mal représentés. Dans ce cas, elle est qualifiée de "clumpy". [50]

Score de Projection (*seulement pour ACP*)

Les informations d'une projection par ACP sont mesurées en comparant la variance totale du data set et celle qui a été retenue après la réduction de dimension. Ce qui la rend très dépendante du nombre de dimensions du jeu de donnée de base.

Ainsi, contrairement aux petits jeux de données, il est quasiment impossible que les trois composantes principales d'un grand jeu de données retiennent 100 pourcents(ou quasiment 100 pourcents) de la variance. Et ce, même si le jeu de données est composé de variables non aléatoires avec des structures très facilement interprétables. Le but de cette méthode est de quantifier l'informativité de la projection non pas par la variance finale, MAIS par l'excès de variance de ce qui est attendu après réduction d'un dataset aléatoire. Pour calculer le "projection score": on

calcule la variance totale qui est retenue par les 3 principaux composants. Ensuite, on estime la valeur de ces mêmes entités, mais pour un set de variable complètement aléatoire.

Enfin, on calcule le score de projection en faisant différence entre la racine carrée de la quantité observée et de la racine carrée de la valeur attendue (celle du random dataset). Ainsi, si le résultat est grand(s'éloigne positivement de zéro) cela veut dire qu'il y a plus d'informations (relatives au calcul de la variance) pour le vrai jeu de données que pour celui du random data set. Par conséquent, la projection est plus susceptible de proposer des structures fidèles et intéressantes, qui ne sont pas dues au hasard[53].

2.3 Conclusion de la partie :

Ces mesures de qualités sont donc utiles pour les personnes non spécialistes, ce qui leur permet d'appréhender au mieux la visualisation obtenue après projection. Ils peuvent également être utilisés de façon automatique dans des algorithmes qui permettent de trier et de déterminer si la projection qui va être affichée est fidèle au jeu de données[54].

Comme nous l'avons brièvement abordé dans l'introduction, les facteurs numériques à eux seuls ne suffisent pas à l'interprétation d'une projection : il faut rajouter le facteur "humain" à l'équation. Malgré l'utilité de toutes ces mesures, le jugement humain reste toujours le plus important[44]. De plus, c'est également l'être humain qui a pour rôle d'interpréter la projection. C'est pour cela qu'il existe des évaluations à faire passer aux humains, qui servent à mesurer l'impact des techniques de projection, la qualité des graphiques et leur compréhension par l'utilisateur. Par exemple ,nous pouvons tester directement les utilisateurs avec plusieurs types de tâches telles que le "data outlier validation", le "clustering validation", le "cluster énumération", et le "class outliers validation" [1]. Ces tests consistent en la détection des structures et des données aberrantes dans les projections. Ainsi si une projection a une bonne qualité (suivant les critères numériques) et permet aux utilisateurs de discriminer rapidement et correctement les structures,elle tend alors vers ce que l'on peut considérer comme étant une "bonne projection".

Conclusion

Dans cet état de l'art, nous avons parlé cette fameuse "curse of dimensionality" appliquée au domaine de la visualisation des grands jeux de données, et plus précisément dans le cas d'une visualisation par projection. Les problèmes de ce genre de projection ne sont dus au fait même de projeter les données, mais plutôt à l'incapacité de les projeter "fidèlement". C'est-à-dire, la difficulté à respecter le rapport des distances des points entre le jeu de données originel (à n -dimensions) et la projection finale (qui a réduit ce dernier en 2 ou 3 dimensions pour rendre cela compréhensible par l'Homme). En effet, plus le nombre de dimensions augmente plus il devient compliqué de respecter les distances après réduction.

Le choix d'une méthode de projection par rapport à une autre dépend d'autant plus de la sémantique et de la structure des données. Ce choix dépendra aussi de ce que l'on souhaite observer : voulons-nous mettre en valeur les classes ? Voulons-nous détecter des valeurs aberrantes ? En fonction de cela, le choix de la méthode et de son paramétrage peut grandement différer. Une fois le jeu de données projeté, il faut s'assurer de la qualité de cette projection. Pour cela nous disposons de nombreux critères qui peuvent donner un indice sur le clustering, outlying et la densité de la projection. Certains critères sont même propres à certaines méthodes (comme le projection score pour l'ACP). Encore une fois, en fonction de ce que nous voulons observer certains critères seront plus utiles que d'autres. De plus, ces critères ne font pas tout : d'une certaine façon, une projection se doit d'être à la fois la plus compréhensible par l'Homme et la plus fidèle possible au jeu de données qu'elle représente.

C'est dans ce but, que dans la partie "pratique" de cette UE, nous souhaitons mettre en place une évaluation des différentes méthodes de projections citées. Pour faire cela, nous envisageons de procéder de la façon suivante :

Au vu de nos nombreuses lectures et des résultats sur notre jeu de données tests (Globe), nous avons déduit que la méthode ClassNeRV est la plus performante et donne les résultats les plus satisfaisants. Par conséquent, nous souhaitons effectuer un étalonnage des différentes autres méthodes de projection, en l'utilisant en tant que référentiel.

Notre idée est donc la suivante : lorsqu'un jeu de données sera projeté avec une autre méthode de projection, le résultat obtenu sera comparé à la projection de ce même jeu de données obtenue via ClassNeRV. Puis, en fonction des divergences entre ces deux jeux de données projetés, certains biais seront mesurés et affichés. De plus dans l'objectif d'être le plus précis possible, nous envisageons également de mesurer les différents critères de la projection (clustering, outlying, forme...) et de croiser les résultats obtenus à ceux de la comparaison des deux projections. Nous pensons aussi qu'il serait judicieux de mettre en place un système de "scoring", qui permettrait de rendre plus intuitive l'appréhension des résultats relatifs à la qualité de la visualisation.

Le but de cette démarche est de proposer un outil capable de comparer sa méthode de projection à d'autres via un tableau de scoring pour se rendre compte de la pertinence et de la fidélité des diverses méthodes de projection.

Bibliography

- [1] N. Heulot, J.-D. Fekete, and M. Aupetit, “Visualizing dimensionality reduction artifacts: An evaluation,” *arXiv preprint arXiv:1705.05283*, 2017.
- [2] M. Card, *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.
- [3] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon, “Visual analytics: Definition, process, and challenges,” in *Information visualization*. Springer, 2008, pp. 154–175.
- [4] N. Heulot, “Etude des projections de données comme support interactif de l’analyse visuelle de la structure de données de grande dimension,” Ph.D. dissertation, Université Paris-Sud, 2014.
- [5] S. Johansson and J. Johansson, “Interactive dimensionality reduction through user-defined combinations of quality metrics,” *IEEE transactions on visualization and computer graphics*, vol. 15, no. 6, pp. 993–1000, 2009.
- [6] K. Koffka, “Principle of gestalt psychology, translated by w. li,” 1997.
- [7] M. Aupetit, “Visualizing distortions and recovering topology in continuous projection techniques,” *Neurocomputing*, vol. 70, no. 7-9, pp. 1304–1330, 2007.
- [8] S. Lespinats and M. Aupetit, “Checkviz: Sanity check and topological clues for linear and non-linear mappings,” in *Computer Graphics Forum*, vol. 30, no. 1. Wiley Online Library, 2011, pp. 113–125.
- [9] P. V. Tran and T. X. Le, “Approaching human vision perception to designing visual graph in data visualization,” *Concurrency and Computation: Practice and Experience*, vol. 33, no. 2, p. e5722, 2021.
- [10] M. F. Deering, “The limits of human vision,” in *2nd International Immersive Projection Technology Workshop*, vol. 2, 1998, p. 1.
- [11] K. A. Barchard and L. A. Pace, “Preventing human error: The impact of data entry methods on data accuracy and statistical results,” *Computers in Human Behavior*, vol. 27, no. 5, pp. 1834–1839, 2011.
- [12] J. Liu, S. Chen, and Z.-H. Zhou, “Progressive principal component analysis,” in *International Symposium on Neural Networks*. Springer, 2004, pp. 768–773.
- [13] L. Micallef, H.-J. Schulz, M. Angelini, M. Aupetit, R. Chang, J. Kohlhammer, A. Perer, and G. Santucci, “The human user in progressive visual analytics,” in *Eurovis (short papers)*, 2019, pp. 19–23.
- [14] D. HETLEROVIĆ, “Detekce anomálií a automatické strojové učení,” Ph.D. dissertation, Masarykova univerzita, Fakulta informatiky, 2018.

- [15] J. Baron, J. Beattie, and J. C. Hershey, "Heuristics and biases in diagnostic reasoning: Ii. congruence, information, and certainty," *Organizational Behavior and Human Decision Processes*, vol. 42, no. 1, pp. 88–110, 1988.
- [16] M. J. Mahoney, "Publication prejudices: An experimental study of confirmatory bias in the peer review system," *Cognitive therapy and research*, vol. 1, no. 2, pp. 161–175, 1977.
- [17] B. Colange, J. Peltonen, M. Aupetit, D. Dutykh, and S. Lespinats, "Steering distortions to preserve classes and neighbors in supervised dimensionality reduction," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 13 214–13 225. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/file/99607461cdb9c26e2bd5f31b12dcf27a-Paper.pdf>
- [18] G. Dzemyda, V. Medvedev, A. Lupeikiene, O. Kurasova, and A. Caplinskas, "Big multidimensional datasets visualization using neural networks—efficient decision support," *Complex Systems Informatics and Modeling Quarterly*, no. 6, pp. 1–11, 2016.
- [19] R. L. Somorjai, B. Dolenko, A. Nikulin, W. Roberson, and N. Thiessen, "Class proximity measures—dissimilarity-based classification and display of high-dimensional data," *Journal of biomedical informatics*, vol. 44, no. 5, pp. 775–788, 2011.
- [20] A.-H. Karimi, A. Wong, and A. Ghodsi, "Srp: Efficient class-aware embedding learning for large-scale data via supervised random projections," *arXiv preprint arXiv:1811.03166*, 2018.
- [21] J.-N. Hwang, H. Li, M. Maechler, R. D. Martin, and J. Schimert, "A comparison of projection pursuit and neural network regression modeling," in *NIPS*, 1991, pp. 1159–1166.
- [22] T. Dickhaus, "Simultaneous statistical inference," in *With applications in the life sciences*. Springer, 2014.
- [23] H. Abdi, "Metric multidimensional scaling (mds): analyzing distance matrices," *Encyclopedia of measurement and statistics*, pp. 1–13, 2007.
- [24] X. Geng, D.-C. Zhan, and Z.-H. Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 35, pp. 1098–107, 01 2006.
- [25] G. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *NIPS*, vol. 15. Citeseer, 2002, pp. 833–840.
- [26] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [27] G. C. Linderman and S. Steinerberger, "Clustering with t-sne, provably," *SIAM Journal on Mathematics of Data Science*, vol. 1, no. 2, pp. 313–332, 2019.
- [28] C. Chui and D. Donoho, "Special issue on diffusion maps and wavelets," pp. 1–2, 2006.
- [29] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [30] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on computers*, vol. 100, no. 5, pp. 401–409, 1969.

How to evaluate the faithfulness of visual data projection ?

- [31] P. Demartines and J. Hérault, “Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets,” *IEEE Transactions on neural networks*, vol. 8, no. 1, pp. 148–154, 1997.
- [32] K. Q. Weinberger and L. K. Saul, “An introduction to nonlinear dimensionality reduction by maximum variance unfolding,” in *AAAI*, vol. 6, 2006, pp. 1683–1686.
- [33] L. Song, A. J. Smola, K. M. Borgwardt, A. Gretton *et al.*, “Colored maximum variance unfolding,” in *Nips*. Citeseer, 2007, pp. 1385–1392.
- [34] L. K. Saul and S. T. Roweis, “An introduction to locally linear embedding,” *unpublished*. Available at: <http://www.cs.toronto.edu/~roweis/lle/publications.html>, 2000.
- [35] M. Belkin and P. Niyogi, “Semi-supervised learning on manifolds,” *Machine Learning Journal*, vol. 1, 2002.
- [36] L. G. Nonato and M. Aupetit, “Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 8, pp. 2650–2673, 2018.
- [37] F. J. García Fernández, M. Verleysen, J. A. Lee, I. Díaz Blanco *et al.*, “Stability comparison of dimensionality reduction techniques attending to data and parameter variations,” in *Eurographics Conference on Visualization (EuroVis)(2013)*. The Eurographics Association, 2013.
- [38] J. B. Kruskal, “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis,” *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.
- [39] T. Schreck, T. Von Landesberger, and S. Bremm, “Techniques for precision-based visual analysis of projected data,” *Information Visualization*, vol. 9, no. 3, pp. 181–193, 2010.
- [40] M. M. Abbas, M. Aupetit, M. Sedlmair, and H. Bensmail, “Clustme: A visual quality measure for ranking monochrome scatterplots based on cluster patterns,” in *Computer Graphics Forum*, vol. 38, no. 3. Wiley Online Library, 2019, pp. 225–236.
- [41] D. Holten, “Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data,” *IEEE Transactions on visualization and computer graphics*, vol. 12, no. 5, pp. 741–748, 2006.
- [42] D. Holten, P. Isenberg, J. J. Van Wijk, and J.-D. Fekete, “An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs,” in *2011 IEEE Pacific Visualization Symposium*. IEEE, 2011, pp. 195–202.
- [43] E. Bertini, A. Tatu, and D. Keim, “Quality metrics in high-dimensional data visualization: An overview and systematization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2203–2212, 2011.
- [44] A. Tatu, P. Bak, E. Bertini, D. Keim, and J. Schneidewind, “Visual quality metrics and human perception: an initial study on 2d projections of large multidimensional data,” in *Proceedings of the International Conference on Advanced Visual Interfaces*, 2010, pp. 49–56.
- [45] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnork, and D. Keim, “Combining automated analysis and visualization techniques for effective exploration of high-dimensional data,” in *2009 IEEE Symposium on Visual Analytics Science and Technology*. IEEE, 2009, pp. 59–66.
- [46] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, “Automatic subspace clustering of high dimensional data for data mining applications,” in *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, 1998, pp. 94–105.

How to evaluate the faithfulness of visual data projection ?

- [47] V. Mallawaarachchi, “How to use dbscan effectively,” *towardsdatascience.com*. [Online]. Available: <https://towardsdatascience.com/how-to-use-dbscan-effectively-ed212c02e62>
- [48] ———, “How to use dbscan effectively,” *towardsdatascience.com*. [Online]. Available: <https://donernesto.github.io/blog/outlier-detection-with-dbscan/>
- [49] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato, “Local affine multi-dimensional projection,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2563–2571, 2011.
- [50] L. Wilkinson, A. Anand, and R. Grossman, “Graph-theoretic scagnostics,” in *Information Visualization, IEEE Symposium on*. IEEE Computer Society, 2005, pp. 21–21.
- [51] M. U. Togbe, M. Barry, A. Boly, Y. Chabchoub, R. Chiky, J. Montiel, and V.-T. Tran, “Anomaly detection for data streams based on isolation forest using scikit-multiflow,” in *International Conference on Computational Science and Its Applications*. Springer, 2020, pp. 15–30.
- [52] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: identifying density-based local outliers,” in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.
- [53] M. Fontes and C. Soneson, “The projection score-an evaluation criterion for variable subset selection in pca visualization,” *BMC bioinformatics*, vol. 12, no. 1, pp. 1–17, 2011.
- [54] E. Bertini, A. Tatu, and D. Keim, “Quality metrics in high-dimensional data visualization: An overview and systematization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2203–2212, 2011.