# Journées mathématiques X-UPS

## Année 2024

## Anayse topologique de données

Mathieu CARRIÈRE

**Application de la persistance en apprentissage automatique**

*Journées mathématiques X-UPS* (2024), p. 79-97.
https://doi.org/10.5802/xups.2024-06



Les Éditions de l'École polytechnique
Route de Saclay
F-91128 PALAISEAU CEDEX
https://www.editions.polytechnique.fr

Centre de mathématiques Laurent Schwartz
CMLS, École polytechnique, CNRS,
Institut polytechnique de Paris
F-91128 PALAISEAU CEDEX
https://portail.polytechnique.edu/cmls/

*Publication membre du*
*Centre Mersenne pour l'édition scientifique ouverte*
www.centre-mersenne.org

## 6. Applications en apprentissage automatique

Dans ce chapitre, nous allons étudier les applications de la théorie de la persistance en apprentissage automatique supervisé. Nous avons déjà vu au chapitre 4 que le théorème de stabilité pouvait directement permettre l'utilisation des diagrammes de persistance en inférence géométrique et statistique. Nous allons maintenant aborder les problématiques qui apparaissent en analyse de données lorsque l'on désire utiliser les diagrammes de persistance pour la construction de modèles prédictifs. Nous commencerons donc par rappeler quelques généralités de l'apprentissage automatique pour les modèles prédictifs en section 6.1. Nous étudierons ensuite quelques représentations classiques de l'analyse topologique de données en section 6.2, et nous présenterons un noyau Gaussien dédié aux diagrammes de persistance en section 6.3, dont nous prouverons quelques propriétés métriques.

***Remarque 6.1.*** — *Comme pour le chapitre 4, nous nous restreignons dans ce chapitre à $T = \mathbb{R}$, $\mathbb{K} = \mathbb{Z}/2\mathbb{Z}$, et aux diagrammes de persistance de* Dgm*, c'est-à-dire de cardinaux finis et dont les coordonnées des points sont finies.*

**6.1. Bases de l'apprentissage automatique supervisé.** — Un *modèle prédictif* est une fonction paramétrée $\hat{f}_\theta : \mathbb{R}^d \to Y$, $\theta \in \mathbb{R}^N$, où $N \in \mathbb{N}^*$ désigne le nombre de paramètres du modèle, et $Y$ est l'espace des prédictions. Parmi les exemples les plus standards, on peut considérer le cas de la *régression*, dans lequel on cherche à prédire les valeurs d'une fonction (inconnue) $f$ à valeurs réelles, et celui de la *classification*, dans lequel $Y = \{l_1, \dots, l_C\}$ contient $C$ catégories non nécessairement comparables entre elles (comme par exemple $Y = \{\text{homme, femme}\}$). Dans un souci de simplification, dans la suite de ce chapitre, nous identifierons les catégories avec leurs indices (arbitraires) $\{1, \dots, C\}$.

Pour construire de tels modèles, le paradigme de *l'apprentissage automatique supervisé* consiste à chercher les modèles qui s'approchent le plus des bonnes prédictions sur des données $\mathcal{D}_n = \{(x_1, y_1), \dots, (x_n, y_n)\} \subset \mathbb{R}^d \times Y$ qui ont déjà été observées, appelées *données d'entraînement*. Formellement, ceci revient à résoudre le problème d'optimisation suivant :

$$\theta^\star = \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L(\theta, x_i, y_i), \tag{15}$$

où $\Theta \subseteq \mathbb{R}^N$ constitue l'ensemble des paramètres, et $L : \Theta \times X \times Y \to \mathbb{R}$ s'appelle la *fonction de perte*.

*Modèles linéaires.* — Un *modèle linéaire* est défini comme une fonction de la forme $\hat{f}_\theta(x) = \langle \theta, \tilde{x} \rangle$, où $\tilde{x} = [1, x_1, \ldots, x_d]^T$ (et le nombre de paramètre $N$ vaut donc $d+1$). Par exemple, dans le cadre de la régression linéaire classique via *la méthode des moindres carrés*, on combine un modèle linéaire avec fonction de perte égale à la différence au carré entre la prédiction du modèle et la valeur observée : $L(\theta, x_i, y_i) = (\hat{f}_\theta(x_i) - y_i)^2$. La solution de ce cas de figure est connue, et on peut montrer que :

$$\theta^\star = (\hat{X}_n^T \hat{X}_n)^{-1} \hat{X}_n^T \hat{Y}_n,$$

où $\hat{X}_n \in \mathbb{R}^{n \times d}$ est la matrice dont les lignes correspondent aux $x_i$, et $\hat{Y}_n := [y_1, \ldots, y_n]^T$.

Dans le cadre de la classification binaire, c'est-à-dire à deux catégories $c_1, c_2$, un modèle linéaire courant est le modèle de *régression logistique* (qui, malgré son nom, est bien un modèle de classification !), qui classe les points en modélisant leurs probabilités d'appartenance à l'une des catégories. Plus formellement, pour un point $x$ dont la catégorie $y$ est inconnue, on a :

$$\begin{aligned} \hat{f}_\theta(x) &:= \operatorname{argmax} \{\mathbb{P}(y = c_1 | x, \theta), \mathbb{P}(y = c_2 | x, \theta)\} \\ &= \operatorname{argmax} \left\{ \frac{1}{1 - \exp(-\langle \theta, \tilde{x} \rangle)}, 1 - \frac{1}{1 - \exp(-\langle \theta, \tilde{x} \rangle)} \right\}. \end{aligned}$$

Dans ce cas, et comme précédemment, $\theta^\star$ peut être calculé explicitement en utilisant comme fonction de perte l'opposé de la log-vraisemblance du modèle sur les données d'entraînement $L(\theta, x_i, y_i) := -\log(\mathbb{P}(y = y_i | x_i, \theta))$.

*Machines à support de vecteurs.* — Un autre paradigme de classification binaire standard à partir d'un modèle linéaire consiste, plutôt que de modéliser une probabilité d'appartenance, à simplement classer les points en fonction de leur position vis-à-vis d'un hyperplan :

$$\hat{f}_\theta(x) = \operatorname{signe}(\langle \theta, \tilde{x} \rangle),$$

où, pour cette méthode uniquement, on utilisera $-1$ et $+1$ pour désigner les deux catégories. On peut ainsi chercher à optimiser la perte $L(\theta, x_i, y_i) = 1 - y_i \hat{f}_\theta(x_i)$. En outre, pour des données linéairement séparables, il existe plusieurs hyperplans possibles qui réalisent une perte nulle sur les données d'entraînement. Ainsi, et pour des raisons de robustesse, on préfèrera un hyperplan dont les *marges*, c'est-à-dire

les plus petites distances des points d'entraînement à l'hyperplan, sont élevées. Voir la figure 28. Ceci permet de transformer le problème général (15) en un problème d'optimisation quadratique via quelques manipulations simples (sur ce point, voir par exemple (Murphy, 2022, Section 17.3.1)), dont on peut trouver des solutions via un solveur numérique. Lorsque les données ne sont pas linéairement séparables, on peut soit résoudre une relaxation du problème d'optimisation quadratique en autorisant les marges à être négatives (et en les pénalisant si c'est le cas), ou utiliser des méthodes à noyaux (voir la section 6.3 ci-dessous).

Nous allons maintenant présenter quelques-uns des modèles non-linéaires les plus fréquents pour la classification (dans la plupart des cas, ces modèles s'adaptent de manière évidente à la régression).

*Modèle des plus proches voisins.* — Un des modèles non-linéaires les plus simples est le modèle des plus proches voisins. Soit $k \in \mathbb{N}^*$. Le *modèle de classification à $k$ voisins* est le modèle suivant :

$$\hat{f}_\theta^{\mathcal{D}_n}(x) = \underset{1\leq c\leq C}{\operatorname{argmax}} \operatorname{card}(\{(x_i, y_i) \in \mathcal{D}_n \,|\, y_i = c \text{ et } x_i \in \mathcal{N}_k(x)\}),$$

où $\theta = \{k\}$ et $\mathcal{N}_k(x)$ dénote les $k$ plus proches voisins de $x$ parmi les données observées. Ce modèle consiste donc simplement, pour prédire la catégorie d'un point $x$, à choisir la catégorie la plus fréquente, ou catégorie majoritaire, parmi les $k$ plus proches voisins de $x$. Voir la figure 28. Ce modèle ne dispose que du paramètre $k \in \mathbb{N}^*$, dont la valeur optimale $k^\star$ peut s'obtenir par *$K$-validation croisée*, c'est-à-dire en partitionant les données d'entraînement en $K$ sous-ensembles $\mathcal{D}_1, \ldots, \mathcal{D}_K$ (si possible de même taille), et en choisissant un $k^\star$ qui minimise la fonction de perte $L(\theta, x_i, y_i) = \delta\left\{\hat{f}_\theta^{\mathcal{D}_{-j_i}}(x_i) = y_i\right\}$, où $j_i$ désigne l'indice du sous-ensemble qui contient $(x_i, y_i)$, $\mathcal{D}_{-j_i} := \mathcal{D}_n \setminus \mathcal{D}_{j_i}$, et $\delta\{A\} = 1$ si $A$ est vraie, $\delta\{A\} = 0$ sinon. La raison pour laquelle on évalue le modèle via des sous-ensembles est que $\hat{f}_\theta^{\mathcal{D}_n}(x_i) = y_i$ par définition ; il est donc impossible de prévoir le comportement de $\hat{f}_\theta^{\mathcal{D}_n}$ si on l'évalue sur les données observées (car la perte sera toujours nulle, un problème que l'on appelle de manière plus générale le *surapprentissage*), on préfère à la place étudier ses propriétés de généralisation via des évaluations de "sous-modèles", c'est-à-dire des instances du modèle qui n'ont pu observer que certains sous-ensembles, et non pas l'intégralité des données d'entraînement.

*Forêts aléatoires.* — Un autre modèle non-linéaire standard de classification consiste à agréger des modèles simples basés sur des arbres.

Un modèle d'arbre de classification peut se définir comme :

$$\hat{f}_\theta(x) = \sum_{j=1}^{J} \theta_j \delta\{x \in R_j\},$$

où $\theta = \{(\theta_j, R_j)\}_{1 \leq j \leq J}$, $J \in \mathbb{N}^*$, et chaque $R_j$ correspond à une région rectangulaire de $\mathbb{R}^d$ de la forme $R_j = \{x \in \mathbb{R}^d \,|\, \alpha_1^j \leq x_{(1)} \leq \beta_1^j, \ldots, \alpha_d^j \leq x_{(d)} \leq \beta_d^j\}$, qui peut s'interpréter comme le noeud d'un arbre binaire $A$ de partitionnement de $\mathbb{R}^d$. Voir la figure 28. Les paramètres $\theta_j$ d'un arbre $A$ sont en général des valeurs réelles pour les arbres de régression, et des catégories pour les arbres de classification, et peuvent être optimisés (ainsi que les seuils $\alpha^j, \beta^j$) sur les données d'entraînement en augmentant la taille de l'arbre de manière itérative pour minimiser la perte $L(\theta, x_i, y_i) = \delta\{f_\theta(x_i) = y_i\}$ (voir par exemple la méthode CART). En pratique, les modèles d'arbre sont assez sensibles aux perturbations et généralisent mal, ce pourquoi on préfère construire des modèles ensemblistes :

$$\hat{f}_\theta(x) = \underset{1 \leq c \leq C}{\operatorname{argmax}} \operatorname{card}(\{A \in \mathcal{A} \,|\, f_{\theta_A}(x) = c\}),$$

où $\theta = \mathcal{A}$ est une famille finie d'arbres (chaque arbre $A$ ayant pour paramètres $\theta_A$) entraînés sur des sous-échantillons des données d'entraînement (pour éviter le surapprentissage).

*Réseaux de neurones.* — Enfin, une vaste famille de modèles, qui compte parmi les plus utilisées actuellement, est la famille constituée des *réseaux de neurones*. Un modèle de réseau de neurones (aussi parfois appelé une *architecture* de réseau) est un modèle de la forme :

$$\hat{f}_\theta(x) = \hat{f}_{W_J, b_J, \sigma_J} \circ \cdots \circ \hat{f}_{W_1, b_1, \sigma_1},$$

où $\theta = \{(W_j, b_j, \sigma_j)\}_{1 \leq j \leq J}$, $J \in \mathbb{N}^*$, et chaque fonction $\hat{f}_{W_j, b_j, \sigma_j}$, appelée *couche du réseau* de taille $l_j \in \mathbb{N}^*$, est définie par : $\hat{f}_{W_j, b_j, \sigma_j}(x) = \sigma_j(W_j x + b_j)$, où $W_j \in \mathbb{R}^{l_j \times l_{j-1}}$ est une *matrice de poids*, $b_j \in \mathbb{R}^{l_j}$ est un *biais*, et $\sigma_j$ est une *fonction d'activation* non-linéaire (comme par exemple une fonction sigmoïde, une fonction ReLU $\sigma_j(x) = \max\{0, x\}$, etc). Remarquons que $l_0 = d$ est la dimension des données, et que la taille de la dernière couche $l_M$ vaut 1 pour la régression, et $C$ (le nombre de catégories) pour la classification, auquel cas on ajoute traditionnellement un argmax à la prédiction du modèle (renormalisée de telle manière que ses coordonnées soient positives et de somme égale à un) pour déterminer la catégorie. Cette architecture est appelée un modèle *complètement connecté* de

réseau, et beaucoup d'autres architectures sont aujourd'hui disponibles. Les paramètres optimaux $W_j^\star, b_j^\star$ sont en général obtenus par descente de gradient (même s'il n'existe pas de garantie générale de trouver un minimum global, ces modèles étant non-convexes) en minimisant, pour la classification, une perte appelée *entropie croisée* : $L(\theta, x_i, y_i) = -\sum_{c=1}^{C} \hat{f}_\theta(x_i)_{(c)} \log(\mathrm{OH}(y_i)_{(c)})$, où OH désigne une représentation *one-hot* de $y_i$, c'est-à-dire un vecteur de taille $C$ dont toutes les coordonnées sont nulles à l'exception de la coordonnée $y_i$ qui vaut 1. Pour la régression, on utilise généralement la perte des moindres carrés. Notons finalement que les tailles ainsi que le nombre de couches sont des hyperparamètres que l'on peut aussi optimiser par validation croisée.

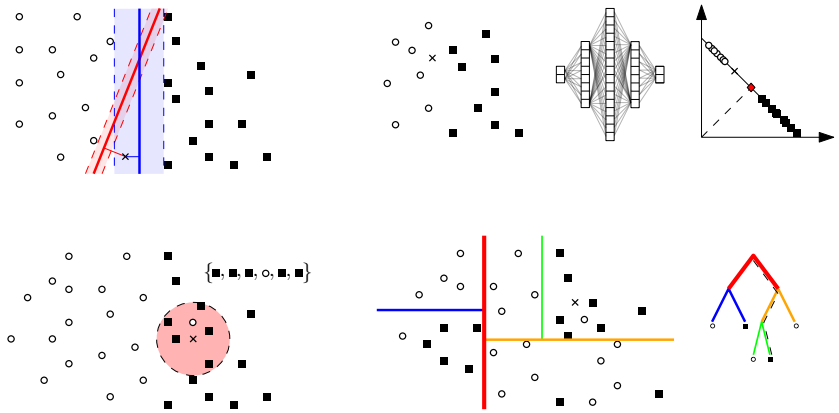


FIGURE 28. Exemples de modèles de classification, et de leurs prédictions sur un point de test (désigné par une croix). **(Gauche, haut)** Machines à support de vecteurs de marge faible (rouge) ou élevée (bleu). **(Gauche, bas)** Modèle de classification à six voisins. **(Droite, haut)** Modèle de réseau de neurones complètement connecté à quatre couches, pour lequel les données sont envoyées sur le segment $x + y = 1, x, y \geq 0$ du plan, et classées en fonction de leurs positions vis-à-vis du milieu du segment (en rouge). **(Droite, bas)** Modèle d'arbre de classification, et chemin emprunté par le point de test dans l'arbre pour la prédiction.

Une dernière remarque importante : il est très fréquent de rajouter un terme $\Omega(\theta)$ au problème général (15). En effet, si les paramètres du modèle ne sont pas contraints, on arrive souvent à des effets de surapprentissage, en particulier quand le nombre de paramètres $N$ est

supérieur au nombre de points $n$. Ces termes sont appelés *termes de régularisation*, et correspondent souvent à la norme 1 ou 2 de $\theta$ (vu comme un vecteur de $\mathbb{R}^N$).

**6.2. Quelques représentations finies classiques des diagrammes de persistance. —** Repassons maintenant dans le cadre des diagrammes de persistance, et supposons que l'on veuille appliquer des modèles prédictifs sur des diagrammes de persistance munis de catégories différentes. Voir par exemple la figure 29 pour un exemple d'application, où l'on désire assigner des catégories à différents points de formes 3D. Pour éviter de dépendre du plongement de la forme, il n'est pas désirable d'utiliser directement les coordonnées des points des formes, mais plutôt d'utiliser des descripteurs *intrinsèques*. Les diagrammes de persistance en font partie, lorsqu'ils sont obtenus par exemple en filtrant les formes 3D via des boules géodésiques : pour chaque point de la forme, une boule géodésique centrée sur le point va changer de topologie en grossissant, ce que l'on peut capturer avec l'homologie persistante, et d'une manière qui ne dépend pas du plongement des formes.

Il apparaît clairement des modèles prédictifs présentés dans la section précédente qu'un pré-requis pour leur utilisation est que les données soient des vecteurs Euclidiens : tous ces modèles font en effet usage des coordonnées des points de données pour bâtir leurs prédictions. Ceci pose un problème pour les diagrammes de persistance, car ils n'ont pas de coordonnées bien définies. Plus généralement, il n'existe pas de notion de somme, moyenne ou produit scalaire pour les diagrammes, et donc les modèles précédents ne peuvent pas s'appliquer. Pour remédier à cela, une partie de la littérature de l'analyse topologique de données est consacrée à la définition de *représentations* des diagrammes de persistance, c'est-à-dire à la définition de transformations explicites $\Phi : \mathrm{Dgm} \to \mathbb{R}^d$, où $d \in \mathbb{N}^*$.

Un soin particulier est de plus apporté aux représentations qui *préservent la stabilité*, c'est-à-dire aux représentations pour lesquelles :

$$\|\Phi(D) - \Phi(D')\|_p \leq K d_q(D, D'),$$

où $1 \leq p, q \leq +\infty$ et $K$ est une constante strictement positive. En effet, pour $q = +\infty$ et en vertu du théorème de stabilité entre diagrammes de persistance, toute représentation préservant la stabilité satisfera $\|\Phi(D(f)) - \Phi(D(g))\|_p \leq K\|f - g\|_\infty$, et sera donc elle-même une représentation stable des données. Dans la suite, nous présentons

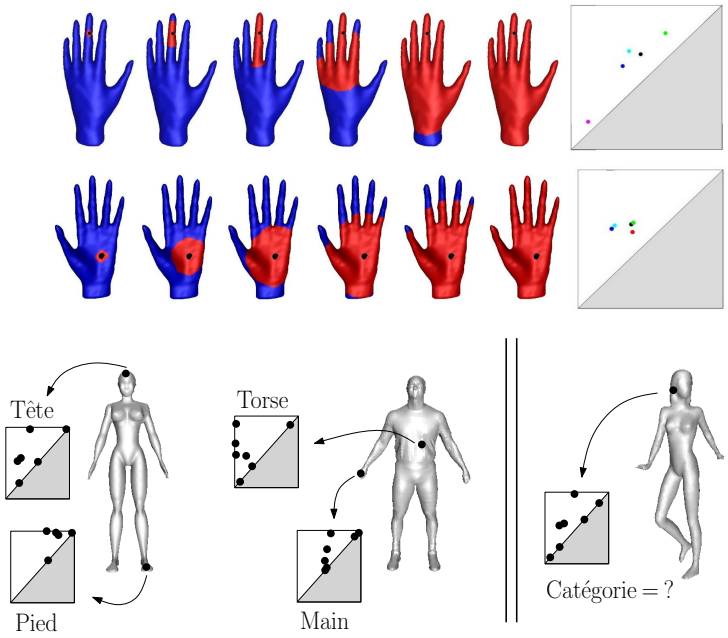



FIGURE 29. Exemple de modèle prédictif pour la segmentation de formes 3D bâti sur des diagrammes de persistance. À partir de filtrations issues de boules géodésiques grandissantes (**haut, milieu**), on peut calculer des descripteurs des points sous la forme de diagrammes de persistance, et les utiliser pour prédire les catégories de points via un modèle prédictif optimisé sur des diagrammes d'entraînement (**bas**).

trois des représentations finies les plus courantes de l'analyse topologique de données.

*Distances à la diagonale.* — Une manière simple mais efficace de représenter les diagrammes consiste simplement à trier les distances à la diagonale.

***Définition 6.1.*** — *Soit* $D \in \mathrm{Dgm}$. *La* suite diagonale $u^D$ *associée à* $D$ *est définie par :*

$$u_k^D = k\max\{\|p - \pi_\Delta(p)\|_\infty \mid p \in D\},$$

*où* $k\max$ *désigne le* $k$*-ème maximum d'un ensemble, et avec la convention que* $u_k^D = 0$ *pour* $k > \mathrm{card}(D)$.

En pratique, pour un jeu de données de diagrammes de persistance $D_1, \dots, D_n$, on tronque les suites à un certain seuil prédéfini $T \in \mathbb{N}^*$, que l'on prend souvent égal à $\max\{\mathrm{card}(D_i) \,|\, 1 \leq i \leq n\}$. Appelons $u^{D_i,T}$ les suites tronquées, considérées comme des vecteurs de $\mathbb{R}^T$.

***Proposition 6.1.*** — *Pour tout $D, D' \in \mathrm{Dgm}$, on a :*

$$\|u^{D,T} - u^{D',T}\|_\infty \leq 2d_b(D, D').$$

*Démonstration.* — Considérons d'abord le cas des suites non tronquées, et utilisons l'ordre de $u^D$ pour fixer un ordre sur $D$ :

$$D = \{p_k \,|\, 1 \leq k \leq \mathrm{card}(D)\} \text{ tel que } u^D_k = \|p_k - \pi_\Delta(p_k)\|_\infty.$$

On va maintenant construire une suite $\tilde{u}^{D'}$ à partir de cet ordre. Soit $\mathcal{C}$ une correspondance partielle entre $D$ et $D'$ qui réalise $d_b(D, D')$.

Supposons $\mathrm{card}(D) \leq \mathrm{card}(D')$. Soit $k \in \mathbb{N}^*$, tel que $k \leq \mathrm{card}(D)$. Si il existe $p' \in D'$ tel que $(p_k, p') \in \mathcal{C}$, on définit $\gamma(p_k) = p'$. Si ce n'est pas le cas (c'est-à-dire si $p_k$ est apparié avec la diagonale), on définit $\gamma(p_k)$ comme un des points de $D'$ aussi apparié à la diagonale, et choisi de telle manière que $\gamma$ soit injective. On définit ensuite $\tilde{u}^{D'}_k := \|\gamma(p_k) - \pi_\Delta(\gamma(p_k))\|_\infty$. Pour $\mathrm{card}(D) \leq k \leq \mathrm{card}(D')$, on utilise les distances à la diagonale des $\mathrm{card}(D') - \mathrm{card}(D)$ points restants de $D'$ (ordonnés de manière arbitraire) pour remplir les entrées de $\tilde{u}^{D'}$, et enfin pour $k > \mathrm{card}(D')$ on définit $\tilde{u}^{D'}_k = 0$. La construction de $\tilde{u}^{D'}$ pour $\mathrm{card}(D) > \mathrm{card}(D')$ est similaire, à savoir qu'on utilise les distances à la diagonale des points de $D'$ obtenus en appliquant $\gamma$ sur les $\mathrm{card}(D')$ premiers points de $D$, et qu'on complète ensuite la suite avec des valeurs nulles. La suite $\tilde{u}^{D'}$ est donc une permutation des entrées de $u^{D'}$.

Soit $k \in \mathbb{N}^*$. Pour étudier $|u^D_k - \tilde{u}^{D'}_k|$, trois cas sont possibles.

— Soit $|u^D_k - \tilde{u}^{D'}_k| = |\|p_k - \pi_\Delta(p_k)\|_\infty - \|\gamma(p_k) - \pi_\Delta(\gamma(p_k))\|_\infty|$ avec $(p_k, \gamma(p_k)) \in M$. Alors, on a (par l'inégalité triangulaire) :

$$\begin{aligned}
\|p_k - \pi_\Delta(p_k)\|_\infty - \|\gamma(p_k) - \pi_\Delta(\gamma(p_k))\|_\infty & \\
&\leq \|p_k - \gamma(p_k)\|_\infty + \|\gamma(p_k) - \pi_\Delta(\gamma(p_k))\|_\infty \\
&\quad + \|\pi_\Delta(\gamma(p_k)) - \pi_\Delta(p_k)\|_\infty - \|\gamma(p_k) - \pi_\Delta(\gamma(p_k))\|_\infty \\
&= \|p_k - \gamma(p_k)\|_\infty + \|\pi_\Delta(\gamma(p_k)) - \pi_\Delta(p_k)\|_\infty \\
&\leq 2\|p_k - \gamma(p_k)\|_\infty \\
&\leq 2d_b(D, D').
\end{aligned}$$

Par symétrie, on obtient finalement $|u^D_k - \tilde{u}^{D'}_k| \leq 2d_b(D, D')$.

— Soit $|u_k^D - \tilde{u}_k^{D'}| = |\|p_k - \pi_\Delta(p_k)\|_\infty - \|\gamma(p_k) - \pi_\Delta(\gamma(p_k))\|_\infty|$ avec $p_k$ et $\gamma(p_k)$ tous les deux appariés à la diagonale. Alors on a : $|\|p_k - \pi_\Delta(p_k)\|_\infty - \|\gamma(p_k) - \pi_\Delta(\gamma(p_k))\|_\infty| \leq \|p_k - \pi_\Delta(p_k)\|_\infty + \|\gamma(p_k) - \pi_\Delta(\gamma(p_k))\|_\infty \leq 2d_b(D, D')$.

— Soit $u_k^D = 0$ ou $\tilde{u}_k^{D'} = 0$. La démonstration du point ci-dessus peut s'utiliser de nouveau pour prouver $|u_k^D - \tilde{u}_k^{D'}| \leq d_b(D, D') \leq 2d_b(D, D')$.

Ainsi, on a $\|u^D - \tilde{u}^{D'}\|_\infty \leq 2d_b(D, D')$. Le résultat final s'obtient en remarquant que trier les entrées d'une suite est une opération Lipschitzienne (exercice laissé au lecteur) :

$$\|u^{D,T} - u^{D',T}\|_\infty \leq \|u^D - u^{D'}\|_\infty \leq \|u^D - \tilde{u}^{D'}\|_\infty \leq 2d_b(D, D').$$

□

Voir la figure 30. On peut aussi compléter ces suites avec les distances calculées entre les différents points du diagramme, en plus des distances à la diagonale. Voir Carrière et al. (2015) pour une étude de cette représentation.

Il est cependant clair que caractériser un diagramme par les distances de ses points à la diagonale, et/ou les distances internes entre ses points, n'est pas une représentation injective : on peut très facilement construire des diagrammes différents dont les représentations correspondantes sont égales (il suffit par exemple de translater les points le long de la diagonale).

*Images de persistance.* — Une autre possibilité plus discriminante que la représentation précédente, introduite par Adams et al. (2017), consiste simplement à transformer les diagrammes de persistance en fonctions intégrables, en convoluant les points des diagrammes avec des Gaussiennes pondérées.

***Définition 6.2.*** — *Soit* $D \in \mathrm{Dgm}$. La surface de persistance $I\rho(D; w, \sigma)$ *de paramètres* $w : \mathbb{R}^2 \to \mathbb{R}$ *et* $\sigma > 0$*, associée à* $D$*, est la fonction suivante :*

$$\rho(D; w, \sigma) : \begin{cases} \mathbb{R}^2 \to \mathbb{R} \\ x \mapsto \displaystyle\sum_{p \in D} w(\tilde{p}) \cdot \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|x - \tilde{p}\|_2^2}{2\sigma^2}\right) \end{cases}$$

*où* $\tilde{p} = (p_x, p_y - p_x)$ *(de telle manière que la diagonale devienne l'axe des abscisses).*

En pratique, pour éviter d'avoir à gérer les fonctions elles-mêmes, on préfère souvent "pixeliser" ces fonctions, en les intégrant sur les cases d'une grille placée sur le plan :

***Définition 6.3.*** — *Soit* $D \in \mathrm{Dgm}$. *Soit* $G = \{g_{i,j} = [a_i, a_{i+1}] \times [b_j, b_{j+1}] \,|\, a_1 \leq \cdots \leq a_{p+1}, b_1 \leq \cdots \leq b_{q+1}\}$ *une grille de taille* $p \times q$, $p, q \in \mathbb{N}^*$. L'image de persistance $I(D; w, \sigma) \in \mathbb{R}^{pq}$ *de paramètres* $w : \mathbb{R}^2 \to \mathbb{R}$ *et* $\sigma > 0$, *associée à* $D$, *est le vecteur dont les coordonnées sont données par* $I(D; w, \sigma)_{(g_{i,j})} = \int_{a_i}^{a_{i+1}} \int_{b_j}^{b_{j+1}} \rho(D; w, \sigma)(x)\mathrm{d}x$.

Voir la figure 30. La fenêtre $\sigma$ et la fonction de poids $w$ sont laissés au choix de l'utilisateur, et peuvent être optimisés comme des hyperparamètres (car en tant que tels ils ne dépendent pas des données d'entraînement), donc par exemple par validation croisée. On peut démontrer que l'image de persistance est stable vis-à-vis de la distance de Wasserstein $d_1$ d'ordre 1 entre les diagrammes de persistance, pour des fonctions de poids bien choisies.

***Proposition 6.2.*** — *Soit* $w$ *une fonction de poids telle que* $w(\tilde{p}) = 0$ *pour tout point* $\tilde{p}$ *tel que* $\tilde{p}_y = 0$ *(c'est-à-dire tel que* $p$ *soit sur la diagonale). Pour tout* $D, D' \in \mathrm{Dgm}$, *on a :*

$$\|I(D; w, \sigma) - I(D'; w, \sigma)\|_1 \leq \|\rho(D; w, \sigma) - \rho(D'; w, \sigma)\|_1 \leq K d_1(D, D'),$$

*où* $K = \left(\sup_u \|\nabla w(u)\| + \sqrt{\frac{2}{\pi}} \frac{\|w\|_\infty}{\sigma}\right)$.

*Démonstration.* — La première inégalité est une conséquence directe de $\|I\|_1 = \sum |\int\int \rho| \leq \sum \int\int |\rho| = \|\rho\|_1$. Prouvons donc le résultat pour les surfaces de persistance. La preuve est basée sur le lemme suivant, dont la preuve est laissée en exercice au lecteur (voir (Adams et al., 2017, Lemma 8)) :

***Lemme 6.1*** **(admis).** — *Soit* $p, p' \in \mathbb{R}^2$. *Alors :*

$$\left\| w(p) \cdot \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|\cdot - p\|_2^2}{2\sigma^2}\right) - w(p') \cdot \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|\cdot - p'\|_2^2}{2\sigma^2}\right) \right\|_1$$
$$\leq \left(\sup_u \|\nabla w(u)\| + \sqrt{\frac{2}{\pi}} \frac{\|w\|_\infty}{\sigma}\right) \|p - p'\|_2.$$

Soit $\mathcal{C}$ une correspondance partielle qui réalise $d_1(D, D')$. Soit $\mathcal{N}_{\tilde{p},\sigma} := \frac{1}{2\pi\sigma^2}\exp\left(-\frac{\|\cdot-\tilde{p}\|_2^2}{2\sigma^2}\right)$. Alors, on a :

$$
\begin{aligned}
&\|\rho(D; w, \sigma) - \rho(D'; w, \sigma)\|_1 \\
&\quad\leq \left\| \sum_{(p,p')\in\mathcal{C}} w(\tilde{p})\cdot\mathcal{N}_{\tilde{p},\sigma} - w(\tilde{p}')\cdot\mathcal{N}_{\tilde{p}',\sigma} \right\|_1 \\
&\quad + \left\| \sum_{p\in D\setminus \mathrm{im}(\pi_1)} w(\tilde{p})\cdot\mathcal{N}_{\tilde{p},\sigma} \right\|_1 + \left\| \sum_{p'\in D'\setminus \mathrm{im}(\pi_2)} w(\tilde{p})\cdot\mathcal{N}_{\tilde{p}',\sigma} \right\|_1 \\
&\quad\leq \sum_{(p,p')\in\mathcal{C}} \left\| w(\tilde{p})\cdot\mathcal{N}_{\tilde{p},\sigma} - w(\tilde{p}')\cdot\mathcal{N}_{\tilde{p}',\sigma} \right\|_1 \\
&\quad + \sum_{p\in D\setminus \mathrm{im}(\pi_1)} \left\| w(\tilde{p})\cdot\mathcal{N}_{\tilde{p},\sigma} - w(\tilde{\pi}_\Delta(p))\cdot\mathcal{N}_{\tilde{\pi}_\Delta(p),\sigma} \right\|_1 \\
&\quad + \sum_{p'\in D'\setminus \mathrm{im}(\pi_2)} \left\| w(\tilde{p})\cdot\mathcal{N}_{\tilde{p}',\sigma} - w(\tilde{\pi}_\Delta(p'))\cdot\mathcal{N}_{\tilde{\pi}_\Delta(p'),\sigma} \right\|_1 \\
&\qquad \text{car } \tilde{\pi}_\Delta(p) := (\pi_\Delta(p)_x, \pi_\Delta(p)_y - \pi_\Delta(p)_x) = (\pi_\Delta(p)_x, 0) \\
&\qquad \text{et donc } w(\tilde{\pi}_\Delta(p)) = 0 \text{ (et pareillement pour } \tilde{\pi}_\Delta(p')) \\
&\quad\leq \underbrace{\left(\sup_u \|\nabla w(u)\| + \sqrt{\frac{2}{\pi}}\frac{\|w\|_\infty}{\sigma}\right)}_{=K} \Big( \sum_{(p,p')\in\mathcal{C}} \|\tilde{p} - \tilde{p}'\|_2 \\
&\quad + \sum_{p\in D\setminus \mathrm{im}(\pi_1)} \|\tilde{p} - \tilde{\pi}_\Delta(p)\|_2 + \sum_{p'\in D'\setminus \mathrm{im}(\pi_2)} \|\tilde{p}' - \tilde{\pi}_\Delta(p')\|_2 \Big) \\
&\qquad \text{d'après le lemme 6.1} \\
&\quad\leq \sqrt{5}K d_1(D, D') \qquad \text{car } \|\tilde{p}\|_2 \leq \sqrt{5}\|p\|_\infty.
\end{aligned}
$$

□

On peut généraliser la construction des surfaces et images de persistance à d'autres fonctions que des Gaussiennes, ainsi qu'à d'autres fonctions de poids, mais il est possible de montrer que préserver la stabilité impose d'utiliser des fonctions de poids qui tendent vers zéro pour des points qui tendent vers l'axe des abscisses (qui, on le rappelle, correspond à la diagonale des diagrammes de persistance). Sur ce point, voir Divol and Lacombe (2020).

*Paysages de persistance.* — Enfin, une dernière construction très usuelle, dûe à Bubenik (2015), peut se définir au niveau des modules de persistance.

***Définition 6.4.*** — *Soit $M = \{\{M_t\}_{t\in\mathbb{R}}, \{m_{t,t'}\}_{t\leq t'\in\mathbb{R}}\}$ un module de persistance indexé sur $\mathbb{R}$. Soit $t < t'$. On appelle $\beta^{t,t'}(M) := \mathrm{rank}(m_{t,t'}) = \dim(\mathrm{im}(m_{t,t'}))$ le* nombre de Betti persistant *entre $t$ et $t'$. Le $k$-ème* paysage *(*landscape *en anglais) de $M$, pour $k \in \mathbb{N}^*$, est la fonction :*

$$\lambda_k^M : \begin{cases} \mathbb{R} \to \mathbb{R} \\ t \mapsto \sup\{s > 0 \,|\, \beta^{t-s,t+s}(M) \geq k\} \end{cases}$$

On peut en réalité démontrer que ces fonctions paysage peuvent se définir au niveau des diagrammes de persistance eux-mêmes, comme les maxima de fonctions tente définies sur les points des diagrammes de persistance.

***Exercice 6.1.*** — *Soit $M$ un module de persistance pdf, et soit $D$ son diagramme de persistance. Montrer que :*

$$\lambda_k^M(t) = k \max\{\Lambda_p(t) \,|\, p \in D\},$$

*où $\Lambda_p(t) = 0$ si $t \leq p_x$ ou $t \geq p_y$, $\Lambda_p(t) = t - p_x$ si $p_x \leq t \leq \frac{p_x+p_y}{2}$ et $\Lambda_p(t) = p_y - t$ si $\frac{p_x+p_y}{2} \leq t \leq p_y$.*

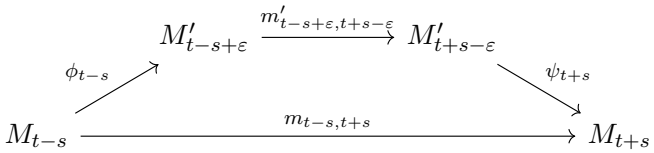
L'intérêt de la définition des paysages de persistance via les nombres de Betti persistants est qu'elle simplifie considérablement la preuve de la stabilité des paysages.

***Proposition 6.3.*** — *Soient $M = \{\{M_t\}_{t\in\mathbb{R}}, \{m_{t,t'}\}_{t\leq t'\in\mathbb{R}}\}$ et $M' = \{\{M'_t\}_{t\in\mathbb{R}}, \{m'_{t,t'}\}_{t\leq t'\in\mathbb{R}}\}$ deux modules de persistance pdf indexés sur $\mathbb{R}$, et soient $D, D'$ leurs diagrammes de persistance. Soit $k \in \mathbb{N}^*$. Alors, on a :*

$$\|\lambda_k^M - \lambda_k^{M'}\|_\infty \leq d_b(D, D').$$

*Démonstration.* — Soit $t \in \mathbb{R}$ et $\varepsilon = d_b(D, D') = d_I(M, M')$. Supposons que $\lambda_k^M(t) \geq \varepsilon$, et soit $s > 0$ tel que $\lambda_k^M(t) \geq s \geq \varepsilon$. Alors, il existe des familles d'applications linéaires $\{\phi_t\}_t, \{\psi_t\}_t$ telles que l'on

a le diagramme commutatif suivant :




Ainsi, on obtient $\beta^{t-s+\varepsilon,t+s-\varepsilon}(M') \geq \beta^{t-s,t+s}(M) \geq k$. D'où l'on déduit $\lambda_k^{M'}(t) \geq s - \varepsilon$ pour tout $s$, et donc $\lambda_k^{M'}(t) \geq \lambda_k^M(t) - \varepsilon$. Si $\lambda_k^M(t) < \varepsilon$, on a trivialement $\lambda_k^{M'}(t) \geq 0 > \lambda_k^M(t) - \varepsilon$. L'argument étant symétrique en $M$ et $M'$, on obtient le résultat désiré. □

Voir la figure 30. Enfin, de la même manière que pour les images de persistance, il est possible de définir les *silhouettes de persistance* en sommant les fonctions tente (voir exercice 6.1), pondérées de telle manière à préserver la stabilité. Il est en outre possible de prouver des résultats de convergence statistique particuliers au moyen de ces représentations, voir sur ce point Chazal et al. (2015).

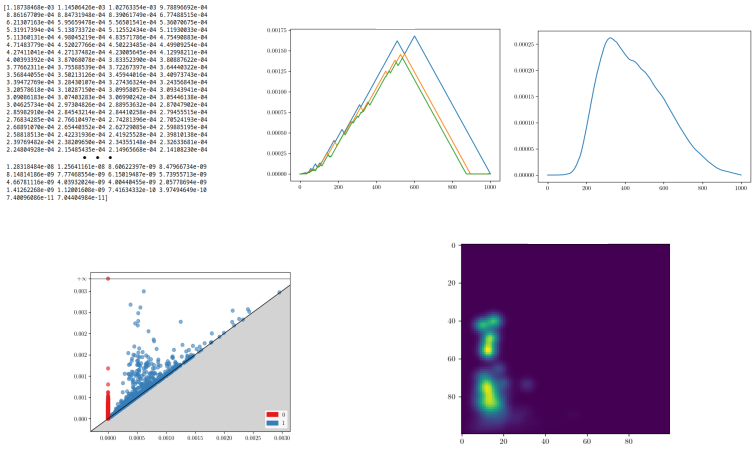


FIGURE 30. Exemples des représentations classiques présentées dans ce chapitre. Pour un diagramme de persistance donné (**bas, gauche**), on peut stocker la liste de ses distances à la diagonale triée (**haut, gauche**), les fonctions linéaires par morceaux de son payage de persistance et sa silhouette (**haut, droite**), et son image de persistance (**bas, droite**).

De manière générale, bien que les paysages, silhouettes et images de persistance contiennent plus d'information que la suite des distances

triées, ces représentations ont leurs propres faiblesses : les paysages sont très redondants (en particulier lorsqu'on les stocke comme des vecteurs en les échantillonnant le long de l'axe des abscisses), et les images sont très creuses, ce qui force à les combiner avec des modèles prédictifs capables de gérer ces propriétés.

**6.3. Méthodes à noyaux pour les diagrammes de persistance.** — Une autre limitation évidente des représentations de diagrammes mentionnées en section 6.2, mais aussi plus généralement des modèles présentés en section 6.1, est leur restriction aux modèles *finis*, c'est-à-dire entièrement caractérisés par un nombre fini de paramètres. Cette limitation est particulièrement remarquable pour les machines à support de vecteurs par exemple, au sens où il est très simple de construire des données non linéairement séparables. En ce qui concerne les diagrammes de persistance, l'ensemble de ces diagrammes, même lorsque l'on se restreint à ceux dont le cardinal est borné supérieurement, est connu pour être de dimension infinie, ainsi que pour sa courbure négative (voir Turner et al. (2014)). Il est donc peu probable que des représentations ou des modèles de dimension finie permettent de capturer l'intégralité des propriétés des diagrammes. Il existe toutefois une classe de modèles non-linéaires que nous n'avons pas encore abordée, et qui permet justement de manipuler des représentations de dimension infinie, les *méthodes à noyaux*, dont nous allons maintenant présenter les bases.

*Bases des méthodes à noyaux.* — L'idée fondamentale des méthodes à noyaux est de représenter les points de données par des vecteurs vivant dans des espaces de Hilbert bien spécifiques, les *espaces à noyau reproduisant* (ENR).

***Définition 6.5.*** — *Soit $X$ un ensemble de données (non nécessairement Euclidiennes). Un ENR $\mathcal{H}$ sur $X$ est un espace de Hilbert de fonctions $\mathcal{H} = \{f : X \to \mathbb{R}\}$ telles que les* fonctionnelles d'évaluation *$\{F_x : \mathcal{H} \to \mathbb{R} \,|\, x \in X\}$, définies par $F_x(f) = f(x)$, soient toutes des fonctions continues vis-à-vis de la distance induite par la norme $\|\cdot\|_{\mathcal{H}}$.*

Cette définition permet de définir simplement des représentations dans $\mathcal{H}$ pour tout point de $X$ ; en effet, par le théorème de représentation de Riesz, il s'ensuit de la continuité des $F_x$ que l'on peut associer un vecteur $\Phi_{\mathcal{H}}(x) \in \mathcal{H}$ à chaque $x \in X$ tel que $F_x(f) = \langle f, \Phi_{\mathcal{H}}(x)\rangle_{\mathcal{H}}$.

Ne reste donc plus qu'à constuire de tels ENR. Un résultat central de Moore et Aronszajn permet de caractériser les ENR via des *noyaux*.

***Définition 6.6.*** — *Un noyau sur $X$ est une fonction $K : X \times X \to \mathbb{R}$ telle que, pour toute famille finie $x_1, \dots, x_n$, $n \in \mathbb{N}^*$, la matrice de Gram $\{\{K(x_i, x_j)\}\}_{i,j}$ soit définie positive.*

***Théorème 6.1*** **(admis).** — *Un espace de Hilbert $\mathcal{H}$ est un ENR sur $X$ si et seulement si il existe un noyau $K$ tel que $\Phi_{\mathcal{H}}(x) = K(x, \cdot)$.*

Il suffit donc de définir un noyau sur $X$ pour définir (implicitement) un ENR sur $X$ et des représentations associées. De plus, le *théorème du représentant* assure que l'on peut résoudre le problème général (15) en n'ayant accès qu'à la matrice de Gram.

***Théorème 6.2*** **(représentant).** — *Soit $\mathcal{H}$ un ENR sur $X$. Alors les solutions (si elles existent) de*

$$f^\star = \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{n} L(f, \Phi_{\mathcal{H}}(x_i), y_i) + \Omega(\|f\|_{\mathcal{H}}),$$

*où $\Omega : \mathcal{H} \to \mathbb{R}_+$ est une fonction croissante de la norme $\|\cdot\|_{\mathcal{H}}$, peuvent s'écrire sous la forme $f^\star = \sum_{i=1}^n \alpha_i \Phi_{\mathcal{H}}(x_i) = \sum_{i=1}^n \alpha_i K(x_i, \cdot)$.*

*Démonstration.* — Soit $f \in \mathcal{H}$. Commençons par décomposer $f$ comme la somme de sa projection sur le sous-espace $\mathcal{H}_n$ engendré par les $\Phi_{\mathcal{H}}(x_j)$ et un terme orthogonal, on obtient $f = \pi_{\mathcal{H}_n}(f) + f^\perp = \sum_{j=1}^n \alpha_j \Phi_{\mathcal{H}}(x_j) + f^\perp$, avec $\langle \Phi_{\mathcal{H}}(x_j), f^\perp \rangle_{\mathcal{H}} = 0$ pour tout $x_j$. Il en résulte que, pour tout point $x_i$ :

$$\begin{aligned} f(x_i) &= \sum_{j=1}^n \alpha_j \Phi_{\mathcal{H}}(x_j)(x_i) + f^\perp(x_i) \\ &= \sum_{j=1}^n \alpha_j \langle \Phi_{\mathcal{H}}(x_j), \Phi_{\mathcal{H}}(x_i) \rangle_{\mathcal{H}} + \langle f^\perp, \Phi_{\mathcal{H}}(x_i) \rangle_{\mathcal{H}} \\ &= \sum_{j=1}^n \alpha_j K(x_j, x_i) \end{aligned}$$

et donc en définitive, on obtient que

$$L(f, \Phi_{\mathcal{H}}(x_i), y_i) = L(\pi_{\mathcal{H}_n}(f), \Phi_{\mathcal{H}}(x_i), y_i) = L(\{\alpha_j\}_{1 \le j \le n}, \Phi_{\mathcal{H}}(x_i), y_i).$$

En ce qui concerne le terme de régularisation, il est clair que $\Omega(\|f\|_{\mathcal{H}}) \geq \Omega(\|\pi_{\mathcal{H}_n}(f)\|_{\mathcal{H}}) = \Omega(\sum_i \sum_j \alpha_i \alpha_j K(x_i, x_j))$ par monotonicité de $\Omega$. Finalement, on peut donc décroître la valeur de la fonction objectif pour tout $f$ en mettant le terme $f^\perp$ à zéro, d'où le résultat. □

Par exemple, si l'on reprend les machines à support de vecteurs, où le but est de trouver un hyperplan optimal dans $\mathcal{H}$ séparant les données (représentées par $\Phi_{\mathcal{H}}$), et caractérisé par un vecteur normal $f^\star \in \mathcal{H}$, le théorème du représentant permet d'affirmer que $f^\star$ s'écrit comme une combinaison linéaire des $K(x_i, \cdot)$ dont il reste à trouver les coefficients $\alpha_1^\star, \ldots, \alpha_n^\star$. Pour ce faire, il s'ensuit du théorème qu'il est suffisant de ne connaître l'évaluation de $f^\star$ que sur les données d'entraînement, et donc de ne connaître que la matrice de Gram.

De nombreuses méthodes existent pour construire des noyaux, et l'une des plus fréquentes repose sur un résultat qui permet de construire des noyaux Gaussiens généraux.

***Théorème 6.3* (admis).** — *Soit $X$ un ensemble de données muni d'une pseudo-distance $d : X \times X \to \mathbb{R}_+$. La fonction*

$$K_\sigma : (x, x') \mapsto \exp\left(-\frac{d(x, x')}{\sigma}\right)$$

*est un noyau sur $X$ pour tout $\sigma > 0$ si et seulement si $d$ est* conditionnellement semi-définie négative *(CSDN), c'est-à-dire que*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j d(x_i, x_j) \leq 0$$

*pour toute famille finie de points $x_1, \ldots, x_n$ et de coefficients $\alpha_1, \ldots, \alpha_n$, $n \in \mathbb{N}^*$, tels que $\sum_{i=1}^n \alpha_i = 0$.*

***Exercice 6.2.*** — *Montrer, dans le cas où $X = \mathbb{R}^d$, que $d(x, x') = \|x - x'\|_2^2$ est CSDN (on pourra utiliser $\|x - x'\|_2^2 = \langle x - x', x - x'\rangle$), ainsi que $d(x, x') = \|x - x'\|_1$ (on pourra utiliser $|x_{(i)} - x'_{(i)}| = x_{(i)} + x'_{(i)} - 2\min\{x_{(i)}, x'_{(i)}\}$).*

*Un noyau Gaussien pour les diagrammes de persistance.* — Malheureusement, on peut facilement générer des contre-exemples qui montrent que les distances de Wasserstein, ainsi que la distance du goulot de bouteille, ne sont pas CSDN (exercice laissé au lecteur). On ne peut donc pas s'en servir pour créer des noyaux Gaussiens. En revanche, il est beaucoup plus pratique de manipuler ces distances lorsqu'on projette les points sur des droites, ce qui est le principe de la *distance de Wasserstein en tranches*.

***Définition 6.7.*** — *Soient $D, D' \in \mathrm{Dgm}$. On définit $\tilde{D} = D \cup \{\pi_\Delta(p') \,|\, p' \in D'\}$ et $\tilde{D}' = D' \cup \{\pi_\Delta(p) \,|\, p \in D\}$. Pour tout $\alpha \in [-\pi/2, \pi/2]$, on définit $u_\alpha^{\tilde{D}}$ comme le vecteur trié des projections*

$\{\langle p, e_\alpha\rangle \,|\, p \in \tilde{D}\}$, *où* $e_\alpha = (\cos(\alpha), \sin(\alpha))$ *dénote le vecteur unitaire d'angle* $\alpha$. *La* distance de Wasserstein en tranches *(*sliced Wasserstein distance *en anglais) entre* $D$ *et* $D'$ *est définie par :*

$$d_{SW}(D, D') = \frac{1}{\pi} \int_{-\pi/2}^{\pi/2} \|u_\alpha^{\tilde{D}} - u_\alpha^{\tilde{D}'}\|_1 \, \mathrm{d}\alpha.$$

*Remarquons que cette distance est bien définie car* $u_\alpha^{\tilde{D}}$ *et* $u_\alpha^{\tilde{D}'}$ *sont des vecteurs de même taille (égale à* $\mathrm{card}(D) + \mathrm{card}(D')$*).*

La raison pour laquelle on préfère manipuler cette distance est qu'on peut montrer qu'elle est CSDN (et donc que l'on peut servir pour créer des noyaux), mais aussi qu'elle se relie facilement à la distance de Wasserstein d'ordre 1 entre diagrammes.

***Proposition 6.4* (admis). —** *La distance* $d_{SW} : \mathrm{Dgm} \times \mathrm{Dgm} \to \mathbb{R}_+$ *est CSDN. La fonction* $k_{SW,\sigma} : \mathrm{Dgm} \times \mathrm{Dgm} \to \mathbb{R}$ *définie par :*

$$k_{SW,\sigma}(D, D') = \exp\left(-d_{SW}(D, D')/\sigma\right)$$

*est donc un noyau Gaussien pour les diagrammes de persistance pour tout* $\sigma > 0$.

La preuve de cette proposition est basée sur deux idées principales : d'une part, la norme 1 est CSDN (voir exercice 6.2), et, d'autre part, pour un ensemble fini de diagrammes de persistance, la dépendance de $u_\alpha^{\tilde{D}}$ à l'égard de $D'$ (qui est en soi une obstruction à une conclusion directe via la norme 1, car les vecteurs $u_\alpha^{\tilde{D}}$ et $u_\alpha^{\tilde{D}'}$ ont des tailles qui varient en fonction de la paire de diagrammes considérés) peut être résolue en ajoutant à tout diagramme de l'ensemble les projections sur la diagonale de tous les autres, et en formulant la norme 1 comme la solution d'un problème de transport optimal entre les deux vecteurs, vus comme des mesures unidimensionnelles discrètes. Voir Carrière et al. (2017).

Finalement, une propriété remarquable de la distance de Wasserstein en tranches est qu'elle est *équivalente*, c'est-à-dire non seulement stable, mais aussi *discriminante*, vis-à-vis de la distance de Wasserstein d'ordre 1.

***Théorème 6.4.* —** *Soient* $D, D' \in \mathrm{Dgm}$ *deux diagrammes de persistance de cardinal borné par* $N$, $N \in \mathbb{N}^*$. *Alors, on a :*

$$\frac{d_1(D, D')}{2M} \leq d_{SW}(D, D') \leq 2\sqrt{2} d_1(D, D'),$$

*où* $M = 1 + 2N(2N - 1)$.

*Démonstration.* — Appelons $\gamma_\alpha$ la bijection entre $\tilde{D}$ et $\tilde{D}'$ induite par les vecteurs triés $u_\alpha^{\tilde{D}}$ et $u_\alpha^{\tilde{D}'}$. Appelons de plus $\gamma$ la bijection entre $\tilde{D}$ et $\tilde{D}'$ qui réalise $d_1(D, D')$.

*Borne supérieure.* Soit $\alpha \in [-\pi/2, \pi/2]$. Rappelons que $\|e_\alpha\|_2 = 1$. On a :

$$\begin{aligned}\|u_\alpha^{\tilde{D}} - u_\alpha^{\tilde{D}'}\|_1 &= \sum_{p\in\tilde{D}} |\langle p - \gamma_\alpha(p), e_\alpha\rangle| \\ &\leq \sum_{p\in\tilde{D}} |\langle p - \gamma(p), e_\alpha\rangle| \leq \sqrt{2}\sum_{p\in\tilde{D}} \|p - \gamma(p)\|_\infty \\ &\leq 2\sqrt{2}d_1(D, D').\end{aligned}$$

La borne supérieure s'ensuit par linérarité.

*Borne inférieure.* L'idée est d'utiliser le fait que $\gamma_\alpha$ est une fonction constante par morceaux de $\alpha$, et qu'elle a au plus $2 + 2N(2N-1)$ valeurs critiques $\alpha_0, \ldots, \alpha_M$ dans $[-\frac{\pi}{2}, \frac{\pi}{2}]$. En effet, ces valeurs sont obtenues en prenant les angles $\alpha$ tels que $\langle p_1 - p_2, e_\alpha\rangle = 0$ pour toutes les paires $p_1, p_2$ de $\tilde{D}$ ou $\tilde{D}'$. Alors, pour une paire d'angles critiques consécutifs, on a :

$$\begin{aligned}&\int_{\alpha_i}^{\alpha_{i+1}} \sum_{p\in\tilde{D}} |\langle p - \gamma_\alpha(p), e_\alpha\rangle| \mathrm{d}\alpha \\ &= \sum_{p\in\tilde{D}} \|p - \gamma_{\alpha_i}(p)\|_2 \int_{\alpha_i}^{\alpha_{i+1}} |\cos(\angle(p - \gamma_{\alpha_i}(p), e_\alpha))| \mathrm{d}\alpha \\ &\geq \sum_{p\in\tilde{D}} \|p - \gamma_{\alpha_i}(p)\|_2 (\alpha_{i+1} - \alpha_i)^2/2\pi \\ &\geq (\alpha_{i+1} - \alpha_i)^2 d_1(D, D')/2\pi,\end{aligned}$$

où l'inégalité utilisée pour borner inférieurement l'intégrale du cosinus provient de la concavité du cosinus. La borne inférieure désirée résulte finalement de l'inégalité de Cauchy-Schwarz. □

## Références

Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., and Ziegelmeier, L. (2017). Persistence images : a stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8) :1–35.

Bubenik, P. (2015). Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(3) :77–102.

Carrière, M., Cuturi, M., and Oudot, S. (2017). Sliced Wasserstein kernel for persistence diagrams. In *34th International Conference on Machine Learning (ICML 2017)*, volume 70, pages 664–673. PMLR.

Carrière, M., Oudot, S., and Ovsjanikov, M. (2015). Stable topological signatures for points on 3D shapes. *Computer Graphics Forum*, 34(5) :1–12.

Chazal, F., Fasy, B., Lecci, F., Rinaldo, A., and Wasserman, L. (2015). Stochastic convergence of persistence landscapes and silhouettes. *Journal of Computational Geometry*, 6(2) :140–161.

Divol, V. and Lacombe, T. (2020). Understanding the topology and the geometry of the persistence diagram space via optimal partial transport. *Journal of Applied and Computational Topology*, 5 :1–53.

Murphy, K. (2022). *Probabilistic Machine Learning : An introduction*. MIT Press.

Turner, K., Mileyko, Y., Mukherjee, S., and Harer, J. (2014). Fréchet means for distributions of persistence diagrams. *Discrete & Computational Geometry*, 52(1) :44–70.