# Proposition de kernel semi-supervisé, et application au clustering visuel interactif

Pierrick Bruneau, Benoît Otjacques

# Proposition de kernel semi-supervisé, et application au clustering visuel interactif

Pierrick Bruneau*, Benoît Otjacques*

*CRP Gabriel Lippmann - Département Informatique
41 rue du Brill, L-4422 Belvaux (Luxembourg)
bruneau,otjacque@lippmann.lu,
http://www.lippmann.lu

**Résumé.** Cet article décrit une nouvelle procédure de transformation de fonction noyau (dénommée *kernel* ci-après). La procédure vise à incorporer la supervision d'un utilisateur directement dans les valeurs de similarité entre objets. En utilisant ces similarités modifiées, l'ensemble d'objets est projeté en 2D grâce à une ACP à noyaux (Schölkopf et al., 1998) (dénommée transformation *kernel PCA* par la suite). Un compromis est ainsi établi entre les données originales et l'expertise d'un utilisateur, tout en offrant un moyen naturel de visualisation et d'interaction. Ces projections semi-supervisées sont évaluées sur des données réelles et synthétiques, dans un contexte simulant une tâche de clustering visuel interactif. L'action d'un utilisateur est reproduite en sélectionnant aléatoirement un sous-ensemble d'objets étiquetés a priori. Les résultats expérimentaux démontrent l'efficacité de la méthode, un seul élément étiqueté pour chaque classe réelle suffisant à introduire des effets manifestes sur la visualisation.

## 1 Introduction

Le clustering est une tâche cruciale dans le contexte d'une analyse de données visuelle, e.g. en simplifiant la visualisation de jeux de données volumineux (Keim et al., 2008). Le cluster est un objet très parlant visuellement (Ware, 2004), et par conséquent un candidat naturel en tant que point d'entrée d'une analyse de données visuelle. Toutefois, un ensemble de clusters doit au préalable être projeté dans un espace à faible dimension (préférentiellement 2D) pour devenir accessible visuellement. La définition d'un système de clustering visuel n'est donc pas triviale, car les données réelles sont souvent associées à une haute dimensionalité. Dans cet article, nous proposons une nouvelle procédure de construction de kernel, combinant les similarités originales entre éléments avec des étiquettes de classes spécifiées a priori. La projection 2D de ce kernel modifié par une transformation kernel PCA permet alors de combiner de manière consistante la topologie intrinsèque des données avec des contraintes spécifiées par un utilisateur. En traitant ces données projetées, un algorithme de clustering peut ainsi prendre en compte le compromis de manière implicite.

En pratique, les données sans étiquette sont souvent les plus abondantes : l'étiquetage reflète souvent une vérité terrain établie manuellement (e.g. par un expert du domaine), donc coûteuse. Dans ce contexte, une tâche d'apprentissage semi-supervisé peut être comprise de deux manières, non-exclusives mais conceptuellement différentes :

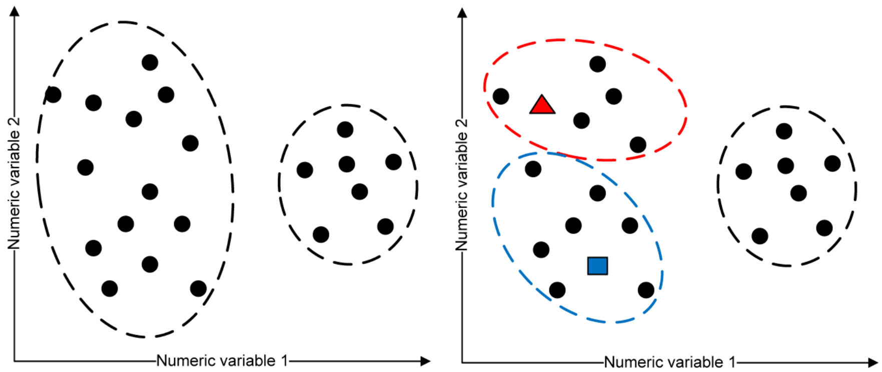


FIG. 1 – À gauche *: résultat potentiel d'un algorithme de clustering.*
À droite *: une autre solution semble privilégiée avec l'ajout de deux exemples étiquetés.*

- comme une tâche d'apprentissage supervisé (i.e. classification) avec un ensemble d'apprentissage de taille très réduite. Cette configuration interdit habituellement l'utilisation de la plupart des algorithmes d'apprentissage supervisé. Cependant, certains auteurs ont proposé d'exploiter la densité des données sans étiquette (i.e. disponibles en grande quantité) pour dépasser cette limitation (Chapelle et al., 2003).
- comme une tâche d'apprentissage non-supervisé aidé par quelques exemples étiquetés, en vue d'incorporer de la connaissance experte (voir figure 1). Cette connaissance peut ne pas être en accord avec la direction imposée par le critère de la procédure d'apprentissage ; le but de la méthode semi-supervisée est alors de traiter ce conflit de manière consistante. Cela peut se faire en utilisant les exemples étiquetés pour l'initialisation du modèle de clustering, et, de manière complémentaire, en forçant leur classe d'appartenance selon cette initialisation tout au long du processus (Basu et al., 2002; Nigam et al., 2006). Dans le contexte des modèles probabilistes, certains auteurs ont transformé un ensemble d'étiquettes en contraintes probabilistes (i.e. *must-link* et *must-not-link*), et ont proposé un algorithme maximisant la vraisemblance de ce modèle (Law et al., 2005).

Les approches de clustering semi-supervisé existantes souffrent des limites suivantes :
- tous les travaux mentionnés ci-dessus se basent sur des transformations linéaires, et des clusters à forme gaussienne, ce qui est parfois trop restrictif dans des situations réelles (e.g. données suivant des variétés non-linéaires, non-gaussiennes),
- certains travaux ont essayé de relâcher l'hypothèse sur la forme des clusters, en autorisant l'association de plusieurs composantes gaussiennes à chaque cluster (Miller et Uyar, 1996). Mais le réglage de l'algorithme résultant s'avère délicat, et semble dépendant du domaine des données.

Notre travail n'entend pas forcer le respect de contraintes de manière explicite, comme cela est fait dans la littérature commentée ci-dessus. Au lieu de cela, nous cherchons plutôt à injecter un compromis dans une projection 2D, qui prenne en compte les similarités originales et un ensemble d'étiquettes spécifiées par un expert. En d'autres termes, notre projection 2D suit la topologie originale des données autant que le permet une information fournie a priori. N'importe quel algorithme de clustering, comme k-means ou EM pour le mélange de gaussiennes (Bishop, 2006) peut ensuite opérer sur ces données numériques continues à faible dimension.

La visualisation de données à grande dimension en utilisant des projections 2D, et les artéfacts de distorsion qui en résultent généralement, sont un sujet d'étude à part entière dans la littérature (Aupetit, 2007), encore actif. Notre contribution peut être vue comme un complément à ce domaine : d'après la terminologie définie dans (Aupetit, 2007), la technique proposée ici pourrait être qualifiée de projection continue non-linéaire.

Dans la section 2, dans un souci d'exhaustivité nous présentons brièvement la transformation kernel PCA, en soulignant l'utilisation de cette technique pour le calcul de projections 2D. Ensuite, dans la section 3, nous proposons une nouvelle procédure de transformation de kernel. Elle permet d'obtenir un compromis entre similarités originales, et étiquetage a priori.

Ce kernel peut s'inscrire dans la tâche de clustering visuel suivante :

1. réaliser une projection 2D avec la transformation kernel PCA, en utilisant notre kernel modifié,
2. effectuer le clustering de ces données projetées.

Dans ce contexte, la semi-supervision serait construite à partir d'interactions avec un utilisateur (e.g. en cliquant et étiquetant des éléments directement sur la visualisation 2D). Dans cet article, l'aspect purement interactif est volontairement laissé de côté, pour se concentrer sur une évaluation expérimentale la plus objective possible du comportement de notre kernel. Nous avons ainsi choisi de le confronter à des sous-ensembles d'éléments étiquetés, sélectionnés aléatoirement dans des jeux de données ayant une vérité terrain connue. Par ce choix, nous entendons identifier les propriétés intrinsèques de notre kernel, et le contraster avec une approche existante. Un kernel classique, non-supervisé, sert de groupe de contrôle pour cette comparaison. Après une discussion critique de nos résultats expérimentaux, nous concluons avec quelques perspectives ouvertes par ce travail dans le domaine de la fouille de données visuelle.

# 2 Projection 2D par transformation kernel PCA

Considérons un ensemble d'éléments $\mathbf{X} = \{\mathbf{x}_i\}_{i\in 1\ldots N}$, prenant ses valeurs dans un domaine $\mathcal{X}$ (appelé *espace original* ci-après), et une transformation non-linéaire $\phi$ projetant un élément $\mathbf{x}_i \in \mathcal{X}$ sur un point $\phi(\mathbf{x}_i) \in \mathbb{R}^M$ (appelé *espace transformé* par la suite). Sous l'hypothèse $\sum_{i=1}^N \phi(\mathbf{x}_i) = \mathbf{0}$, la matrice de covariance empirique de l'image de $\mathbf{X}$ dans l'espace transformé est donnée par :

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^N \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T,$$

les vecteurs propres associés à cette matrice vérifiant alors :

$$\mathbf{C}\mathbf{v}_m = \lambda_m \mathbf{v}_m,\ m = 1\ldots M.$$

Suivant en cela les travaux de (Schölkopf et al., 1998) et (Bishop, 2006), ce calcul peut être transformé en :

$$\mathbf{K}\mathbf{a}_m = \lambda_m N \mathbf{a}_m,\ m = 1\ldots M, \tag{1}$$

avec $k(\mathbf{x},\mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}')$ un kernel, $\mathbf{K}$ la matrice $N\times N$ telle que $\mathbf{K}_{ij} = k(\mathbf{x}_i,\mathbf{x}_j)$ (nommée *matrice de kernel* ci-après), et $\mathbf{a}_m$ un vecteur dans $\mathbb{R}^N$. Après avoir résolu (1), i.e.

trouvé ses vecteurs et valeurs propres, un ensemble de $M$ fonctions de projection peut être défini comme suit :

$$y_m(\mathbf{x}) = \sum_{i=1}^{N} a_{mi} k(\mathbf{x}, \mathbf{x}_i). \tag{2}$$

En supposant les valeurs propres ordonnées de manière décroissante, la projection 2D qui capture le maximum de variance dans l'espace transformé est alors construite avec $y_1$ et $y_2$. L'hypothèse $\sum_{i=1}^{N} \phi(\mathbf{x}_i) = \mathbf{0}$ peut être relâchée en utilisant l'expression modifiée suivante en lieu de matrice du kernel (Bishop, 2006) :

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N\mathbf{K} - \mathbf{K}\mathbf{1}_N + \mathbf{1}_N\mathbf{K}\mathbf{1}_N,$$

avec $\mathbf{1}_N$ la matrice $N \times N$ dont toutes les cellules égalent $\frac{1}{N}$. L'application $\phi$ n'a en général pas à être explicitement définie : en effet, toute matrice semi-définie positive $\mathbf{K}$ a été démontrée comme résultant de produits scalaires dans un espace transformé, possiblement à dimension infinie (Bishop, 2006). Ainsi, en pratique des fonctions kernel sont définies directement, en s'assurant simplement que les matrices de kernel induites sont bien semi-définies positives.

Le kernel gaussien vérifie cette propriété, et est défini comme suit :

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\sigma^2}\right).$$

Remarquons que cette expression requiert un calcul de norme euclidienne, allouant implicitement $\mathbb{R}^d$ à $\mathcal{X}$. Ce kernel a été largement utilisé dans la littérature ; toutefois, expérimentalement nous l'avons trouvé inadapté pour le traitement de données à grande dimension ($d > 100$). Ce problème a déjà été identifié dans la littérature (François et al., 2005). En tant qu'alternative, les auteurs proposent la fonction p-gaussienne :

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{d_{L2}(\mathbf{x}, \mathbf{x}')^p}{\sigma^p}\right), \tag{3}$$

avec $d_{L2}(.,.)$ la distance euclidienne. Les paramètres $p$ et $\sigma$ sont estimés par des formules empiriques, calibrées de sorte que la distribution des valeurs de kernel s'accorde avec celle des distances dans l'espace original, indépendamment de sa dimension :

$$p = \frac{\ln\left(\frac{\ln 0.05}{\ln 0.95}\right)}{\ln \frac{d_{L2}^{95\%}}{d_{L2}^{5\%}}}, \quad \sigma = \frac{d_{L2}^{95\%}}{(-\ln 0.05)^{\frac{1}{p}}} = \frac{d_{L2}^{5\%}}{(-\ln 0.95)^{\frac{1}{p}}}, \tag{4}$$

avec $d_{L2}^{5\%}$ (respectivement $d_{L2}^{95\%}$) le quantile à 5% (respectivement 95%) de la distribution cumulée de $d_{L2}$ [1]. Dans le reste de cet article, le kernel (3) sera utilisée en tant que base non-supervisée.

Sur la figure 2, un échantillon de données généré par 3 gaussiennes 2D se chevauchant partiellement est illustré, dans son espace original, et selon sa projection utilisant les équations (4), (3), and (2). Dans cet exemple, les données semblent “gonflées” par la transformation : la distribution des distances reste semblable après transformation, mais la topologie intrinsèque (i.e. clusters gaussiens) est maintenant exacerbée.

1. Dans le papier référencé, $d_{L2}^{5\%}$ et $d_{L2}^{95\%}$ ont été échangés par erreur dans les expressions de $\sigma$. Une version corrigée est apportée ici.

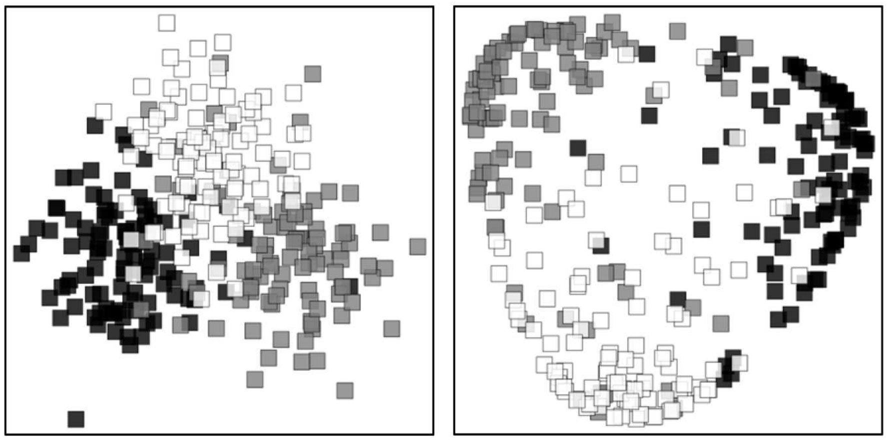

FIG. 2 – À gauche *: Jeu de données 2D original. La composante gaussienne à l'origine de chaque élément est identifiée par une teinte de gris caractéristique.*
À droite *: projection 2D par transformation kernel PCA de ce jeu de données, utilisant la fonction p-gaussienne.*

# 3 Proposition de kernel semi-supervisé

Dans cette section, les valeurs retournées par le kernel sont supposées appartenir à $[0, 1]$. Cette hypothèse est assez conventionnelle (François et al., 2005), et est respectée par la fonction p-gaussienne. Une tâche de clustering revient en partie à affecter des étiquettes (inconnues a priori) à une collection d'éléments. Le but recherché est alors d'obtenir un étiquetage le plus proche possible d'une vérité terrain. Formellement, pour un échantillon de données $\mathbf{X}$ tel que défini dans la section précédente, nous introduisons une fonction d'étiquetage associant chaque élément à une classe parmi $R$ :

$$\begin{aligned} l : \mathbf{X} &\to \{1, \ldots, R\} \\ \mathbf{x} &\to l(\mathbf{x}). \end{aligned}$$

Dans cet article nous supposons un contexte semi-supervisé, i.e. avec un étiquetage potentiellement incomplet : seul un ensemble $\mathbf{X}_L \in \mathbf{X}$ est associé à une étiquette. $l$ ne sera donc utilisée le plus souvent qu'au travers de sa restriction $l' = l|_{\mathbf{X}_L}$. Notons que $l'$ peut définir n'importe quel niveau de supervision, d'une totale absence d'étiquetage (i.e. $\mathbf{X}_L = \emptyset$), à un contexte complètement supervisé (i.e. $\mathbf{X}_L = \mathbf{X}$), en passant par toutes les situations intermédiaires. Notre intuition est de transformer un kernel selon les plus proches voisins étiquetés de ses arguments respectifs. La fonction suivante implémente en partie cette intuition, en retournant le plus proche élément étiqueté de n'importe quel élément dans $\mathbf{X}$ :

$$\begin{aligned} s : \mathbf{X} &\to \mathbf{X}_L \\ \mathbf{x} &\to s(\mathbf{x}) = \begin{cases} \emptyset & \text{if } \mathbf{X}_L = \emptyset \\ \arg\max_{\mathbf{x}' \in \mathbf{X}_L} k(\mathbf{x}, \mathbf{x}') & \text{sinon.} \end{cases} \end{aligned}$$

$l$ et $s$ sont utilisées pour transformer $k$ comme suit :

$$k'(\mathbf{x},\mathbf{x}') = \begin{cases} k(\mathbf{x},\mathbf{x}') & \text{si } |\mathrm{Im}(l')| \leq 1 \\ k(\mathbf{x},\mathbf{x}')^{\frac{1}{\alpha}} & \text{si } |\mathrm{Im}(l')| > 1 \wedge l'\big(s(\mathbf{x})\big) \neq l'\big(s(\mathbf{x}')\big) \\ k(\mathbf{x},\mathbf{x}')^{\alpha} & \text{si } |\mathrm{Im}(l')| > 1 \wedge l'\big(s(\mathbf{x})\big) = l'\big(s(\mathbf{x}')\big), \end{cases} \tag{5}$$

avec $\alpha \in \mathbb{N}^*$, et Im$(l')$ l'image de $l'$ [2]. Intuitivement, avec $k$ retournant des valeurs dans $[0, 1]$ comme requis, et $\alpha > 1$, transformer $k$ en $k'$ revient à augmenter (respectivement diminuer) la similarité entre éléments ayant la même image (respectivement une image différente) par $l' \circ s$, tout en restant dans l'intervalle voulu. Les illustrations et expériences de cet article utilisent la fonction p-gaussienne (voir l'équation (3)), mais remarquons que l'expression (5) pourrait être appliquée à n'importe quel kernel semi-défini positif prenant ses valeurs dans $[0, 1]$, ceci sans perte de généralité. L'inspection stricte des règles de construction de fonctions kernel valides (voir e.g. (Shawe-Taylor et Cristianini, 2004) ou (Bishop, 2006)) semble indiquer que la fonction définie par l'expression (5) (voire même la fonction p-gaussienne) n'est pas un kernel valide (i.e. n'implique pas nécessairement des matrices de kernel semi-définies positives). Toutefois, des fonctions kernel invalides ont déjà été utilisées avec succès dans la littérature (Vapnik, 1995). De plus, dans ce travail nous n'utilisons que les deux premières dimensions de l'espace propre (i.e. la projection 2D), les valeurs propres desquelles ne seraient pas toutes deux réelles et significativement positives que pour des données extrêmement dégénérées.

# 4 Protocole expérimental

## 4.1 Description de la tâche

La procédure de transformation de kernel proposée dans la section précédente est incluse dans la tâche de clustering visuel interactif suivante :

1. une projection 2D initiale (équation (2)) est calculée avec la matrice de kernel p-gaussienne (équation (3)),
2. des étiquettes (i.e. valeurs de classe) sont associées à tous les éléments par un algorithme de clustering,
3. l'utilisateur met à jour ces étiquettes, ainsi que la sémantique associée, selon ses préférences,
4. les étiquettes modifiées par l'utilisateur sont utilisées pour transformer la matrice de kernel initiale (équation (5)),
5. cette nouvelle matrice de kernel est utilisée pour mettre à jour la projection 2D via une transformation kernel PCA,
6. retour à l'étape 2, à moins que l'utilisateur ne soit satisfait de la projection et du clustering courants.

Dans cet article, nous laissons l'aspect purement interactif de côté, pour nous concentrer sur une évaluation approfondie du comportement de notre kernel transformé, en le confrontant à des sous-ensembles aléatoires d'éléments étiquetés a priori. Pour une meilleure estimation

2. La modification du kernel n'est sensée que si l'image de $l'$ contient plus d'une valeur d'étiquette : la condition $\mathbf{X}_L = \emptyset$ n'est donc pas assez forte.

des effets de notre transformation de kernel, nous la contrastons avec les deux alternatives suivantes :

– la matrice de kernel issue de la fonction p-gaussienne, sans supervision (qui jouera le rôle de groupe de contrôle),
– une méthode de clustering semi-supervisé existante (Basu et al., 2002) transposée sur un kernel. En quelques mots, cette approche revient originellement à contraindre l'appartenance des éléments étiquetés par l'utilisateur, en biaisant ensuite l'algorithme de clustering avec ces affectations statiques. Dans les termes du présent article, nous implémentons ce principe en utilisant l'équation (5) sans la fonction de voisinage, i.e. la valeur de kernel $k(\mathbf{x}, \mathbf{x}')$ est transformée ssi $\mathbf{x}$ et $\mathbf{x}'$ sont tous deux dans $\mathbf{X}_L$.

## 4.2 Mesure de la qualité des résultats

La performance de ces méthodes est évaluée avec les mesures suivantes :

– le nombre de classes inféré par l'algorithme de clustering (**nclass** dans la table 1)
– la pureté des clusters (**purity** dans la table 1).

Ainsi qu'évoqué en introduction, le présent travail est à relater avec la littérature traitant de visualisation et de projections 2D. Dans ce contexte, nous mesurons également les distorsions engendrées par les projections, afin de dénoter des compressions et étirements (**compress** et **stretch** dans la table 1) relatifs aux distances dans l'espace original. Ces mesures de distorsion sont normalisées dans l'intervalle $[0, 1]$, 1 indiquant la distorsion maximale. Des détails au sujet du calcul de ces mesures peuvent être trouvés dans (Aupetit, 2007). En rapport à cette référence, remarquons que contrairement à ce qui y est préconisé en cas de données à grande dimension, nous n'avons pas employé de mesures basées sur le rang : nous avons en effet déjà traité ce problème en employant un kernel y étant peu sensible.

## 4.3 Jeux de données choisis, et utilisation

Un jeu de données synthétique et deux jeux de données réels issus du dépôt UCI ont été utilisés pour nos expériences. Celles-ci ont été implémentées avec R.

– **Gaussian** : 3000 points générés selon 3 gaussiennes 2D se recouvrant partiellement. 1000 éléments sont échantillonnés selon chaque composante. Un sous-échantillon de ce jeu de données a déjà été illustré sur la figure 2.
– **Pima** : ce jeu de données a été établi à partir d'enregistrements médicaux de patients venant de la tribu Indienne Pima. Il est défini sur 8 variables numériques, et une variable de classe binaire (i.e. présence ou absence du diabète). Il contient 500 exemples négatifs, et 268 exemples positifs.
– **Isolet** : ce jeu de données a été créé à partir d'enregistrements audio de personnes prononçant des lettres isolées. Chaque enregistrement est décrit par 617 variables numériques. Nous avons extrait les enregistrements de voyelles : cela revient donc à considérer 5 classes, avec 300 exemples dans chacune d'entre elles.

Chaque expérience consiste tout d'abord à tirer un sous-échantillon, sans remplacement, dans un de ces jeux de données. 100 éléments sont pris dans chaque classe (exception faite des exemples négatifs de *Pima*, parmi lesquels nous tirons 200 éléments, ceci afin de reproduire au mieux l'équilibre du jeu de données original). La vérité terrain est ignorée pour tous les exemples du sous-échantillon, sauf pour un nombre donné $n_{\text{lab}}$ d'entre eux pour chaque classe, ceci afin de simuler l'interaction avec l'utilisateur. Une expérience est paramétrée par $\alpha$ (voir équation (5)), et $n_{\text{lab}}$. Nous autorisons $\alpha \in \{2, 3, 5, 10\}$, et $n_{\text{lab}} \in \{1, 2, 5, 10\}$. Notons que le pourcentage de supervision associé prend ses valeurs dans $[1\%, 10\%]$.

Une expérience est aussi paramétrée par une transformation de kernel, parmi :

- **unsupervised** : la fonction p-gaussienne, sans supervision,
- **simple** : l'approche semi-supervisée de référence (Basu et al., 2002),
- **neighbors** : notre approche de kernel semi-supervisé sensible au voisinage étiqueté (équation (5)).

Les mesures de compression et d'étirement sont calculées pour chaque expérience. Afin de ne produire qu'une mesure pour chaque expérience, nous conservons la médiane des mesures de compression (respectivement étirement) spécifiques à chaque expérience. Les données projetées sont ensuite fournies à un algorithme de clustering, de manière non-supervisée. Nous utilisons l'algorithme EM bayésien implémenté dans le package VBmix (Bruneau, 2012) pour obtenir un mélange de gausiennes représentant notre clustering. Le nombre de composantes trouvé a posteriori sert d'estimateur pour notre mesure de qualité basée sur le nombre de classes. Le mélange de gaussiennes est utilisé pour inférer l'étiquette de chaque élément, et leur comparaison à la vérité terrain sert à estimer la pureté des clusters obtenus. In fine, une condition expérimentale est caractérisée par un tuple (jeu de données, transformation, $\alpha$, $n_{\text{lab}}$). Pour chaque condition, nous réalisons 20 expériences. L'algorithme de clustering utilisé est connu pour souffrir de problèmes de minima locaux. Pour contourner ce problème, 10 exécutions en sont réalisées, et un critère pseudo-BIC permet de sélectionner le meilleur parmi ce pool.

# 5 Résultats et discussion

Un test ANOVA *three-way independent* est effectué sur les résultats de nos expériences. Les trois variables indépendantes identifiées sont ordonnées comme suit : *transformation*, $\alpha$, et $n_{\text{lab}}$. Pour *transformation*, nous définissons le contraste de *contrôle* entre *unsupervised* et les méthodes semi-supervisées, ainsi que le contraste *expérimental* entre *simple* et *neighbors*. Un contraste polynomial est appliqué à $\alpha$ and $n_{\text{lab}}$. Beaucoup de conditions expérimentales sont associées à des distributions de mesures pour lesquelles l'hypothèse gaussienne est inacceptable, ou provoquent l'échec du test de Levene pour l'homogénéité de la variance. Toutefois, nous avons choisi d'effectuer le même nombre d'expériences (i.e. 20) dans toutes les conditions expérimentales, ce qui assure la robustesse du test ANOVA (Donaldson, 1968; Lunney, 1970).

Le test a été exécuté indépendamment pour chaque jeu de données et mesure de qualité : les résultats en sont résumés dans la table 1. Les conclusions suivantes peuvent en être tirées :

- L'influence de la transformation *simple* sur la topologie des données projetées est généralement insignifiante.
- L'analyse des mesures de distorsion montre que la proposition de kernel semi-supervisé entraîne des modifications drastiques de la projection 2D obtenue. Cette influence est très perceptible avec ne serait-ce qu'un seul élément étiqueté par classe, et un supplément d'effet modéré pour de plus grands échantillons étiquetés (voir la figure 3). Cette propriété est plutôt conforme à ce qu'un utilisateur est susceptible d'attendre, en rendant ses actions rapidement tangibles.
- Les distorsions sont très fortement influencées par la variation de $\alpha$. Même pour une valeur faible, les artéfacts de projection inhérents à la fonction p-gaussienne sont soit

| | |
|---|---|
| **compress** | – Le contraste expérimental est très significatif ($p < 10^{-10}$, avec $p < 0.01$ seulement pour *Isolet*).<br>– $\alpha$ induit très significativement une tendance linéaire ($p < 10^{-10}$).<br>– $n_{\text{lab}}$ induit plus faiblement une tendance linéaire ($p \simeq 10^{-3}$).<br>– Ces tendances sur $\alpha$ et $n_{\text{lab}}$ interagissent presque exclusivement avec le contraste expérimental. |
| **stretch** | – Le contraste expérimental est très significatif ($p < 10^{-10}$).<br>– $\alpha$ induit très significativement une tendance linéaire ($p < 10^{-10}$).<br>– $n_{\text{lab}}$ induit plus faiblement une tendance linéaire ($p \simeq 10^{-3}$), plus fortement avec *Gaussian* ($p < 10^{-10}$).<br>– Ces tendances sur $\alpha$ et $n_{\text{lab}}$ interagissent presque exclusivement avec le contraste expérimental. |
| **purity** | – Les deux contrastes sur la transformation sont significatifs ($p \simeq 10^{-2}$, seulement le contraste expérimental pour *Isolet*).<br>– $\alpha$ induit modérément une tendance linéaire ($p < 10^{-3}$).<br>– $n_{\text{lab}}$ induit significativement une tendance linéaire ($p < 10^{-5}$), plus faiblement pour *Pima* ($p < 0.1$).<br>– Selon le jeu de données, il peut y avoir une interaction entre une tendance linéaire sur $\alpha$ et le contraste de contrôle ($p < 0.1$ pour tous les jeux de données), ou le contraste expérimental ($p < 10^{-5}$ pour *Gaussian* et *Isolet*).<br>– Excepté avec *Pima* (interaction faible, $p < 0.1$), une forte interaction entre le contraste expérimental et une tendance linéaire selon $n_{\text{lab}}$ a été relevée ($p < 10^{-6}$). |
| **nclass** | – Le contraste expérimental est très significatif ($p < 10^{-10}$). Le contraste de controle est plus faiblement significatif, et seulement avec *Gaussian* et *Isolet* ($p < 0.1$).<br>– $\alpha$ induit très significativement une tendance linéaire ($p < 10^{-10}$), plus modérément pour *Isolet*.<br>– $n_{\text{lab}}$ n'a pas d'influence si considéré isolément.<br>– Une interaction significative entre le contraste expérimental et une tendance linéaire selon $\alpha$ a été relevée ($p < 10^{-3}$). |

TAB. 1 – *Résultats des tests ANOVA, agrégés par mesure de qualité.*

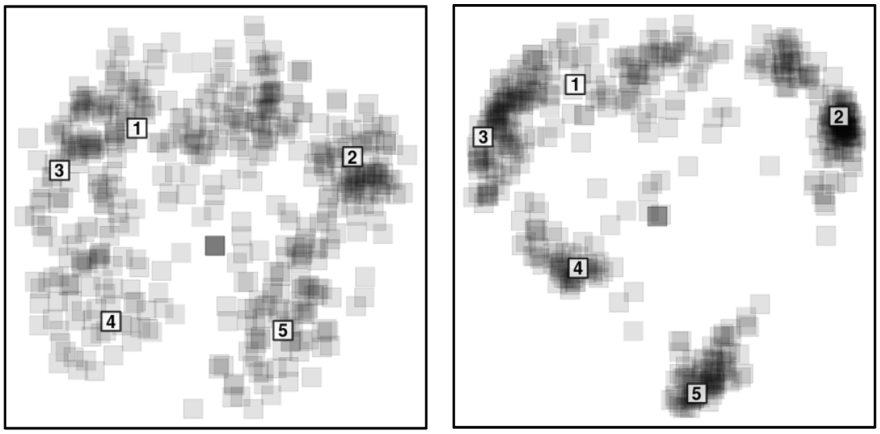


FIG. 3 – À gauche *: projection d'un sous-ensemble d'*Isolet *avec la transformation* unsupervised*. La teinte de gris indique la densité de données dans la projection, et un élément de chaque classe est surligné avec un chiffre distinctif.*
À droite *: projection du même sous-ensemble avec la transformation* neighbors*, utilisant l'ensemble surligné en tant que $\boldsymbol{X}_L$, et $\alpha = 3$ (voir équation* (5)*). Ceci accentue la séparation de l'échantillon en groupes, d'où résulte une plus grande compression de ses éléments par la projection.*

allégés (i.e. dans le cas de l'étirement), soit accentués (i.e. dans le cas de la compression). Cette tendance suit fortement une tendance linéaire, ce qui souligne davantage le rôle essentiel de $\alpha$ en tant que paramètre ajustable.
– Une augmentation de $\alpha$ tend à diminuer le nombre de clusters inférés, avec parfois des conséquences négatives en termes de pureté de clusters. Remarquons tout de même que de manière mécanique, une meilleure pureté est plus facile à obtenir en utilisant un plus grand nombre de clusters. De manière plus générale, une amélioration de la pureté devrait raisonnablement être attendue avec une méthode semi-supervisée : toutefois, la sélection aléatoire des éléments étiquetés de nos expériences a considérablement dégradé les performances de notre méthode de ce point de vue.
– Utiliser davantage d'éléments étiquetés a une influence visible seulement pour la transformation *neighbors*. Cette influence est plutôt négative sur la pureté des clusters pour des valeurs faibles de $n_{\text{lab}}$ : ce handicap est toutefois rattrapé et dépassé rapidement avec l'augmentation du nombre d'éléments étiquetés. Cela confirme accessoirement l'influence tangible qu'auraient des interactions avec l'utilisateur en employant notre kernel semi-supervisé.

# 6 Conclusion

Dans cet article, une nouvelle transformation de kernel semi-supervisée a été décrite et évaluée. Celle-ci est essentiellement basée sur le voisinage des données étiquetées. Comme nos expériences le montrent, très peu d'éléments étiquetés a priori sont suffisants pour influencer fortement une projection kernel PCA subséquente, tout en préservant la topologie originale des données. Cette réactivité permet de minimiser le nombre d'interactions avec l'utilisateur, tout en lui fournissant un retour tangible dans le contexte d'une visualisation. La séparation en clusters est également renforcée par notre méthode, une caractéristique plutôt intéressante pour faciliter la caractérisation visuelle de ces objets.

Les expériences ont également permis de montrer l'importance du paramètre ajustable $\alpha$ dans le contexte de notre kernel semi-supervisé. Son augmentation tend a diminuer de nombre de clusters estimés sur les données projetées, avec une augmentation linéaire des artéfacts de projection en contrepartie, et pas d'avantage apparent sur le plan de la pureté des clusters. Aucun paramétrage n'est donc optimal de manière évidente : dans un contexte interactif, il est donc préférable de proposer une valeur intermédiaire par défaut (e.g. 3), tout en permettant à l'utilisateur de l'ajuster ensuite selon sa préférence. Nous avons démontré la grande sensibilité de notre proposition aux éléments étiquetés a priori. Cette propriété a un revers : si un seul élément par classe est fourni, et est également mal choisi (e.g. outlier de sa classe), notre proposition a des conséquences plutôt négatives sur la pureté des clusters estimés par la suite. Toutefois, l'augmentation du nombre d'éléments étiquetés compense rapidement ce handicap initial.

Au travers de ce travail, nous avons voulu décrire et évaluer en détail notre méthode de projection semi-supervisée. Nous voulons inscrire celle-ci dans la construction d'un système de clustering visuel interactif, mais nous avons d'abord voulu étudier ses propriétés indépendamment de considérations liées aux interactions avec un utilisateur. Nous avons cependant tracé les contours d'une potentielle implémentation de ce système dans la section 4.1. Dans ce contexte, la vérité terrain serait l'expertise que l'utilisateur possède sur les données, et la performance de la méthode serait idéalement mesurée selon sa faculté à s'approcher efficacement d'un clustering respectant au mieux la vérité terrain spécifique de l'utilisateur. L'idée générale derrière un tel système serait de permettre à un utilisateur d'étiqueter les éléments de manière interactive, directement au travers de la projection 2D, puis d'adapter celle-ci de manière dynamique à ces actions. Au-delà de la définition de cinématiques adéquates pour ces interactions, nous tenons à souligner que notre travail n'est pas adaptable tel quel à ce contexte interactif. En effet, chaque interaction transforme la matrice de kernel de manière non-linéaire. En considérant une approche naïve, le calcul de la projection modifiée requiert $O(N^3)$ opérations en pratique. Des optimisations sont déjà possibles du fait que seules les deux premiers vecteurs et valeurs propres sont nécessaires ; mais il doit exister un algorithme (ou au moins une heuristique) permettant de mettre à jour *en ligne* la projection définie par l'équation (2) selon le différentiel de chaque interaction, minimisant ainsi le coût calculatoire.

# Références


Aupetit, M. (2007). Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 1304–1330.

Basu, S., A. Banerjee, et R. Mooney (2002). Semi-supervised clustering by seeding. *Proceedings of 19th International Conference on Machine Learning*.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bruneau, P. (2012). VBmix : a R package for Variational-Bayes mixture learning. Technical report, LINA (CNRS UMR 6241).

Chapelle, O., J. Weston, et B. Schölkopf (2003). Cluster kernels for semi-supervised learning. *Advances in Neural Information Processing Systems 15*, 585–592.

Donaldson, T. S. (1968). Robustness of the F-test to errors of both kinds and the correlation between the numerator and denominator of the F-ratio. *Journal of the American Statistical Association*, 660–676.

François, D., V. Wertz, et M. Verleysen (2005). About the locality of kernels in high-dimensional spaces. *International Symposium on Applied Stochastic Models and Data Analysis*, 238–245.

Keim, D., G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, et G. Melançon (2008). Visual analytics : Definition, process, and challenges. In *Information Visualization*, pp. 154–175. Springer.

Law, M. H. C., A. Topchy, et A. K. Jain (2005). Model-based clustering with probabilistic constraints. *Proceedings of SIAM Data Mining*.

Lunney, G. H. (1970). Using analysis of variance with a dichotomous dependent variable : An empirical study. *Journal of Educational Measurement*, 263–269.

Miller, D. J. et D. J. Uyar (1996). A mixture of experts classifier with learning based on both labelled and unlabelled data. *Advances in Neural Information Processing Systems 9*, 571–577.

Nigam, K., A. McCallum, et T. Mitchell (2006). Semi-supervised text classification using EM. *In Semi-Supervised Learning (Chapelle, Schölkopf and Zien eds.)*.

Schölkopf, B., A. Smola, et K.-R. Müller (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 1299–1319.

Shawe-Taylor, J. et N. Cristianini (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer.

Ware, C. (2004). *Information Visualization : Perception for Design*. Elsevier.

## Summary

In this paper, a new kernel transformation procedure is described. It aims at incorporating a degree of supervision directly in the original pairwise similarities of a data set. The modified similarities can then be projected using a 2D kernel PCA (Schölkopf et al., 1998), so as to reflect the compromise between genuine data and user knowledge, while being affordable for visualization and interaction. Such semi-supervised projections are evaluated with synthetic and real data, in the context of a simulated visual clustering task. Randomly selected subsets of elements are chosen to hold a label, thus reproducing actual user interactions. The results show the effectiveness of the method, with as few as one labelled element per class inducing tangible effects.