
PROJET 1

JEAN-THOMAS BAILLARGEON
CHRISTOPHER BLIER-WONG
POUR LE COURS STT-7330
MÉTHODE D'ANALYSE DES DONNÉES

PRÉSENTÉ LE 25 MARS 2018 À LA PROFESSEURE

ANNE-SOPHIE CHAREST

*Département de mathématiques et de statistiques
Faculté des sciences et de génie
Université Laval*



UNIVERSITÉ
LAVAL

JEAN-THOMAS BAILLARGEON
CHRISTOPHER BLIER-WONG
FACULTÉ DES SCIENCES ET DE GÉNIE
ÉCOLE D'ACTUARIAT
UNIVERSITÉ LAVAL
HIVER 2018

Table des matières

1	Introduction	2
2	Fonction noyau	2
3	PCA avec noyau	3
3.1	Normalisation des données	3
4	Application pratique	4
5	Conclusion	4
6	Annexe : Preuve du PCA avec noyau	4

1 Introduction

L'objectif de l'analyse par composantes principales est d'obtenir une représentation des données dans un espace plus restreint en conservant la plus grande quantité d'information possible. Plus précisément, on considère les combinaisons linéaires des variables mesurées pour l'espace restreint. Par contre, on observe souvent que la relation entre les données n'est pas linéaire. L'analyse par composantes par noyau généralise ce modèle pour des combinaisons non-linéaires des attributs. Au lieu de faire une décomposition par valeurs et vecteurs propres sur la matrice de covariance des données centrées

$$\Sigma = \text{Var}(X) = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T,$$

on fait une décomposition par valeurs et vecteurs propres sur la matrice de covariance des données projetées sur un nouvel espace d'attributs.

2 Fonction noyau

L'idée de l'espace des attributs est de projeter les données originales par une fonction non linéaire vers un nouvel espace. Formellement, on a

$$\begin{aligned} \Phi : \mathbb{R}^N &\rightarrow \mathcal{F} \\ \mathbf{x} &\mapsto \Phi(\mathbf{x}) \end{aligned}$$

où les données $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^N$ est projeté vers un espace d'attributs \mathcal{F} [Muller et al., 2001]. Souvent, la dimension de \mathcal{F} est beaucoup plus élevée que l'espace originale. L'apprentissage statistique peut maintenant être fait sur les données $(\Phi(\mathbf{x}_1), y_1), (\Phi(\mathbf{x}_2), y_2), \dots, (\Phi(\mathbf{x}_n), y_n)$.

Le produit scalaire entre deux espaces d'attributs peut être reformulé en terme d'une fonction noyau k par

$$k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})).$$

Dans plusieurs problèmes d'apprentissage, le "truc du noyau" permet d'éviter de calculer directement les nouvelles données $\Phi(\mathbf{x}_n)$. En effet, pour certains algorithmes d'apprentissage, il n'est pas nécessaire d'avoir toutes les données car on peut reformuler les équations de mise à jour par le produit scalaire entre différentes données et ainsi les remplacer par la fonction de noyau. Des exemples de noyaux communément utilisés sont présentés dans la table 1.

Nom	$k(\mathbf{x}, \mathbf{y})$
Gaussien (RBF)	$\exp\left(\frac{-\ \mathbf{x}-\mathbf{y}\ ^2}{c}\right)$
Polynomial	$((\mathbf{x} \cdot \mathbf{y} + \theta))^d$
Sigmodoidal	$\tanh(\kappa(\mathbf{x} \cdot \mathbf{y}) + \theta)$
Multiquadrique inversé	$\frac{1}{\sqrt{\ \mathbf{x}-\mathbf{y}\ ^2 + c^2}}$

TABLE 1 – Noyaux communs

3 PCA avec noyau

Dans le contexte de l'analyse par composantes principales linéaire, on trouvait les valeurs et vecteurs propres qui correspondaient à la matrice de covariance

$$\text{Var}(X) = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T.$$

[Schölkopf et al., 1997] proposent de répéter cette analyse sur une transformation des données originales, i.e. trouver les valeurs et vecteurs propres de

$$\text{Cov}(\Phi(X)) = \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T. \quad (3.1)$$

Ensuite, en appliquant le "truc du noyau" présenté dans la section (2), on évite de calculer explicitement les données $\Phi(\mathbf{x})$, il suffit de calculer la matrice des noyaux.

3.1 Normalisation des données

Dans le contexte de l'analyse par composantes principales, on conseil souvent de centrer et réduire les données. Par contre, dans le contexte de projection des données dans l'espace \mathcal{F} et profiter du truc du noyau, on ne peut pas calculer $\tilde{\Phi}(\mathbf{x}_i) = \Phi(\mathbf{x}_i) - \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_i)$. Soit la matrice K , où

$$K_{ij} = (k(\mathbf{x}_i, \mathbf{x}_j))_{ij}.$$

Il est possible que $\frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}) \neq 0$. On doit alors centrer les données selon la passe du p'tit cochon qui tousse suivante :

$$\begin{aligned} \tilde{k}(\mathbf{x}_i, \mathbf{x}_j) &= \tilde{\Phi}(\mathbf{x}_i)^T \tilde{\Phi}(\mathbf{x}_j) \\ &= \left(\Phi(\mathbf{x}_i) - \frac{1}{N} \sum_{l=1}^N \Phi(\mathbf{x}_l) \right)^T \left(\Phi(\mathbf{x}_j) - \frac{1}{N} \sum_{l=1}^N \Phi(\mathbf{x}_l) \right) \\ &= k(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{N} \sum_{l=1}^N k(\mathbf{x}_i, \mathbf{x}_l) - \frac{1}{N} \sum_{l=1}^N k(\mathbf{x}_j, \mathbf{x}_l) + \frac{1}{N^2} \sum_{l,k} k(\mathbf{x}_l, \mathbf{x}_k). \end{aligned}$$

Alors, on peut centrer la matrice \tilde{K} selon

$$\tilde{K}_{ij} = k - \mathbb{1}_N K - K \mathbb{1}_N + \mathbb{1}_N K \mathbb{1}_N,$$

où $(\mathbb{1}_N)_{ij} := \frac{1}{N}$.

4 Application pratique

5 Conclusion

6 Annexe : Preuve du PCA avec noyau

On s'inspire de la preuve en [Schölkopf et al., 1997] pour dériver les équations de l'ACP avec noyau. Soit les données $\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_N)$ et que ces données sont centrées, i.e. que $\sum_{i=1}^N \Phi(\mathbf{x}_i) = 0$. Soit la matrice de covariance des données transformées sur le nouvel espace

$$\overline{C} = \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T. \quad (6.1)$$

On doit trouver les valeurs propres λ et les vecteurs propres V qui satisfait

$$\lambda V = \overline{C} V.$$

On peut présenter les vecteurs propres selon une combinaison linéaire des attributs

$$V = \sum_{i=1}^N \alpha_i \Phi(\mathbf{x}_i).$$

On obtient ensuite

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T \left(\sum_{j=1}^N \alpha_{ij} \Phi(\mathbf{x}_i) \right) &= \lambda_i \sum_{j=1}^N \alpha_{ij} \Phi(\mathbf{x}_j) \\ \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_i) \left(\sum_{j=1}^N \alpha_{ij} k(\mathbf{x}_i, \mathbf{x}_j) \right) &= \lambda_i \sum_{j=1}^N \alpha_{ij} \Phi(\mathbf{x}_j). \end{aligned}$$

Ensuite, en multipliant les deux côtés de l'équation par $\Phi(\mathbf{x}_l)$, on obtient

$$\begin{aligned}
\frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{x}_l)^T \Phi(\mathbf{x}_i) \left(\sum_{j=1}^N \alpha_{ij} k(\mathbf{x}_i, \mathbf{x}_j) \right) &= \lambda_i \sum_{j=1}^N \alpha_{ij} \Phi(\mathbf{x}_l)^T \Phi(\mathbf{x}_j) \\
\frac{1}{N} \sum_{i=1}^N k(\mathbf{x}_i, \mathbf{x}_j)^T \left(\sum_{j=1}^N \alpha_{ij} k(\mathbf{x}_i, \mathbf{x}_j) \right) &= \lambda_i \sum_{j=1}^N \alpha_{ij} k(\mathbf{x}_i, \mathbf{x}_j)^T \\
K^2 \alpha_i &= N \lambda_i K \alpha_i \\
K \alpha_i &= N \lambda_i \alpha_i.
\end{aligned}$$

Avec la condition que $v_j^T v_j = 1$, on a

$$\sum_{i=1}^N \sum_{l=1}^N \alpha_{jl} \alpha_{ji} \Phi(\mathbf{x}_l)^T \Phi(\mathbf{x}_i) = 1,$$

qui devient

$$\alpha_j^T K \alpha_j = 1.$$

En appliquant la condition précédente et en multipliant par α_j , on obtient

$$\lambda_j N \alpha_i^T \alpha_i = 1.$$

Pour obtenir la projection d'un nouveau point, on applique

$$\Phi(\mathbf{x})^T v_j = \sum_{i=1}^N \alpha_{ji} \Phi(\mathbf{x})^T \Phi(\mathbf{x}_i) = \sum_{i=1}^N \alpha_{ji} k(\mathbf{x}, \mathbf{x}_i).$$

Références

- [Muller et al., 2001] Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., and Schölkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, 12(2) :181–201.
- [Schölkopf et al., 1997] Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer.