Références / page pub

- **Introduction au Machine Learning**, Chloé-Agathe Azencott, Dunod InfoSup
  Version électronique (PDF) sans exercices disponible gratuitement sur
  http://cazencott.info [lien cliquable]
  (nouvelle édition en préparation)

- **Le Machine Learning avec scikit-learn**, Aurélien Géron, Dunod InfoSup

- Textes et vidéos sur **OpenClassrooms :**
  Textes accessibles librement ; vidéos accessibles avec un compte gratuit.
  - Parcours Ingénieur Machine Learning [lien cliquable]
  - Objectif IA (grand public) [lien cliquable]

Ressources complémentaires

- **Sources de jeux de données**
  - Kaggle [lien cliquable]
  - Le module `datasets` de `scikit-learn` [lien cliquable]
  - Le UCI Machine Learning Repository [lien cliquable]
- **MOOC scikit-learn** [lien cliquable]

Cours 4 – Méthodes à noyaux

1. Noyaux

1.1 Produit scalaire

- $\langle \vec{x}, \vec{x}' \rangle = \sum_{j=1}^{p} x_j x'_j$
- $\langle \vec{x}, \vec{x}' \rangle = ||\vec{x}||_2 \, ||\vec{x}'||_2 \cos\theta \qquad ||\vec{x}||_2^2 = \langle \vec{x}, \vec{x} \rangle$
- Interprétable comme **mesure de similarité**
- **Forme définie positive bilinéaire :**
  - $\langle \vec{x}, \vec{x}' \rangle = \langle \vec{x}', \vec{x} \rangle$ pour tout $\vec{x}, \vec{x}' \in \mathcal{X}$
  - $\langle \vec{x} + \vec{z}, \vec{x}' \rangle = \langle \vec{x}, \vec{x}' \rangle + \langle \vec{z}, \vec{x}' \rangle$ pour tout $\vec{x}, \vec{x}', \vec{z} \in \mathcal{X}$
  - $\langle a\vec{x}, \vec{x}' \rangle = a \langle \vec{x}, \vec{x}' \rangle$ pour tout $a \in \mathbb{R}$
  - $\langle \vec{x}, \vec{x} \rangle \geq 0$ et $\langle \vec{x}, \vec{x} \rangle = 0$ ssi $\vec{x} = 0$
- Apparaît dans de nombreux algorithmes d'apprentissage.

1.2 Noyau

- **Généralisation** du produit scalaire : $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$
  - **sémantiquement** une similarité
  - **mathématiquement** une forme définie positive
- Aronszajn: Si k est semi-définie positive[1], il existe un espace de Hilbert $\mathcal{H}$ et une application $\varphi : \mathcal{X} \rightarrow \mathcal{H}$ telle que $k(\vec{x}, \vec{x}') = \langle \varphi(\vec{x}), \varphi(\vec{x}') \rangle_{\mathcal{H}}$ pour tout $\vec{x}, \vec{x}', \vec{z} \in \mathcal{X}$

[1] Pour tout $m \in \mathbb{N}^*$, pour tout $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_m\} \in \mathcal{X}$, la matrice $K \in \mathbb{R}^{m \times m}$ telle que $K_{il} = k(\vec{x}_i, \vec{x}_l)$ est semi-définie positive

1.3 Astuce du noyau

$k(\vec{x}, \vec{x}') = \langle \varphi(\vec{x}), \varphi(\vec{x}') \rangle_{\mathcal{H}}$

- Si un algorithme ne fait intervenir les éléments de $\mathcal{X}$ que dans des produits scalaires, **remplacer ces produits scalaires** par k est équivalent à appliquer l'algorithme dans $\mathcal{H}$ après avoir appliqué φ
- Utile **si k est plus simple à calculer que φ**

1.3 Astuce du noyau

$$k(\vec{x}, \vec{x}') = \langle \varphi(\vec{x}), \varphi(\vec{x}') \rangle_{\mathcal{H}}$$

- Si un algorithme ne fait intervenir les éléments de $\mathcal{X}$ que dans des produits scalaires, **remplacer ces produits scalaires** par k est équivalent à appliquer l'algorithme dans $\mathcal{H}$ après avoir appliqué φ

- Utile **si k est plus simple à calculer que φ**

- Exemple : **noyau quadratique** $k : (\vec{x}, \vec{x}') \mapsto (1 + \langle \vec{x}, \vec{x}' \rangle)^2$

  Équivaut à $\varphi : (x_1, x_2, \dots, x_p) \mapsto$
  $(1, x_1, \dots, x_p, x_1^2, x_2^2, \dots, x_p^2, \sqrt{2}x_1x_2, \dots, \sqrt{2}x_{p-1}x_p)$.

1.4 Régression ridge à noyau

- **Régression ridge :** $\arg\min_{\vec{\beta} \in \mathbb{R}^{p+1}} \frac{1}{n} \left(\vec{y} - X\vec{\beta}\right)^\top \left(\vec{y} - X\vec{\beta}\right) + \lambda \left|\left|\vec{\beta}\right|\right|_2$
  - Solution : $\vec{\beta}^* = \left(X^\top X + \lambda I_p\right)^{-1} X^\top \vec{y}$
  - Modèle : $f : \vec{x} \mapsto \langle \vec{\beta}^*, \vec{x} \rangle$
- Reformulation : $f : \vec{x} \mapsto \vec{x} X^\top \left(\lambda I_n + X X^\top\right)^{-1} \vec{y}$

1.4 Régression ridge à noyau

- **Régression ridge :** $\arg\min_{\vec{\beta}\in\mathbb{R}^{p+1}} \frac{1}{n}\left(\vec{y} - X\vec{\beta}\right)^\top \left(\vec{y} - X\vec{\beta}\right) + \lambda \left|\left|\vec{\beta}\right|\right|_2$
  - Solution : $\vec{\beta}^* = \left(X^\top X + \lambda I_p\right)^{-1} X^\top \vec{y}$
  - Modèle : $f : \vec{x} \mapsto \langle \vec{\beta}^*, \vec{x}\rangle$
- Reformulation : $f : \vec{x} \mapsto \vec{x}X^\top \left(\lambda I_n + XX^\top\right)^{-1} \vec{y}$
  - $\vec{x}X^\top \in \mathbb{R}^n$ a pour i-ème entrée : $\langle \vec{x}, \vec{x}_i\rangle$
  - $XX^\top \in \mathbb{R}^{n\times n}$ a pour entrée (i, l) : $\langle \vec{x}_i, \vec{x}_l\rangle$

1.4 Régression ridge à noyau

- **Régression ridge :** $\arg\min_{\vec{\beta} \in \mathbb{R}^{p+1}} \frac{1}{n} \left(\vec{y} - X\vec{\beta}\right)^\top \left(\vec{y} - X\vec{\beta}\right) + \lambda \left|\left|\vec{\beta}\right|\right|_2$
  - Solution : $\vec{\beta}^* = \left(X^\top X + \lambda I_p\right)^{-1} X^\top \vec{y}$
  - Modèle : $f : \vec{x} \mapsto \langle \vec{\beta}^*, \vec{x} \rangle$
- Reformulation : $f : \vec{x} \mapsto \vec{x} X^\top \left(\lambda I_n + X X^\top\right)^{-1} \vec{y}$
  - $\vec{x} X^\top \in \mathbb{R}^n$ a pour i-ème entrée : $\langle \vec{x}, \vec{x}_i \rangle$
  - $X X^\top \in \mathbb{R}^{n \times n}$ a pour entrée (i, l) : $\langle \vec{x}_i, \vec{x}_l \rangle$
- **Kernelization :** remplacer $\vec{x} X^\top$ par $\kappa \in \mathbb{R}^n$ tel que $\kappa_i = k(\vec{x}, \vec{x}_i)$ et $X X^\top$ par $K \in \mathbb{R}^{n \times n}$ telle que $K_{il} = k(\vec{x}_i, \vec{x}_l)$

  équivaut à transformer les données par φ puis apprendre une régression ridge, **pour environ le même coût algorithmique**

1.5 Exemples de noyaux

– **Noyau quadratique** $k : (\vec{x}, \vec{x}') \mapsto (1 + \langle \vec{x}, \vec{x}' \rangle)^2$

φ calcule tous les monômes de degré au plus 2 de $x_1, x_2, \dots, x_p$

1.5 Exemples de noyaux

– **Noyau quadratique** $k : (\vec{x}, \vec{x}') \mapsto (1 + \langle \vec{x}, \vec{x}' \rangle)^2$

φ calcule tous les monômes de degré au plus 2 de $x_1, x_2, \dots, x_p$

– **Noyau polynomial** $k : (\vec{x}, \vec{x}') \mapsto (1 + \langle \vec{x}, \vec{x}' \rangle)^d$

φ calcule tous les monômes de degré au plus d de $x_1, x_2, \dots, x_p$

1.5 Exemples de noyaux

- **Noyau quadratique** $k : (\vec{x}, \vec{x}') \mapsto (1 + \langle \vec{x}, \vec{x}' \rangle)^2$

  φ calcule tous les monômes de degré au plus 2 de $x_1, x_2, \ldots, x_p$

- **Noyau polynomial** $k : (\vec{x}, \vec{x}') \mapsto (1 + \langle \vec{x}, \vec{x}' \rangle)^d$

  φ calcule tous les monômes de degré au plus d de $x_1, x_2, \ldots, x_p$

- **Noyau RBF gaussien** $k : (\vec{x}, \vec{x}') \mapsto \exp\left(-\frac{||\vec{x}-\vec{x}'||_2^2}{\sigma^2}\right)$

  φ calcule tous les monômes de $x_1, x_2, \ldots, x_p$ et $\mathcal{H}$ est de dimension infinie.

1.5 Exemples de noyaux

- **Noyau pour chaîne de caractères** $k : (\vec{x}, \vec{x}') \mapsto \sum_{u \in \mathcal{A}^m} \psi_u(\vec{x})\psi_u(\vec{x}')$
  - $\mathcal{A}^m$ = l'ensemble des chaînes de m caractères sur l'alphabet $\mathcal{A}$
  - $\psi_u(\vec{x}) = 1$ si u est une sous-chaîne de $\vec{x}$ et 0 sinon.

1.5 Exemples de noyaux

- **Noyau pour chaîne de caractères** $k : (\vec{x}, \vec{x}') \mapsto \sum_{u \in \mathcal{A}^m} \psi_u(\vec{x})\psi_u(\vec{x}')$
  - $\mathcal{A}^m$ = l'ensemble des chaînes de m caractères sur l'alphabet $\mathcal{A}$
  - $\psi_u(\vec{x}) = 1$ si u est une sous-chaîne de $\vec{x}$ et 0 sinon.
- $k(\vec{x}, \vec{x}')$ est le nombre de chaînes de m caractères communes à $\vec{x}$ et $\vec{x}'$ et peut être calculé en $\mathcal{O}(|\vec{x}|)$ au lieu de $\mathcal{O}(|\mathcal{A}|^m)$.

1.5 Exemples de noyaux

- **Noyau pour chaîne de caractères** $k : (\vec{x}, \vec{x}') \mapsto \sum_{u \in \mathcal{A}^m} \psi_u(\vec{x})\psi_u(\vec{x}')$
  - $\mathcal{A}^m$ = l'ensemble des chaînes de m caractères sur l'alphabet $\mathcal{A}$
  - $\psi_u(\vec{x}) = 1$ si u est une sous-chaîne de $\vec{x}$ et 0 sinon.
- $k(\vec{x}, \vec{x}')$ est le nombre de chaînes de m caractères communes à $\vec{x}$ et $\vec{x}'$ et peut être calculé en $\mathcal{O}(|\vec{x}|)$ au lieu de $\mathcal{O}(|\mathcal{A}|^m)$.
- Si $m = 8$, $|\mathcal{A}| = 20$ et en moyenne $|\vec{x}| = 485$, on compare 25,6 milliards d'opérations à moins de 500.

2. Machines à vecteurs de support

2.1 Formulation primale

- Classification binaire, $\mathcal{Y} = \{-1, 1\}$

$$\underset{\vec{\beta} \in \mathbb{R}^p, b \in \mathbb{R}}{\arg\min} \frac{1}{2} \left|\left|\vec{\beta}\right|\right|_2^* + C \sum_{i=1}^{n} \max\left(0, 1 - y_i \left(\langle \vec{\beta}, \vec{x}_i \rangle + b\right)\right)$$

2.2 Formulation duale

$$\arg\max_{\vec{\alpha}\in\mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{l=1}^{n} \alpha_i \alpha_l y_i y_l \langle \vec{x}_i, \vec{x}_l \rangle$$

sous les contraintes $\sum_{i=1}^{n} \alpha_i y_i = 0$ et $0 \leq \alpha_i \leq C$

– Modèle : $f : \vec{x} \mapsto \sum_{i=1}^{n} \alpha_i y_i \langle \vec{x}_i, \vec{x} \rangle$

2.3 SVR

- Perte ε-insensible :

$$L(y, f(\vec{x})) = \max(0, |f(\vec{x}) - y| - \varepsilon)$$