MAP553 Apprentissage Statistique

PC9 : convexification, SVM, boosting

1 Support Vector Machine (SVM)

Considérons $R > 0$, $\mathcal{W}$ un RKHS sur $\mathcal{X}$ de noyau k et $\mathcal{F} = \{f \in \mathcal{W} : |f|_{\mathcal{W}} \leq R\}$. Nous allons étudier le classifieur $\hat{h}_{\varphi,\mathcal{F}}(x) = \mathsf{signe}(\hat{f}_{\varphi,\mathcal{F}}(x))$ où $\hat{f}_{\varphi,\mathcal{F}}$ est obtenu en résolvant le problème de minimisation convexe

$$\hat{f}_{\varphi,\mathcal{F}} = \operatorname*{argmin}_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \varphi(-y_i f(x_i))$$

avec $\varphi(x) = (1+x)_+$ la perte *hinge*. Ce problème est équivalent au problème suivant

$$\hat{f}_{\varphi,\mathcal{F}} = \operatorname*{argmin}_{f\in\mathcal{W}} \left\{ \frac{1}{n}\sum_{i=1}^{n}(1 - y_i f(x_i))_+ + \lambda |f|^2_{\mathcal{W}} \right\}$$

pour un choix adapté de $\lambda > 0$.

1. Montrer que $\hat{f}_{\varphi,\mathcal{F}}$ appartient à l'espace $V = \mathsf{Vect}\{k(x_i,.) : i = 1,\dots,n\}$.
2. Montrer que $\hat{f}_{\varphi,\mathcal{F}}(x) = \sum_{j=1}^{n} \hat{\beta}_j k(x_j, x)$ où $\hat{\beta} = [\hat{\beta}_1,\dots,\hat{\beta}_n]^T$ est solution de

   $$\hat{\beta} = \operatorname*{argmin}_{\beta\in\mathbb{R}^n} \left\{ \frac{1}{n}\sum_{i=1}^{n}(1 - y_i(K\beta)_i)_+ + \lambda \beta^T K \beta \right\}$$

   avec K la matrice $K = [k(x_i,x_j)]_{i,j=1,\dots,n}$.
3. Montrer que ce problème est équivalent au problème de minimisation

   $$\hat{\beta} = \operatorname*{argmin}_{\substack{\beta,\xi\in\mathbb{R}^n \text{ tels que} \\ y_i(K\beta)_i \geq 1-\xi_i \\ \xi_i \geq 0}} \left\{ \frac{1}{n}\sum_{i=1}^{n}\xi_i + \lambda\beta^T K\beta \right\}$$

4. Montrer que les conditions de Kuhn-Tucker de ce problème donnent $\hat{\beta}_i = y_i\hat{\alpha}_i/(2\lambda)$, pour $i = 1,\dots,n$ avec les $\hat{\alpha}_i$ vérifiant $\min(\hat{\alpha}_i, y_i(K\hat{\beta})_i - (1-\xi_i)) = 0$ et $\min(1/n - \hat{\alpha}_i, \xi_i) = 0$.
5. Montrer qu'on a les propriétés suivantes
   - si $y_i\hat{f}_{\varphi,\mathcal{F}}(x_i) > 1$ alors $\hat{\beta}_i = 0$
   - si $y_i\hat{f}_{\varphi,\mathcal{F}}(x_i) < 1$ alors $\hat{\beta}_i = y_i/(2\lambda n)$
   - si $y_i\hat{f}_{\varphi,\mathcal{F}}(x_i) = 1$ alors $0 \leq \hat{\beta}_i y_i \leq 1/(2\lambda n)$
6. Interpréter géométriquement ce résultat dans le cas où k est le noyau linéaire sur $\mathbb{R}^d$.
7. Quelle est l'interprétation géométrique dans le cas général ?
8. A partir de la propriété de dualité forte, montrer que les $\hat{\alpha}_i$ sont solutions du problème dual

   $$\hat{\alpha} = \operatorname*{argmax}_{0\leq\alpha_i\leq 1/n} \left\{ \sum_{i=1}^{n}\alpha_i - \frac{1}{4\lambda}\sum_{i,j=1}^{n} K_{i,j} y_i y_j \alpha_i \alpha_j \right\}.$$

2 AdaBoost

Soit n points labellés $(x_i, y_i)_{i=1,\dots,n} \in \mathcal{X}^n \times \{-1,+1\}^n$, une famille $\mathcal{H} = \{h_1, \dots, h_M\}$ de classifieurs à valeurs dans $\{-1,+1\}$ et

$$\mathcal{F} = \left\{ f = \sum_{j=1}^{M} \theta_j h_j : \theta \in \mathbb{R}^M \right\}.$$

AdaBoost est un algorithme calculant une solution approximative du classifieur

$$\hat{h}_{\varphi,\mathcal{F}} = \mathsf{signe}(\hat{f}_{\varphi,\mathcal{F}}) \quad \text{avec} \quad \hat{f}_{\varphi,\mathcal{F}} = \operatorname*{argmin}_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \varphi(-y_i f(x_i)) \quad \text{et} \quad \varphi(x) = \exp(x).$$

La solution approchée $\hat{f}_M$ de $\hat{f}_{\varphi,\mathcal{F}}$ est calculée à l'aide de l'algorithme suivant :

Initialisation : $\hat{f}_0 = 0$
Itérer : For $m = 1, \dots, M$

$$\hat{f}_m = \hat{f}_{m-1} + \beta_m h_{j_m} \quad \text{où} \quad (\beta_m, h_{j_m}) = \operatorname*{argmin}_{\substack{h\in\mathcal{H}\\ \beta\in\mathbb{R}}} \frac{1}{n}\sum_{i=1}^{n} \varphi\left(-y_i\big(\hat{f}_{m-1}(x_i) + \beta h(x_i)\big)\right).$$

On supposera dans la suite que pour chaque $h \in \mathcal{H}$ il existe i tel que $y_i \neq h(x_i)$.

1. En posant $w_i^{(m)} = n^{-1}\exp(-y_i \hat{f}_{m-1}(x_i))$, montrer que

$$\frac{1}{n}\sum_{i=1}^{n} \varphi\left(-y_i\big(\hat{f}_{m-1}(x_i) + \beta h(x_i)\big)\right) = (e^{\beta} - e^{-\beta}) \sum_{i=1}^{n} w_i^{(m)} \mathbf{1}_{h(x_i)\neq y_i} + e^{-\beta}\sum_{i=1}^{n} w_i^{(m)}.$$

2. En déduire que

$$h_{j_m} = \operatorname*{argmin}_{h\in\mathcal{H}} \sum_{i=1}^{n} w_i^{(m)} \mathbf{1}_{h(x_i)\neq y_i}.$$

3. Montrer que

$$\beta_m = \frac{1}{2}\log\left(\frac{1-\mathrm{err}_m(h_{j_m})}{\mathrm{err}_m(h_{j_m})}\right) \quad \text{où} \quad \mathrm{err}_m(h) = \frac{\sum_{i=1}^{n} w_i^{(m)} \mathbf{1}_{h(x_i)\neq y_i}}{\sum_{i=1}^{n} w_i^{(m)}}.$$

4. En remarquant que $-y_i h(x_i) = 2\mathbf{1}_{y_i \neq h(x_i)} - 1$ montrer que $\hat{f}_M$ peut être obtenu avec l'algorithme suivant :

AdaBoost

Initialisation : $w_i^{(1)} = 1/n$, pour $i = 1, \dots, n$
Itérer : For $m = 1, \dots, M$

$$\begin{aligned} h_{j_m} &= \operatorname*{argmin}_{h\in\mathcal{H}} \mathrm{err}_m(h) \\ 2\beta_m &= \log(1-\mathrm{err}_m(h_{j_m})) - \log(\mathrm{err}_m(h_{j_m})) \\ w_i^{(m+1)} &= w_i^{(m)} \exp(2\beta_m \mathbf{1}_{h_{j_m}(x_i)\neq y_i}), \quad i = 1, \dots, n \end{aligned}$$

Résultat : $\hat{f}_M(x) = \sum_{m=1}^{M} \beta_m h_{j_m}(x)$.