TP/TD 9 : MÉTHODES À NOYAUX : APPLICATION À LA RECONNAISSANCE FACIALE

COURS D'APPRENTISSAGE, ECOLE NORMALE SUPÉRIEURE, 20 NOVEMBRE 2015

Jean-Baptiste Alayrac
jean-baptiste.alayrac@inria.fr

Résumé. Ce TP a pour but de montrer que les méthodes à noyaux peuvent s'appliquer avec succès à des données réelles. On apprendra ici divers classifieurs en appliquant les méthodes vues en cours et on démontrera leur performance sur la classification de visages.

1. Régression ridge en dimension 1 avec noyau Gaussien

Commencer par considérer le noyau Gaussien en dimension 1 (i.e, $\mathcal{X} = \mathbb{R}$). On rappelle que ce noyau est défini comme $k(x_1, x_2) = \exp(-\frac{(x_1-x_2)^2}{2\sigma^2})$.

1) Vérifier que ce noyau est bien défini positif. On pourra pour cela remarquer que ce noyau correspond au produit scalaire dans un espace de Hilbert $\mathcal{H}$ de dimension infinie (on notera cet espace, qui est, par définition, un RKHS $\mathcal{H}$).

Passons maintenant à quelques rappels sur la régression ridge dans le cadre des méthodes à noyaux. On considère n paires $(x_i, y_i) \in \mathbb{R}^2$. On cherche maintenant à apprendre notre fonction de régression f dans $\mathcal{H}$. On rappelle que le problème de minimisation de la régression ridge à noyaux s'écrit :

$$\min_{f\in\mathcal{H}} \frac{1}{2n}\sum_i (y_i - f(x_i))^2 + \frac{\lambda}{2}\|f\|_{\mathcal{H}}. \tag{1}$$

Grâce au théorème du représentant on rappelle qu'on peut se limiter à chercher f sous la forme $f(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$. On rappelle que, si $K \in \mathbb{R}^n$ désigne la matrice de Gramm des x_i relativement au noyau k, la norme de $f = \sum_{i=1}^n \alpha_i k(., x_i)$ dans le RKHS, se calcule par $\|f\|_{\mathcal{H}} = \alpha^T K\alpha$. On peut donc récrire le problème (1) de cette manière :

$$\min_{\alpha\in\mathbb{R}^n} \frac{1}{2n}\|Y - K\alpha\|_2^2 + \frac{\lambda}{2}\alpha^T K\alpha,$$

où $Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$.

Alors, en annulant le gradient (matriciellement !), on montre qu'une solution de la régression ridge à noyaux peut s'écrire $f(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$ avec :

$$\alpha = (K + n\lambda I)^{-1} y.$$

2) Remarquer que la fonction `sqdist` disponible sur la page web du cours `http://www.di.ens.fr/~slacoste/teaching/apprentissage-fall2014/TP/sqdist.m` permet de calculer en une seule ligne de Matlab le noyau Gaussien. Implémenter la régression ridge à noyau en fixant arbitrairement $\lambda = 10^{-5}$. On pourra considérer le modèle de génération des données suivant : les x_i sont pris suivant une loi uniforme dans $[0, 1]$ et les y_i sont liés aux x_i par la formule suivante : $y_i = f(x_i) + \epsilon$ où ϵ est un bruit Gaussien centré et f une fonction de votre choix.

3) Faire varier la largeur de bande σ et, en séparant données d'apprentissage et données de test, observer les régimes de sur apprentissage et de sous-apprentissage sur ces données.

2. Problème

4) Télécharger les données à l'adresse `http://vis-www.cs.umass.edu/lfw/lfw.tgz` ainsi que le code à remplir à l'adresse `http://www.di.ens.fr/~slacoste/teaching/apprentissage-fall2015/TP/code_TP9.tar`. Vous pouvez explorez le répertoire du jeu de données afin de comprendre son architecture. Afin de vous faciliter le déroulé du TD, vous n'aurez à coder que les parties importantes. Commencez par regarder le code et exécutez le tout début afin d'afficher une célébrité du jeu de données.

Maintenant, on va choisir deux "personnalités" dans la base de données que l'on étudiera dans la suite du TP. Le but sera d'apprendre des classifieurs permettant de discriminer les images des deux personnalités choisies . Parmi les nombreux choix possibles, des individus comme David Beckham, Jacques Chirac, George W. Bush ou Tony Blair sont de bons choix car on dispose de suffisamment d'exemples pour travailler dans de bonnes conditions. Chacune des deux personnalités sera encodée par une sortie $y \in \{-1, 1\}$.

On va d'abord s'intéresser à la manière de représenter une image numériquement pour la tâche qui nous intéresse (on emploie souvent le terme de "features" pour désigner de tels attributs.) De manière simple on va considérer deux types d'attributs pour ces images. Ils ne sont pas forcément les choix les plus adaptés et on préfère souvent en utiliser de plus sophistiqués. Cependant ici le but est de se concentrer plus sur les méthodes de classification plutôt que sur la représentation.

5) Une fois que vous avez choisi vos deux personnalités, complétez la partie du code qui vous permet de créer les matrices de designs. On rappelle que la fonction `imread` sur ces données a pour sortie un tenseur de dimension trois représentant les intensités de couleur en RVB (rouge-vert-bleu). On considèrera ici deux types de "features", la première est l'intensité lumineuse (`illumination` dans le code), elle consiste en la moyenne des valeurs d'intensité RVB. La seconde est l'intensité des couleurs en RVB (`colorIntensity` dans le code).

6) Séparer les données que vous avez choisies en deux échantillons l'un d'entraînement et l'autre de validation/test (voir code pour les détails).

7) Normaliser la matrice de design.

8) On va maintenant implémenter une régression ridge avec un noyau linéaire sur les données en utilisant comme "features" soit l'intensité lumineuse soit les intensités de couleurs. Pour cela on vous demande de tout d'abord calculer le noyau linéaire sur les données, puis implémentez la fonction `getErrors`. La fonction `getErrors` doit vous renvoyer un tableau d'erreur d'entraînement et d'erreur de test pour différentes valeurs de λ (`lambdas`) étant donné des labels `Y`, une matrice de noyau `K`, des indices d'entraînement `ntrain` et de test `ntest` (voir code pour les détails).

9) Représenter en fonction du paramètre de régularisation λ de la régression ridge, les erreurs d'entraînement et de validation/test. On pourra regarder une plage de valeurs logarithmique entre 10^{-10} et 10^{10} (voir fonction `logspace` dans le code). Commentez ce que vous observez.

10) Afficher les exemples mal classés avec le pire score ainsi que les exemples bien classés ayant le meilleur score. Pour cela complétez la fonction `displayHighScores` dans le code. Regarder également les exemples qui sont classés avec une confiance faible en valeur absolue. Pour cela complétez la fonction `displayLowScores` dans le code.

Désormais, on va travailler avec un noyau Gaussien (aussi appelé RBF pour "Radial Basis Function"). On rappelle qu'un noyau Gaussien de paramètre σ est défini par $\forall x, y \in \mathbb{R}^p, K(x,y) = \exp(-\frac{\|x-y\|^2}{2\sigma^2})$.

11) En utilisant les attributs de votre choix (ceux qui vous ont paru les plus adaptés dans la partie précédente), calculer le noyau associé. On pourra fixer le paramètre σ^2 à 10^6 pour commencer. Réutilisez la fonction `getErrors` afin de représenter l'évolution en fonction de λ, paramètre de régularisation, des erreurs d'entraînement et de validation/test.

12) Fixer λ et étudiez l'effet de la variation du paramètre σ du noyau Gaussien.