

Data Challenge Kernel Methods

Samy FERRAT et Nolan ALBANET

Université Paris Dauphine - PSL

samy.ferrat@dauphine.eu & nolanalbanet@sfr.fr

Abstract

Le but du data challenge est d'apprendre comment implémenter des algorithmes d'apprentissage automatique, mieux les comprendre et les adapter à des données réelles. Dans le cadre du projet l'on fait face à une classification d'images avec 10 classes d'objets.

1. Introduction

Le but du Data Challenge est de faire une classification d'images en 10 classes. Les images sont représentées comme des vecteurs de 3072 coordonnées, l'ensemble d'entraînement contient 5000 images tandis que l'ensemble de test 2000 images.

2. Preprocessing

Afin de réduire la dimension des images et de faire de l'extraction de caractéristiques, et d'accélérer l'entraînement nous avons utilisé une version simplifiée de l'algorithme SIFT (Scale-invariant feature transform) sur les images en noir et blanc "moyennées". Dans cette version, un seul feature est extrait en prenant pour origine et taille caractéristique du patch le centre et la largeur de l'image. Cela produit un feature de dimension $num_bins^2 \times num_angles = 300$ (nous avons fait varier les hyperparamètres num_bins et num_angles afin d'obtenir de meilleurs résultats de prédiction).

3. Entraînement du modèle

3.1. Algorithme SVM

Pour l'entraînement nous avons implémenté un SVC, et utilisé les noyaux pour implémenter l'algorithme comme présenté dans le cours, nous avons utilisé la version "one vs all" qui est très populaire pour les problèmes multi-classes car économe en coût et donc en temps de calcul. Pour chaque label, on attribue à l'image un score de compatibilité [1], le label retenu à la fin étant celui amassant le plus gros score.

3.2. Noyau

Nous avons utilisé les noyaux suivants pour le challenge :

- Le noyau linéaire : $K(x, y) = x^T y$
 - Le noyau polynomial : $K(x, y) = (x^T y)^k$ où k est le degré du noyau polynomial.
 - Le noyau RBF : $K(x, y) = \exp(-\frac{1}{2} \|x - y\|^2)$
 - Le noyau chi-2 : $K(x, y) = \exp(-c\chi(x, y))$
- où $\chi(x, y) = \sum_{i=1}^n \frac{(x_i - y_i)^2}{x_i + y_i}$ et c est un hyperparamètre.

4. Resultat

Le noyau linéaire donne de faibles résultats environ 0.23 de précision, le noyau chi-2 quant à lui met beaucoup de temps à tourner, nous avons donc découpé les données et fait l'entraînement avec 1250 images ce qui donne un résultat d'environ 0.38 sur le leaderbord, le noyau RBF semble quant à lui être le bon compromis en effet il nous a donné un score de 0.58600 sur le private leaderbord et de 0.577 sur le public leaderbord tout en faisant tourner l'algorithme en un temps raisonnable contrairement au noyau CHI 2.

Nous avons introduit une seconde couche de neurones en fixant les paramètres de la première couche et en entraînant ceux de la couche suivante par la méthode SVC pour différents types de noyaux à partir du tableau des scores [1]. Les résultats ne sont pas meilleurs qu'avec une simple couche avec noyau RBF.