Morpho Challenge INFMDI341

Jonathan Ohayon

22 juin 2015

Description du challenge

Les données de ce challenge sont fournies par Morpho, une société du groupe Safran partenaire de la chaire Machine Learning for Big Data qui travaille notamment sur l'identification faciale. Le challenge porte sur la reconnaissance de visage : en particulier, le but est de mettre au point un système qui soit capable de prédire si deux images de visage correspondent ou non à la même personne, même si ces personnes n'ont pas été vues en apprentissage. Pour cela, on va utiliser une mesure de distance (ou de similarité) entre les images : si la distance est inférieure à un certain seuil ?, on prédit qu'il s'agit de la même personne. Afin d'obtenir une bonne performance, il est nécessaire d'utiliser des techniques d'apprentissage de métrique (introduites dans le 1er TP) afin d'avoir une mesure de distance mieux adaptée au problème qu'une simple distance Euclidienne par exemple.

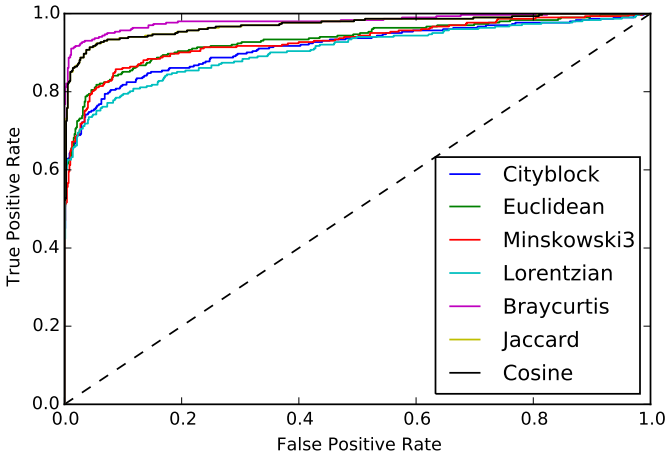
Par souci de confidentialité, Morpho ne fournit pas directement les données sous forme d'images, mais sous la forme de descripteurs extraits depuis les images (la procédure d'extraction des descripteurs n'est pas divulguée).

Les données sont séparées en deux bases : une base facile et une base difficile. Pour ce challenge, il y a aura ainsi un leaderboard pour chaque base, avec un critère de performance spécifique à chaque base (voir plus bas).

Algorithmes et Resultat

Mesure de Similarité

Dans un premier temps, il était intéressant de mesurer les différents scores obtenues par les différentes similarité existante.



Comme nous pouvons le voir, les meilleurs résultats sont obtenus avec la "distance" de Braycurtis. Elle obtient un résultat de 0.21 sur le challenge. J'ai essaye une grande partie des similarité en suivant l'article [Cha07].

Mahalanobis

A la fin du tp1 de machine learning de cette année, il y avait le code de métrique learning avec la distance de Mahalanobis. Pour le faire tourner sur les données du chalenge, j'ai change le nombre d'exemple vu par iteration, et j'ai évité de faire la projection sur le cone des matrice semi-definie positive a chaque iteration pour des raisons de temps de calcul.

LMNN

En suivant les recommendations sur le ipython notebook, j'ai regardé l'algorithme LMNN [WBS05]. Malheureusement sur le facile, l'algorithme marche mal vu qu'il a besoin d'un nombre de classe assez grand. On essaye toujours ici de trouver une distance de Mahalanobis mais nous avons la possibilité de faire la descente avec une matrice de rang plus faible en utilisant la decomposition de $M = L^T L$. J'ai implémenté une version en descente stochastique, sur les données faciles l'algorithmes ne donne pas de résultat très satisfaisant. Voir lmmn_sgd.py . A noter qu'il existe aussi des versions non linéaires de cette algorithme [KTS+12]. J'ai notamment essayé d'implementer la GB-LMNN mais ma version ne marche pas et par manque de temps, j'ai abandonne cette méthode.

- Score Facile = 0.28
- Score Difficile = 0.27

Neural Network

Suite a un workshop de deep learning, j'ai essaye d'implementer un réseau de neurone en utilisant une Amazon Machine Image qui m'a été introduite lors du Workshop. Cette image contient un package python avec sklearn, Theano, itorch, et cuda pour pouvoir effectuer des calculs sur GPU. En suivant l'article [MSMB08], j'ai implémenté un similarity neural network. Comme je n'ai pas réussi a partager les poids entre les unités cachées, j'ai tout simplement pris des entrées qui avaient deja cette symétrie un peu a la façon du RandomForestDistance, les inputs sont $(x + y)$ et $(x - y)$, nous avons donc 3000 entrées et 50 unités cachés sur une couche. Finalement, j'utilise la proba des classes pour définir une dissimilarité un peu de la meme façon que le RandomForestDistance.

- Score Facile = 0.181
- Score Difficile = 0.2501

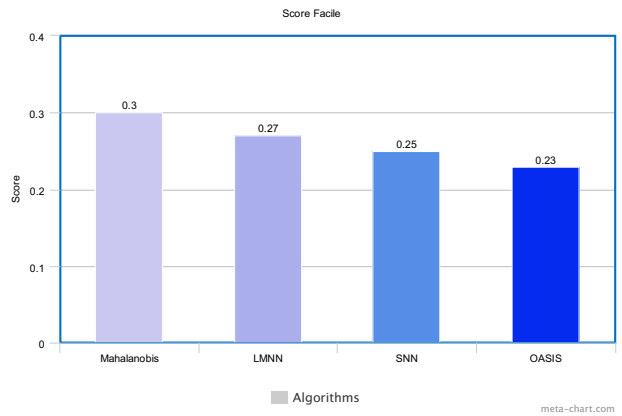
OASIS

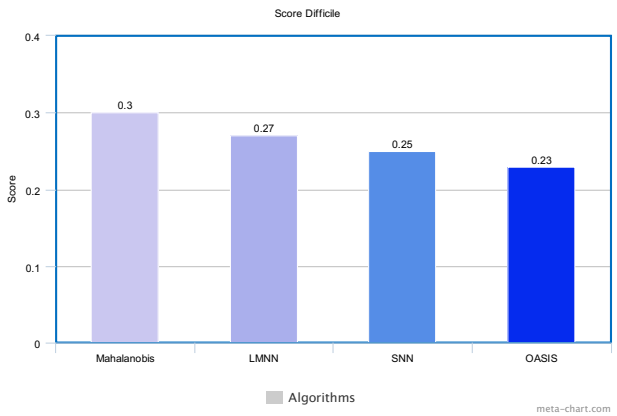
Finalement, je me suis intéressé a OASIS un algorithme de métrique learning en mode online [CSSB10]. Il existe une très bonne implementation de cette algorithme sur github
https ://gist.github.com/gwtaylor/94412c4808421acbd1d0
Le but de cette algorithme est d'optimiser le produit scalaire entre x et y, a l'aide d'une matrice W. cette matrice ne définie pas forcement une pseudo distance. Il convient quand meme de faire une projection sur le cone des matrices semi-definie positive de temps en temps pour éviter un phénomène de sur-apprentissage.
j'ai fait tourner le code sur le cluster AWS que j'avais ouvert pour le réseau de neurone, car les calculs mettaient trop de temps sur mon ordinateur.

Sur le score facile, en utilisant comme similarité la cosine, nous obtenons un score de 0.21 environ mais il s'avère quand utilisant plutôt une distance de Mahalanobis avec la PSD de W et en divisant par la norme L1 de x et y nous obtenons un meilleur score de 0.153. Cependant la base difficile, c'eat la cosine qui l'emporte.

- Score Facile = 0.153
- Score Difficile = 0.230

un petit récapitulatif :





Axe d'amelioration

- Il serait intéressant de regarder un peu plus en details, les réseaux de neurones notamment de modifier les paramètres de pas de patience ainsi que le nombre de d'unites cachées. Il serait aussi intéressant de changer la fonction de cout pour ajuster la precision de l'algorithme.
- J'ai aussi essaye sans succès de faire tourner un Kernel PCA avant les différents algorithmes de métrique learning. Ja n'ai malheureusement pas eu le temps de vraiment valider ces méthodes.
- Le code n'est pas complètement mis au propre. Il serait bien d'arranger les différents algorithmes pour avoir un code plus stable.

Références

[Cha07] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions, 2007.

[CSSB10] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *The Journal of Machine Learning Research*, 11 :1109–1135, 2010.

[KTS+12] Dor Kedem, Stephen Tyree, Fei Sha, Gert R Lanckriet, and Kilian Q Weinberger. Non-linear metric learning. In *Advances in Neural Information Processing Systems*, pages 2573–2581, 2012.

[MSMB08] Stefano Melacci, Lorenzo Sarti, Marco Maggini, and Monica Bianchini. A neural network approach to similarity learning. In *Artificial Neural Networks in Pattern Recognition*, pages 133–136. Springer, 2008.

[WBS05] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005.