Techniques d'apprentissage de fonctions de distance
(Proposition de stage X2005 — 2007)

Frank NIELSEN
(E-mail :Frank.Nielsen@acm.org)

Thématique	algorithmique
Laboratoire	LIX, École Polytechnique, Paris
Durée	4 à 5 mois (Avril 2008 $\sim$, anglais ou français)

Positionnement du stage

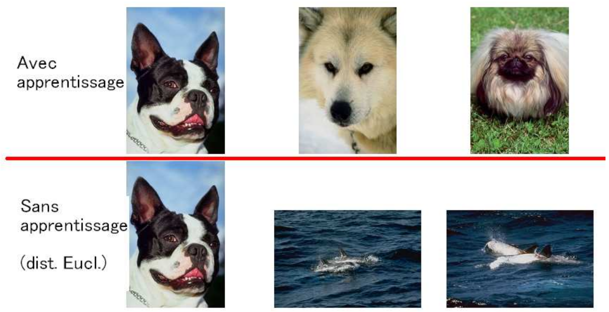
La plupart des problèmes rencontrés dans la vision par ordinateur nécessite de "choisir" une fonction de distance, ou une fonction de perte (loss function) adéquate. Le designer d'algorithmes et/ou programmeur est alors confronté à la situation de définir la distance entre deux vecteurs d'information (ou bien de définir la fonction objective correspondante), et les solutions sont *ad-hoc*, variant de la simple distance Euclidienne familière à une "alchimie" manipulant des puissances de celles-ci (hand-crafting distances). Cela a comme inconvénient de figer, c'est-à-dire de coder en dur, la fonction distance une fois pour toutes. Autrement dit, la fonction distance dépend uniquement de la tâche à accomplir et non pas des entrées possibles.

Toutefois, comme les jeux de données rencontrés en pratique sont de grandes dimensions, les points représentant l'information de l'entrée sont souvent distribués de manière non-uniforme, et il est quasi-impossible (voire inhumain !) de pouvoir deviner "la" meilleure fonction de distance pour résoudre le problème. On souhaiterait plutôt que l'algorithme passe une partie de son temps à apprendre à partir des données la fonction de distance la plus pertinente pour ce jeu (cf. algorithmes adaptatifs auto-améliorants [1]).

L'apprentissage de fonctions de distances est récurrent à tous les domaines de l'informatique : que cela soit sur des problèmes de bioinformatique, ou de traitement d'images, de musiques, etc.

Objectifs du stage

Dans ce stage nous nous proposons d'examiner des classes *paramétriques* de distance élargissant le cadre usuel, et de regarder l'impact en pratique sur quelques problèmes de vision/classification. Un état de l'art résumant bien le principe sur l'apprentissage des distances est décrit dans [2]. Le plus souvent, on considére la distance de Mahalanobis généralisant la distance euclidienne (la matrice Σ étant la matrice identité) : $d(P,Q) = \sqrt{(P-Q)^T \Sigma^{-1} (P-Q)}$. On cherche à élargir ce cadre en regardant tout d'abord les distances de Bregman définies pour des fonctions géneratrices strictement convexes et différentiables F par $d(P,Q) = F(P) - F(Q) - (P-Q)^T \nabla F(Q)$, avec ∇F le gradient de F. Les distances de Bregman unifient de manière élégante la distance euclidienne au carré (ou de Mahalanobis, en prenant $F(X) = X^T \Sigma^{-1} X)$ avec de nombreuses fonctions entropiques comme la divergence de Kullback-Leibler pour $F(X) = \sum_i x_i \log x_i$. De plus, il existe une bijection entre ces fonctions de distances et une grande famille de distributions statistiques usuelles (Gaussiennes, Bernoulli, Poisson, Gamma, etc.), appelée famille exponentielle [3]. On pourra donc aussi apprendre le type de distributions combinatoires à prendre en compte dans les algorithmes de soft clustering de mixtures [3]. L'apprentissage de distance peut être *guidé* également par l'utilisateur en indiquant des couples de vecteurs similaires ou dissimilaires, orientant l'optimisation de la fonction distance.



Ce stage à dominante théorique permettra de se familiariser avec un domaine de recherche d'actualité pour lequel il reste encore des progrès spectaculaires à réaliser. Il pose des questions fondamentales sur la nature d'une distance (axiomatisation débouchant sur la notion de projection/orthogonalité généralisant le théorème de Pythagore, véritable fondateur de la science moderne), mais aussi sur des questions simples restant pourtantes difficiles à résoudre comme par exemple : comment comparer deux distances entre elles (notion d'information mutuelle normalisée, cf. [4]), etc. En guise d'application, on regardera l'apprentissage de distances pour le clustering avec contraintes pour un moteur de recherche d'images semi-supervisé [5] (DistBoost) : chaque utilisateur est capable de définir sa perception de similarité dans les images. Le stage permettra également de se familiariser avec les techniques d'apprentissage et de classification automatique (support vector machines, AdaBoost), et s'appuiera sur des bibliothéques logicielles existantes (eg., Weka[1] pour Java ou d'autres en C++).

Profil/prérequis

Mots clefs: apprentissage par ordinateur, distance.
Outils: Java ou C++ (au choix).

Bibliographie

1. Nir Ailon, Bernard Chazelle, Seshadhri Comandur, Ding Liu: Self-improving algorithms. SODA 2006: 261-270

2. L. Yang and R. Jin. Distance metric learning: A comprehensive survey. Technical report, Department of Computer Science and Engineering, Michigan State University, 2006.

3. Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, Joydeep Ghosh: Clustering with Bregman Divergences. Journal of Machine Learning Research 6: 1705-1749 (2005)

4. J.P. Pluim, J.B.A. Maintz and M.A. Viergever, Mutual information-based registration of medical images: a survey, IEEE Trans. Med. Imaging 22 (2003), pp. 986-1004.

5. Tomer Hertz, Aharon Bar-Hillel, Daphna Weinshall: Learning Distance Functions for Image Retrieval. CVPR (2) 2004: 570-577

[1] `http://www.cs.waikato.ac.nz/ml/weka/` ou `http://www.support-vector-machines.org/SVM_soft.html`