

DIIC - INC, 3^e année
Module RAPP
TP 1 - Classification non supervisée : k -means

Hervé JÉGOU, Guy LORETTE

25 janvier 2006

LE but de ce TP est d'observer le comportement d'un algorithme de classification non supervisée : les k -means. Cet algorithme est utilisé pour séparer les données sources en k classes distinctes. Vous en trouverez une description et une discussion dans le livre de David J.C. MacKay (pages 285–290) : "Information Theory, Inference, and Learning Algorithms". Ce livre est disponible en ligne, sous certaines conditions, à l'URL <http://wol.ra.phy.cam.ac.uk/mackay/itila/book.html>. L'algorithme du k -means y est détaillé p.286.

Dans un premier temps, vous allez implanter cet algorithme et le tester sur un jeu de données fourni sur la page enseignement du TP. Vous pouvez accéder à cette dernière à partir de l'URL <http://www.irisa.fr/temics/Equipe/Jegou/Teaching/>. Les données fournies sont supposées correspondre à une variable temporelle dont on ne connaît pas la nature et qu'on aimerait classer en plusieurs catégories. La seule chose que l'on sait est que cette variable peut prendre $\Omega = 10$ valeurs différentes, de 0 à 9. Remarquez que les échantillons ne sont pas de longueur identique. Ces données sont également fournies dans un fichier de paramètre lisible par matlab ou la librairie libit.

Pour vous faciliter le travail, un squelette de programme vous est fourni. Dans l'état actuel, il lit les $N = 29$ séquences et les affiche à l'écran.

Vous écrirez votre compte-rendu au fur et à mesure du TP et vous le rendrez à la fin des quatre heures.

Étude demandée

L'étude que vous allez mener se déroulera en trois parties :

1. la génération des descripteurs associés aux séquences ;
2. l'implantation de l'algorithme de classification en k classes distinctes ;
3. l'interprétation des résultats et la mesure de leur robustesse aux conditions initiales et aux paramètres de l'algorithme.

Afin de comparer des vecteurs de taille identique, les descripteurs générés sont définis comme les histogrammes associés aux séquences.

Question 1. *Quelle est alors la taille des vecteurs de description ? Pensez-vous que ces descripteurs proposés soient pertinents ? Justifiez votre réponse. Dans tous les cas, proposez une amélioration possible et retenez-la pour la suite.*

Question 2. *Calculez les descripteurs associés à chacune des N séquences.*

Question 3. *Implantez une fonction `nearest_neighbor` qui prend en entrée un ensemble de n vecteurs, un vecteur requête r et un entier m qui permet de sélectionner une norme de Minkowski donnée¹. Cette fonction renvoie l'index du descripteur le plus proche. Vérifiez le bon fonctionnement de cette fonction sur quelques exemples.*

Question 4. *Écrivez une fonction `k_means` qui implémente l'algorithme du k -means. L'entrée de cette fonction est constituée d'un ensemble de N vecteurs, d'une norme m ainsi que d'un paramètre k . Ce dernier fixe le nombre de classes produites par l'algorithme.*

Note : on vous demande de choisir les conditions initiales aléatoirement. Expliquez comment vous faites.

Question 5. *Exécutez votre algorithme en faisant varier les paramètres ($m = 1, 2$ et $k = 2, 3, 4$) et en faisant varier les conditions initiales. Notez les résultats obtenus pour les dix premières séquences (les 29 si vous êtes courageux). Commentaires ?*

Question 6. *Quelle distance vous paraît la mieux adaptée au problème ? Quelle autre mesure de différence entre descripteurs pourrait paraître naturelle dans ce contexte ?*

Question 7. *Proposez un moyen de mesurer la variabilité au sein d'une classe.*

Un point délicat dans l'algorithme du k -means est le choix du nombre de classes.

Question 8. *Proposez une solution (heuristique) qui permet de sélectionner ce nombre k .*

Question 9. *On suppose désormais que chacune des classes représente une chaîne de Markov. Pour chacune des classes, calculez la matrice de probabilités de transition.*

Question 10. *Calculez ensuite la probabilité a posteriori d'apparition de chacune des séquences pour chacun des modèles. Concluez.*

¹On rappelle que si $m = 2$, cette norme correspond à la distance euclidienne ; si $m = 1$, il s'agit de la distance de Manhattan. Cette fonction est calculée par la fonction `vec_distance_norm` dans `libit`.