Prototypes et k plus proches voisins (*kppv (kNN)*)

Eric Gaussier

LIG - MIAI
Université Grenoble Alpes
Eric.Gaussier@imag.fr

Table des matières

Learning Vector Quantization (1)

Algorithme en ligne (*on-line*) dans lequel des prototypes sont placés stratégiquement par rapport aux frontières de classes.

L'idée sous-jacente (due à Kohonen (1989)) est que les exemples d'apprentissage attirent et repoussent les prototypes des mêmes classes et des classes différentes.

Learning Vector Quantization (2)

Algorithme - LVQ1

1. Choisir r prototypes initiaux par classe $(m_1(k), \cdots, m_r(k), 1 \leq k \leq K))$
2. Tirer au hasard, avec replacement, un exemple d'apprentissage $(x^{(i)}, y^{(i)})$; soit $m_j(k)$ le prototype le plus proche de $x^{(i)}$:
   - Si $y^{(i)} = k$, rapprocher le prototype vers l'exemple :
   $$m_j(k) \leftarrow m_j(k) + \epsilon(x^{(i)} - m_j(k))$$
   - Sinon, éloigner le prototype de l'exemple :
   $$m_j(k) \leftarrow m_j(k) - \epsilon(x^{(i)} - m_j(k))$$
3. Répéter l'étape 2 en faisant décroître ϵ vers 0

Démontrer qu'on rapproche (resp. éloigne) bien les prototypes

Learning Vector Quantization (3)

La mise à jour des prototypes dépend de l'exemple considéré. Le tirage aléatoire avec replacement (comment le coder ?) permet d'avoir une version stable de l'algorithme.

Il n'y a pas de fonctions objectifs optimisée dans le cadre de LVQ. Ceci pose problème car on n'a pas de cadre théorique pour étudier le comportement de l'algorithme.

Distinction entre *modélisation de la classe* et *modélisation de la frontière entre classes*. LVQ se situe plutôt dans la deuxième lignée, avec toutefois une certaine généralisation.

Learning Vector Quantization (4)

Illustration

N^o exple	valeur att. 1 (x_1)	valeur att. 2 (x_2)	classe
1	3	3	-
2	4	3.8	-
3	5	3	-
4	6	3.5	-
5	3	4	+
6	4	4.2	+
7	4	6	+
8	6	5	+

On choisit les exemples 2 et 6 comme prototypes initiaux des classes $-$ et $+$ ($r = 1$). $\epsilon = 0.5$ et est divisé par 2 à chaque itération. Dérouler les premières étapes de l'algorithme.

Table des matières

Les k plus proches voisins (*kppv*) (1)

L'idée sous-jacente est de faire reposer la classification d'un exemple sur la base de la classification des exemples proches, ce qui permet une modélisation fine de la frontière entre classes (parfois trop fine) - Illustration au tableau.

C'est une approche dite *memory-based*, ou *fondée sur la mémoire*. On parle aussi de *lazy learning*, c'est-à-dire d'*apprentissage paresseux* car il n'y a pas d'apprentissage proprement dit : les exemples d'apprentissage sont simplement stockés et réutilisés lors de la classification.

Les k plus proches voisins (*kppv*) (2)

Algorithme standard

Soit un objet x à classer.

1. Déterminer $N_k(x)$, l'ensemble des k plus proches voisins de x
2. Choisir la classe de x sur la base d'un vote majoritaire dans $N_k(x)$

Exemple

Données précédentes : $k = 3,\ x^T = (5, 3.5)$

- Quelle est la classe de x ?
- Dessiner la frontière entre classes pour $k = 1$

Les k plus proches voisins (*kppv*) (3)

Améliorations

Le vote majoritaire est parfois remplacé par un vote majoritaire pondéré par la distance (ou similarité) entre exemples/objets/éléments.

Il est possible d'apprendre (malgré tout) certains éléments :

- Seuil sur chaque classe - on utilise $N_k^c(x)$, ensemble des voisins de x dans la classe c et la distance (ou similarité) obtenue sur $N_k^c(x)$
- Métrique complète (exemple avec la distance de Mahalanobis)

Les k plus proches voisins (*kppv*) (4)

Implantation dans le cas de matrices creuses
Utilisation du fichier (ou de la table) inverse
Illustration au tableau

Les k plus proches voisins (*kppv*) (4)

La malédiction des grandes dimensions (*Curse of dimensionality*)

On suppose que l'on dispose uniformément des points dans un hypercube de côté 1 d'un espace vectoriel de dimension p de façon à ce que, pour chaque point, il existe un voisin situé à une distance de 10^{-1}.

Combien faut-il de points pour réaliser cela dans un esapce vectoriel de dimension 1, 2, p ?

Les k plus proches voisins (*kppv*) (4)

La malédiction des grandes dimensions (*Curse of dimensionality*)

- De manière générale, dans un espace vectoriel de grande dimension, la distance entre un point donné x et son point le plus proche dans un ensemble d'apprentissage est comparable (proche) de celle entre x et son point le plus éloigné dans l'ensemble d'apprentissage (malédiction des grandes dimensions).
- En pratique, il n'est pas certain que nous soyons frappé par cette malédiction ! Les techniques de réduction de dimension et de sélection d'attributs permettent de s'en affranchir.