Programmation linéaire pour l'apprentissage par K-plus-proches-voisins

Yuzhen Wang[1,2], Pierre Lemaire[1], Iragaël Joly[1], Nadia Brauner[2]

[1] Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, Grenoble, France
[2] Univ. Grenoble Alpes, INRIAE, Grenoble INP, GAEL, Grenoble, France
prenom.nom@grenoble-inp.fr

Mots-clés : *programmation linéaire, K-NN, distance euclidienne pondérée*

1 Présentation du problème

Un algorithme d'apprentissage supervisé vise à classifier des observations inconnues à l'aide des observations déjà classifiées. L'un des algorithmes les plus connus est celui des plus proches voisins (K-Nearest Neighbors, K-NN) [6], pour lequel la classe d'une nouvelle observation est déterminée comme la classe majoritaire parmi ses k plus proches voisins. La valeur k et la distance utilisée sont des paramètres de l'algorithme. Dans ce qui suit, nous considérons un jeux de données avec N observations et M attributs. Chaque observation X_i ($i = 1..N$) est définie par sa classe y_i et les valeurs des attributs $x_{i,m}$ ($m = 1..M$).

Le plus souvent, la distance utilisée est la distance euclidienne. C'est un choix simple mais rarement optimal, en particulier lorsque les différents attributs n'ont pas le même ordre de grandeur. Il est alors possible d'améliorer les performances de l'algorithme en utilisant d'autres distances. Dans ce travail nous nous intéressons à des distances euclidiennes pondérées, c'est-à-dire que la distance entre deux observations X_1 et X_2 est donnée par $\sum_{m=1}^{M} w_m(x_{1,m} - x_{2,m})^2$. Notre but est de trouver une façon efficace d'attribuer les poids w_m (a priori, $w_m \geq 0$).

Hocke et Martinetz [5] ont proposé un modèle linéaire pour déterminer ces poids. L'idée de leur modèle est de minimiser la plus grande distance au sein d'une même classe (la *distance intra-classe*). Afin de faciliter la formulation, on pré-calcule la matrice $a_{i,j,m} = (x_{i,m} - x_{j,m})^2$. Les variables de décisions sont les poids à déterminer ($w_m \geq 0, m = 1..M$) et la distance intra-classe ($r \geq 0$). On obtient :

$$\begin{array}{lllll} \min & r & & & \\ s.t. & \sum_{m=1}^{M} w_m\, a_{i,j,m} & \geq & 1 & (y_i \neq y_j) \\ & \sum_{m=1}^{M} w_m\, a_{i,j,m} & \leq & r & (y_i = y_j) \end{array} \qquad \begin{array}{lllll} \min & r + C\sum_{i=1}^{N} \xi_i & & & \\ s.t. & \sum_{m=1}^{M} w_m\, a_{i,j,m} & \geq & 1 & (y_i \neq y_j) \\ & \sum_{m=1}^{M} w_m\, a_{i,j,m} & \leq & r + \xi_i & (y_i = y_j) \end{array}$$

Le modèle de droite est une version souple avec des tolérances (variables $\xi_i \geq 0$) qui autorise des erreurs mais les pénalise grâce au paramètre C.

2 Nouveaux modèles linéaires

Dans notre étude, nous avons proposé plusieurs variantes des modèles initiaux, en particulier en ajoutant la notion de *distance inter-classe*, c'est-à-dire la plus petite distance entre deux observations de classes différentes (distance que l'on veut maximiser).

De même les tolérances peuvent être adaptées aux deux types d'erreurs (inter ou intra classes) pour aboutir à différents modèles. Ainsi, le modèle ci-dessous maximise la distance inter-classe et autorise les deux formes d'erreurs :

$$\begin{array}{lllll} \max & r - C_{inter}\sum_{i=1}^{N} \xi_i^{inter} - C_{intra}\sum_{i=1}^{N} \xi_i^{intra} & & & \\ s.t. & \sum_{m=1}^{M} w_m a_{i,j,m} & \geq & r - \xi_i^{inter} & (y_i \neq y_j) \\ & \sum_{m=1}^{M} w_m a_{i,j,m} & \leq & 1 + \xi_i^{intra} & (y_i = y_j) \end{array}$$

3 Analyses et résultats expérimentaux

Au final, nous avons proposé 8 modèles différents (ou 7 car deux peuvent être prouvés équivalents) pour déterminer les poids w_m. L'enjeu est alors dans l'usage de ces différents modèles, selon 2 critères de performance principaux : la qualité de la classification résultante (mesurée par le taux d'erreur de K-NN) et le temps de calcul.

Pour cela nous proposons une étude expérimentale sur des jeux de données de la littérature [1, 2, 3], en nous basant sur les recommandations de Demšar [4] pour les comparaisons.

Les modèles étant similaires, leurs performances restent proches mais avec des différences significatives. Ainsi, certains modèles sont systématiquement dominés par d'autres et peuvent être ignorés. Par contre, les meilleures performances ne sont pas systématiquement obtenues avec les mêmes modèles, il n'y a donc pas de choix de modèle évident. Enfin, les modèles les plus longs à calculer donnent souvent les meilleurs poids, mais au prix de temps de calcul parfois excessifs.

Des analyses complémentaires, en cours, visent à améliorer les performances.

D'une part, déterminer le bon paramétrage (C, C_{inter} et/ou C_{intra} selon les modèles) est fait en explorant systématiquement 11 valeurs ($\{2^{-x}, x = 0..10\}$) proposées par [5]. Il faut donc s'assurer que ces valeurs sont suffisantes pour obtenir des poids de qualité, et nécessaires pour ne pas faire de calculs inutiles. Les premiers résultats tendent à montrer que l'ensemble de valeurs considéré est suffisant, mais non nécessaire, en particulier lorsqu'il s'agit de déterminer un couple (C_{inter}, C_{intra}).

D'autre part, on remarque que les poids de certains attributs sont toujours à 0, et que ces attributs peuvent donc être supprimés. Différentes heuristiques sont envisagées pour procéder à une sélection des attributs afin de diminuer la taille et donc la complexité du problème à traiter.

Au delà des performances en termes de prédiction et temps de calcul, une perspective est de proposer une analyse plus qualitative des résultats, en particulier d'extraire de la connaissance des poids calculés. Par exemple, est-il possible de hiérarchiser les attributs ou au moins distinguer ceux qui sont essentiels de ceux qui sont inutiles, sur la base de ces poids ? Les enjeux de reproductibilité des résultats et de confiance sont alors encore plus sensibles que pour valider la performance globale.

Références

[1] Broad institute - cancer program datasets. `http://portals.broadinstitute.org/cgi-bin/cancer/datasets.cgi`, 2020-10-13.

[2] Kaggle : Your machine learning and data science community. `https://www.kaggle.com/`, 2020-10-13.

[3] Uc irvine machine learning repository. `https://archive.ics.uci.edu/ml/index.php`, 2020-10-13.

[4] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7 :1–30, 2006.

[5] Jens Hocke and Thomas Martinetz. Maximum distance minimization for feature weighting. *Pattern Recognition Letters*, 52 :48–52, 2015.

[6] Ian H Witten and Eibe Frank. Data mining : practical machine learning tools and techniques. *Morgan Kaufmann*, 2005.