Heading for Interpretable Adaptive Nearest Neighbour Classifiers

Frank Klawonn, Katharina Tschumitschew
Department of Computer Science
University of Applied Science
Salzdahlumer Str. 46/48
D-38302 Wolfenbuettel, Germany
{f.klawonn, katharina.tschumitschew}@fh-wolfenbuettel.de

Abstract

Nearest neighbour classifiers are a very simple classification tool. They are interpretable in case-based reasoning sense, i.e. the decision for a classification is carried out on the basis of cases stored in a database. However, although a large data base of sample cases might lead to a good classification performance it does not contribute to the interpretability of the classifier. In this paper we propose an algorithm to reduce the database of cases for a nearest neighbour classifier incorporating an adaptation of the distance measure in terms of scaling. This scaling can be used to construct fuzzy rules to provide a better interpretation of the classifier.

1. Introduction

In supervised learning like classification as well as regression, in the field of data mining and modelling the trade-off between accuracy and interpretability has been discussed widely (see for instance [1]). Although complicated and black box models might lead to a better performance, they might not be interpretable and will not be accepted by a user. Complicated models also tend to overfitting. Therefore, it is desirable, to construct simple and interpretable models in supervised learning.

In this paper we concentrate on nearest neighbour classifiers that solve classification tasks on the basis of a database of classified sample cases. A new data object is classified on the basis of the class(es) of its closest neighbour(s) in the sample database. Although a large database of sample cases might enhance the performance of a nearest neighbour classifier, it leads to higher computational costs, when classifying new data, and it does not contribute to the interpretability of the classifier. Simplicity and interpretability can only be achieved, when the database contains a reasonably small set of representative sample cases or prototypes. Therefore, in this paper we propose a heuristic method to reduce the number of cases in the sample database.

Another problem in the context of nearest neighbour classifiers is the question how to measure the distance or similarity between an object from the sample database and a new object to be classified. Here we propose an adaptive nearest neighbour classifier that uses a scaled distance similar as it is sometimes used in fuzzy systems.

The paper is organized as follows. In section 2 we briefly review the concept of nearest neighbour classifiers. Our algorithm to construct a simplified nearest neighbour classifier is explained in section 3. Section 4 shows the performance of our nearest neighbour classifier on some example data sets. In section 5 we outline the relation to fuzzy classifiers. The final conclusions contain perspectives for future work.

2. Nearest neighbour classifiers

A nearest neighbour classifier performs a classification task, i.e. it defines a mapping from an input space $\mathcal{S}$ to a finite output space $\mathcal{C}$ of classes based on a database $\mathcal{B} \subseteq \mathcal{S} \times \mathcal{C}$ of samples and distance measure $d : \mathcal{S}^2 \rightarrow \Re_0^+$. A k-nearest neighbour classifier (where k is a positive integer) determines the class of an object $s \in \mathcal{S}$ by computing the k closest objects in the sample database (nearest neighbours) and assigning the class to s which occurs most often among the k nearest neighbours. A 1-nearest neighbour classifier simply assigns the class of the closest sample in the database to an object s.

The definition of a suitable distance measure d is crucial for the performance of a nearest neighbour classifier. If the input space consists of a set of p continuous-valued attributes, i.e. $\mathcal{S} \subseteq \Re^p$, then commonly the (squared) Euclidean distance is chosen.

However, better performance can be achieved, when the distance measure is adapted, leading to an adaptive nearest

neighbour classifier. Many of these have been proposed in the literature, like for instance [2] where the distance measure is adapted for each object in the sample database based on a (local) linear discriminant analysis. Although discriminant analysis is a powerful tool for classification, the resulting classifier (or the resulting distance measure in case of our nearest neighbour classifier) is not easy to interpret. Therefore, we propose the following strategy to obtain a simplified nearest neighbour classifier with an adaptive distance.

3. Reducing the number of cases and adapting the distance

For reasons of simplicity, we restrict our considerations to 1-nearest neighbour classifiers. However, the described algorithm can also be applied in the case of arbitrary k-nearest neighbour classifiers.

In the first step we construct a preliminary sample database from a set of available examples. We choose between the following strategies.

(a) When the set of available examples – the training set – is not too large, the whole training set builds the initial sample database. We then reduce the number of objects in our sample database step by step and remove an object, when the misclassification rate does not exceed a specified limit, when removing this object from the database.

(b) In the case of a large training set, we start with an empty sample database and add samples from the training set step by step. We add samples from the training set, until the misclassification error of the sample database is less than a specified limit. As long as we have not reached this limit, in each step we add the data object from the training with which we obtain the lowest misclassification rate. Note that in some cases, this might even lead to a temporary increase of the misclassification rate. However, our experiments have shown that we have to accept this temporary drop of the performance in order to obtain a good classifier in the end.

When no further objects can be removed from the sample database in case (a) or when we finally fall below the acceptable misclassification rate in case (b), we adapt the distance measure of each object in the following way. Here we assume that all attributes are continuous-valued and that we use

$$d(s, s') \;=\; \sum_{i=1}^{p} |s_i - s'_i|,$$

where s' is the object to be classified, s is an object from the sample database, and s_i and s'_i is the ith attribute of s, respectively s'. In order to keep the distance adaptation interpretable, we allow for each object s' in the sample database and for each of its attributes an individual scaling in both directions. This means, for s and each of its attributes, we have two scaling factors $c_s^{(l)}$ and $c_s^{(r)}$. We define the scaled distance of an object s' to s by

$$d_i(s, s') \;=\; \begin{cases} c_s^{(l)} \cdot |s_i - s'_i| & \text{if } s'_i \le s_i \\ c_s^{(r)} \cdot |s_i - s'_i| & \text{if } s'_i \ge s_i \end{cases}$$

The overall scaled distance between s and s' is

$$d(s, s') \;=\; \sum_{i=1}^{p} d_i(s, s').$$

In the beginning all scaling factors are set to the value one. When we have constructed the sample preliminary database according to case (a) or (b), we adapt the scaling factors. For each object that is still in the sample database and for each of its attributes and both directions (associated with the scaling factors $c_s^{(l)}$ and $c_s^{(r)}$, respectively), we check whether

$$c_s^{(l)}(\text{new}) \;=\; \sigma \cdot c_s^{(l)}(\text{old})$$

or

$$c_s^{(l)}(\text{new}) \;=\; \frac{1}{\sigma} \cdot c_s^{(l)}(\text{old})$$

(and analogously for $c_s^{(r)}$) leads to a reduction of the misclassification rate. $0 < \sigma < 1$ is a learning rate. Of course, we could also think about some kind of random or adaptive strategy to find the best scaling factors. But in order to keep the algorithm simple and the computational complexity low, we apply this simple learning strategy.

After the scaling factors have been adapted, we try to further reduce the sample database, i.e. we remove an object from the sample database, if the misclassifiaction rate of the nearest neighbour classifier with the reduced sample database does not exceed the specified misclassification rate. When this is finished, we again adapt the scaling factors and repeat this procedure until we cannot further reduce the sample database.

4. Results

In order to demonstrate the performance of our nearest neighbour classifier, we have carried out experiments with two simple data sets from the UCI Machine Learning Repository (http://www.ics.uci.edu/∼mlearn/MLRepository.html), namely the Iris and the glass data set. We have applied

data set	misclassification rate without scaling	misclassification rate with scaling	no. of selected cases without scaling	no. of selected cases with scaling
iris	1.0	1.7	11.4	5.2
glass	7.9	8.2	67.0	42.9

Table 1. Performance of the classifier

N	instances
1	6.0, 3.4, 4.5, 1.6, Iris Versicolor
2	4.9, 3.0, 1.4, 0.2, Iris Setosa
3	6.3, 2.8, 5.1, 1.5, Iris Virginica
4	6.2, 2.8, 4.8, 1.8, Iris Virginica

Table 2. A sample database for the Iris data set

10-fold (stratified) cross validation. This means, the data set was partitioned into 10 equally sized subsets. The distribution of the classes in each subset is the same as in the whole data set. Then we take out one of the 10 subsets and construct our nearest neighbour classifier based one the remaining 90% of the data. After the classifier has been constructed, we test its performance on the 10% of the data that were left out. This procedure is applied to each of the 10 subsets. Table 1 shows the average performance as well as the average size of the sample database. The table shows that the use of the scaling factors leads to a reduction of the sample database.

Table 2 shows one result for the Iris data set with only four cases in the sample database.

It is interesting to take a closer at the scaling factors for these four cases, shown in tables 3 and 4. It can be seen that some of the attributes do not play an important role for

no. of instance	$Atribut1$		$Atribut2$	
	left	right	left	right
1	1.0	0.003	0.015	0.003
2	0.003	0.003	0.003	0.003
3	0.092	0.303	0.092	0.092
4	0.003	0.003	0.003	0.015

Table 3. Scaling factors for the first two attributes

no. of instance	$Atribut3$		$Atribut4$	
	left	right	left	right
1	0.008	1.0	0.003	0.003
2	0.003	1.0	0.003	0.003
3	6.011	0.092	0.092	0.092
4	0.003	0.003	6.011	0.003

Table 4. Scaling factors for the last two attributes

a sample case. For example, for instance no. 2 the first two attributes have a scaling factor of at least one for both the right and the left-hand side. For most of the attributes, one of the two scaling factors is large. This means that the corresponding class extends only in the direction of the small scaling factor.

5. The relation to fuzzy classifiers

Although looking at the scaling factors provides already some interesting information, it might be desirable, to find a simpler representation for non-expert users. Fuzzy sets and fuzzy rules provide a very intuitive framework that is easily understandable for a non-expert user.

There is a close connection between (triangular) fuzzy sets and metrics [3]. Given a metric $\delta : X \times X \to [0, \infty)$ on a set X, we can construct an equality or similarity relation $E_\delta : X \times X \to [0, 1]$ by

$$E_\delta(x, y) \;=\; 1 - \min\{\delta(x, y), 1\}.$$

with respect to the Łukasiewicz t-norm. A similarity relation E on a set X with respect to the Łukasiewicz t-norm $*$ is a mapping $E : X \times X \to [0, 1]$ satisfying

(E1) $E(x, x) = 1$, (reflexivity)

(E2) $E(x, y) = E(y, x)$, (symmetry)

(E3) $E(x, y) * E(y, z) \leq E(x, z)$. (transitivity)

where the Łukasiewicz t-norm $* : [0, 1] \times [0, 1] \to [0, 1]$ is defined by $\alpha * \beta = \max\{\alpha + \beta - 1, 0\}$. In the context of fuzzy systems, the Łukasiewicz t-norm can be interpreted as the truth function of a conjunction (see for instance [4]). Vice versa, we have that any equality relation E satisfying (E1), (E2) and (E3) induces a (pseudo-)metric δ_E by

$$\delta_E(x, y) \;=\; 1 - E(x, y).$$

Under certain conditions a fuzzy set can be interpreted as the (fuzzy) set of all elements that are similar to a certain

element x_0 with respect to a suitable equality relation. This means, the fuzzy set μ_{x_0} induced by the element x_0 is the (fuzzy) set of all elements that are similar to x_0 with respect to the equality relationj E, i.e.

$$\mu_{x_0}(x) \;=\; E(x_0, x).$$

Viewing these ideas in the context of our nearest neighbour classifier, we can establish the following connection to fuzzy sets. For an object s in the sample database we use the distance function (to an arbitrary object s')

$$d(s,s') \;=\; \sum_{i=1}^{p} d_i(s,s') \;=\; \sum_{i=1}^{p} d_i(s,s')$$

where

$$d_i(s,s') \;=\; \begin{cases} c_s^{(l)} \cdot |s_i - s_i'| & \text{if } s_i' \le s_i \\ c_s^{(r)} \cdot |s_i - s_i'| & \text{if } s_i' \ge s_i. \end{cases}$$

In terms of the object s, we use the (scaled) metric d_i for the ith attribute. Using this metric, we can associate the fuzzy set $\mu_i^{(s)}(x) = |s_i - x|$ with the object s for the ith attribute. Combining these fuzzy sets with the Łukasiewicz t-norm and assuming for the moment that the d_i values are quite small, we obtain the overall similarity degree of a new object s' to the object s from the sample database by

$$E(s,s') \;=\; \sum_{i=1}^{p} (1 - d_i(s_i, s_i')) \;-\; (p-1). \qquad (1)$$

Associating with s the fuzzy sets

$$\mu_i(x) \;=\; 1 - \min\{d_i(s,x), 1\},$$

and using the Łukasiewicz t-norm to combine the fuzzy sets, we can interpret $E(s,s')$ as the firing degree of the fuzzy rule that the object s' is similar to the object s from the sample database. In this sense, instead of assigning to s' the class of the closest object from the sample database.

The only problem that occurs here is that equation (1) applies only in the case, when the values $d_i(s_i, s_i')$ are small enough. This can be avoided, by applying an additional (suitably small) scaling factor to all objects and attributes to guarantee for overall small distances. With this additional scaling factor we obtain a formal one-to-one correspondence between fuzzy rules and nearest neighbour classifiers. However, the additional scaling factor would lead to very wide (triangular) fuzzy sets. Therefore, for the visualisation and user friendly interpretation of the nearest neighbour classifier, we propose not to use the corrected fuzzy sets (incorporating the additional scaling factor), but to use the original ones just taking the individual scaling factors $c_s^{(l)}$ and $c_s^{(r)}$ into account.

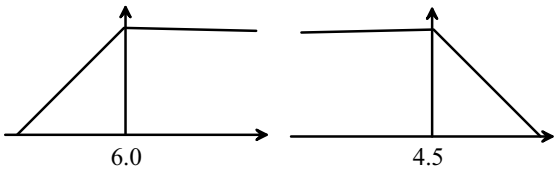

Figure 1. The fuzzy sets corresponding to the first attribute (left) and third (right) attribute of instance no. 1 of table 3

In this way, we can construct suitable fuzzy sets for each object in the sample database and each of its attributes. A fuzzy set for an attribute of an instance is constructed by choosing the membership degree one at the corresponding value of the attribute and decreasing the membership degrees with a slope proportional to the right scaling factor to the right and with a slope proportional to the left scaling factor to the left. An important observation is that most of the scaling factors in tables 3 and 4 are almost zero. This means that the corresponding fuzzy set remains (almost) constantly one in the direction of the corresponding zero scaling factor. If the left and the right scaling factors are both (almost) zero, this means, that the corresponding fuzzy set can be interpreted as "any value". Such fuzzy sets can be left out, when we construct fuzzy rules. In this sense, the first instance of the sample database shown in table 2 induces the fuzzy rule

> If attribute 1 is *at least 6.0 or not much smaller*
> and attribute 3 is *at most 4.5 or not much bigger*,
> then class is Iris Versicolor.

The fuzzy sets corresponding to the expressions *at least 6.0 or not much smaller* and *at most 4.5 or not much bigger* are shown in figure 1. The slopes of the fuzzy sets are determined by the corresponding scaling factor in tables 3 and 4. The fuzzy sets for attributes 2 and 4 are not needed, since all their corresponding scaling factor are almost zero.

The third instance of the sample database shown in table 2 induces the fuzzy rule

> If attribute 3 is
> *at least 5.1 or only very slightly smaller*
> then class is Iris Virginica.

Figure 2 shows the fuzzy set that corresponds to the expression *at least 5.1 or only very slightly smaller*.

6. Conclusions

We have demonstrated, how an interpretable adaptive nearest neighbour classifier can be constructed. Further research will concentrate on more sophisticated methods to

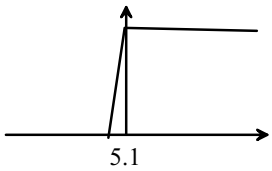

Figure 2. The fuzzy sets corresponding to the third attribute of instance no. 3 of table 3

construct the classifier that might replace our simple approach that is more or less based on a greedy strategy.

References

[1] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, editors. *Interpretability Issues in Fuzzy Modelling*. Springer, Berlin, 2004.

[2] T. Hastie and R. Tibshirani. Discriminant nearest neighbour classification. *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 6:607–616, 1996.

[3] F. Klawonn and R. Kruse. Equality relations as a basis for fuzzy control. *Fuzzy Sets and Systems*, 54:147–156, 1993.

[4] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*. Chichester, 1994.