Weighted k-Nearest-Neighbor Techniques and Ordinal Classification

Klaus Hechenbichler
hechen@stat.uni-muenchen.de
Institut für Statistik, Ludwig-Maximilians-Universität München,
Akademiestraße 1, 80799 München, Germany

Klaus Schliep
k.p.schliep@massey.ac.nz
Allan Wilson Centre for Molecular Ecology and Evolution, Massey University,
Private Bag 11222, Palmerston North, New Zealand

13th October 2004

Abstract

In the field of statistical discrimination k-nearest neighbor classification is a well-known, easy and successful method. In this paper we present an extended version of this technique, where the distances of the nearest neighbors can be taken into account. In this sense there is a close connection to LOESS, a local regression technique. In addition we show possibilities to use nearest neighbor for classification in the case of an ordinal class structure. Empirical studies show the advantages of the new techniques.

1 Introduction

Based on the common nearest neighbor technique for classification we develop a much more flexible tool, that extends the basic method in two directions. First we introduce a weighting scheme for the nearest neighbors according to their similarity to a new observation that has to be classified. Based on the fact, that the voting of nearest neighbors is equivalent to the mode of the class probability distribution, the second extension uses the median or the mean of that distribution, if the target variable shows an ordinal or even higher scale level. A *R* package called *kknn* with implementations for our technique is in preparation and will be published soon.

One special combination of these two extensions, a weighted mean estimation, builds the connection to the local regression technique LOESS and especially to the Nadaraya-Watson estimator. Both are nicely summarized for example in *Chen et al.* (2004) and *Cleveland and Loader* (1995).

After a short description of the common kNN classification method in section 2, we introduce our weighted technique in section 3 and the extension to ordinal target variables in section 4. Then the empirical part in section 5 compares the results of a study with four standard datasets for classification, one large microarray problem and finally one set with ordinal structure in the target variable.

2 k-Nearest-Neighbor Techniques (kNN)

The nearest neighbor method (*Fix and Hodges* (1951), see also *Cover and Hart* (1967)) represents one of the simplest and most intuitive techniques in the field of statistical discrimination. It is a nonparametric method, where a new observation is placed into the class of the observation from the learning set that is closest to the new observation, with respect to the covariates used. The determination of this similarity is based on distance measures.

Formally this simple fact can be described as follows: Let

$$L = \{(y_i, x_i), i = 1, \ldots, n_L\}$$

be a training or learning set of observed data, where $y_i \in \{1, \ldots, c\}$ denotes class membership and the vector $x_i' = (x_{i1}, \ldots, x_{ip})$ represents the predictor values. The determination of the nearest neighbors is based on an arbitrary distance function $d(.,.)$. Then for a new observation (y, x) the nearest neighbor $(y_{(1)}, x_{(1)})$ within the learning set is determined by

$$d(x, x_{(1)}) = \min_i (d(x, x_i))$$

and $\hat{y} = y_{(1)}$, the class of the nearest neighbor, is selected as prediction for y. The notation $x_{(j)}$ and $y_{(j)}$ here describes the jth nearest neighbor of x and its class membership, respectively.

For example, such typical distance functions are the Euclidean distance

$$d(x_i, x_j) = \left(\sum_{s=1}^{p} (x_{is} - x_{js})^2 \right)^{\frac{1}{2}}$$

or the absolute distance

$$d(x_i, x_j) = \sum_{s=1}^{p} |x_{is} - x_{js}| \quad .$$

In general, both measures can be seen as special cases of the so-called Minkowski distance

$$d(x_i, x_j) = \left(\sum_{s=1}^{p} |x_{is} - x_{js}|^q \right)^{\frac{1}{q}} \quad .$$

The Euclidean distance results for the selection $q = 2$, the absolute distance for the parameter value $q = 1$.

The method has been explained by the random occurrence of the learning set, as described in *Fahrmeir et al.* (1996). The class label $y_{(1)}$ of the nearest neighbor $x_{(1)}$ of a new case x is a random variable. So the classification probability of x into class $y_{(1)}$ is $\mathrm{P}(y_{(1)}|x_{(1)})$. For large learning sets x and $x_{(1)}$ coincide very closely with each other, so $\mathrm{P}(y_{(1)}|x_{(1)}) \approx \mathrm{P}(y|x)$ results approximately. Therefore the new observation x is predicted as belonging to the true class y with the probability approximately $\mathrm{P}(y|x)$.

A first extension of this idea, that is widely and commonly used in practice, is the so-called k-nearest neighbor method. Here not only the closest observation within the learning set is referred for classification, but also the k most similar cases. The parameter k has to be selected by the user. Then the decision is in favour of the class label, most of these neighbors belong to.

Let k_r denote the number of observations from the group of the nearest neighbors, that belong to class r:

$$\sum_{r=1}^{c} k_r = k \quad .$$

Then a new observation is predicted into the class l with

$$k_l = \max_r (k_r) \quad .$$

This prevents one singular observation from the learning set deciding about the predicted class. The degree of locality of this technique is determined by the parameter k: For $k = 1$ one gets the simple nearest neighbor method as maximal local technique, for $k \to n_L$ a global majority vote of the whole learning set results. This implies a constant prediction for all new observations, that have to be classified: Always the most frequent class within the learning set is predicted.

3 Weighted k-Nearest-Neighbors (wkNN)

This extension is based on the idea, that such observations within the learning set, which are particularly close to the new observation (y, x), should get a higher weight in the decision than such neighbors that are far away from (y, x). This is not the case with kNN: Indeed only the k nearest neighbors influence the prediction; however, this influence is the same for each of these neighbors, although the individual similarity to (y, x) might be widely different. To reach this aim, the distances, on which the search for the nearest neighbors is based in the first step, have to be transformed into similarity measures, which can be used as weights.

Standardization of covariates

Thus again in the first step the k nearest neighbors are selected according to the Minkowski distance. As before, for that purpose one needs two parameters: The number of neighbors k and the Minkowski parameter q for selection of the distance measure.

To put equal weight on each covariate in computing the distances, one has to standardize the values. In the case of ratio or difference scale level, this aim is reached simply by dividing the variables by their standard deviation. Subtraction of the mean is not necessary, as this operation has no influence on the distances between observations.

For ordinal covariates with m classes we offer two procedures: They can be treated in the same way as variables of ratio scale level, or be transformed into $m-1$ dummy variables. For example, if there are 5 ordinal classes, the following dummy variables $v_1, \ldots, v_4$ result from this transformation:

class	v_1	v_2	v_3	v_4
1	1	1	1	1
2	-1	1	1	1
3	-1	-1	1	1
4	-1	-1	-1	1
5	-1	-1	-1	-1

When computing differences between two observations, the number of non zero columns corresponds to the difference of order between them. This second approach always treats distances in a proportional, linear way independently of the Minkowski parameter.

In a similar way dummy variables for nominal covariates with m classes can be derived. As there is no reference category when working with distances, one needs m dummy variables:

class	v_1	v_2	v_3	v_4	v_5
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1

Here there is either no difference if the observations are identical, or two non zero columns if they belong to different classes.

Now the problem arises of how to standardize these dummy variables. We offer a standardization technique for both kinds of dummy variables, that is based on the trace of the covariance matrix of the corresponding dummies. We ignore the correlation structure and use the term

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}\mathrm{var}(v_i)} \qquad \text{rsp.} \qquad \sqrt{\frac{1}{m-1}\sum_{i=1}^{m-1}\mathrm{var}(v_i)}$$

as a normalizing divisor for nominal, and respectively ordinal, covariates. This standardization of all corresponding dummy variables with the same (averaged) standard deviation is necessary, as differences between classes should be treated symmetrically, regardless of the differences in the standard deviations of the single dummy variables.

Furthermore, without an additional correction covariates with many classes would get more weight than others, as they produce more dummy variables, which all would contribute in the same way to the distance measure as one single metric variable. So when computing the distances, all differences between corresponding dummies are weighted by $\frac{1}{m-1}$ or $\frac{1}{m}$ respectively, if the original covariate has ordinal or nominal scale level.

Of course, standardization with the mean variances of the dummy variables will differ from the exact treatment of metric variables, but this approach seems better than not handling categorical variables at all.

The standardization of all kinds of covariates is only based on the observations from the learning set. One could also add the x values of all new cases that have to be classified before the standardization step, but we believe that it is more consistent and comparable to standardize all new observations by the same factors. Then the results only depend on the values within the learning set.

Few authors address the issue of how to include nominal and ordinal covariates within distance measures, but alternative treatments for categorical variables can be found for example in *Fahrmeir et al.* (1996) and *Cost and Salzberg* (1993).

Weighting scheme for neighbors

The transition from distances to weights then follows in the second step according to any arbitrary kernel function. These are functions $K(.)$ of the distances d with maximum in $d = 0$ and values, that get smaller with growing absolute value of d. Thus the following properties must hold:

- $K(d) \geq 0$ for all $d \in \mathbb{R}$
- $K(d)$ gets its maximum for $d = 0$
- $K(d)$ descents monotonously for $d \rightarrow \pm\infty$

Typical examples for this kind of function are the following:

- rectangular kernel $\frac{1}{2} \cdot \mathrm{I}(|d| \leq 1)$
- triangular kernel $(1 - |d|) \cdot \mathrm{I}(|d| \leq 1)$
- Epanechnikov kernel $\frac{3}{4}(1 - d^2) \cdot \mathrm{I}(|d| \leq 1)$
- quartic or biweight kernel $\frac{15}{16}(1 - d^2)^2 \cdot \mathrm{I}(|d| \leq 1)$

- triweight kernel $\frac{35}{32}(1-d^2)^3 \cdot \mathrm{I}(|d| \leq 1)$
- cosine kernel $\frac{\pi}{4}\cos(\frac{\pi}{2}d) \cdot \mathrm{I}(|d| \leq 1)$
- Gauss kernel $\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{d^2}{2}\right)$
- inversion kernel $\frac{1}{|d|}$

In the case of distances, which are defined as strictly positive values, of course only the positive domain of K has to be used. In this sense the choice of the kernel is the third parameter of this technique. But from experience the choice of a special kernel (apart from the special case of the rectangular kernel, that gives equal weights to all neighbors) is not crucial.

Every kernel function needs either a window width, if the values become zero in a certain distance from the maximum value, or a dispersion parameter, if the values are larger than zero for all $d \in \mathbb{R}$. In wkNN both are selected automatically according to the distance of the first neighbor $x_{(k+1)}$, that is not taken into consideration any more. This is done implicitly by standardization of all other distances with the distance of the $(k+1)$th neighbor:

$$D(x, x_{(i)}) = \frac{d(x, x_{(i)})}{d(x, x_{(k+1)})} \quad \text{for} \quad i = 1, \ldots, k$$

These standardized distances always take values within the interval $[0, 1]$. In our implementation we add a small constant $\epsilon > 0$ to $d(x, x_{(k+1)})$ in order to avoid weights of 0 for some of the nearest neighbors. This could happen if one or more of these neighbors show exactly the same distance as the $(k+1)$th, as most of the kernels become 0 at the window boundary $D = 1$. This band width of 1 is an adequate choice, as all observations with a larger distance from x than the kth neighbor have no influence on the prediction. So the choice of the band width is adaptively based on the data.

Summary of wkNN

After determination of the similarity measures for the observations in the learning set, each new case (y, x) is classified into the class with the largest added weight

$$\max_r \left(\sum_{i=1}^{k} K(D(x, x_{(i)})) I(y_{(i)} = r) \right) \quad .$$

Both kNN and NN can be seen as special cases of wkNN: kNN results for a choice of the rectangular kernel, NN results for $k = 1$, independently of the chosen kernel function.

The main target of this extended method is to gain a technique, that up to a certain degree is independent of a bad choice for k resulting in a high misclassification error. Now this number of nearest neighbors is implicitly hidden in the weights: If k is too large k is adjusted to a lower value automatically. In this case a small number of neighbors with large weights dominates the other neighbors, whose classes have no influence on the prediction because of their low weights.

The algorithmic structure of wkNN is shown below as a summary. As mentioned before, the common nearest neighbor techniques are special cases of this algorithm.

Weighted k-Nearest-Neighbor classification (wkNN)

1. Let $L = \{(y_i, x_i), i = 1, \ldots, n_L\}$ be a learning set of observations x_i with given class membership y_i and let x be a new observation, whose class label y has to be predicted.

2. Find the $k+1$ nearest neighbors to x according to a distance function $d(x, x_i)$.

3. The $(k+1)$th neighbor is used for standardization of the k smallest distances via
$$D_{(i)} = D(x, x_{(i)}) = \frac{d(x, x_{(i)})}{d(x, x_{(k+1)})} \quad .$$

4. Transform the normalized distances $D_{(i)}$ with any kernel function $K(.)$ into weights $w_{(i)} = K(D_{(i)})$.

5. As prediction for the class membership y of observation x choose the class, which shows a weighted majority of the k nearest neighbors
$$\hat{y} = \max_r \left(\sum_{i=1}^{k} w_{(i)} I(y_{(i)} = r) \right) \quad .$$

In general these methods, wkNN and also simpler nearest neighbor techniques can be seen as voting or ensemble methods in this sense: Some potential classifiers (the nearest neighbors) are aggregated by a (weighted) majority vote and this aggregated result is used as prediction. This shows a certain similarity to modern ensemble techniques like bagging or boosting (*Breiman* (1996), *Friedman et al.* (2000)).

4 Using wkNN for Ordinal Classification

A second extension, that is independent of the weighting method, results from the question how to cope with target variables with different scale level. The classification version of wkNN described above is concieved to predict nominal classes and works with a weighted majority vote of the nearest neighbors. This proceeding can also be described as using the mode of the estimated class probability distribution, that results from the standardized added weights for each class label:

$$\hat{P}(y = r|x, L) = \frac{\sum_{i=1}^{k} w_{(i)} I(y_{(i)} = r)}{\sum_{i=1}^{k} w_{(i)}}$$

Based on this fact, it seems only natural to use the median of this distribution for the prediction of an ordinal target variable. Furthermore the prediction of a metric target variable with an even higher scale level could be done via the mean of the distribution.

Using this mean shows a strong connection to the local regression technique LOESS. Here the residual sum of squares of a localized regression problem is minimized:

$$\min_\beta \sum_{i=1}^{n_L} (y_i - \beta_0 - \beta_1(x_i - x))^2 \, K\left(\frac{x_i - x}{d(x, x_{(k)})}\right)$$

If no covariates are considered, one gets the special case of the Nadaraya-Watson estimator, a local smoothing technique that uses piecewise constant regression functions. This means that the prediction is simply the weighted mean of all observations within the local window:

$$\hat{y} = \mathrm{E}(y|x) = \frac{\sum_{i=1}^{n_L} K\left(\frac{x_i - x}{d(x, x_{(k)})}\right) y_i}{\sum_{i=1}^{n_L} K\left(\frac{x_i - x}{d(x, x_{(k)})}\right)} = \frac{\sum_{i=1}^{k} w_{(i)} y_{(i)}}{\sum_{i=1}^{k} w_{(i)}}$$

This is exactly the behaviour of wkNN when using the mean of the class distribution.

In this sense Nadaraya-Watson can also be seen as a special case of wkNN and forms the point of intersection with LOESS. The only differences are that on the one hand, wkNN offers a large variety of possible kernel functions in order to produce different weighting schemes, while LOESS works only with the tricube kernel

$$K(d) = \frac{70}{81}(1 - |d|^3)^3 \cdot \mathrm{I}(|d| \leq 1) \quad .$$

On the other hand the standardization of the distances is based on the $(k+1)$th neighbor instead of the kth in LOESS. Thus the kth neighbor still has influence on the prediction, which fits in a better way to a nearest neighbor technique that has its origin in the common kNN method.

As working with the mean of the class distribution, in other words with Nadaraya-Watson or LOESS without covariates, does not lead to new insights, the crucial point of this work is the classification context with a special view on the field of ordinal prediction via the median of the class distribution. This point is not covered by the regression technique LOESS, which is designed for metric variables. Nevertheless, we point out that Nadaraya-Watson estimation will be completely available within the *kknn*-package.

5 Empirical Studies

5.1 Study Design

In this section we compare weighted nearest neighbor approaches to simpler variants, that do not use weights on the data. For that purpose the number k of nearest neighbors as well as the Minkowski parameter q and the kernel function $K(.)$ are changed systematically. The results of one example with ordinal class structure are of special interest.

The evaluation of the methods is based on the raw misclassification error rate $\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$. In the case of ordinal class structure additional measures should be used, which take into account, that a larger distance is a more severe error than a wrong classification into a neighbor class. Therefore we use the mean absolute value of the differences $\frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$ and the mean squared difference $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, which penalizes larger differences even harder.

Of course these measures are based on test error, as the resubstitution error always is zero when using nearest neighbor techniques. Therefore we divide the dataset at random into two parts consisting respectively of one third and two thirds of the observations. The larger (learning) dataset is used as set of prototypes and the observations of the smaller (test) dataset are predicted. We use 50 different random splits into learning and validation set and give the mean over these splits as result.

5.2 Datasets

The *Wisconsin breast cancer dataset*, originally collected at the University of Wisconsin Hospitals in Madison, is taken from the extensive data archives of the University of California in Berkeley. It is a standard dataset, which has been used many times as an example in evaluating new classification techniques, for example *Breiman* (1998), and is available on the internet via
`http://www.ics.uci.edu/~mlearn/MLRepository.html`.
The target variable is binary and describes the malignancy of a tumor, 9 ordinal biochemical measures are used as covariates. All observations with missing values were deleted, so 683 out of 699 observations remain in the dataset.

The *glass identification dataset* is taken from the data archives of the University of California. The target variable is the industrial usage of glass material, which is measured in 6 classes, the independent variables are 9 chemical indicators. The dataset consists of 214 cases without missing values.

The *ionosphere dataset* from the data archives in Berkeley describes radar signals, that are measured by 34 metric variables. The binary target variable is the quality of the admission (good or bad). All in all 351 observations without missing values are considered.

The *soybean dataset* again is one of the standard datasets from the data archives of the University of California. In contrast to the other datasets the target variable has a large number of different classes (15). After deletion of all observations with one or more missing values 266 cases remain. While the dependent variable describes the kind of disease of a soybean plant, the 35 covariates represent climatic as well as other biological factors.

The *SRBCT dataset* is presented in *Kahn et al.* (2001) and contains expression values of 2308 genes for 83 *Single Round Blue Cells Tumor* patients, each coming from one out of 4 different classes (EWS, BL, NB and RMS). These data were published via
`http://www.thep.lu.se/pub/Preprints/01/lu_tp_01_06_supp.html`.
After preprocessing the selected genes were standardized. This is a typical example for the problem of microarray analysis, where there are a huge number of variables, but only a few observations are available.

Finally, the *scapula dataset* (Feistl & Penning, not published yet) are part of a dissertation written at the *Institut für Rechtsmedizin der LMU München*. The aim was to predict age of dead bodies only by means of the scapula. Therefore a lot of measures, implying angles, lengths, descriptions of the surface, etc. were provided. We preselected 15 important covariates to predict age, which was splitted into 8 distinct ordinal classes, each covering ten years. The dataset consists of 153 complete observations.

5.3 Results

k	q	kernel	test error	k	q	kernel	test error
1	1	rectangular	0.035	5	1	rectangular	0.033
		triangular	0.035			triangular	0.033
		biweight	0.035			biweight	0.035
	2	rectangular	0.043		2	rectangular	0.032
		triangular	0.043			triangular	0.036
		biweight	0.043			biweight	0.040
3	1	rectangular	0.032	7	1	rectangular	0.034
		triangular	0.035			triangular	0.030
		biweight	0.034			biweight	0.033
	2	rectangular	0.034		2	rectangular	0.032
		triangular	0.040			triangular	0.032
		biweight	0.042			biweight	0.038

Table 1: Misclassification error for Wisconsin breast cancer data

We now consider the misclassification errors for the first 4 datasets. In the Wisconsin breast cancer data (Table 1) there is almost no variation in the error rate. This classification problem seems to be too simple to produce significant differences between the techniques. For the glass (Table 2), ionosphere (Table 3) and soybean (Table 4) datasets, a small value of k seems to be the best

k	q	kernel	test error	k	q	kernel	test error
1	1	rectangular	0.276	5	1	rectangular	0.330
		triangular	0.276			triangular	0.274
		biweight	0.276			biweight	0.269
	2	rectangular	0.304		2	rectangular	0.356
		triangular	0.304			triangular	0.305
		biweight	0.304			biweight	0.302
3	1	rectangular	0.305	7	1	rectangular	0.345
		triangular	0.276			triangular	0.277
		biweight	0.279			biweight	0.271
	2	rectangular	0.330		2	rectangular	0.355
		triangular	0.308			triangular	0.307
		biweight	0.307			biweight	0.300

Table 2: Misclassification error for glass data

choice. Without weighting the error rates increase with growing number of k. By using weights, no matter which kind of kernel is used, the results for higher k again reach the optimal results. Thus the advantage of the weighting method shows in the fact, that a kind of automatic adjustment of k takes place: If k is chosen too high the weights reduce the influence of neighbors that are too far away from the new observation. Thus this desired property could be verified empirically.

k	q	kernel	test error	k	q	kernel	test error
1	1	rectangular	0.096	5	1	rectangular	0.119
		triangular	0.096			triangular	0.099
		biweight	0.096			biweight	0.098
	2	rectangular	0.136		2	rectangular	0.163
		triangular	0.136			triangular	0.133
		biweight	0.136			biweight	0.130
3	1	rectangular	0.111	7	1	rectangular	0.125
		triangular	0.099			triangular	0.102
		biweight	0.099			biweight	0.100
	2	rectangular	0.156		2	rectangular	0.172
		triangular	0.136			triangular	0.135
		biweight	0.134			biweight	0.128

Table 3: Misclassification error for ionosphere data

For the microarray data, the most interesting point is the comparison of the nearest neighbor results with other classification techniques. Therefore we employ a variety of classical and modern methods. First, in addition to our nearest neighbor techniques we apply a recent method called prediction analysis of microarray (PAM) which was especially designed for high-dimensional microarray data (*Tibshirani et al.* (2002)) and works without gene selection. PAM is based on shrunken centroids and necessitates the choice of the shrinkage parameter

k	q	kernel	test error	k	q	kernel	test error
1	1	rectangular	0.116	5	1	rectangular	0.177
		triangular	0.116			triangular	0.127
		biweight	0.116			biweight	0.120
	2	rectangular	0.137		2	rectangular	0.197
		triangular	0.137			triangular	0.135
		biweight	0.137			biweight	0.126
3	1	rectangular	0.157	7	1	rectangular	0.201
		triangular	0.118			triangular	0.135
		biweight	0.116			biweight	0.125
	2	rectangular	0.172		2	rectangular	0.218
		triangular	0.125			triangular	0.141
		biweight	0.133			biweight	0.130

Table 4: Misclassification error for soybean data

δ. The number of genes used to compute the shrunken centroids depends on this parameter. A possible choice is $\delta = 0$: All genes are used to compute the centroids. Also a selection method for an optimal value of δ by cross-validation is proposed. In our study, we apply both approaches. The PAM method is implemented in the *R* library *pamr*. Next we use a promising technique called partial least squares (PLS): New components are determined by PLS dimension reduction and LDA is performed on these new components (*Nguyen and Rocke* (2002)). We use successive numbers of PLS components. These results were already published in *Boulesteix* (2004). Furthermore some modern techniques like bagging (*Breiman* (1996)) and different boosting algorithms (*Friedman et al.* (2000)), both based on standard classification trees (CART), as well as support vector machines SVM (*Furey et al.* (2000)) are applied. For SVM we use the implementation from the *R* library *e1071*. Finally, classical linear discriminant analysis (LDA) is performed on these data.

For most of these techniques we need a variable or gene selection method, as coping with more than two thousand variables is not possible. Here the genes are ranked according to the $\frac{BSS}{WSS}$-statistic, that for gene s is computed as follows:

$$\frac{\mathrm{BSS}_s}{\mathrm{WSS}_s} = \frac{\sum_{r=1}^{c} \sum_{i:y_i=r} (\hat{\mu}_{sr} - \hat{\mu}_s)^2}{\sum_{r=1}^{c} \sum_{i:y_i=r} (x_{is} - \hat{\mu}_{sr})^2}$$

BSS means *between group sum of squares*, WSS *within group sum of squares*. Furthermore $\hat{\mu}_s$ denotes the sample mean of x_s, while $\hat{\mu}_{sr}$ is the sample mean of x_s within class r. This kind of variable selection is performed separately for the learning data of each random split of the data set.

This comparison in Table 5 shows, that CART and also ensemble techniques based on this method like bagging and boosting, which are known to improve CART significantly, do not perform very well. For this data LDA and PAMR give lower error rates and SVM reduces the error beyond 2 %. Results for PLS improves dramatically when a larger number of components are used. Finally,

technique	test error
PAMR ($\delta = 0$)	0.066
PAMR (optimal δ)	0.024
1 PLS	0.362
3 PLS	0.052
5 PLS	0.008
LDA (10 genes)	0.085
LDA (20 genes)	0.041
SVM (100 genes)	0.013
CART (100 genes)	0.177
Bagging	0.070
Discrete AdaBoost	0.069
Gentle AdaBoost	0.065
NN (100 genes)	0.009
5NN (100 genes)	0.013
w5NN (100 genes)	0.006

Table 5: Misclassification error for SRBCT data

nearest neighbor performs best, and our weighting technique shows the lowest misclassification error over all methods tested. This is a very satisfying result, as microarray problems play a very important role in modern biostatistics.

When interpreting the ordinal results concerning the scapula data (Table 6), the distance measures between true and predicted values are of special interest, as these measures take into account the ability to consider ordinal structure. Here the choice $k = 1$ seems to give the best results, if one ignores ordinality and simply uses mode instead of median. For higher k the error rates of kNN increase, but can again be lowered by the use of weighting techniques.

When using the suitable median in order to take ordinality within the target variable into account the error rates for every single parameter combination decrease in comparison to the mode, which is a very strong result. However, surprisingly now mostly the unweighted kNN results seem to dominate.

Apart from the satisfying empirical results, the major advantage of nearest neighbor techniques is the computation time. In comparison to other modern ordinal classification techniques, for example ordinal boosting (*Tutz & Hechenbichler* (2004)), which sometimes give slightly better results, the corresponding wkNN results can be computed within only a few seconds.

6 Concluding Remarks

In summary the results of wkNN for classification of nominal scaled target variables are very promising, as choosing a too large value of k gets implicitly corrected by using a weighting scheme. Especially in the topical and difficult case of microarray data, nearest neighbor techniques in general, but especially

technique	k	q	kernel	misclassification	absolute distance	squared distance
nominal (mode)	1	1	rectangular	0.653	0.967	1.841
			triangular	0.653	0.967	1.841
			biweight	0.653	0.967	1.841
		2	rectangular	0.629	0.914	1.695
			triangular	0.629	0.914	1.695
			biweight	0.629	0.914	1.695
	3	1	rectangular	0.665	1.047	2.147
			triangular	0.647	0.958	1.824
			biweight	0.651	0.962	1.831
		2	rectangular	0.653	1.021	2.051
			triangular	0.632	0.918	1.709
			biweight	0.630	0.915	1.697
	5	1	rectangular	0.647	0.985	1.918
			triangular	0.642	0.938	1.740
			biweight	0.648	0.949	1.780
		2	rectangular	0.647	0.977	1.881
			triangular	0.626	0.909	1.685
			biweight	0.633	0.922	1.717
	7	1	rectangular	0.655	0.993	1.924
			triangular	0.633	0.916	1.666
			biweight	0.643	0.936	1.735
		2	rectangular	0.646	0.996	1.968
			triangular	0.615	0.877	1.573
			biweight	0.631	0.919	1.709
ordinal (median)	1	1	rectangular	0.653	0.967	1.841
			triangular	0.653	0.967	1.841
			biweight	0.653	0.967	1.841
		2	rectangular	0.629	0.914	1.695
			triangular	0.629	0.914	1.695
			biweight	0.629	0.914	1.695
	3	1	rectangular	0.607	0.847	1.451
			triangular	0.648	0.941	1.762
			biweight	0.651	0.959	1.817
		2	rectangular	0.620	0.852	1.432
			triangular	0.632	0.896	1.613
			biweight	0.628	0.902	1.644
	5	1	rectangular	0.610	0.822	1.355
			triangular	0.628	0.865	1.482
			biweight	0.642	0.915	1.670
		2	rectangular	0.620	0.832	1.357
			triangular	0.625	0.847	1.420
			biweight	0.629	0.889	1.585
	7	1	rectangular	0.622	0.824	1.329
			triangular	0.622	0.828	1.360
			biweight	0.632	0.876	1.531
		2	rectangular	0.617	0.820	1.330
			triangular	0.608	0.806	1.305
			biweight	0.623	0.864	1.502

Table 6: Error rates for scapula data

wkNN, perform very well. Finally, in the field of ordinal classification the use of the median instead of the mode improves the results of nearest neighbor techniques. By using the mean instead of median or mode one gets the classical Nadaraya-Watson estimator for local smoothing, which is also included in our wkNN package.

In general local techniques are known to be inadequate for high dimensional data because of the curse of dimensionality. It must be emphasized that in many practical problems, especially in the field of microarray data, that are always extremely high dimensional, nearest neighbor techniques give quite good results. One possible explanation for this statement is, that the problem of high dimensionality appears especially when estimating a statistical model with many parameters for the huge amount of covariates. On the contrary nearest neighbor techniques do not estimate any parameters; instead the prediction is based on prototypes. For this reason further investigation of nearest neighbor techniques (using mode, median or mean of the class probability distribution) seems to be a worthwhile task.

Alternative weighting schemes are mentioned in other works, but they show fundamental differences to our wkNN method. For example, the technique in *Fahrmeir et al.* (1996) selects a fixed number of neighbors for every possible class r ($r = 1, \ldots, c$) and the classification is based on the mean distance of a new observation to these class representatives. On the other hand, in wkNN, the question, "how many of the nearest neighbors out of the complete dataset fall into the different classes", is of great importance. Thus there is a much closer connection to the classical kNN technique, which is exclusively based on these counts. The weights in wkNN only play a role in the final classification step.

Paik and Yang (2004) use combinations of many kNN classifiers with different values for k and different subsets of covariates to improve the results of one single kNN prediction. This method is called adaptive classification by mixing (ACM). It also works with weights, but instead of weighting the observations of the learning set, a weighting scheme for the whole classifiers based on their classification probabilities is computed.

Flexible metric nearest neighbor classification by *Friedman* (1994) introduces a third completely different idea for a weighting scheme: Here local flexible weights for the covariates are used in order to consider their local relevance, that is estimated by recursive partitioning techniques. So again no weighting of the observations occurs.

Finally, we note that one important problem still can not be solved by using weights in nearest neighbor techniques: The critical point of variable selection. Too many covariates, that vary completely at random and have no predictive power for the target variable, can disturb the prediction severely.

References

[1] Boulesteix (2004): *PLS dimension reduction for classification with microarray data*; Discussion Paper 392, SFB 386 der Ludwig-Maximilians-Universität München.

[2] Breiman (1996): *Bagging predictors* in **Machine Learning 24**; p.123–140.

[3] Breiman (1998): *Arcing classifiers* in **Annals of Statistics 26**; p.801–849.

[4] Chen, Härdle and Schulz (2004): *Prognose mit nichtparametrischen Verfahren*; Technical Report, Humboldt-Universität zu Berlin.

[5] Cleveland and Loader (1995): *Smoothing by local regression: Principles and methods*; Technical Report, AT&T Bell Laboratories, Murray Hill, NY.

[6] Cost and Salzberg (1993): *A weighted nearest neighbor algorithm for learning with symbolic features* in **Machine Learning 10**; p.57–78.

[7] Cover and Hart (1967): *Nearest neighbor pattern classification* in **IEEE Transactions on Information Theory 13**; p.21–27.

[8] Fahrmeir, Hamerle and Tutz (1996): *Multivariate statistische Verfahren*; Walter de Gruyter & Co Verlag; Berlin.

[9] Fix and Hodges (1951): *Discriminatory analysis, nonparametric discrim-*

ination: Consistency properties; Technical Report 4, U.S. Air Force, School of Aviation Medicine, Randolph Field, TX.

[10] Friedman (1994): *Flexible metric nearest neighbor classification*; Technical Report 113, Stanford University, Statistics Department.

[11] Friedman, Hastie and Tibshirani (2000): *Additive logistic regression: A statistical view of boosting* in **Annals of Statistics 28**; p.337–374.

[12] Furey, Cristianini, Duffy, Bednarski, Schummer and Haussler (2000): *Support vector machine classification and validation of cancer tissue samples using microarray expression data* in **Bioinformatics 16**; p.906–914.

[13] Kahn, Wie, Ringner, Saal, Ladanyi, Westermann, Berthold, Schwab, Antonescu, Peterson and Meltzer (2001): *Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks* in **Nature Medicine 7**; p.673–679.

[14] Nguyen and Rocke (2002): *Tumor classification by partial least squares using microarray gene expression data* in **Bioinformatics 18**; p.39–50.

[15] Paik and Yang (2004): *Combining nearest neighbor classifiers versus cross-validation selection* in **Statistical Applications in Genetics and Molecular Biology 3**; Article 12.

[16] Tibshirani, Hastie, Narasimhan and Chu (2002): *Diagnosis of multiple cancer types by shrunken centroids of gene expression* in **PNAS 99**; p.6567–6572.

[17] Tutz and Hechenbichler (2003): *Aggregating classifiers with ordinal response structure*; Discussion Paper 359, SFB 386 der Ludwig-Maximilians-Universität München.