ESTIMATING FEATURE WEIGHTS FOR DISTANCE-BASED CLASSIFICATION*

J.M. Sotoca, J.S. Sánchez and F. Pla
Dept. Llenguatjes i Sistemes Informàtics
Universitat Jaume I, Campus Riu Sec, E-12071 Castelló (SPAIN)
e-mail: {sotoca,sanchez,pla}@lsi.uji.es

Key words: Feature weighting; Feature relevance; Nearest neighbour rule; Generalized least squares; Classification.

Abstract: This paper presents a new feature weighting method for distance-based classifiers. It is based on a generalized least squares minimization of a criterion function to estimate a feature relevance metric. Experiments over both artificial and real data sets illustrate the behaviour of this algorithm when irrelevant attributes and/or features with varying relevance are present. Effectiveness of the proposed technique is compared with that of other weighting methods. We also provide an empirical study on the effect of the training set size on performance of feature weighting models.

1 INTRODUCTION

Supervised learning algorithms use a collection of instances or training set (TS) to estimate the class label of new input samples. The instances are generally represented by a number of attributes or features and a class value. After learning from the TS, the performance of the induced classifier is usually assessed on an independent testing set.

The Nearest Neighbour (NN) classification rule (Dasarathy, 1990) constitutes one of the most appealing examples of this kind of learning algorithms. This technique assigns to a given sample the same class than the closest instance in the TS according to a certain measure of similarity in the feature space. Apart from other advantages common to most non-parametric classification approaches, the NN rule and its extension to k neighbours (or k-NN rule, in which a sample is classified by taking the majority vote from its k closest instances) combine their simplicity in implementation with an appropriate behaviour in their expected performance.

In practice, an important drawback of these classifiers comes from the possible inclusion of irrelevant, redundant, interacting, or noisy attributes, which can drastically reduce the resulting classification performance. It is well-known that NN classifiers are specially sensitive to irrelevance since the corresponding distance function calculates an average similarity measure across all the attributes (Aha, 1992). Furthermore, the number of instances needed to maintain an appropriate error rate grows exponentially with the number of attributes (Langley and Iba, 1993).

These observations have motivated to propose many feature weighting techniques in order to scale the influence of an attribute over the similarity measure used by the NN rules according to the relevance of each feature. For instance, (Cost and Salzberg, 1993) use differences in probability distributions across classes to modify the distance metric for nominal attributes, (Kononenko, 1994) use a distance criterion between neighbours, (Wettschereck et al., 1997) employ mutual information to compute coefficients on numeric attributes, (Kohavi et al., 1997) utilize a simple wrapper approach to heuristically determine appropriate weights for use in NN classification, (Domeniniconi et al., 2002) propose a locally adaptive metric for computing neighbourhoods based on a Chi-squared distance analysis.

In this paper, we explore a quite different approach to determining feature weights for NN classification. From a practical point of view, we apply a parametric model based on a generalized non-linear least squares algorithm to estimate a feature relevance metric.

The organization of the rest of this paper is as fol-

*This work has been supported by grants: P1-1B2002-07 from *Fundació Caixa Castelló - Bancaixa*, DPI2001-2956-C02-02 and TIC2000-1703-C03-03 from CICYT *Ministerio de Ciencia y Tecnología*, and IST-2001-37306 from *IST Project European Union*.

lows. Section 2 provides a brief review of the generalized least squares minimization method and introduces the new feature weighting method. Section 3 describes the experiments carried out, along with the databases used in the present paper. Section 4 reports the empirical results, including a study on the effect of the number of instances in the TS over the behaviour of different feature weighting methods. Finally, some concluding remarks and further extensions are depicted in Section 5.

2 THE ESTIMATION METHOD

In this section, we briefly describe the generalized least squares estimate method to minimize a given function obtained from a class-intensity-based model. Afterwards, we introduce the new feature weighting algorithm based on these two concepts.

2.1 The generalized least squares estimator

The generalized least squares (GLS) method (Amemiya, 1985) is a procedure to find a consistent estimation by means of a parametric model. This estimator has already been used in the computer vision domain to estimate, for instance, a parametric model of lines or planes (Danuser and Stricker, 1998) or, motion estimation (Montoliu and Pla, 2001).

The problem can be described as a set of vector-values functions *f* which depend on a set of observed data l and an unknown model parameter *x*. They can be written in the implicit form as follows:

$$f(x, l) = 0 \tag{1}$$

When the real-world is observed, we see how the observation vectors l are perturbed by noise and other source errors. Thus, we can model $l = \widetilde{l} + e$ where $\widetilde{l}$ is the vector actually measured and *e* is a vector with the residual of the observation *l*. The estimation procedure consists of an iterative scheme with an initial state $x(t = 0) = \widetilde{x}$ in the model parameter and with the first estimate $l(t = 0) = \widetilde{l}$.

At each step *t* of the iterative process, we have the following change over the model:

$$\begin{array}{c} \triangle x = x(t+1) - x(t) \\ \triangle l = l(t+1) - l(t) = e \end{array} \tag{2}$$

The function *f* in Eq. (1) can be linearized using the Taylor expansion and neglecting the second and higher order terms as follows:

$$f^{t+1}(x, l) = f^t(x, l) + \frac{\partial f^t}{\partial x}\triangle x + \frac{\partial f^t}{\partial l}\triangle l = 0 \tag{3}$$

Previous equation can be rewritten in matrix form to solve the equation system

$$A\triangle x + Be = W \tag{4}$$

where $A = \frac{\partial f^t}{\partial x}$, $B = \frac{\partial f^t}{\partial l}$ and $W = -f^t(l, x)$.

To solve the Eq. (4), we need to minimize the expression $e^T Q_{ll}^{-1} e$ subject to the condition $f^t(x, l) = 0$, where Q_{ll}^{-1} can be interpreted as a weight associated with the actual observations. Applying Lagrange multipliers (Danuser and Stricker, 1998), $\triangle x$ at each step can be worked out as follows:

$$\triangle x = (A^T Q_{ww}^{-1} A)^{-1} A^T Q_{ww}^{-1} W \tag{5}$$

where the matrix $Q_{ww} = BQ_{ll}B^T$ is introduced to simplify the notation in Eq. (5).

The solution of the previous equation is obtained by an iterative process that stops when the residuals are small enough. Together with the estimation of the parameters, the estimates for the unperturbed observations l are updated after each iteration by:

$$\triangle l = Q_{ll} B^T Q_{ww}^{-1} (W - A\triangle x) \tag{6}$$

Thus, the previous equation proceeds iteratively and the difference between the values of the vector l and its estimation by the model are shorter at each iteration.

2.2 A class-intensity-based model

In general, we have a TS with *N* instances, where each instance x is a point $x = \{x_1, \ldots, x_d\} \in \Re^d$, in a d-dimensional feature space, and whose label $c(x)$ is a member of a set of classes *J*. Then, for a given sample, the aim is to assign it to the correct class using its measured value *x*.

Basically, the k-NN classifier consist of assigning an input sample q to the class most frequently represented among the k closest instances in the TS, according to a certain similarity measure.

In our case, the goal is to find a model that, through the use of a set of weights applied over each feature, we can minimize the misclassification risk. For this task, the use of the GLS method is proposed, which allows to find a global solution that will be robust with respect to outliers or misclassified elements in the training data.

Therefore, we suppose the following hypothesis related to the concept of electric field. This idea has been formulated in terms of the so-called *potential function rules* (Aizerman et al., 1964) as kernel classification rules. Using such an idea, we can apply the leave-one-out method over the TS and analyze the purity of TS in order to detect which dimensions have more relevance. Thus, for one instance x and its neighbourhood, we have prototypes with the same class as x (elements with a unit positive electrical

charge) and others with different class (elements with a unit negative electrical charge). We define the class intensity associated with an instance x as the sum of the influences that each neighbour p with class label $c(p)$ has over x. This influence is given by its charge class *C* and the inverse of the squared distance *D* according to the following expression:

$$E_x(w,\delta) = \sum_{p \in S} \frac{C(c(x),c(p))}{D(w,\delta)} \quad (7)$$

where S is a subset that contains the prototypes of the neighbourhood selected around x. w is the weight vector, and δ is the observation vector in the TS. The charge class is defined in the following way:

$$C(c(x),c(p)) = \begin{cases} 1 & \text{if } c(p) = c(x) \\ -1 & \text{if } c(p) \neq c(x) \end{cases}$$

and $D(w,\delta)$ represents the distance between the element x and a prototype $p \in S$. The distance can be expressed as follows:

$$D(w,\delta) = \sum_{i=1}^{d} w_i \delta_i(p_i,x_i)^2 \quad (8)$$

where $\delta_i()$ is

$$\delta_i(p_i,x_i) = \begin{cases} \frac{|p_i - x_i|}{max(x_i) - min(x_i)} & \text{if numerical value} \\ 0 & \text{if categorical value and } p_i = x_i \\ 1 & \text{if categorical value and } p_i \neq x_i \end{cases} \quad (9)$$

The feature weight w_i defines the model parameter. $\delta_i(p_i,x_i)$ is the normalized distance for feature i between the instance x and neigbour p, where all instances $x \in N$ constitute our set of observations. The previous expression (see Eq. (9)) is a standard function (i.e., subtract the minimum and divide by the observed range) to normalize all continuous values. This ensures that the range of $\delta_i()$ falls in [0,1] for all features. Thus, they have equal maximum and minimum potential effects on distance computations. This also means that each redundant, irrelevant and noisy feature has as much potential impact on the distance function as any other feature does.

2.3 Feature Weight Estimation

In the present approach, we start from an initial state in the TS supposing that the feature space is isotropic and all attributes have the same relevance. If the performance of the k-NN is optimal, all prototipes in the neighbourhood have the same class label, and the class intensity $E_x^2(w_a,\delta)$ is

$$E_x^2(w_a,\delta) = \sum_{p \in S} \frac{1}{D(w_a,\delta)} \quad (10)$$

where w_a is a vector of constant values that collects the proportion of importance for each feature. Through, we have a loss of classification caused by mislabelled instances, with an estimation of the class intensity in the neighbourhood of x as seen in Eq. (7). Applying Eq. (1) to such a problem, we propose to solve the following criterion function F_x:

$$F_x = E_x^2(w_a,\delta) - E_x^1(w,\delta) = 0 \quad (11)$$

The Jacobian matrices A and B, along with the residual of the functions W, defined for each instance *x*, have the following forms:

$$A = \left\{ -\frac{\partial E_x^1}{\partial w_1}, \ldots, -\frac{\partial E_x^1}{\partial w_d} \right\}$$

where

$$-\frac{\partial E_x^1}{\partial w_i} = \sum_{p \in S} \frac{C(c(x),c(p))\delta_i(p_i,x_i)^2}{D(w,\delta)^2}$$

and

$$B = \left\{ \frac{\partial F_x}{\partial \delta_{11}}, \ldots, \frac{\partial F_x}{\partial \delta_{1d}}, \ldots, \frac{\partial F_x}{\partial \delta_{k1}}, \ldots, \frac{\partial F_x}{\partial \delta_{kd}}, \frac{\partial F_x}{\partial E_x^1}, \frac{\partial F_x}{\partial E_x^2} \right\}$$

where to the neighbour p_k for the dimension i

$$\frac{\partial F_x}{\partial \delta_{ki}} = \frac{-2w_i\delta_{ki}(p_{ki},x_i)(1 - C(c(x),c(p_k)))}{D(w,\delta_k)^2}$$

Thus, the vector of residual functions is defined as follows:

$$W = -F_x = -(E_x^2(w_a,\delta) - E_x^1(w,\delta))$$

At each stage t, the new values obtained for the model parameter w_i are assigned to the vector w_a, in such a way that the iterative procedure tends to minimize the vector of residual functions.

3 DESCRIPTION OF DATA SETS AND EXPERIMENTS

From the UCI Repository (Murphy, 1995), twelve artificial and real data sets have been picked to compare the behaviour of the feature weighting model proposed here with that of other well-known methods. The main characteristics of the data sets are summarized in Table 1 (the number of irrelevant features are given in brackets). To increase statistical significance of the results in domains with a limited number of instances, 5-fold cross-validation has been employed: each database is divided into five blocks, using four blocks as TS and the remaining block as a test set. Results reported here correspond to the average over the five partitions.

The six artificial databases (Led+17, Waveform, Waveform+40, and Monk1-3) have been chosen to

Table 1: The data sets used in the experiments.

	Features	Classes	Instances
Led+17	24 (17)	10	2,000
Waveform	21	3	5,000
Waveform+40	40 (19)	3	5,000
Monk1	6 (3)	2	556
Monk2	6	2	601
Monk3	6 (3)	2	494
Diabetes	8	2	768
Glass	9	6	214
Heart	13	2	270
Vowel	10	10	528
Vehicle	18	4	848
Wine	13	3	178

evaluate the performance of each algorithm under controlled conditions: these data sets are known to have irrelevant and redundant features along with attributes with varying relevance. But, on the other hand, to ensure practical applicability, we have also wanted to analyze the proposed method in real-world domains. The data sets have both continuous and nominal features.

The Waveform+40 data set is identical to the Waveform database, but adding 19 completely irrelevant features. In them, all attributes include some noise, giving different levels of relevance (Breiman et al., 1984). On the other hand, both Monk1 and Monk3 contain 3 irrelevant features but, in the case of Monk3, noise has been introduced to the data set by changing the class label of some instances.

4 EMPIRICAL RESULTS

In this section, we present classification accuracy of several feature weighting methods when applied to the databases described in Table 1. Apart from the approach proposed here, we have also employed the well-known *ReliefF* algorithm (Kononenko, 1994), a method called *Class Weighted*-L_2 (CW-L_2) (Paredes and Vidal, 2000), and non-weighted k-NN classification using equal weights for all features ($w_i = 1.0$).

The ReliefF scheme assigns low weights to features that are completely irrelevant, but it is not clear to what extent it can detect redundant or highly interacting attributes. The CW-L_2 method obtains a set of weights (one weight per attribute and class) by means of gradient-descent minimization of an appropriate criterion index.

For the GLS algorithm, all weights are initially set to 1.0. Although the method consists of minimizing the criterion function given in Eq. (11), it has to be noted that a low value of F_x does not always result in a low leaving-one-out 1-NN error rate. This suggests that, rather than using the weight settings obtained at the end of the minimization procedure, in general a better choice would be providing the weights that led to the minimum 1-NN error rate.

Table 2 provides the classification accuracy obtained by each method. The first five columns correspond to the results when using the 1-NN rule, while the last columns are those from the best k-NN classifier (with k ranging from 1 through 21). Highlight indicates the best method for each particular domain, both with 1-NN and k-NN rules. In all cases, except for the CW-L_2 scheme, we have used a normalized Euclidean distance function.

From the results reported in Table 2, it can be noted two situations. First, in the case of databases with completely irrelevant attributes (Led+17, Waveform+40, Monk1, and Monk3), all feature weighting methods outperform non-weighted k-NN classification (i.e., $w_i = 1.0$). Analogously, when attribute relevance varies (Waveform and Monk2), the weighting algorithms also achieve a better performance than the non-weighted classification. On the other hand, over most real data sets, improvement with the feature weighting models is not statistically significant, probably because all attributes are relevant.

In Led+17, all feature weighting methods obtain a 100% of classification accuracy, suggesting that detection of irrelevant attributes does not constitute a hard problem for this particular domain. In general, both ReliefF and GLS models obtain the highest increase in performance with respect to the non-weighted classification. In fact, when comparing the columns of the best k-NN, one can observe that ReliefF and GLS provide the highest classification accuracy in 8 out of the 12 databases. On the other hand, CW-L_2 algorithm clearly outperforms the other methods only in the case of the Glass database.

4.1 Effect of TS Size on Feature Weighting

In this section, we study the effect of using different TS sizes on the classification accuracy of the feature weighting methods. More specifically, we are interested in knowing the number of instances needed to achieve a sufficiently high classification accuracy. In other words, this will determine which feature weighting algorithm has the fastest learning rate: if two models show a similar classification accuracy but one needs a smaller number of instances than the other, the former will be deemed as the best method for a particular problem. To perform this experiment, we selected the two best models (i.e., ReliefF and GLS) according to the results reported in Table 2.

In the Led+17 database, for each TS used in the previous experiment (that is, for each of the five

Table 2: Accuracy performances of several feature weighting methods.

	1-NN				k-NN			
	$w_i = 1$	ReliefF	GLS	CW-L_2	$w_i = 1$	ReliefF	GLS	CW-L_2
Led+17	76.7	**100**	**100**	**100**	96.9	**100**	**100**	**100**
Waveform	76.8	**78.4**	77.1	77.0	**84.8**	**84.8**	**84.8**	80.4
Waveform+40	73.3	**79.7**	75.3	77.8	83.2	**85.1**	84.1	80.1
Monk1	73.1	**100**	99.2	96.2	76.1	**100**	99.2	96.2
Monk2	81.5	82.1	**89.8**	88.0	81.5	82.1	**89.8**	88.0
Monk3	85.6	93.5	**96.4**	**96.4**	96.0	97.1	**99.1**	98.2
Diabetes	**71.5**	70.5	70.4	69.3	74.3	73.0	**74.7**	73.4
Glass	68.2	69.1	68.2	**76.2**	68.2	69.1	68.2	**76.2**
Heart	75.8	76.9	76.6	**79.1**	**82.9**	79.1	82.8	82.5
Vowel	97.8	**98.1**	97.8	97.4	97.8	**98.1**	97.8	97.4
Vehicle	67.8	**68.7**	67.8	68.2	**68.9**	68.7	68.6	68.8
Wine	94.9	97.1	94.9	**97.7**	97.1	97.1	97.1	**97.7**

blocks), we randomly picked up a number of instances, ranging from 15 to 210, maintaining the a priori probabilities of the classes in the original TS. Then, for each training subset, we employed ReliefF and GLS to estimate the feature weights and finally, the average 1-NN classification accuracy for those different TS sizes has been included in the figures. In the case of the Monk2 database, the TS size ranges from 32 to 448 instances. For comparison purposes, we have also included in this study the classification accuracy achieved by the non-weighted 1-NN rule ($w_i = 1.0$).

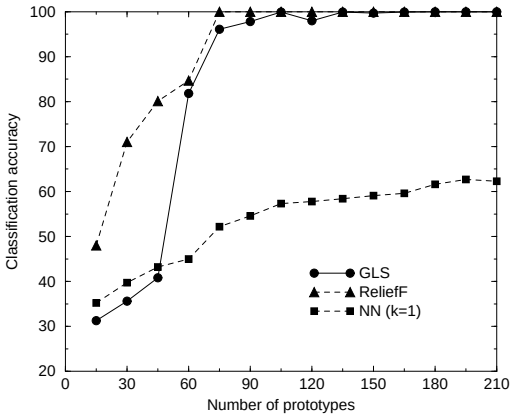

Figure 1: Classification accuracy with varying the number of instances over the Led+17 database.

Figure 1 illustrates the classification accuracy over the Led+17 data set when varying the TS size. As can be seen, while ReliefF achieves the highest classification performance of 100% with 75 instances, GLS reaches the optimal classification rate when using the training subset with 105 instances. Thus, for this particular domain with 17 irrelevant attributes, ReliefF presents a learning rate somewhat faster than GLS. It is worth noting that the performance of the non-weighted NN rule is far enough from the optimal accuracy obtained by both feature weighting methods in almost all TS sizes. In fact, there is a significant difference between the non-weighted 1-NN classifier and the weighting algorithms when the number of instances is 60 and higher.

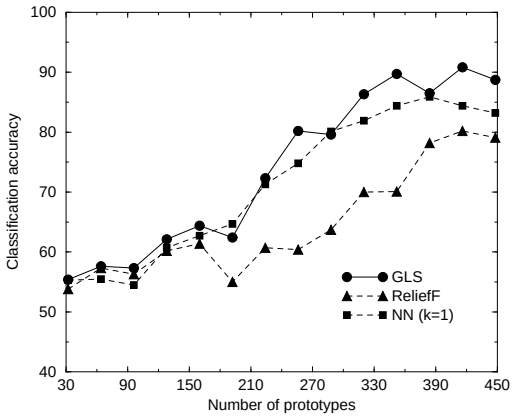

Figure 2: Classification accuracy with varying the number of instances over the Monk2 database.

Figure 2 shows the classification performance over the Monk2 database. In this case, both ReliefF and GLS achieved the highest classification accuracy by using 416 instances. Nevertheless, GLS obtains a classification rate of 80.2% (that is, the best result of ReliefF) with only 256 instances. Results in this experiment suggest that, when all attributes are relevant, GLS performs faster than ReliefF and, even more important, it obtains a higher classification accuracy (the highest rate obtained by GLS is 90.8%). On the other hand, as expected, the non-weighted 1-NN rule exhibits a behaviour substantially better then that shown in the case of the Led+17 database: in the present case with all attributes being relevant, the 1-NN classifier provides high enough accuracy rate.

5 CONCLUDING REMARKS

In this paper, a new feature weighting method has been introduced. It basically consists of determining a set of weights by means of a generalized least squares minimization of a criterion function. The aim of this approach is to estimate a feature relevance metric for distance-based classifiers, in such a way that neighbourhood is elongated along the least relevant features and constricted along the most important ones.

From the experiments carried out, it can be observed that the behaviour of the GLS algorithm proposed here is similar to that of the well-known ReliefF approach. Both of them are capable of detecting the irrelevant attributes and also determining the relevance of each feature. These methods generally outperform the non-weighted k-NN classification rule ($w_i = 1.0$) and other feature weighting approaches (in particular, the CW-L_2 algorithm).

When studying the learning rate of ReliefF and GLS models, it seems that the former achieves the optimal classification accuracy faster than GLS in presence of irrelevant attributes, while GLS algorithm is able to obtain better results than ReliefF when all attributes are relevant.

Further work includes to compare the behaviour of feature weighting methods with respect to that of the feature selection models (i.e., all weights are set to 0 or 1). We are also interested in exploring the concurrent use of both approaches, that is, firstly applying a feature selection scheme to eliminate the irrelevant and redundant attributes and afterwards, employing a feature weighting algorithm to determine the different relevance levels.

The topic of this paper concerns methods that set feature weights in the distance functions. An alternative and frequently used approach for improving classification accuracy involves assigning weights to instances themselves (Salzberg, 1991; Brighton and Mellish, 2002). In brief, instance weighting bias the prediction of classifiers by emphasizing the contributions of some instances over others. According to this framework, future research is also addressed to determine the applicability of several feature weighting methods to instance weighting.

REFERENCES

Aha, D. W. (1992). Tolerating noise, irrelevant and novel attributes in instance-based learning algorithms. *Int. Journal of Man-Machine Studies*, 36(2):267–287.

Aizerman, M., Braverman, E., and Rozonoer, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:917–936.

Amemiya, T. (1985). *Advanced Econometrics*. Harvard University Press, Cambridge, MA.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Chapman & Hall, New York.

Brighton, H. and Mellish, C. (2002). Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery*, 6(2):153–172.

Cost, S. and Salzberg, S. L. (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10(1):57–78.

Danuser, G. and Stricker, M. (1998). Parametric model-fitting: from inlier characterization to outlier detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):263–280.

Dasarathy, B. V. (1990). *Nearest neighbor (NN) norms: NN pattern classification techniques*. IEEE Computer Society Press, Los Alamitos, CA.

Domeniniconi, C., Peng, J., and Gunopulos, D. (2002). Locally adaptive metric nearest-neighbor classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9):1281–1285.

Kohavi, R., Langley, P., and Yun, Y. (1997). The utility of feature weighting in nearest-neighbor algorithm. In *Proc. of 9th European Conf. on Machine Learning*, pages 85–92, Prague, Czech Republic. Springer Verlag.

Kononenko, I. (1994). Estimating attributes: analysis and extensions of relief. In *Proc. of 7th European Conf. on Machine Learning*, pages 171–182, Catania, Italy. Springer Verlag.

Langley, P. and Iba, W. (1993). Average-case analysis of a nearest-neighbor algorithm. In *Proc. of 13th Int. Joint Conf. on Artificial Intelligence*, pages 889–894.

Montoliu, R. and Pla, F. (2001). Multiple parametric motion model estimation and segmentation. In *Proc. of Int. Conf. on Image Processing*, pages 933–936, Thessaloniki, Greece.

Murphy, P. M. (1995). Uci repository of machine learning. http://www.ics.uci.edu/AI/ML/MLDBRepository.html. Department of Information and Computer Science, University of California, Irvine, CA.

Paredes, R. and Vidal, E. (2000). A nearest neighbor weighted measure in classification problems. In *Pattern Recognition and Applications. Frontiers in Artificial Intelligence and Applications*, pages 44–50, Amsterdam, The Netherlands. IOS Press.

Salzberg, S. L. (1991). A nearest hyperrectangle learning method. *Machine Learning*, 6(3):251–276.

Wettschereck, D., Aha, D. W., and Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for class of lazy learning algorithms. *Artificial Intelligence Review*, 11(1-5):273–314.