# A Nearest Neighbor Weighted Measure In Classification Problems

R. Paredes and E. Vidal
Instituto Tecnológico de Informática,
Universidad Politécnica de Valencia, 46071 Valencia, Spain.
rparedes@iti.upv.es, evidal@iti.upv.es

*Proceedings of the VII SNRFAI, Bilbao 1999*

**Abstract**

A weighted dissimilarity measure in vectorial spaces is proposed to optimize the performance of the nearest neighbor classifier. An approach to find the required weights based on gradient descent is presented. Experiments with both synthetic and real data shows the effectiveness of the proposed technique.

**Keywords:** Nearest Neighbour, Classification, Weighted Measures.

## 1 Introduction

The great effectiveness of the Nearest Neighbor (NN) rule when the number of *prototypes*, labeled pints in a vector space, is going to infinity is well known [1]. However in most real situations the number of prototypes uses to be very small and often leads to a dramatic loose of NN performance.

Many researches have studied this small-data-set problem and have proposed variations of the NN rule and the k-NN rule to improve classification error rate [2, 3, 4, 5, 6, 7, 8, 9, 10].

Consider the following general statistical statement of a Pattern Recognition classification problem with two-class and identical a priori probabilities: Let $D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be a training data set (prototypes) of independent identically distributed random variable pairs, where $Y_i \in \{0, 1\}, 1 \leq i \leq n$, and let $X$ be an observation from the same distribution. Let the label of $X$ be estimated through a classification rule $g_n$, based on $D_n$, as $Y = g_n(X)$. The probability of error is $R_n = P\{Y \neq g_n(X)\}$. Devroye states that, for any integer $n$ and classification rule $g_n$, there exists a distribution of $(X, Y)$ with Bayes risk $R^* = 0$ such that the expection of $R_n$ is $E(R_n) \geq \frac{1}{2} - \varepsilon$, where $\varepsilon > 0$ is an arbitrary small number. This theorem states that even though we have rules, such as the K-NN rule, that are universally consistent, that is, they *asymptotically* provide optimal performance for any distribution, their *finite* sample performance can always be extremely bad for some distributions.

For these reasons a new distance measure is here proposed that tries to improve the NN classification in small data sets situations.

The proposed weighted measure can be seen as a generalization of the simple weighted $L_2$ dissimilarity in a d-dimensional space:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{d} {\sigma_j}^2 (x_j - y_j)^2} \tag{1}$$

where $\sigma_i$ is the weight of the i-th dimension. Assuming a classification problem into $m$ different classes, a natural extension of (1) is:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^{d} \sigma_{cj}^2 (x_j - y_j)^2} \tag{2}$$

where $c = class(\mathbf{x})$. We will refer to this extension as *"WL2 dissimilarity"*. If $\sigma_{ij} = 1$, $1 < i < m$, $1 < j < d$, the weighted measure is just the $L_2$ metric. On the other hand, if the weights are the inverse of the variances in each dimension, the Mahalanobis distance is obtained. Weights can also be computed as class-dependent inverse variances, leading to a measure that will be refered to as *class-dependent Mahalanobis dissimilarity*.

In the general case (2) does not behave as a metric, since $d(\mathbf{x}, \mathbf{y})$ can be different from $d(\mathbf{y}, \mathbf{x})$ if $class(\mathbf{x}) \neq class(\mathbf{y})$. In this most general setting, we are interested in finding an $m \times d$ weight matrix, $M$, which optimizes the WL2-based NN classification performance.

$$M = \begin{pmatrix} \sigma_{11} & \dots & \sigma_{1d} \\ \vdots & & \vdots \\ \sigma_{m1} & \dots & \sigma_{cd} \end{pmatrix}$$

# 2 Approach

In order to optimize the error rate of the NN classifier with the WL2 dissimilarity measure a specific criterion *index* is proposed. This index is pretended to be binded to the error rate of the classifier, and the problem is to find the $M$ matrix that minimizes this index.

## 2.1 Index

Under the proposed framework, we can expect that NN performance can be improved if distances between points belonging to the same class are made to be small while inter-class distances are made to be large. Given a fixed training data set, $S$, this suggest trying to optimize the following criterion index:

$$J(M) = \sum_{\mathbf{x} \in S} \frac{d(\mathbf{x}^{=}_{nn}, \mathbf{x})}{d(\mathbf{x}^{\neq}_{nn}, \mathbf{x})} \tag{3}$$

where $\mathbf{x}^{=}_{nn}$ is the nearest neighbor of $\mathbf{x}$ in the same class, $(class(\mathbf{x}) = class(\mathbf{x}^{=}_{nn}))$ and $\mathbf{x}^{\neq}_{nn}$ is the nearest neighbor of $\mathbf{x}$ in a different class, $(class(\mathbf{x}) \neq class(\mathbf{x}^{\neq}_{nn}))$.

## 2.2 Gradient Descent

In order to find a matrix $M$ that minimizes (3) a gradient descent procedure is proposed:

$$\sigma_{ij}^{(k+1)} = \sigma_{ij}^{(k)} - \frac{\partial(J(M))}{\partial \sigma_{ij}^{(k)}} \tag{4}$$

where $\sigma_{ij}^{(k)}$ denotes the value of $\sigma_{ij}$ at iteration $k$ of the descent algorithm. By developing the partial derivatives in (4) the following update equations are obtained:

$$\sigma_{ij}^{(k+1)} = \sigma_{ij}^{(k)} - \frac{\mu_{ij}\sigma_{ij}^{(k)}(x^{=}_{nn_j} - x_j)^2}{d(x, x^{=}_{nn})d(x, x^{\neq}_{nn})} \quad \forall \mathbf{x} \in S : class(\mathbf{x}) = i \tag{5}$$

$$\sigma_{ij}^{(k+1)} = \sigma_{ij}^{(k)} + \frac{d(x, x^{=}_{nn})\mu_{ij}\sigma_{ij}^{(k)}(x^{\neq}_{nn_j} - x_j)^2}{d(x, x^{\neq}_{nn})^3} \quad \forall \mathbf{x} \in S : class(\mathbf{x}) \neq i \wedge clase(\mathbf{x}^{\neq}_{nn}) = i \tag{6}$$

where $\mu_{ij}$ is a step factor (or "learning rate") associated to dimension $j$ in class $i$ (typically $\mu_{ij} = \mu \ \forall i, j$). This descent procedure stops when no significant change in $J(M)$ id observed.

It is interesting to note that the computations involved in (5) and (6) implicitly entail computing the NN of each $\mathbf{x} \in S$, according to the WL2 dissimilarity corresponding to the current values of the weights $\sigma_{ij}$. Therefore, as a byproduct, a Leaving-One-Out estimation of the error rate of the NN classifier whit the weighted measure can readily be obtained.

Figure 1 shows a typical evolution of this algorithm, as applied the so called "monkey problem" data set which will be described in Section 3.

## 2.3 Finding adequate solutions in adverse situations.

A potential drawback of the proposed gradient descent algorithm is that if the impact of the additive factor in (6) is not sufficiently important, the algorithm tends to set all $\sigma_{ij}$ to zero.

Consider the following two-classes problem, each class having 500 two-dimensional points (figure 2). Class A is a mixture of two Gaussian distributions, the first distribution has a standard deviation of $\sqrt{10}$ in the $x_1$ dimension and a unit standard deviation in the $x_2$ dimension, while the second distribution has a unit standard deviation in the $x_1$ dimension and a standard

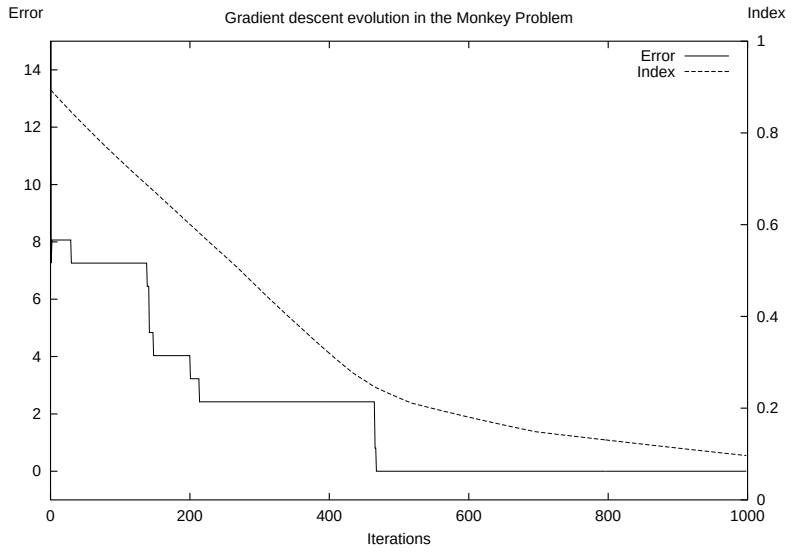


Figure 1: Typical behaviour of the gradient descent algorithm as applied to the "monkey problem" data set. Classification error is estimated through Leaving-One-Out.

deviation of $\sqrt{10}$ in the $x_2$ dimension, both distributions centered at (0,0). Class B is a Gaussian distribution with unit standard deviation in the $x_1$ dimension and a standard deviation of $\sqrt{10}$ in the $x_2$ dimension, centered at (6,0). Note the relatively large interclass overlapping on the $x_1$ dimension.

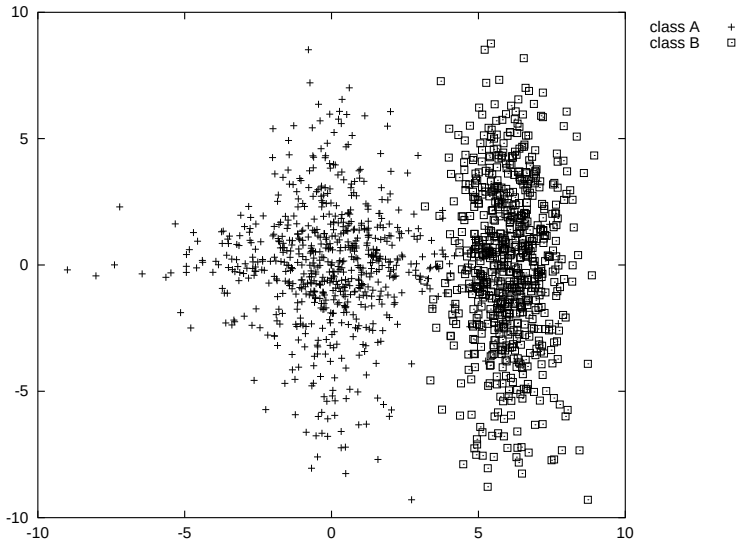


Figure 2: Two-class problem with mixture Gaussian distributions and interclass overlapping.

In situations like this one, the convergence of the proposed gradient descent procedure can be problematic. In fact, as shown in Figure 3, with this data set (and using just unit initialization weights a constant value for the step factor $\mu$), the estimated error rate tends to worsen as the

proposed criterion index $J(M)$ (3) does increase through successive iterations.

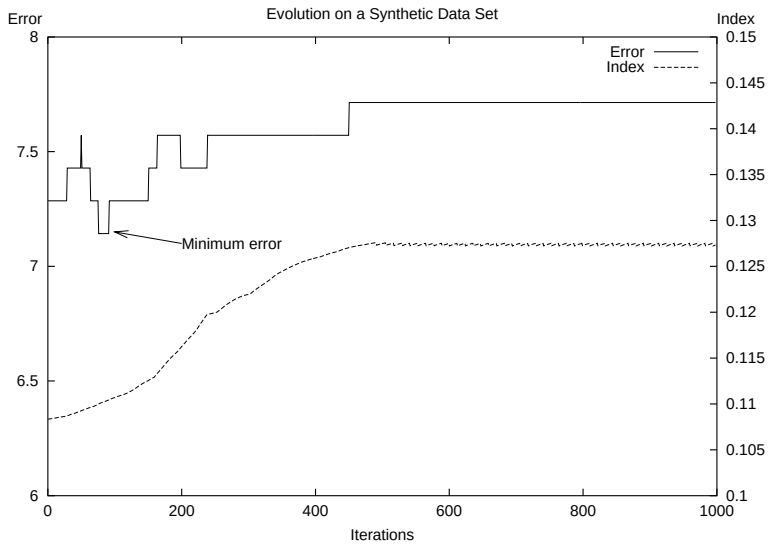


Figure 3: Divergent evolution of the gradient descent algorithm with the "adverse" synthetic data of Figure 2. The error rate tends to increase, but there is a step in which it is minimum.

This undesirable behaviour is due to the fact that all $\sigma_{ij}$ tend to zero until the algorithm stops. Nevertheless, it is interesting to note that, even through this divergent behaviour, a minimum error estimate is achieved at a certain step of the procedure, as it can be seen in Figure 3. In other words, a low value of $J(M)$ does not always necessarily mean a low value of the NN classifier error rate; as mentioned in Section 2, it was only an assumption. This suggests us that, rather than supplying the weight values obtained at the end of the descent procedure, a better choice for $M$ in general would be supplying the weights that led to the minimum estimated error rate. In typical cases, such as that shown in Figure 1, this minimum is achieved at the convergence point of the descent procedure, while in adverse situations, such as that in Figure 3, the minimum-error weights will hopefully be a better choice than the standard ($L_2$ or Mahalanobis) distance and, certainly, will not be as bad as the weights obtained at the end of the process.

# 3 Experiments

Different corpora have been used from the UCI Repository Of Machine Learning Databases and Domain Theories [12]. A short description of these corpora is given below:

- Australian: 690 prototypes, 14 features, 2 classes. Divided into 10 sets for cross-validation.
- Balance: 625 prototypes, 4 features, 3 classes. Divided into 10 sets for cross-validation.

- DNA: Training set of 400 prototypes (20% of the original set) [1]. Test set of 1186 prototypes, 180 features, 3 classes.
- Diabetes: 768 prototypes, 8 features, 2 classes. Divided into 11 sets for cross-validation.
- Heart: 270 prototypes, 13 features, 2 classes. Divided into 9 sets for cross-validation.
- Letter: Training set of 3000 prototypes (20% of the original set), Test set of 5000 prototypes, 16 features, 26 classes.
- Monkey-Problem-1: Training of 124 prototypes, Test of 432 prototypes, 6 features, 2 classes.
- Vehicle: 846 prototypes, 18 features, 4 classes. Divided into 9 sets for cross-validation.

These data sets have both numeric and categorical features. Each categorical feature has been replaced by $n$ binary features, where $n$ is the range of the categorical feature. For example, in a hypothetical set of data with two features: Age (Continuous) and Sex (Categorical: M,F), the categorical feature would be replaced by two binary features; i.e., Sex=M will be represented as (1,0) and Sex=F as (0,1). The continuous feature will not suffer any change, leading to an overall representation in three dimensions.

Most of the UCI data sets are small. In these cases *N-Fold Cross-Validation* [13] has been applied to obtain the classification results. Each corpus is divided into $N$ blocks using $N-1$ blocks as training set and the remaining block as a test set. Therefore, each block is used exactly once as a test set. The number of cross validation blocks, $N$, is specified for each corpus in the UCI documentation. For the larger DNA, Monkey and Letter corpora a single specific partition into training and test sets is provided by UCI and, in these cases, no cross validation was necessary.

# 4 Results

Experiments with both the NN and the k-NN rules have been carried out using the $L_2$, a "*class-dependent*" Mahalanobis, and our WL2 dissimilarity measures. Class-dependent Mahalanobis consists in weighting each dimension by the inverse of the variance of this dimension in each class.

Both in the case of the Mahalanobis and the WL2 dissimilarities, computation singularities can appear when dealing with categorical features, which often exhibit *null* class-dependent variances. This problem was solved by using the overall variance as a "back-off" smoothing for the null values.

[1] The size of the training data has been intentionally reduced for saving computing time and showing the capabilities of our method for working with small-data-sets.

In the case of k-NN, the results reported for each method correspond to the value of $k, 1 < k < 21$, which gave best results for this method. Initialization values for training the WL2 weights were selected according to the following simple rule, based on leaving-one-out 1-NN performance on the training data of conventional methods: If raw $L_2$ outperforms Mahalanobis, then set all initial $\sigma_{ij} = 1$; otherwise, set them to the inverse of the corresponding training data standard deviation. Similarly, the step factors, $\mu_{ij}$, are set to a small constant (0.001) in the former case and to the inverse of the standard deviation in the later. Results are summarized in table 1.

| | $NN - L_2$ | $NN - MAH$ | **NN-WL2** | $K - L_2$ | $K - MAH$ | **K-WL2** |
|---|---|---|---|---|---|---|
| Australian | 65.73% | 82.94% | **82.64%** | 69.26% | 85.29% | **85.14%** |
| Balance | 78.83% | 68.0% | **69.16%** | 91.16% | 91.16% | **91.16%** |
| Diabetes | 69.94% | 68.3% | **67.34%** | 76.5% | 73.77% | **74.18%** |
| DNA | 70.82% | 77.23% | **91.9%** | 82.2% | 77.23% | **91.9%** |
| Heart | 52.10% | 71.26% | **72.41%** | 59.0% | 79.31% | **79.31%** |
| Letter | 87.82% | 84.02% | **89.48%** | 87.82% | 84.02% | **89.48%** |
| Monkey | 78.7% | 87.04% | **100%** | 83.33% | 87.33% | **100%** |
| Vehicle | 65.3% | 66.79% | **69.75%** | 66.54% | 70.25% | **70.49%** |

Table 1: Performance of several methods on different data sets. Results in boldface correspond to the here proposed WL2 techniques.

WL2 outperforms conventional methods in many cases. The greatest improvement is obtained in *monkey-problem1*, a categorical corpus with a small number of features and only two classes. Similarly good improvement is obtained for the *DNA* corpus, which is also a corpus with categorical data, but with far more features (180) and 3 classes. For the *letter* corpus, with 16 continuous features and 26 classes, a moderate but significant gain of classification accuracy is obtained. For other data sets, only marginal improvement or very similar results with respect to conventional techniques are obtained. The only case in which (k-NN) results are slightly but significantly worse for the WL2 method is the *Diabetes* corpus.

Since the weights are the same for all points in the same class, results were expected to be data-distribution dependent. That is, the accuracy of the method depends on the distribution of the points in each class and, most importantly, on the amount of interclass overlapping, as it was suggested by the example shown in figures 2 and 3.

# 5 Concluding remarks

A general weighted measure for the NN classification rule has been presented. In order to obtain a good matrix of weights, gradient-descent minimization of an appropriate criterion index is

proposed. Results obtained for several data sets are promising. Nevertheless, other optimization methods can be devised to minimize the proposed index and new indexes can be proposed which would probably lead to improved performance.

These issues will be studied in future work. Another issue to be studied is the behaviour of a new weighting scheme, where weights are assigned to each *prototype* –rather than (or in addition to) each *class*. This more "local" configuration of the dissimilarity function is expected to lead to a more data-independent overall behaviour of the corresponding k-NN classifiers.

# References


[1] T.M. Cover and P.E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Infromation Theory*, 13(1):21–27, January 1967.

[2] Ivan Tomek. A generalization of the k-nn rule. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(2):121–126, February 1976.

[3] Keinosuke Fukunaga and Thomas E. Filck. The 2-nn rule for more accurate nn risk estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(1):107–112, January 1985.

[4] Andrew Luk and James E.S. Macleod. An alternative nearest neighbour classification scheme. *Pattern Recognition Letters*, 4:375–381, 1886.

[5] K. Urahama and Y.Furukawa. Gradient descent learning of nearest neighbor classifiers with outlier rejection. *Pattern Recognition*, 28(5):761–768, 1995.

[6] Robert D. Short and Keinosuke Fukunaga. A new nearest neighbor distance measure. In *Proc. 5th IEEE Int. Conf. Pattern Recognition.*, pages 81–86, Miami Beach, FL., 1980.

[7] Robert D. Short and Keinosuke Fukunaga. An optimal distance measure for nearest neighbour classification. *IEEE Trans.. Info. Theory*, 27:622–627, 1981.

[8] K. Fukunaga and T.E.Flick. A parametrically defined nearest neighbour measure. *Patter Recognition Lett.*, 1:3–5, 1982.

[9] K. Fukunaga and T.E.Flick. An optimal global nearest neighbour metric. *IEEE Trans. Pattern Recognition Mach. Intell. PAMI*, 6:314–318, 1984.

[10] J.P.Myles and D.J. Hand. The multi-class metric problem in nearest neighbour discrimination rules. *Pattern Recognition*, 23(11):1291–1297, 1990.

[11] László Györfi Luc Devroye and Gábor Lugosi. *A probabilistic theory of pattern recognition.* Springer-Verlag New York, Inc., 1996.

[12] UCI Repository Machine Learning: http://www.ics.uci.edu/AI/ML/MLDBRepository.html.

[13] S.J.Raudys and A.K.Jain: "Small Sample Effects in Statistical Pattern Recognition: Recomendations for Practitioners and Open Problems". IEEE Trans on PAMI, vol. 13, n. 3, pp. 252-263, 1991.