Intelligent Information Systems VII
Proceedings of the Workshop held in
Malbork, Poland, June 15-19 , 1998

pages 288- 291

A framework for similarity-based classification methods.

Włodzisław Duch

Department of Computer Methods, Nicholas Copernicus University, Grudziądzka 5,
87-100 Toruń, Poland; e-mail: duch@phys.uni.torun.pl

Abstract. A general framework for similarity-based (SB) classification methods is presented. Neural networks, such as the Radial Basis Function (RBF) and the Multilayer Perceptrons (MLPs) models, are special cases of SB methods. Many new versions of minimal distance methods are derived from this framework.

Keywords: k-NN, minimal distance methods, neural networks, classification

1 Introduction

Recently one of us [1] presented a general framework for similarity-based classification methods. This framework is extended here and connections between neural methods and SB methods pointed out. Some of the simplest classification algorithms applicable to pattern recognition problems are based on the *k*-nearest neighbor (*k*-NN) rule [2]. Each training data vector is labeled by the class it belongs to and is treated as a reference vector. During classification *k* nearest reference vectors to the unknown (query) vector **X** are found, and the class of vector **X** is determined by a 'majority rule'. The probability of assigning a vector **X** to class C_i is $p(C_i|\mathbf{X}) = N_i/k$. The Hamming neural network [3] is explicitly based on the nearest neighbor rule, computing distances for the binary patterns and finding the maximum overlap (minimum distance) with the prototype vectors. A general framework for similarity based methods is presented here and relations with neural methods are explained. A short discussion is presented at the end.

2 A framework for the minimal distance methods

The problem of classification is stated as follows: given a set of class-labeled training vectors $\{\mathbf{X}^p, \mathbf{C}(\mathbf{X}^p)\}, p = 1..K$, where $\mathbf{C}(\mathbf{X}^p)$ is the class of $\mathbf{X}^p$, and a vector **X** of an unknown class, use the information provided in the distance $d(\mathbf{X},\mathbf{X}^p)$ to estimate probability of classification $p(C_i|\mathbf{X};M)$, where M describes the classification model used (parameter values and procedures employed). The empirical risk matrix $R(C_i,C_j)$ measures the risk of misclassification (in the simplest case $R(C_i,C_j)=\delta_{ij}$) and the error function that should be minimized is:

$$E(M)=\sum_p\sum_k R(C(\mathbf{X}_p),C_k)p(C_k|\mathbf{X}_p;M) \tag{1}$$

where p runs over all vectors in the training dataset, k runs over all classes and $C(\mathbf{X}_p)$ is the true class of the vector $\mathbf{X}_p$. Parameters and procedures defining the model M should be found that minimize the probability of misclassification and thus maximize the probability of correct classification (although there are some differences between the two approaches they will be ignored here).

A general model of an adaptive system used for classification may include all or some of the following:
$M=\{k,d(\cdot;r),G(d(\cdot)),\{\mathbf{D}^n\},E[\cdot],K(\cdot)\}$, where
k is the number of reference vectors taken into account in the neighborhood of $\mathbf{X}$;
r is the maximum size of the neighborhood considered;
$d(\cdot;r)$ is the function used to compute similarities (distances);
$G(d(\mathbf{X},\mathbf{X}^p))$ is the weighting function estimating contribution of reference vector $\mathbf{X}^p$ to the classification probability;
$\{\mathbf{D}^n\}$ is the set of reference vectors created from the training set of vectors $\{\mathbf{X}^p\}$;
$E[\cdot]$ is the total cost function that is minimized at the training stage;
$K(\cdot)$ is a kernel function, scaling the influence of the error, for a given training example, on the total cost function.

In addition an adaptive system may include several models M_l and an interpolation procedure to select between different models or combine results of a committee of models. In the simplest version $p(C_i|\mathbf{X};M)$ is parametrized by $p(C_i|\mathbf{X};k,d(\cdot),\{\mathbf{X}^n\}\})$, and the k-NN method is obtained. Instead of enforcing exactly k neighbors the radius r may be used as an adaptive parameter. The number of classification errors, or the probability of classification $p(C_i|\mathbf{X};r)=N_i/\sum_j N_j$, is then optimized using the leave-one-out method or a validation set. Introduction of variable radiuses r_i for each reference vector instead of one universal radius in the input space increases the number of adaptive parameters significantly. Development along this line leads to the Restricted Coulomb Energy (RCE) classifier introduced by Reilly, Cooper and Elbaum [4] which may be treated as the hard limit approximation of the Gaussian-based RBF network.

The **the conical radial function** (favorite fuzzy logic membership function) is suitable for the weighting function: it is zero outside the radius r and $1-d(\mathbf{X},\mathbf{D})/r$ inside this radius. Classification probability:

$$p(C_i|\mathbf{X};r)=\frac{\sum_{n\in C_i}G(\mathbf{X};\mathbf{D}^n,r)}{\sum_n G(\mathbf{X};\mathbf{D}^n,r)};G(\mathbf{X};\mathbf{D},r)=\max\left(0,1-\frac{d(\mathbf{X},\mathbf{D})}{r}\right) \tag{2}$$

$G(\mathbf{X};\mathbf{D},r)$ is the weight estimating contribution of reference vector at the distance $d(\mathbf{X},\mathbf{D})$. Radial Basis Function (RBF) networks using Gaussian or inverse multiquadratic transfer functions are a particular example of the soft weighting minimal distance algorithm. Other useful weighting functions include a combination of two sigmoidal functions: $\sigma(||\mathbf{X}-\mathbf{D}^{n}||-r)-\sigma(||\mathbf{X}-\mathbf{D}^{n}||+r)$. If r_k is the distance to the k-th neighbor and $r_k \geq r_i, i=1..k-1$ then a conical weighting function $G(d)=1-d/\alpha r_k, \alpha>1$ has values $G(0)=1$ and $G(r_k)=1-1/\alpha$; for large α the cone is very broad and all vectors receive the same attention; for α approaching 1 the furthest neighbor has weight approaching zero. Multilayer perceptron (MLP), the most common neural networks models, compute in each node:

$$\sigma(\mathbf{W}\cdot\mathbf{X}) = \sigma\left(\frac{1}{2}(||\mathbf{W}||^2+||\mathbf{X}||^2-||\mathbf{W}-\mathbf{X}||^2)\right) = \sigma(I_{max}-d(\mathbf{W},\mathbf{X})) \tag{3}$$

For normalized input vectors (normalization in $N+1$ dimensional space is recommended) sigmoidal functions evaluate the influence of reference vectors $\mathbf{W}$, depending on their distance $d(\mathbf{W},\mathbf{X})$, on classification probability $p(C_i|\mathbf{X};\{\mathbf{W},\theta\})$. As a function of a distance $\sigma(I_{max}-d(\mathbf{W},\mathbf{X}))$ monotonically decreases, at $d(\mathbf{W},\mathbf{X})=I_{max}$ reaching the value of 0.5. For normalized $\mathbf{X}$ but arbitrary $\mathbf{W}$ the range of sigmoid argument is in the $[-|\mathbf{W}|,+|\mathbf{W}|]$ interval. Unipolar sigmoid has a maximum curvature around ± 2.4, therefore smaller weights of the norm mean that the network operates in almost linear regime. Sigmoidal functions used in MLPs estimate the influence of weight vectors according to the distance between weight and the training vectors, combining many such estimations to compute the final output.

Calculation of similarity is most often based on Minkowski's metric $d(\mathbf{X},\mathbf{X}';g)^{\alpha}=\sum_i^N g_i(X_i-X_i')^{\alpha}$. In the simplest RBF version with Gaussian functions only one parameter – dispersion – is optimized. Independent optimization of all N components of dispersion vector has the same effect as optimization of the scaling factors g_i in soft-weighted NN-r method. Calculation of distances may also be parameterized in a different way around each reference vector, providing a large number of adaptive parameters. To select features useful for classification and to lower the complexity of the classification model the cost function should have an additional penalty term, such as the sum of all g_i^2. Features for which the product of the scaling factors $g_i \max_{jk}|X_i^{(j)}-X_i^{(k)}|$ is small may be deleted without significant loss of accuracy. Nonlinear similarity function that is approximately constant within a cluster and rapidly changes between clusters belonging to different classes is obtained replacing (X_i-X_i') terms in the Minkovsky metric with $\sum_{j=1}^{n_i} a_{ij}\sigma(b_{ij}(X_i-X_i')-c_{ij})$. Such similarity measures provide useful a,b,c parameters for minimization of in-class variances and maximization of between-class variances.

Active selection of reference vectors starts from K-means or other clusterization techniques to select a relatively small number of initial reference vectors close to the cluster centers. Classification accuracy is checked on the remaining set (using k-NN or NN-r rule) and each time an error is made the vector is moved from the remaining to the reference set. An alternative approach starts from the whole training set and removes those vectors that have all k nearest vectors from the same class. The reference vector $\mathbf{D}^n$ in the neighborhood of a training vector $\mathbf{X}$ may be updated using: $\mathbf{D}^n_{new}=\mathbf{D}^n_{old}+\eta\delta_{\pm}(C(\mathbf{X}),C(\mathbf{D}^n_{old}))(\mathbf{X}-\mathbf{D}^n_{old})$, where η is the learning rate, decreasing during training, and $\delta_{\pm}$ is $+1$ if $\mathbf{X}$ and $\mathbf{D}^n_{old}$ belong to the same class or -1 otherwise. Virtual Support Vectors added to the reference set may improve classification rates.

The choice of kernel function in the measure of classification error may be based on local regression [6]: $E(\mathbf{X};M)=\sum_i K(d(\mathbf{X}^i,\mathbf{X}^{ref}))(F(\mathbf{X}^i;M)-y^i)^2$, where y^i are the desired values for $\mathbf{X}^i$ and $F(\mathbf{X}^i;M)$ are the values predicted by the model M; here the kernel function $K(d)$ measures the influence of the reference vectors on the total error.

3 Discussion

The similarity based framework accommodates many classification methods. We have found very few methods in the literature that try to improve upon the simple *k*-NN scheme. Value Difference Metric is a probabilistic similarity measure gaining popularity [5]. Hastie and Tibshirani [8] write about adaptive *k*-NN classification from the linear discriminant point of view, advocating the use of several local metrics. Friedman [9] proposed adaptation of metric based on a tree-like interactive partitioning of the data. Atkenson, Moor and Schaal [6] discuss locally weighted regression techniques, various metric and kernel functions applied to approximation problems.

All these proposals and many more may be accommodated in the general framework presented here. Identification of the best combination of procedures and adaptive parameters should allow for improvements of *k*-NN as well as neural classifiers. Performance of various methods described here depends on the nature of the data given for classification and remains a subject of further empirical study. So far we have tested only a few simplest methods obtaining very encouraging results.

Acknowledgments: Support of the KBN is gratefully acknowledged.

References

1. W. Duch, Neural minimal distance methods, Proc. 3-rd Conf. on Neural Networks and Their Applications, Kule, Poland, Oct. 14-18, 1997
2. P.R. Krishnaiah, L.N. Kanal, eds, Handbook of statistics 2: classification, pattern recognition and reduction of dimensionality (North Holland, Amsterdam 1982)
3. P. Floreen, The convergence of Hamming memory networks, *Trans. Neural Networks* 2 (1991) 449–457
4. D.L. Reilly, L.N. Cooper, C. Elbaum, A neural model for category learning, *Biological Cybernetics* 45 (1982) 35–41
5. D.L. Waltz, Memory-based reasoning, in: M. A. Arbib, ed, *The Handbook of Brain Theory and Neural Networks* (MIT Press 1995), pp. 568–570
6. C.G. Atkenson, A.W. Moor and S. Schaal, Locally weighted learning, *Artificial Intelligence Review* (submitted, 1997)
7. H.J. Hamilton, N. Shan, N. Cercone, RIAC: a rule induction algorithm based on approximate classification, Tech. Rep. CS 96-06, Regina University 1996
8. T. Hastie, R. Tibshirani, Discriminant adaptive nearest neighbor classification, *IEEE PAMI* 18 (1996) 607-616
9. J. H. Friedman, Flexible metric nearest neighbor classification, *Technical Report, Dept. of Statistics, Stanford University* 1994