An Evaluation of the WPE Algorithm Using Tangent Distance.

R. Paredes, E. Vidal and D. Keysers
Instituto Tecnológico de Informática,
Universidad Politécnica de Valencia, 46071 Valencia, Spain.
rparedes@iti.upv.es, evidal@iti.upv.es
keysers@I6.Informatik.RWTH-Aachen.DE

Abstract

Weighting Prototype Editing (WPE) is a novel approach to edit a given set of prototypes so that the resulting set can outperform the original one in terms of the Nearest Neighbor (NN) classification accuracy. This technique is applied in this work along with an interesting dissimilarity measure between pixel maps, known as Tangent Distance (TD). Experiments on the USPS handwriting digits benchmark corpus are presented, with results showing the capability of the WPE to improve the already good results based on TD NN classification.

Keywords: Editing, Condensing, Nearest Neighbour, Weighted Prototypes, Tangent Distance.

1 Introduction

The Nearest neighbor (NN) rule is a very common and successful approach for many pattern recognition applications. While the asymptotic optimality of this rule is well know [1], when the number of prototypes is not large enough performance can degrade dramatically. Unfortunately, this is quite often the case in real applications. One idea to circumvent this problem is the use of *Editing Techniques* [11, 10, 8, 2, 6, 3] which attempt to "clean" interclass overlap regions, thereby leading to smoother NN-based decision boundaries between classes and hopefully increasing classification accuracy.

In [7] a new editing technique called *"Weighting Prototype Editing (WPE)"* was introduced [1]. Rather than aiming at asymptotically good performance as most editing techniques do, the WPE tries to obtain a good editing rule for *each given prototype set*. This is achieved by first learning an adequate assignment of a weight to each prototype and then pruning out those prototypes having large weights. As a result, WPE was expected to outperform other traditional editing techniques when the number of available prototypes is small. Moreover, since the prototype weights are explicitly optimized for each prototype set, performance was expected to be uniformly good for varying sizes and/or dimensionalities of the training sets of prototypes.

These expectations could be successfully confirmed in [7] throughout a series of experiments using common benchmark synthetic data sets. Moreover, as compared with *Wilson, MultiEdit* and *Cross-Validation Editing*, only WPE was actually able to achieve error rates consistently close to the corresponding Bayes bounds, despite significantly decreasing the number of prototypes and increasing the data dimension.

An interesting feature of WPE, observed in these experiments, is that the optimization algorithm tends to assign large weights not only to the prototypes laying on the interclass Bayes confusion regions (as required for the editing mechanism), but also to prototypes which are deeply embedded into their corresponding Bayes acceptance regions. Correspondingly, by pruning prototypes with large weights, a certain degree of prototype *Condensing* is achieved along with the *Editing* effect initially aimed at.

We should emphasize that this combined Editing/Condensing effect is achieved by WPE with *complete independence of the* metric *adopted*. Therefore, it can be generally used to improve the results of many Pattern Recognition tasks for which good, may be sophisticated classification techniques are already available. If these techniques can be seen under a NN-based classification scheme then, no matter how complex (even *non-vectorial*) data representation is used, or how elaborate the metric to compare these representations is, WPE is easily applied. If the available training data contains confusing and/or redundant prototypes, the WPE can take care of removing the required prototypes such that the expected test-set error rates become lower.

In the present work WPE is applied to a real task for which good results have already been achieved using appro-

[1] C source code is available at: "http:\\www.iti.upv.es\~rparedes\wpedit"

priate techniques. It consists in the classification of handwriting characters from the USPS corpus. This corpus is known to be a hard corpus, for which a 2.5% human error rate has been measured. One of the most successful automatic techniques that have been applied to this corpus is the *Tangent Distance* [9], which achieves error rates as low as 3.4%, as compared, for example, with 5.6% obtained by NN classification using the Euclidean Distance between normalized pixel maps.

As we will see, the WPE technique can be straightforwardly applied along the the TD. Given the relatively high intrinsic error rate of USPS (as assessed by its human error rate), it can be expected that a good prototype Editing/Condensing process will actually help improving the performance over that achieved by using the raw training data.

2 Tangent Distance

Tangent Distance (TD) is a locally invariant distance measure, introduced by SIMARD et al. (see e.g. [9]), which proved to be especially effective in the domain of digit recognition [4]. When an image is transformed (e.g. scaled and rotated), the set of all transformed patterns is a manifold in pattern space. The distance between two patterns can now be defined as the minimum distance between their respective manifolds, but exact computation of this measure is a hard non-linear optimization problem. Instead, small transformations of the pattern can be approximated by a tangent subspace to the manifold. This first-order approximation of the manifold is spanned by a set of tangent vectors that can be computed as the derivatives of the respective transformations or estimated from the data within a statistical framework [5]. Distances to the linear subspaces and between them can be efficiently calculated.

The distances used in the present experiments were calculated using the derivatives of the affine transformations (six vectors for translations, scaling, rotation and axis deformations) and the derivative of the line-thickness transformation [4].

3 Weighted Prototype Editing

Let $\mathcal{S}$ be a representation space[2] with m classes and let $d : \mathcal{S} \times \mathcal{S} \rightarrow \Re$ be an appropriate *dissimilarity* in $\mathcal{S}$. Let $T = \{(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2), \ldots, (\mathbf{x}_n, c_n)\}$ be a training set, where $\mathbf{x}_i \in \mathcal{S}$ and $c_i \in \{1, 2, ..., m\}$ and let $\mathbf{y} \in \mathcal{S}$ be a *test sample*. The NN rule assigns $\mathbf{y}$ to a class c_j such that $d(\mathbf{y}, \mathbf{x}_j)$ is minimum.

[2]Note that $\mathcal{S}$ needs not be a *vector space*. Nevertheless, for that sake of clarity, elements of $\mathcal{S}$ will be typeset in boldface.

Following [7], a *"Weighted Prototype"* dissimilarity measure is defined as:

$$d_w(\mathbf{y}, \mathbf{x}) = \sigma_x d(\mathbf{y}, \mathbf{x}) \tag{1}$$

where $\sigma_x \in [0, \infty]$ is a weight associated to the prototype $\mathbf{x}$. According to [7], optimal weights are those which minimize the following criterion index:

$$J(\boldsymbol{\sigma}) = \sum_{\mathbf{x} \in T} \sum_{i=1}^{K} \frac{d_w(\mathbf{x}, \mathbf{x}^{=}_{i-nn})}{d_w(\mathbf{x}, \mathbf{x}^{\neq}_{i-nn})} \tag{2}$$

where $\mathbf{x}$ is a prototype of class c, $\mathbf{x}^{=}_{i-nn}$ the i-th nearest prototype of $\mathbf{x}$ in c, and $\mathbf{x}^{\neq}_{i-nn}$ the i-th nearest prototype of $\mathbf{x}$ in a different class. Both $\mathbf{x}^{=}_{i-nn}$ and $\mathbf{x}^{\neq}_{i-nn}$ are assumed to be computed using the weighted distance function d_w.

To find a vector $\hat{\boldsymbol{\sigma}} = [\hat{\sigma}_{x_1}, \ldots, \hat{\sigma}_{x_n}]$ which minimizes (2), a gradient descent method is adopted. To this end, (approximate) partial derivatives of $J(\boldsymbol{\sigma})$ with respect to $\sigma_{\mathbf{x}}$ $\forall \mathbf{x} \in T$ can be easily derived, leading to the following update equations:

$$\sigma_{\mathbf{x}^{=}_{i-nn}} = \sigma_{\mathbf{x}^{=}_{i-nn}} - \frac{\mu \cdot d(\mathbf{x}, \mathbf{x}^{=}_{i-nn})}{\sigma_{\mathbf{x}^{\neq}_{i-nn}} \cdot d(\mathbf{x}, \mathbf{x}^{\neq}_{i-nn})} \tag{3}$$

$$\sigma_{\mathbf{x}^{\neq}_{i-nn}} = \sigma_{\mathbf{x}^{\neq}_{i-nn}} + \frac{\mu \cdot \sigma_{\mathbf{x}^{=}_{i-nn}} \cdot d(\mathbf{x}, \mathbf{x}^{=}_{i-nn})}{\sigma^2_{\mathbf{x}^{\neq}_{i-nn}} \cdot d(\mathbf{x}, \mathbf{x}^{\neq}_{i-nn})} \tag{4}$$

where μ is an appropriate "learning rate" or step factor.

Note that, as a byproduct of computing (3) and (4), a leaving-one-out error rate estimation (LOOER) of the NN classifier with the current d_w is obtained at each step of the gradient descent process [7]. Therefore, by selecting $\hat{\boldsymbol{\sigma}}$ as a vector whose LOOER is the lowest among all $\boldsymbol{\sigma}$'s produced throughout the descent process, the finally supplied weights are guaranteed to outperform the LOOER of the original dissimilarity measure d.

Once the prototype weights are obtained, the actual WPE simply consists in pruning out those prototypes whose weights exceed a certain threshold. By decreasing the threshold value, different *editing/condensing degrees* can be obtained.

4 Experiments

All the results presented here were obtained using the well known US Postal Service handwritten digits recognition corpus (USPS). It contains normalized grey scale images of size 16×16, divided into a training set of 7291 images and a test set of 2007 images. A human error rate of

2.5% performance shows that it is a hard recognition task. Many techniques have been applied to this corpus [4]. The TD technique discussed in section 2 is among the most successful approaches. It achieves a 3.39% error rate by plain NN classification.

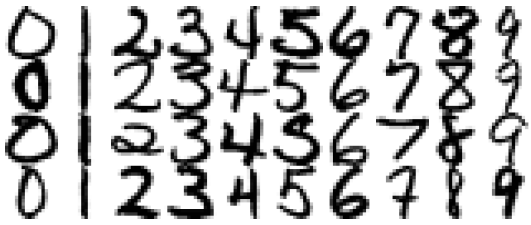

Figure 1. Some examples of the USPS corpus.

The TD has been used as a "black box" distance function for the editing techniques tested; that is, a square matrix of 7291×7291 distances between every pair of training prototypes has been computed using the TD procedures. For comparison purposes, these distances are supplied both to the well known Wilson editing technique [11] and to the WPE technique here proposed.

In the test phase, TDs between test and training images are used for direct NN classification, as well as for classification with the sets edited by Wilson's and the WPE techniques.

Wilson's editing technique needs a parameter k which is the number of NNs used for deciding whether a training prototype is edited or not. In the experiments, values of k ranging from 2 to 10 have been tested. On the other hand, the WPE technique also needs a (not critical [7]) parameter k, and a pruning weight threshold which, in the experiments, has been tested for values ranging from 1 to 1.5.

5 Results

Wilson editing performance is shown in 2. The best result is achieved using the edited training set with $k = 2$. With this value, only 110 prototypes are eliminated from the original set of size 7291.

As it can be seen in figure 2, only for $k = 2$ the result is (slightly) better than that with 1-NN using the whole training set. It is worth noting that this value of k is critical, given the observed rapid degradation of classification results for grater values of k.

The best result achieved by WPE technique is 3.18, a relative improvement of 6.2% over the plain application of the TD method. This very same best result is achieved for all values of k tested: $1 \leq k \leq 10$. The results for $k = 4$ are shown in 3. In this case, the best test result is achieved for training sets edited with threshold values between 1.27 and

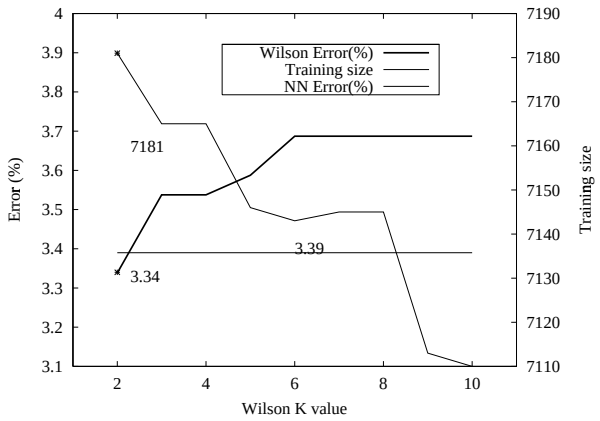

Figure 2. Wilson editing results for different k values.

1.30. Using the threshold 1.27 the training set size is reduced down to 6987 prototypes, that is, 304 training prototypes are eliminated, nearly three times the number of prototypes eliminated by the Wilson technique.

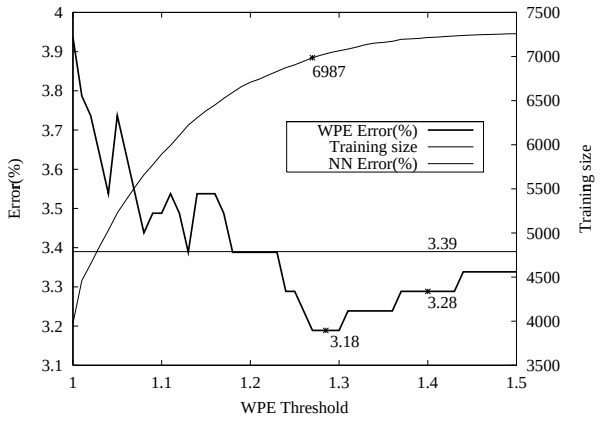

Figure 3. WPE results for different threshold values.

It is important to note that, in this case, the threshold parameter is far less critical than the value of k in Wilson's technique: there are fairly wide range of threshold values with error rates clearly lower than those of NN using the whole training set.

In order to gain some qualitative insight into the capabilities of WPE, figure 4 and 5 shows a selection of images from the set of 304 training prototypes eliminated by WPE with a threshold value of 1.27. All the images fall into one of two subsets: a) images of confusing, badly written digits and b) images of digits written with very typical writing style. Training images of the first subset are very prone to lead to NN classification errors of frequent, well written test digits. This subset corresponds to true *Editing*. On the other

hand, it is very unlikely that prototypes of the second subset are really needed for correct NN classification of any test image, given the large amount of other training-set images very similar to them. These images have been eliminated as a result of the *Condensing* side effect actually achieved by the WPE technique.

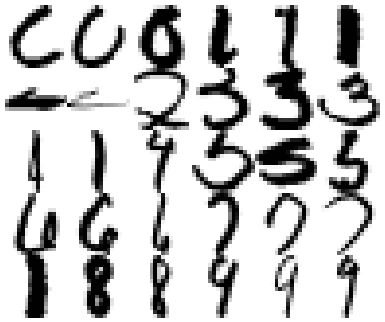

Figure 4. A selection of the badly written digits eliminated by the WPE technique.

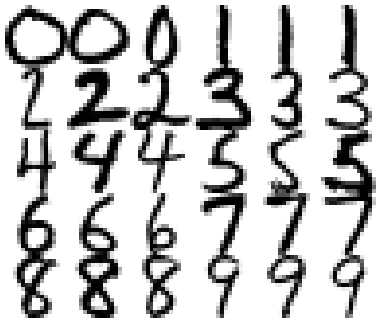

Figure 5. A selection of the very typical writing digits eliminated by the WPE technique.

6 Concluding remarks

The capability of the Weighting Prototypes Editing technique to improve the results of a good NN classifier have been demonstrated on the well known benchmark USPS corpus. The baseline NN classifier provided an error rate as low as 3.39%, using the Tangent Distance metric. This figure was improved by weighting Prototype Editing to 3.18%, a 6.2% relative improvement. This editing procedure also achieved a significant condensing effect, resulting in the elimination of nearly three times more prototypes than with the Wilson's traditional editing technique. A qualitative analysis of the prototypes automatically eliminated by this editing technique provides clear insights into the kind of prototypes whose elimination helps improving the classification accuracy.

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] P.A. Devijver and J. Kittler. *Pattern Recognition. A Statistical Approach*. NJ: Prentice Hall, 1982.

[3] Gyorfi Devroye and Lugosi. *A Probabilistic Theory of Pattern Recognition.* Berlin, Germany: Springer-Verlag, 1995.

[4] Daniel Keysers, Jörg Dahmen, Thomas Theiner, and Hermann Ney. Experiments with an extended tangent distance. In *Proceedings 15th International Conference on Pattern Recognition*, volume 2, pages 38–42, Barcelona, Spain, September 2000.

[5] Daniel Keysers, Wolfgang Macherey, Jörg Dahmen, and Hermann Ney. Learning of variability for invariant statistical pattern recognition. In *ECML 2001, 12th European Conference on Machine Learning*, volume 2167 of *Lecture Notes in Computer Science*, pages 263–275, Freiburg, Germany, September 2001. Springer.

[6] J. Koplowitz and T. Brown. On the relation of the performance to editing in nearest neighbor rules. *PAttern Recognition*, 13(3):251–255, 1981 1981.

[7] R. Paredes and E. Vidal. Weighting prototypes. a new editing approach. In *XV International Conference on Pattern Recognition. 15th ICPR.*, 2000.

[8] C. Penrod and T. Wagner. Another look at the edited nearest neighbor rule. *IEEE Trans. Syst., Man, Cyber.*, SMC-7:92–94, 1977.

[9] P. Simard, Y. Le Cun, J. Denker, and B. Victorri. Transformation invariance in pattern recognition — tangent distance and tangent propagation. In Genevieve Orr and Klaus-Robert Müller, editors, *Neural networks: tricks of the trade*, volume 1524 of *Lecture Notes in Computer Science*, pages 239–274, Heidelberg, 1998. Springer.

[10] I. Tomek. An experimetn with the edited nearest neighbor rule. *IEEE Trans. Syst., Man, Cyber.*, SMC-6(2):121–126, 1976.

[11] D.L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trnas. Syst., Man, Cyber*, SMC-2:408–421, May/June 1972.