

# Character Recognition by Adaptive Statistical Similarity

Thomas M. Breuel  
PARC, Inc.  
3333 Coyote Hill Rd.  
Palo Alto, CA 94304, USA

**Abstract** *Handwriting recognition and OCR systems need to cope with a wide variety of writing styles and fonts, many of them possibly not previously encountered during training. This paper describes a notion of Bayesian statistical similarity and demonstrates how it can be applied to rapid adaptation to new styles. The ability to generalize across different problem instances is illustrated in the Gaussian case, and the use of statistical similarity Gaussian case is shown to be related to adaptive metric classification methods. The relationship to prior approaches to multitask learning, as well as variable or adaptive metric classification, and hierarchical Bayesian methods, are discussed. Experimental results on character recognition from the NIST3 database are presented.*

## 1 Introduction

Two common approaches to solving classification problems are Bayesian methods and nearest neighbor methods. In Bayesian methods, we model class conditional distributions and use those estimates for finding minimum error rate discriminant functions. In nearest neighbor methods, we classify unknown feature vectors based on their proximity in feature space (usually, some Euclidean space,  $\mathbb{R}^d$ ) to previously classified samples.

Nearest neighbor methods can be understood in a Bayesian framework if we view the nearest neighbor procedure as implicitly using a non-parametric approximation of class conditional densities. Asymptotically, the error rate of nearest neighbor procedures is known to be within a factor of two of the Bayes optimal error rate [4]. But perhaps more important in practice than the asymptotic error rate is the performance of a nearest neighbor classifier given only a limited amount of training data. For example, an OCR system confronted with a novel font should be able to learn how to recognize characters in that font from perhaps only a small number of training examples of those characters derived from contextual information or user corrections.

To improve the performance of nearest neighbor methods, a number of authors (e.g., [7, 3]) have proposed us-

ing similarity functions other than the Euclidean distance in nearest neighbor classification, and given on-line or off-line procedures for computing such similarity functions<sup>1</sup>. A number of other techniques for multitask learning and transfer of knowledge between learning tasks have been described in the literature [2]; we will return to a discussion of that work to the methods described in this paper in the Discussion section.

This paper describes a notion of similarity grounded in Bayesian statistics that is learnable based on training examples using a wide variety of existing density estimation methods and classifiers. It then discusses the relationship between such a notion of statistical similarity and nearest neighbor classification methods. The paper uses feature vectors derived from the NIST3 database to demonstrate the ability of the method to create nearest neighbor classifiers with greatly improved classification performance on limited numbers of prototypes.

## 2 Bayesian Statistics

To establish notation and background, let us briefly review a few aspects of Bayesian decision theory relevant to classification problems. Bayesian decision theory[4] tells us that for the minimum error rate classifier (classification under a zero-one loss function), we should pick the class with the maximum posterior probability. That is, let  $\Omega$  be a finite set of possible classes and our feature vectors  $x$  be vectors in  $\mathbb{R}^d$ . Given the class conditional densities  $P(\omega|x)$ , choose the class  $\omega \in \Omega$  that has the maximum posterior probability given the sample  $x \in \mathbb{R}^d$ .

The differences among different classification methods come down to different tradeoffs and approaches in estimating and modeling  $P(\omega|x)$ .  $P(\omega|x)$  is usually estimated from a large set of training samples  $\{(x_1, \omega_1), \dots, (x_n, \omega_n)\}$ , the training set. Here, the  $x_i$  are

<sup>1</sup>They are often referred to as “adaptive similarity metrics”, but they do not satisfy the metric axioms and to avoid confusion, we refer to them here as “similarity functions”.

measurements or feature vectors, and the  $\omega_i$  are the corresponding classes.

One of the most common ways of estimating  $P(\omega|x)$  is to estimate  $P(x|\omega)$  and then apply Bayes rule:

$$P(\omega|x) = \frac{P(x|\omega)P(\omega)}{P(x)} \quad (1)$$

For example, if samples  $x$  are generated by picking a per-class prototype  $x_\omega$  and adding Gaussian random noise  $N \sim G(0, \Sigma)$  to it ( $\sim$  means “distributed according to”), then  $x \sim x_\omega + N$  or, equivalently,  $P(x|\omega) = G(x_\omega, \Sigma)$ . This may be extended to allowing multiple prototypes per class, giving mixture of Gaussian models  $P(x|\omega) = \sum_i G(x_{\omega,i}, \Sigma)$ . Another approach to modeling  $P(x|\omega)$  is that of many generic density estimation techniques like multi-layer perceptrons or logistic regression.

Since we are only interested in  $\arg \max_\omega P(\omega|x)$ , instead of  $P(\omega|x)$ , we can use any of a large number of equivalent decision functions  $D_\omega(x)$  such that classifying according to  $\omega(x) = \arg \max_\omega D_\omega(x)$  results in minimum error rates. Such an approach is taken by, for example, linear discriminant analysis and support vector machines.

### 3 Bayesian Similarity and Classification

The motivation for the Bayesian statistical similarity model introduced in this paper is the following. Assume we are performing nearest neighbor classification. We are given a prototype  $x'$  together with its class label  $\omega'$  and an unknown vector  $x$  to be classified. If we could estimate the probability that  $x$  and  $x'$  represent the same class, then we could use this to determine the probability that vector  $x$  comes from class  $\omega'$ .

Let us write this probability as  $P(S|x, x')$ , where  $S$  is a binary variable;  $S = 1$  means  $x$  and  $x'$  come from the same class, and  $S = 0$  otherwise. We can express  $P(S|x, x')$  in terms of  $P(\omega|x)$  and  $P(\omega|x')$  and use this as the definition of Bayesian statistical similarity.

**Definition 1** The (Bayesian) statistical similarity function  $S(x, x')$  is the conditional distribution  $P(S = 1|x, x')$ , where

$$S(x, x') = P(S = 1|x, x') = \sum_{\omega \in \Omega} P(\omega|x)P(\omega|x') \quad (2)$$

Note that in this definition, the distributions  $P(\omega, x)$  and  $P(\omega, x')$  need not be the same.

There are a number of properties we should observe. First, unlike, say, Euclidean distances, which assume values in the range  $[0, \infty)$  statistical similarity functions assume values in the interval  $[0, 1]$ , and increase with increasing similarity. A value of 1 means that two feature vectors  $x$

and  $x'$  are known to be in the same class (up to a set of measure zero). However,  $S(x, x')$  can be less than one, namely when the feature vector  $x$  cannot be classified unambiguously.

We should note that statistical similarity is problem dependent; a similarity function  $S(x, x')$  trained on one problem will not be optimal for another problem. However, as we will see in examples below, statistical similarity functions can often generalize to a wider range of classification problems than traditional classifiers.

Let us now look at the classification rule. For this, we first need another definition.

**Definition 2** Given some  $\omega \in \Omega$ , let us call  $x_\omega$  an unambiguous exemplar for class  $\omega$  iff  $P(\omega|x_\omega) = 1$ ; because of normalization, this also means that  $P(\omega'|x_\omega) = 0$  when  $\omega' \neq \omega$ , or  $P(\omega'|x_\omega) = \delta(\omega', \omega)$ .

If  $x_0$  is an unambiguous exemplar for class  $\omega_0$ , then

$$P(S = 1|x, x_0) = \sum_{\omega} P(\omega|x)P(\omega|x_0) \quad (3)$$

$$= \sum_{\omega} P(\omega|x)\delta(\omega, \omega_0) \quad (4)$$

$$= P(\omega_0|x) \quad (5)$$

Therefore, we have shown the following:

**Theorem.** If  $x_0$  is an unambiguous exemplar for class  $\omega_0$ , then  $P(\omega_0|x) = P(S = 1|x, x_0)$ .

In fact, if we estimate  $P(S = 1|x, x')$  by choosing  $x'$  from a training set of unambiguous exemplars  $T = \{x_{\omega_1}, \dots, x_{\omega_N}\}$ , we can write down a closed form expression for  $P(S = 1|x, x')$ :

$$P(S = 1|x, x') = \sum_{x' \in T} P(\omega|x)\delta(x_\omega, x') \quad (6)$$

What this shows is that if we have a perfect model of  $P(S|x, x')$  together with a set of unambiguous exemplars, then nearest neighbor classification using a statistical similarity function is completely equivalent to Bayes-optimal classification. However, when estimating conditional distributions from training data, the two approaches are not equivalent. First, estimating  $P(S|x, x')$  from training data is a different problem from estimating  $P(\omega|x)$ , both in the kind of models and in the kind of training data we can use. We will explore this in a Gaussian noise example below. Second, even if we cannot estimate an optimal model for  $P(S = 1|x, x')$ , we can use  $P(S = 1|x, x')$  as a similarity function in a nearest neighbor approach to classification. When used that way, the ability to use and select multiple prototypes makes up for modeling errors (in fact, let us state without proof that if  $P(S = 1|x, x')$  is sufficiently smooth when expanded in  $x - x'$ , we can perform a Taylor series expansion and show that it performs asymptotically no worse

than nearest neighbor classification using a Euclidean metric).

Another advantage of estimating  $P(S = 1|x, x')$  instead of class conditional densities is that it can be carried out with unlabeled training data in some important cases. In handwriting recognition, we can use adaptive clustering techniques like those described in [1] to cluster together exemplars from a single writer and use those to bootstrap a statistical similarity measure. Other examples of cases where class labels are unavailable but information about whether two samples are in the same class can be derived are found in information retrieval and visual object recognition.

#### 4 The Gaussian Case

Consider a classification problem in which the observed vectors are distributed according to  $x \sim x_\omega + N$ , where  $x_\omega$  is a class prototype and  $N$  is i.i.d. noise, independent of the object class. Then,  $P(x|\omega) = N(x - x_\omega)$ . Since  $P(S|x, x_\omega) = P(\omega|x) = \frac{P(x|\omega)P(\omega)}{P(x)} = \frac{N(x-x_\omega)P(\omega)}{\sum_{\omega'} N(x-x_{\omega'})}$ , we see that  $P(S|x, x_\omega)$  is translation invariant: if we translate  $x$  and the prototypes  $x_\omega$ , classification will be carried out the same way.

Furthermore, staying with this example, if the prototypes  $x_\omega$  are displaced by different amounts  $\Delta_\omega$ ,  $P(S|x, x_\omega + \Delta_\omega)$  may not be an accurate estimate of  $P(\omega|x)$  anymore. But we see that numerator is unaffected by a class dependent translation, and the denominator, the sample distribution  $P(x)$ , is affected equally for all classes, leaving the likelihood ratio, and hence the classification rule, unaffected.

In practice,  $N$  may not be completely independent of  $x$ , but if it varies slowly, we can choose models of  $P(S|x, x')$  that take advantage of this fact.

In fact, the Gaussian case provides a connection with adaptive metric models. Consider a simple adaptive metric model in which we optimize a quadratic form  $Q$  for our metric in order to minimize the error rate; that is, we use as our decision rule

$$\omega(x) = \arg \min_{\omega} (x - x_\omega) \cdot Q \cdot (x - x_\omega) \quad (7)$$

If our decision rule is  $\hat{\omega}(x) = \arg \max_{\omega} P(S|x, x_\omega)$  and our noise model  $N$  is a Gaussian  $G(0, \Sigma)$ , then, by the above argument,

$$\omega(x) = \arg \max_{\omega} P(S|x, x_\omega) \quad (8)$$

$$= \arg \max_{\omega} \frac{P(\omega)}{P(x)} G(x - x_\omega, \Sigma) \quad (9)$$

$$= \arg \max_{\omega} G(x - x_\omega, \Sigma) \quad (10)$$

$$= \arg \max_{\omega} e^{-\frac{1}{2\|\Sigma\|} (x-x_\omega) \cdot \Sigma^{-1} \cdot (x-x_\omega)} \quad (11)$$

$$= \arg \min_{\omega} (x - x_\omega) \cdot \Sigma^{-1} \cdot (x - x_\omega) \quad (12)$$

By comparing Equation 12 and Equation 7, we see that we can use  $\Sigma^{-1}$  as the quadratic form  $Q$  (the choice is not entirely unique).

#### 5 Character Recognition

The above ideas were tested experimentally on an isolated handwritten character recognition task using the NIST 3 database [5] (see also [6] for a state-of-the-art character recognition system and comparisons of a large number of classifiers). Similar experiments have been used in other works on variable and adaptive metric methods (e.g., [3]). The overall idea is to estimate  $P(S = 1|x, x')$  using multi-layer perceptrons (MLPs) as a simple and well-studied trainable model of posterior probabilities. Then, we use  $P(S = 1|x, x')$  as our “distance” in a  $k$ -nearest neighbor classifier and compare its performance with the performance of a standard nearest neighbor classifier.

The images used in these experiments were images of handwritten digits from the NIST 3 database. The NIST 3 database is used as a convenient source of real-world samples exhibiting writer-dependent variations to test the ability of these methods to adapt to novel writers. However, it is used in this paper in a different way from the way it is used traditionally for training classifiers. Traditionally, nearest neighbor or other classifiers are trained on as many samples as possible and achieve excellent writer-independent performance. In this paper, we use samples from the NIST 3 database to examine the ability of nearest neighbor and Bayesian similarity classifiers to generalize from a small number of prototypes (200 in one experiment, 10 in the other).

For all training, only images from the first 1000 writers were used; all testing was carried out the next 200 writers, a distinct population from the test set. For feature extraction, bounding boxes for characters were computed and the characters were rescaled uniformly to fit into a  $40 \times 40$  image. The resulting character image was slant corrected based on its second order moments. The uncorrected and slant corrected images form the first two feature maps. Derivatives were estimated along multiples of  $\frac{\pi}{5}$  degrees, resulting in five feature maps. Additionally, feature maps of interior regions, skeletal endpoints, and skeletal junction points were computed. Each of the resulting feature maps was anti-aliased and scaled down to a  $10 \times 10$  grid. This results in 10  $10 \times 10$  feature maps, or a 1000 dimensional feature vector. (Experiments were also carried out with subsets of these feature maps consisting of only the raw image, 100 dimensional, or the raw image, the slant corrected image, and derivatives, 700 dimensional, with similar results.)

To obtain statistical similarity models a multi-layer perceptron (MLP) was trained using gradient descent training. It has been shown (see [4] p.304) that training a multi-layer

Statistical Similarity Nearest Neighbor	Euclidean Nearest Neighbor
2.6%	9.5%

**Table 1. An experimental comparison of the performance of Euclidean nearest neighbor methods with statistical similarity based nearest neighbor methods. The error rates are derived from 5000 test samples, using 200 prototypes selected as described in the paper.**

perceptron under a least square error criterion and binary output variables results in an approximation to the posterior probability distribution. The feature vectors  $x$  and  $x'$  from each image were presented as a single vector  $(x - x', x + x')$  for input and training to the MLP (this is a fixed linear transformation of  $(x, x')$  and does not affect the ability of the MLP to model the conditional probability). The MLP used in the experiments had 30 hidden units. When the classes corresponding to the feature vectors in the NIST database were the same, the target output during training was set to 1, otherwise 0. This results in an estimate of  $P(S = 1|x, x')$ , where  $x$  and  $x'$  are both drawn from the same prior distribution of characters  $P(x)$  and  $P(\omega)$ .

Therefore, after estimating a statistical similarity function this way, the statistical similarity function was used in a simple nearest neighbor classifier. To select the prototypes for the nearest neighbor classifier, feature vectors from the training set were compared to the set of prototypes (initially empty) and the class associated with the most similar, according to the statistical similarity function, was returned as the classification. Whenever the classification was incorrect, the incorrectly classified feature vector was added to the set of prototypes. This process was stopped when the set of prototypes had grown to 200 prototypes.

To estimate misclassification rates, 5000 feature vectors were selected from a separate test set and classified like the training vectors (however, misclassified feature vectors were not added during the set of prototypes). As a control, the same training and testing process was carried out using Euclidean distance. The results of these experiments are shown in Table 1. They show a 3.7-fold improvement of using statistical similarity over Euclidean distance.

In a second set of experiments, the statistical similarity function was trained not on randomly selected pairs of feature vectors, but only on pairs of feature vectors from the same writer. This means that the statistical similarity function characterizes the variability in character shape for individual writers, as opposed to characterizing it for the whole population of writers. For testing, feature vectors from 200

Statistical Similarity Nearest Neighbor	Euclidean Nearest Neighbor
5.1%	22.6%

**Table 2. An experimental comparison of the performance of Euclidean nearest neighbor methods with statistical similarity based nearest neighbor methods on a rapid writer adaptation problem. The error rates are derived from 8767 test samples, using 10 prototypes selected as described in the paper.**

writers not in the training set were used. For each writer, the first instance of each character was used as a prototype, resulting in 10 prototypes per writer. These prototypes were then used to classify the remaining samples from the same writer. These results are shown in Table 2. The results show a 4.4-fold improvement of statistical similarity over Euclidean nearest neighbor methods.

These experimental results demonstrate that using statistical similarity functions can result in greatly improved recognition rates compared to Euclidean nearest neighbor classification methods—statistical similarity functions are an effective “adaptive metric” for these kinds of problems. However, that is all these initial experiments were designed to test, and several important experiments remain to be done; we will return to this issue in the Discussion.

## 6 Discussion

This paper has introduced the notion of statistical similarity based on the conditional distribution  $P(S = 1|x, x') = \sum_{\omega} P(\omega|x)P(\omega|x')$ . It was shown that classification using a statistical similarity function and a set of unambiguous exemplars is equivalent to Bayesian minimum error rate classification.

The paper has also connected variable metric nearest neighbor methods with statistical similarity measures. A relationship between class conditional distributions and similarity measures has also been observed by a number of previous authors. However, those methods attempted to construct various forms of distance functions or distance metrics, often with specific parametric forms. This paper, in contrast, eliminates any notion of “distance” entirely: statistical notions of object similarity were identified with the conditional distribution  $P(S = 1|x, x')$  and were justified directly in terms of Bayesian minimum error rate classification. Furthermore, this paper has demonstrated, both theoretically and empirically, that the required conditional distribution functions can be learned easily by training some convenient probabilistic model—for example, a multi-layer

perceptron—on pairs of input vectors.

The idea of learning  $P(S = 1|x, x')$  and using it for classification has been previously demonstrated on an optical character recognition task [1], but without exploring the connection with Bayesian methods presented in this paper. That paper also demonstrates how such models can be used when no writer or font-specific prototypes are available. The use of hierarchical Bayesian classification is also closely related to notions of statistical similarity—in the hierarchical Bayesian framework: the covariance models can be viewed as a representation of a distance or similarity measure. Hierarchical Bayesian methods have been suggested a number of times as the basis for generalizing across learning tasks in the literature and been demonstrated for knowledge transfer between related learning tasks in optical character recognition task in [8]. The Gaussian model used in the latter paper is closely related to the Gaussian model described in Section 4. Furthermore, the method described in [1] can be interpreted as a hierarchical Bayesian method, using  $P(S = 1|x, x')$  as a kind of covariance model, and using an uninformative prior for the class means.

The experiments in the paper compared the performance of statistical similarity with a Euclidean nearest neighbor classifier on two handwritten character recognition problems and demonstrated a 2.7 and 4.4-fold improvement relative to Euclidean nearest neighbor methods, both in a writer-independent and a writer dependent recognition tasks. These results show that statistical similarity is an effective method for constructing problem or domain-specific similarity measures. However, the database of handwritten characters used in the experiments was used simply as a convenient source of feature vectors; the experiments did not attempt to demonstrate state-of-the art handwriting recognition performance. Doing so will require using a much larger number of prototypes, as well as additional computational methods—such as tree-structured codebooks—to reduce the amount of time required to compare an unknown sample against a large number of prototypes. This remains to be explored in future work.

Another question that might be raised is how adaptive nearest neighbor methods like those described in the literature [7, 3] perform relative to the methods described in this paper. In some sense, the question can be answered trivially: this paper has presented a means by which we can transform any procedure for estimating conditional probabilities into a means for obtaining an adaptive or variable metric similarity function; many of the techniques described in the literature can simply be viewed as specific choices of estimators for  $P(S = 1|x, x')$ . But we might ask the specific question: does the use of multi-layer perceptrons for estimating  $P(S = 1|x, x')$ , as used in the experiments above, result in better performance than the variable metrics used in the literature? And can we construct other

estimators for  $P(S = 1|x, x')$  that might result in better performance than either MLPs or those prior method? These questions still remain to be answered in future work.

Overall, by demonstrating the utility of statistical similarity theoretically and its practical learnability using standard classification methods, achieving knowledge transfer between related classification tasks becomes much simpler than for prior methods. This is of particular importance in handwriting recognition and document analysis, which are related by the need for rapid adaptation of classifiers to stylistic variations: using the kinds of statistically based similarity measures described in this paper allows us to combine the ability of methods like multi-layer perceptrons to adapt to specific problems with the robustness and rapid adaptation of nearest-neighbor methods to new problems.

## References

- [1] T.M. Breuel. Classification by probabilistic clustering. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (ICASSP 2001)*, pages 1333–1336, 2001.
- [2] Rich Caruana. Algorithms and applications for multi-task learning. In *International Conference on Machine Learning*, pages 87–95, 1996.
- [3] Carlotta Domeniconi, Jing Peng, and Dimitrios Gunopulos. An adaptive metric machine for pattern classification. In *NIPS*, pages 458–464, 2000.
- [4] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd ed)*. Wiley Interscience, 2001.
- [5] M. D. Garris and R. A. Wilkinson. NIST Special Database 3: Binary Images of Handwritten Segmented Characters, 1992. 5.25" CD-ROM with documentation, available from: Standard Reference Data, national Institute of Standards and Technology, 221/A323, Gaithersburg, MD 20899 .
- [6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.
- [7] David G. Lowe. Similarity metric learning for a variable-kernel classifier. *Neural Computation*, 7(1):72–85, 1995.
- [8] C. Mathis and T. M. Breuel. Classification using a hierarchical bayesian approach. In *Proceedings of the International Conference on Pattern Recognition (ICPR'02), Quebec City, Quebec, Canada, 2002*.