Adaptive Kernel Metric Nearest Neighbor Classification

Jing Peng
EE&CS Department
Tulane University
New Orleans, LA 70118
jp@eecs.tulane.edu

Douglas R. Heisterkamp & H.K. Dai
Computer Science Department
Oklahoma State University
Stillwater, OK 74078
{doug,dai}@cs.okstate.edu

Abstract

Nearest neighbor classification assumes locally constant class conditional probabilities. This assumption becomes invalid in high dimensions due to the curse-of-dimensionality. Severe bias can be introduced under these conditions when using the nearest neighbor rule. We propose an adaptive nearest neighbor classification method to try to minimize bias. We use quasiconformal transformed kernels to compute neighborhoods over which the class probabilities tend to be more homogeneous. As a result, better classification performance can be expected. The efficacy of our method is validated and compared against other competing techniques using a variety of data sets.

1. Introduction

In pattern classification, we are given l training samples, where the training samples consist of q feature measurements $\mathbf{x} = (x_1, \cdots, x_q)^t \in \Re^q$ and the known class labels, $L \in \{0, \cdots, J\}$. Here we assume that there are only two classes, i.e., $L \in \{0, 1\}$; The goal is to predict the class label of a given query $\mathbf{x}'$.

A simple approach to this problem is nearest neighbor (NN) classification. Such a method produces continuous and overlapping neighborhoods and uses a different neighborhood for each individual query. Furthermore, empirical evaluation to date shows that the NN rule is a rather robust method. In addition, it has been shown [5] that the one NN rule has asymptotic error rate that is at most twice the Bayes error rate. NN rules assume that locally the class (conditional) probabilities are approximately constant. However, this assumption is often invalid in practice due to the curse-of-dimensionality. Severe bias can be introduced in the NN rule in a high dimensional input space. As such, the choice of a distance measure is critical and adaptive NN classification becomes attractive [4, 6, 7].

This paper presents an adaptive NN method to try to minimize bias in high dimensions. We estimate a metric for computing neighborhoods based on quasiconformal transformed kernels. The adaptation yields class conditional probabilities that tends to be constant in the modified neighborhoods. The closer match to the assumptions of NN provides a better classification performance.

2. Related Work

Friedman [6] proposes a technique for capturing local feature relevance as a reduction in prediction errors. Induced flexible metrics show improvement in performance over the simple NN method. In [7], Hastie and Tibshirani propose an adaptive NN method based on linear discriminant analysis. The method computes a distance metric as a product of properly weighted within and between sum of squares matrices. They show that the resulting metric approximates the weighted *Chi-squared* distance [7, 9, 11]. Amari and Wu [1] describe a method for improving SVM performance by increasing spatial resolution around the decision surface based on the Riemannian geometry. Viewed under the same light, our goal is to expand the spatial resolution around samples whose class probabilities are different from the query and contract the spatial resolution around samples whose class probability distribution is similar to the query. The effect is to make the space around samples farther from or closer to the query, depending on their class (conditional) probability distributions.

3. Kernel Distance

The kernel trick has been applied to numerous problems [3, 10]. The kernel allows an algorithm to work in a fea-

ture space of high dimension. If $\phi(\mathbf{x})$ is a mapping of a point in the input space to the feature space, then the kernel calculates the dot product $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}')$, where $\cdot$ denotes the dot product. Common kernels are Gaussian $K(\mathbf{x}, \mathbf{x}') = e^{-\gamma||\mathbf{x}-\mathbf{x}'||^2}$ and polynomial $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x} \cdot \mathbf{x}')^d$. Distance in the feature space may be calculated by means of the kernel [3, 12]. With $\mathbf{x}$ and $\mathbf{x}'$ in the input space then the (squared) feature space distance is $D(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}, \mathbf{x}) - 2K(\mathbf{x}, \mathbf{x}') + K(\mathbf{x}', \mathbf{x}')$.

4. Quasiconformal Kernel

It is a straight forward process to create a new kernel from existing kernels [3]. However, we desire to create a new kernel such that, for each input query $\mathbf{x}'$, the class posterior probability in the neighborhood induced by the kernel metric tend to be constant. We therefore look to quasiconformal mappings [2].

Previously, Amari and Wu [1] have modified a support vector machine with a quasiconformal mapping. If $c(\mathbf{x})$ is a positive real valued function of $\mathbf{x}$, then a new kernel can be created by

$$\tilde{K}(\mathbf{x}, \mathbf{x}') = c(\mathbf{x})c(\mathbf{x}')K(\mathbf{x}, \mathbf{x}'). \quad (1)$$

We call it a quasiconformal kernel. Note that if the restriction on $c(\mathbf{x})$ being positive is removed, $\tilde{K}$ is still a valid kernel [3].

The question becomes which $c(\mathbf{x})$ do we wish to use? We can change the Riemannian metric by the choice of $c(\mathbf{x})$. The metric $g_{i,j}$ associated with kernel K becomes the metric $\tilde{g}_{i,j}$ associated with kernel $\tilde{K}$ by the relationship [1]: $\tilde{g}_{i,j}(\mathbf{x}) = c_i(\mathbf{x})c_j(\mathbf{x}) + c(\mathbf{x})^2 g_{i,j}(\mathbf{x})$x, where $c_i(\mathbf{x}) = \frac{\partial c(\mathbf{x})}{\partial x_i}$.

Amari and Wu [1] expanded the spatial resolution in the margin of a support vector machine by using the following $c(\mathbf{x}) = \sum_{i \in SV} \alpha_i e^{\frac{-||\mathbf{x}-\mathbf{x}_i||^2}{2\tau^2}}$, where SV is the set of support vectors, $\mathbf{x}_i$ is the ith support vector, α_i is a positive number representing the contribution of $\mathbf{x}_i$, and τ is a free parameter. Since the support vectors are likely to be at the boundary of the margin, this creates an expansion of spatial resolution in the margin and a contraction elsewhere.

5. Adaptive Quasiconformal Kernel Nearest Neighbors

Our adaptive quasiconformal kernel nearest neighbor (AQKNN) algorithm is motivated as follows. Suppose that the original kernel K is a radial kernel, then $K(\mathbf{x}, \mathbf{x}) = 1$. Thus, the quasiconformal kernel (squared) distance can be written as

$$D(\mathbf{x}, \mathbf{x}') = c(\mathbf{x})^2 - 2c(\mathbf{x})c(\mathbf{x}')K(\mathbf{x}, \mathbf{x}') + c(\mathbf{x}')^2 \quad (2)$$

Our goal is to produce neighborhoods where the class conditional probabilities tend to be homogeneous. That is, we want to expand the spatial resolution around samples whose class probabilities are different from the query and contract the spatial resolution around samples whose class probability distribution is similar to the query. The effect is to make the space around samples farther from or closer to the query, depending on their class (conditional) probability distributions. An appealing candidate for a sample $\mathbf{x}$ with a query $\mathbf{x}'$ is

$$c(\mathbf{x}) = \frac{\Pr(j_{\bar{m}}|\mathbf{x})}{\Pr(j_m|\mathbf{x}')} \quad (3)$$

where $j_m = \arg\max_j \Pr(j|\mathbf{x}')$ and $j_{\bar{m}} = 1 - j_m$. It is based on the magnitude of the likelihood ratio of two conditional class probabilities: the maximum likelihood class (j_m) of the query versus the complementary class ($j_{\bar{m}}$) of the sample.

Note that: (1) The multiplier $c(\mathbf{x})$ for a sample $\mathbf{x}$ yields a contraction effect $c(\mathbf{x}) < 1$ when and only when $\Pr(j_m|\mathbf{x}) + \Pr(j_m|\mathbf{x}') > 1$, that is, both conditional likelihoods are consistent. (2) The multiplier $c(\mathbf{x}')$ for a query $\mathbf{x}'$ measures the degree of uncertainty in labeling $\mathbf{x}'$ with its maximum likelihood class. Substituting $c(\mathbf{x})$ (3) into (2) and simplifying, we obtain

$$D(\mathbf{x}, \mathbf{x}') = \left(\frac{\Pr(j_m|\mathbf{x}') - \Pr(j_m|\mathbf{x})}{\Pr(j_m|\mathbf{x}')}\right)^2 + 2c(x')c(x)(1 - K(x, x')).$$

To understand the above distance we look at the distance using a second order Taylor expansion of a general Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^t \Sigma^{-1}(\mathbf{x}-\mathbf{x}')}$ at the query point, $\mathbf{x}'$, $K(\mathbf{x}, \mathbf{x}') \approx 1 - \frac{1}{2}(\mathbf{x} - \mathbf{x}')^t \Sigma^{-1}(\mathbf{x} - \mathbf{x}')$. Substituting the Taylors expansion into $D(\mathbf{x}, \mathbf{x}')$ yields

$$\begin{aligned} D(\mathbf{x}, \mathbf{x}') &\approx \frac{[\Pr(j_m|\mathbf{x}') - \Pr(j_m|\mathbf{x})]^2}{\Pr(j_m|\mathbf{x}')^2} \\ &+ c(\mathbf{x}')c(\mathbf{x})(\mathbf{x} - \mathbf{x}')^t \Sigma^{-1}(\mathbf{x} - \mathbf{x}') \end{aligned} \quad (4)$$

The first term in the above equation is a *Chi-squared* (appropriately weighted) distance, while the second term is the weighted quadratic (Mahalanobis) distance. The distance (4) is more effient computationally than the kernel-based one, and is thus used in AQKNN in our experiments. When $c(\mathbf{x}) \approx c(\mathbf{x}')$, the quasiconformal kernel distance is reduced to the weighted Mahalanobis distance, with a weighting factor of $c(\mathbf{x}')c(\mathbf{x})$ depending on degrees of class-consistency in $c(\mathbf{x})$ and of labeling-uncertainty in $c(\mathbf{x}')$.

Let us examine $c(\mathbf{x}')$. Clearly, $0 \leq c(\mathbf{x}') \leq 1$. When $c(\mathbf{x}') \approx 0$ there is a high degree of certainty in which case c(**x'**) is aggressive in modifying the Mahalanobis distance and applies a large contraction. On the other hand, when

Table 1. Average classification error rates.

	Iris(4)	Sonar(60)	Liver(6)	Pima(8)	Vote(16)	OQ(16)	Cancer(9)	Ionosphere(34)	Hepatitis(19)
AQKNN	**4.0**	8.7	**15.7**	**17.7**	4.7	3.4	**2.3**	4.8	**11.6**
C4.5	8.0	23.1	19.5	22.4	3.5	9.4	4.7	5.4	19.4
DANN	6.0	**7.7**	15.9	22.2	3.2	4.2	2.5	4.8	**11.6**
KNN	6.0	12.5	16.6	21.7	7.8	6.0	2.7	5.8	14.8
Machete	5.0	21.2	20.1	21.5	6.5	6.7	2.9	6.0	13.5
Parzen	6.0	12.5	16.1	19.9	7.3	5.2	2.5	5.4	12.9
Scythe	**4.0**	16.3	20.8	21.1	14.7	5.4	2.5	7.2	13.5
SVM-R	**4.0**	12.5	16.1	21.3	**3.0**	**3.0**	2.4	**3.0**	13.6

$c(\mathbf{x}') \approx 1$ there is a low degree of certainty. The *Chi-squared* term achieves little statistical information, in which case c(**x'**) is cautious in modifying the Mahalanobis distance and applies little or no contraction. Now consider the effect of $c(\mathbf{x})$ on the distance (4). It is not difficult to show that

$$c(\mathbf{x}) = c(\mathbf{x}') \pm \sqrt{[\Pr(j_m|\mathbf{x}') - \Pr(j_m|\mathbf{x})]^2 / \Pr(j_m|\mathbf{x}')^2},$$

where $\pm$ represents the algebraic sign of $[\Pr(j_m|\mathbf{x}') - \Pr(j_m|\mathbf{x})]$. For a given $\mathbf{x}'$, $c(\mathbf{x}')$ is fixed. Thus the dilation/contraction of the Mahalanobis distance due to variations in $c(\mathbf{x})$ is proportional to the square root of the *Chi-squared* distance with the dilation/contraction determined by the direction of variation of $\Pr(j|\mathbf{x})$ from $\Pr(j|\mathbf{x}')$. That is, $c(\mathbf{x})$ attempts to compensate for the *Chi-squared* distance ignorance of the direction of variation of $\Pr(j|\mathbf{x})$ from $\Pr(j|\mathbf{x}')$ and is driving the neighborhood closer to homogeneous class conditional probabilities.

Estimation From a nearest neighborhood of K_M points around $\mathbf{x}'$ in the simple Euclidean distance, we take the maximum likelihood estimate for $\Pr(j)$. To estimate $p(\mathbf{x}|j)$, we use simple non-parametric density estimation: Parzen Windows estimate with Gaussian kernels [5]. We place a Gaussian kernel over each point $\mathbf{x}_i$ in class j. The estimate $\hat{p}(\mathbf{x}|j)$ is then simply the average of the kernels. For simplicity, we use identical Gaussian kernels for all points with covariance $\Sigma = \tau^2 I$. More precisely,

$$\hat{p}(\mathbf{x}|j) = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \frac{1}{\tau^q (2\pi)^{q/2}} e^{-\frac{1}{2\tau^2}(\mathbf{x}-\mathbf{x}_i)^t(\mathbf{x}-\mathbf{x}_i)} \quad (5)$$

where C_j represents the set of training samples in class j. Together, $\hat{\Pr}(j)$ and $\hat{p}(\mathbf{x}|j)$ define $\hat{\Pr}(j|\mathbf{x})$ through $\hat{\Pr}(j|\mathbf{x}) = \frac{\hat{p}(\mathbf{x}|j)\hat{\Pr}(j)}{\sum_i \hat{p}(\mathbf{x}|i)\hat{\Pr}(i)}$. Using the estimates in (5), we obtain an empirical estimate of (3) for each data point $\mathbf{x}$.

6. Empirical Results

We compare several competing classification methods using data sets from the UCI Machine Learning Database Repository: (1) AQKNN method described above; here we use a simplified form of AQKNN: we set Σ in (4) to $\sigma^2 I$. Thus, Equation (4) simply reduces to $D(\mathbf{x}, \mathbf{x}') \approx \frac{[\Pr(j_m|\mathbf{x}') - \Pr(j_m|\mathbf{x})]^2}{\Pr(j_m|\mathbf{x}')^2} + \frac{1}{\sigma^2} c(\mathbf{x}')c(\mathbf{x})\|\mathbf{x} - \mathbf{x}'\|^2$; (2) SVM method with radial kernels [8]; (3) Simple K-NN method; (4) C4.5;(5) Machete [6]; (6) Scythe [6]; (7) DANN–discriminant adaptive nearest neighbor classification [7]; and (8) Local Parzen Windows method–a nearest neighborhood of K_M points around the query $\mathbf{x}'$ is used to estimate $\Pr(j|\mathbf{x})$s through (5), from which the Bayes method is applied.

The features are first normalized over the training data to have zero mean and unit variance, and the test data features are normalized using the corresponding training mean and variance. Procedural parameters for each method were determined empirically through cross-validation. Whenever SVMlight is involved, the values of γ in the radial basis kernel $\exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|)$, and c were chosen empirically.

Table 1 shows the error rates for the eight methods on the nine data sets, where the number next to each problem denotes the dimensions of the problem. Note that the average error rates for the Iris, Sonar, Vote, and Hepatitis data sets (the number of data points is 100, 208, 232 and 155, respectively) were based on leave-one-out cross-validation, whereas the error rates for the remaining data sets were based on 10 independent runs of a random selection of 200 training data and 200 testing data, since larger data sets are available in these five cases (the numbers are 345, 768, 1536, 683 and 351 for Liver, Pima, OQ, Cancer and Ion, respectively).

Table 1 shows that AQKNN achieved the best or near best performance over the nine data sets, followed by SVM-

R. To measure robustness, for each method m we compute the ratio b_m of its error rate e_m and the smallest error rate over all methods being compared in a particular example: $b_m = e_m / \min_{1 \leq k \leq 8} e_k$. The larger the value of b_m, the worse the performance of the method is in relation to the best one for that example, among the methods being compared. The distribution of the b_m values for each method m over all the examples, therefore, indicates its robustness.

Figure 1 plots the distribution of b_m for each method over the nine data sets. The dark area represents the lower and upper quartiles of the distribution that are separated by the median. The outer horizontal lines show the entire range of values for the distribution. It is clear that AQKNN is most robust over the data sets. Additional improvement in performance is possible when the Mahalanobis distance in (4) is fully employed. It is also interesting note that the local Parzen Windows method, while never achieved best performance, is rather robust.

7 Summary and Conclusions

This paper presents an adaptive kernel distance for nearest neighbor classification. This method estimates a distance based on quasiconformal transformed kernels. The effect of the kernel distance is to move samples having similar class posterior probabilities to the query closer to it, while moving samples having different class posterior probabilities farther away from the query. As a result, the class conditional probabilities tend to be more homogeneous in the modified neighborhoods. The experimental results demonstrate that the AQKNN algorithm can potentially improve the performance of K-NN and recursive partitioning methods in some classification and data mining problems. The results are also in favor of AQKNN over similar competing methods such as Machete and DANN.

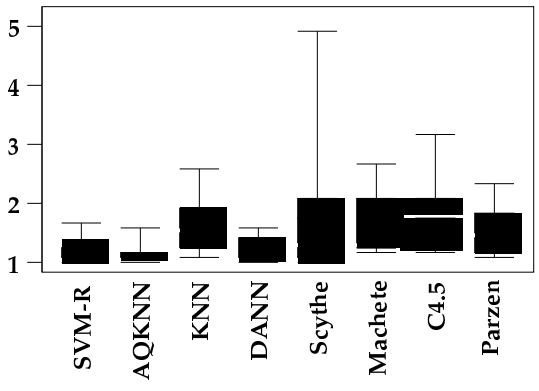

Figure 1. Performance distributions.

References

[1] S. Amari and S. Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12(6):783–789, 1999.

[2] G. D. Anderson, M. K. Vananamurthy, and M. K. Vuorinen. *Conformal Invariants, Inequalities, and Quasiconformal Maps*. Canadian Mathematical Society Series of Monographs and Advanced Texts. John Wiley & Sons, Inc., New York, 1997.

[3] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, UK, 2000.

[4] C. Domeniconi, J. Peng, and D. Gunopulos. Locally adaptive metric nearest neighbor classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2002.

[5] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., 1973.

[6] J. H. Friedman. *Flexible Metric Nearest Neighbor Classification*. Tech. Report, Dept. of Statistics, Stanford University, 1994.

[7] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(6):607–615, 1996.

[8] T. Joachims. Making large-scale svm learning practical. In B. Schvlkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, volume 13. The MIT Press, 1999.

[9] J. P. Myles and D. J. Hand. The multi-class metric problem in nearestneighbor discrimination rules. *Pattern Recognition*, 723:1291–1297, 1990.

[10] B. Schölkopf. The kernel trick for distances. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 301–307. The MIT Press, 2001.

[11] R. Short and K. Fukunaga. Optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory*, 27:622–627, 1981.

[12] V. N. Vapnik. *Statistical learning theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, New York, 1998.