Minimum Expected Risk Estimation for Near-neighbor Classification

Maya R. Gupta , Santosh Srivastava, Luca Cazzanti
{gupta}@ee.washington.edu

Dept of EE, University of Washington
Seattle WA, 98195-2500

UW
Electrical
Engineering

UWEE Technical Report
Number UWEETR-2006-0006
April 2006

Department of Electrical Engineering
University of Washington
Box 352500
Seattle, Washington 98195-2500
PHN: (206) 543-2150
FAX: (206) 543-3842
URL: http://www.ee.washington.edu

Minimum Expected Risk Estimation for Near-neighbor Classification

Maya R. Gupta , Santosh Srivastava, Luca Cazzanti
{gupta}@ee.washington.edu

Dept of EE, University of Washington
Seattle WA, 98195-2500

University of Washington, Dept. of EE, UWEETR-2006-0006
April 2006

Abstract

We consider the problems of class probability estimation and classification when using near-neighbor classifiers, such as k-nearest neighbors (kNN). This paper investigates minimum expected risk estimates for neighborhood learning methods. We give analytic solutions for the minimum expected risk estimate for weighted kNN classifiers with different prior information, for a broad class of risk functions. Theory and simulations show how significant the difference is compared to the standard maximum likelihood weighted kNN estimates. Comparisons are made with uniform weights, symmetric weights (tricube kernel), and asymmetric weights (LIME kernel). Also, it is shown that if the uncertainty in the class probability is modeled by a random variable, and the expected misclassification cost is minimized, the result is equivalent to using a classifier with a minimum expected risk estimate. For symmetric costs and uniform priors, it is seen that minimum expected risk estimates have no advantage over the standard maximum likelihood estimates. For asymmetric costs, simulations show that the differences can be striking.

1 Introduction

We consider the standard supervised statistical learning problem of classifying test samples given a labeled database of training examples, a known finite set of possible class labels, and a matrix of misclassification costs. This paper proposes and analyzes robust classification and estimation for near-neighbor learning, such as k-nearest neighbors (kNN). Near-neighbor learning is sometimes also called instance-based learning, memory-based learning, case-based reasoning, or lazy learning.

We assume that a set of training sample points (neighbors), and weights on those training samples have been chosen for a particular test point. The paper's focus is on how to make optimal estimates of a class label and class probabilities given the set of weighted neighbors. We show that the standard estimation procedure is a maximum likelihood estimate. Maximum likelihood estimates can lack robustness. Laplace smoothing, or other smoothing, is sometime used heuristically in statistical learning. Such smoothing can be theoretically justified as Bayesian minimum expected risk (MER) estimation. The two-class analytic MER solution for weighted near-neighbors was given in a recent workshop paper [1]. In this paper we investigate more deeply; we give the multi-class solution, incorporate different prior information, and show that the same classification results from replacing the class probability estimate with a random variable and directly minimizing the expected cost. After defining notation in Section 2, we establish that standard near-neighbor learning uses maximum likelihood estimates in Section 3. MER estimates for the class probability estimates are given in Section 4. We discuss different prior information scenarios in Section 5. Classifying by directly minimizing expected misclassification cost is proposed and solved in Section 6. In Section 7, we show by theory and simulation how much difference the MER estimation can make. Section 8 is a summary and discussion of open questions. All the proofs of the mathematical result are given in the Appendix.

2 Notation

Supervised statistical learning is based on a set of given training pairs $\mathcal{T} = \{X_i, Y_i\}$, where $X_i \in \mathcal{R}^d$ and $Y_i \in \{1, 2, \ldots, \mathcal{G}\}$, where $\mathcal{G}$ is finite. The training samples $\{(X_i, Y_i)\}$ and test sample (X, Y) are assumed to be drawn independently and identically from some sufficiently nice joint distribution $P_{(X,Y)}$. One classification problem is to estimate the probability of each class for a test feature vector, $P(Y = g|X = x)$ for $g \in \{1, 2, \ldots, \mathcal{G}\}$, based on the given training pairs $\mathcal{T}$. A related classification problem is to classify X as label Y given $\mathcal{T}$ and a $\mathcal{G} \times \mathcal{G}$ misclassification cost matrix C, where $C(g, h)$ specifies the cost of classifying a test sample as class g when the truth is class h.

It will be convenient to treat the unknown $P_{Y|x}$ as a random vector Θ with $\mathcal{G}$ random components Θ_g, where each Θ_g represents an unknown $P(Y = g|x)$ for $g \in \{1, \ldots, \mathcal{G}\}$. A particular realization of the random vector Θ will be the probability mass function (pmf) θ. The random vector Θ is distributed with density $f(\theta)$, which we constrain to have a mathematically nice and relevant formulation (as stated in the following sections). The focus of this paper is on robust classification estimates $\hat{Y}$, and robust estimates of the class pmf $\hat{\theta}$ with components $\hat{\theta}_g$ for $g \in \{1, \ldots, \mathcal{G}\}$.

Lastly, we note that in this paper we differentiate between *near neighbors* and nearest neighbors, where *nearest neighbors* refers to the closest samples from a training set, with closest measured in terms of Euclidean distance if not otherwise noted. Near neighbors is used to more generally mean a set of near (but not necessarily nearest) neighbors.

3 Standard Near-neighbor Learning

There are many approaches to supervised statistical learning; we focus here on near-neighbor methods. Such methods are known to perform well in practice, are intuitive, and can achieve optimal error rates asymptotically [2]. Non-parametric neighborhood methods weight the training samples in a neighborhood around the test point x. It is not important for this paper how one defines a neighborhood; common definitions are to use the k-nearest neighbors, or all neighbors within a defined radius. Let the sample pairs be re-indexed by their distance to the test sample x, so that X_j is the jth nearest neighbor to x. Given a neighborhood, a weighted kNN classifier assigns a weight w_j to each neighbor, usually by evaluating a kernel that assigns weight based on the distance from x to X_j [2], though we will also report results on a recent asymmetric kernel [3]. The kNN classifier assigns equal weights to every neighbor. Our formulation will hold for any weighted kNN classifier where the weights satisfy $\sum_{j=1}^{k} w_j = 1$ and $w_j \geq 0$.

3.1 Standard Estimates for Near-neighbor Learning

From the weights and the neighborhood sample pairs, it is standard to form an estimate of the probability of each class,

$$\hat{\theta}_g = \sum_{j=1}^{k} w_j I_{(Y_j = g)} \tag{1}$$

for $g \in \{1, \ldots, \mathcal{G}\}$, where $I_{(\cdot)}$ is an indicator function that equals one when its argument is true, and equals zero otherwise. This standard formula for class probability estimates dates back at least to 1977 [4].

The class pmf estimate $\hat{\theta}$ for the test sample x can be used to choose a class label $\hat{y}$. Given a class pmf estimate $\hat{\theta}$ and a misclassification cost matrix C, it is standard to classify x as the class $\hat{Y}$ which minimizes the expected misclassification cost with respect to the estimated pmf $\hat{\theta}$ such that $\hat{Y}$ solves

$$\underset{g}{\text{argmin}} \sum_{h=1}^{\mathcal{G}} C(g, h)\hat{\theta}_h. \tag{2}$$

For uniform weights, the estimate (1) is the maximum likelihood estimate of θ given the neighborhood data samples $Y_1, Y_2, \ldots, Y_k$ of the test feature vector x under the assumption that these near-neighbor samples were all drawn from the same pmf. That is, if m_g of the near-neighbors are of each class g out of a total of k near-neighbors, then the estimate (1) is $\hat{\theta}_g = m_g/k$, which maximizes the likelihood $f(\theta)$ of independently identically drawing those neighbors, where

$$f(\theta) = \left(\frac{k!}{\prod_{g=1}^{\mathcal{G}} m_g!} \right) \prod_{g=1}^{\mathcal{G}} \theta_g^{m_g}. \tag{3}$$

Maximum likelihood (ML) estimates can be quite unrepresentative of the underlying likelihood distribution when small sample sizes are used. Near-neighbor algorithms are often run with small neighborhood sizes, which yields small sample sizes for the estimation done in (1), and thus we hypothesize that a different estimation principle could make a difference in practice. See [5, pgs. 300-310] and [6, ch. 15] for further discussions of the problems with maximum likelihood estimation.

Next, we define a weighted likelihood function and show that for nonuniform weight vectors w, the estimate (1) is similarly the maximum weighted likelihood estimate, and can be expected to have similar problems of high estimation variance when used with relatively small neighborhood sizes k.

Let $f(\theta)$ be the likelihood of drawing the neighborhood samples with each neighborhood sample weighted by w_j:

$$f(\theta) = \prod_{j=1}^{k} \prod_{g=1}^{\mathcal{G}} \theta_g^{w_j k I_{(Y_j=g)}} = \prod_{g=1}^{\mathcal{G}} \theta_g^{k \sum_{j=1}^{k} w_j I_{(Y_j=g)}}, \tag{4}$$

where the multiplicative constant term of $f(\theta)$ has been dropped because it will not affect the minimization problems we will solve to estimate θ. Note that for kNN the weights are uniform— $w_j = 1/k$ for all j — and then the formulas (4) and (3) are equivalent (up to a normalization constant).

Lemma 1 *The class probability estimate given in (1) maximizes the weighted likelihood given in (4) subject to the constraints*

$$\sum_{g=1}^{\mathcal{G}} \theta_g = 1, \sum_{j=1}^{k} w_j = 1. \tag{5}$$

4 Minimum Expected Risk Estimates

In order to form more robust estimates of the class label and the class pmf for a test sample, we propose applying to near-neighbor learning a principle of estimation more robust than maximum likelihood estimation. Minimizing the expectation of a relevant risk R where the expectation is taken over all possible pmf's will yield robust results in terms of average performance. Minimizing the maximum error could further bound the possible error, but at the expense of suboptimal estimates on average. We apply a Bayesian minimum expected risk (MER) principle [7, ch. 4] to estimate the class probabilities. The MER estimate of the class conditional probability $\hat{\theta}_{MER}$ solves

$$\underset{\hat{\theta}}{\operatorname{argmin}} \int R(\theta, \hat{\theta}) f(\theta) d\theta, \tag{6}$$

where $f(\theta)$ is the probability of pmf θ being the true underlying pmf, and R is some non-negative function (such as mean-squared error, or relative entropy) suitable for measuring distortion.

The minimization problem (6) can be rewritten as

$$\underset{\hat{\theta}}{\operatorname{argmin}} E_{\Theta}[\ R(\Theta, \hat{\theta})\ ]$$

where Θ is distributed with probability density $f(\theta)$.

Let $f(\theta)$ be the likelihood of the weighted neighbor class labels as given in (4). This is equivalent to defining $f(\theta)$ to be the posterior with a uniform prior on the random variable Θ. Theorem 1 establishes the analytic solution for estimates for the class of Bregman divergence risk functions R [8], which include the standard squared error loss and relative entropy.

Definition of Bregman Divergence [9] Let $\psi \in \mathcal{C}^2 : \theta \in [0,1]^{\mathcal{G}} \to R$ be a strictly convex function. The Bregman divergence d_ψ is defined as

$$d_\psi(\theta, \phi) = \psi(\theta) - \psi(\phi) - (\theta - \phi)^T \nabla\psi(\phi). \tag{7}$$

Theorem 1 *Let the probability of the multinomial pmf of θ be of the form*

$$f(\theta) = \gamma \prod_{g=1}^{\mathcal{G}} \theta_g^{\alpha_g}, \quad \gamma = \textit{normalization constant.}$$

The MER estimate $\hat{\theta}$ of the unknown pmf solves,

$$\hat{\theta} = \operatorname*{argmin}_{\phi} \int_\theta R(\theta, \phi) f(\theta) d\theta,$$

where $\int_\theta$ denotes a multiple integral with region of integration such that $\sum_{g=1}^{\mathcal{G}} \theta_g = 1, \theta_g \geq 0$.

Then for any Bregman divergence risk $R(\theta, \phi) = d_\psi(\theta, \phi)$, the MER estimate gives the probability of the gth class,

$$\hat{\theta}_g = \frac{\alpha_g + 1}{\sum_{g=1}^{\mathcal{G}} \alpha_g + \mathcal{G}}, \quad g = 1, \ldots, \mathcal{G}. \tag{8}$$

In accordance with the above theorem, the MER estimate for uniform weights for either mean-squared R or relative entropy R is

$$\hat{\theta}_g = \frac{m_g + 1}{k + \mathcal{G}}. \tag{9}$$

for $g \in \{1, 2, \ldots, \mathcal{G}\}$. This estimate is equivalent to *Laplace correction* for estimating multinomial distributions [10, pg. 272], also called *Laplace smoothing*. Appropriately, the history of Laplace correction goes back to Laplace himself; Jaynes offers historical information and more details about alternate derivations [11, pgs 154-165]. Laplace correction has been shown to be useful for class probability estimation in decision trees [12, 13, 14, 15], and with Naive Bayes [16]. Laplace correction was incorporated in the CN2 rule learner [17], and Domingos used it to break ties in a unified instance-based and rule-based learner [18]. Many different smoothing approaches are used in speech recognition [19].

More generally, applying Theorem 1 to find the MER estimate for weighted neighbors yields

$$\hat{\theta}_g = \frac{k \sum_{j=1}^{k} w_j I_{(Y_j = g)} + 1}{k + \mathcal{G}} \tag{10}$$

for any Bregman divergence. More information on the Bregman divergences can be found in recent papers [20, 21].

5 Prior Distributions on Θ

Thus far a uniform prior distribution on Θ has been assumed, so that the posterior $f(\theta)$ is the likelihood of θ, as per (4). However, consider a two-class problem and uniform prior class probabilities $\{\pi_1 = .5,\ \pi_2 = .5\}$; there are still many possible prior probabilities over the class pmf θ. At one extreme, the prior is $q(\theta) = .5\delta([\theta_1, 1-\theta_2]) + .5\delta([1-\theta_1, \theta_2])$, where we use the standard Dirac delta generalized function notation to express that the prior probability has half of its support on the pmf $[0,1]$, and half on the pmf $[1,0]$. At the other extreme is the uniform prior $q(\theta) = 1/\sqrt{2}$, which corresponds to every possible θ being equally likely. Both yield the same marginal class prior probabilities $\{\pi_1 = .5, \pi_2 = .5\}$.

Better performance can be expected with a prior that better represents the prior knowledge. In this section we consider some cases and approaches to prior information.

5.1 Zero Bayes' Risk

Suppose the Bayes' risk for a learning problem was known to be zero, so that the true class pmf θ for any test sample x is $\theta_g = 1$ for some class g and $\theta_h = 0$ for all other classes h. Then a uniform prior on Θ will yield grossly inaccurate estimates, as the MER estimation takes into account the likelihood of a large set of θ's that will never occur in this problem. (The likelihood of those non-possible θ's will sometimes be non-zero because the underlying assumption that x's neighbors have been drawn from the same θ as x's class does not hold). Using MER with the correct prior information for this case is trivially equivalent to using ML.

5.2 Limited Class Probabilities

Another interesting example will be covered in a simulation in Section 7.4. In that two-class simulation, $\theta_1 < .8$ for all samples. Using a uniform prior for θ includes the likelihood of θ_1 from .8 to 1, and this incorrect inclusion leads to some inaccurate estimates. More generally, in a practical learning problem, one may know that there are no feature vectors x for which the probability of being class one is greater than some a. Here we give an analytic result for the two-class case for this type of prior information.

Lemma 2 *For a two-class classification problem, let the probability of θ be the product $f(\theta)q(\theta)$, where $q(\theta)$ is uniform over the region $[0, a]$ for some $a < 1$, and $f(\theta)$ has the form,*

$$f(\theta) = \gamma \prod_{g=1}^{\mathcal{G}} \theta_g^{\alpha_g},$$

where γ is a normalization constant.

Then the MER estimate $\hat{\theta}$ of the unknown pmf solves

$$\hat{\theta} = \operatorname*{argmin}_{\phi} \int_{\theta} R(\theta, \phi) f(\theta) q(\theta) d\theta,$$

where $\int_\theta$ denotes an integral with region of integration such that $\theta_1 + \theta_2 = 1$, and $\theta_1 \geq 0$, $\theta_2 \geq 0$.

Then for any Bregman divergence risk, the MER estimate gives the probability of class one to be

$$\hat{\theta}_1 = \frac{\mathcal{B}(a, \alpha_1 + 2, \alpha_2 + 1)}{\mathcal{B}(a, \alpha_1 + 1, \alpha_2 + 1)}, \tag{11}$$

where $\mathcal{B}$ is the incomplete beta function.

Applying Lemma 2 to the case of interest, $\alpha_1 = k \sum_{j=1}^{k} w_j I_{(Y_j=1)}$ and $\alpha_2 = k \sum_{j=1}^{k} w_j I_{(Y_j=2)}$, then the estimate (11) is,

$$\hat{\theta}_1 = \frac{\mathcal{B}(a, k\sum_{j=1}^{k} w_j I_{(Y_j=1)} + 2, k\sum_{j=1}^{k} w_j I_{(Y_j=2)} + 1)}{\mathcal{B}(a, k\sum_{j=1}^{k} w_j I_{(Y_j=1)} + 1, k\sum_{j=1}^{k} w_j I_{(Y_j=2)} + 1)}. \tag{12}$$

Of course, when $a = 1$, then (11) is equivalent to (8) for the two-class problem.

In practice, one may or may not have an idea of what prior information to encode. Incorrect prior assumptions can significantly reduce the effectiveness of MER estimation, as we will see in Subsection 7.2 on simulations. In the next subsections we discuss how to use global likelihood information for prior information.

5.3 Prior Based on Global Likelihood

The estimated $\hat{\theta}$ is based only on a small set of data that are local enough to the test point x to be considered relevant. This local sample set may be too small and random to accurately communicate the true local θ. The entire training set of n training samples is much larger, and may thus provide a more accurate assessment of the class probabilities P_Y unconditioned on any test point x. Thus the entire sample set may be used to form a prior for the class probabilities θ

that is more accurate than a uniform prior. Given no prior information, it may be useful to use the global likelihood of the class labels as a prior over the class probabilities random variable Θ. The global likelihood is

$$\ell(\theta) = \left(\frac{n!}{\prod_{g=1}^{\mathcal{G}} r_g!} \right) \prod_{g=1}^{\mathcal{G}} \theta_g^{r_g} \tag{13}$$

where there are r_g training samples of class g out of n total training samples. Because n is relatively large compared to k, this global likelihood distribution will have a relatively narrow peak. If such a peaked distribution is used as a prior, no local likelihood will affect the estimate significantly, since the local likelihood is based on a subset of the training samples and thus will be less peaked.

For this reason, we propose using the global likelihood as a prior but giving it less weight, which will make it less narrowly peaked. Based on the global likelihood, define a prior function on θ,

$$\ell(\theta) = \prod_{g=1}^{\mathcal{G}} \theta_g^{v(\frac{r_g}{n})}, \tag{14}$$

where the unneeded normalization constants have been dropped and the variable v acts as an artificial number of sample points that the global likelihood prior is based on. For $v = n$, the global likelihood prior is the same as the global likelihood (up to a constant). In practice, v will be set much smaller so that the prior does not overwhelm the local likelihood in the MER estimation. Using this prior (14), the local posterior $f(\theta)$ used for test sample x is then the posterior on θ formed by multiplying the weighted likelihood (4) by the global likelihood prior (14):

$$f(\theta) = \prod_{g=1}^{\mathcal{G}} \left(\theta_g^{k \sum_{j=1}^{k} w_j I_{(Y_j = g)}} \right) \left(\theta_g^{v(\frac{r_g}{n})} \right).$$

It is useful to rewrite this as

$$f(\theta) = \prod_{g=1}^{\mathcal{G}} \theta_g^{k \sum_{j=1}^{k} w_j I_{(Y_j = g)} + v(\frac{r_g}{n})}.$$

Using the above $f(\theta)$ and applying Theorem 1, the MER estimate is

$$\hat{\theta}_g = \frac{k \sum_{j=1}^{k} w_j I_{(Y_j = g)} + v(\frac{r_g}{n}) + 1}{k + v + \mathcal{G}} \tag{15}$$

for either mean-squared error R or relative entropy R, or other Bregman divergence risk.

The value of v should represent how accurate and useful the practitioner thinks the global likelihood is as a prior distribution, versus the uncertainty of using the local k neighbors to estimate θ. In practice, we propose using cross-validation to train both v and the number of neighbors k.

Using the empirical global likelihood still implies that, a priori to the empirical global likelihood information, there was a uniform prior on Θ. Established alternatives to this are the invariant prior $\prod_{g=1}^{\mathcal{G}} \frac{1}{\theta_g}$, and the concept of a *data translated likelihood* that leads to the *noninformative prior* [22] $\prod_{g=1}^{\mathcal{G}} \theta_g^{-\frac{1}{2}}$.

Incorporating global likelihood information was reported to work well on a related problem for rule learning [23]. That approach was called "m-estimates" based on Cestnik's earlier work on Naive Bayes [16]. Their m-estimate is a weighted average of the rule's empirical training samples and the global training samples. Interpreting a probability estimate as a weighted average of the empirical distribution and some more general distribution dates back to [24]; we discuss Carnap's viewpoint and how it adds intuition to our presented estimation results in Subsection 5.6.

5.4 Weighted Global Likelihood Prior

An extension of the proposed global likelihood prior given in (14) is to weight the entire training set to form a weighted global likelihood prior $\ell(\theta)$ that is specific to the test point x. For example, one could weight the entire training set based on each feature vector X_i's distance to the test point x. Let this weighted prior be

$$\ell(\theta) = \prod_{g=1}^{\mathcal{G}} \theta_g^{v \sum_{i=1}^{n} u_i I_{(Y_i=g)}}$$

where constants have been dropped, u_i is the weight placed on the ith training sample and may be a function of x, and v acts as the number of points drawn from the weighted global empirical class distribution.

Multiplying the above weighted global likelihood prior by the local likelihood forms a posterior over the class pmf θ:

$$f(\theta) = \prod_{g=1}^{\mathcal{G}} \theta_g^{k \sum_{j=1}^{k} w_j I_{(Y_j=g)} + v \sum_{i=1}^{n} u_i I_{(Y_i=g)}}. \tag{16}$$

Applying Theorem 1 to solve the MER estimation problem with the weighting $f(\theta)$ from (16) yields the estimate

$$\hat{\theta}_g = \frac{k \sum_{j=1}^{k} w_j I_{(Y_j=g)} + v \sum_{i=1}^{n} u_i I_{(Y_i=g)} + 1}{k + v + \mathcal{G}}. \tag{17}$$

The u_i weights can be used to "localize" the global likelihood by decaying the weight on points with distance to the test point. One idea would be to only weight points "somewhat" local to the test point, in an attempt to optimize a trade-off between a greater number of samples for estimation accuracy and sample locality to the test point for estimation relevance.

5.5 Maximum A Posteriori (MAP) Estimates

The weighted global likelihood given in (16) can similarly be used to create a MAP estimate:

Lemma 3 *The class pmf estimate*

$$\hat{\theta}_g = \frac{k \sum_{j=1}^{k} w_j I_{(Y_j=g)} + v \sum_{i=1}^{n} u_i I_{(Y_i=g)}}{k + v}, \tag{18}$$

is the maximum a posteriori estimate for the weighted global likelihood prior given in (16), subject to the constraints

$$\sum_{g=1}^{\mathcal{G}} \theta_g = 1, \ \sum_{j=1}^{k} w_j = 1$$

Problems can occur with the MAP estimate when using standard non-uniform priors. For example, the well-established noninformative prior $\prod_{g=1}^{\mathcal{G}} \theta_g^{-1/2}$ [22] leads to a MAP estimate of

$$\hat{\theta}_g = \frac{m_g - 1/2}{k - 1},$$

which for one near-neighbor will be infinite.

5.6 Carnapian Interpretation

The proposed MER estimate with a global likelihood prior can be interpreted within an estimation framework proposed by Carnap in 1952. Although Carnap's views were not Bayesian, he proposed a general continuum of induction rules that correspond to a Bayesian minimum expected risk estimation framework using a range of different prior information [11, pg. 279]. Carnap noted that there were two extremes to the multinomial estimation problem (Carnap and Jaynes both gave binomial examples, but their logic extrapolates straightforwardly to the multinomial case). At the one extreme is the empirical distribution $\hat{\theta}_g = m_g/k$. At the other extreme is what Carnap refers to as a logical factor, which corresponds to an uninformed guess, such as the estimate $\hat{\theta}_g = 1/\mathcal{G}$. Carnap noted that experts in his time agreed that the best estimate is somewhere between those two extreme estimates, and considered a convex weighting of the two extreme estimates to form a continuum of inductive rules. Any point on Carnap's continuum is seen to correspond to a different prior in the Bayesian minimum expected risk framework.

The Laplace correction estimate (9) lies along this continuum, as pointed out by Carnap himself [24, pg. 35]. Rewrite (9) as

$$\hat{\theta}_g = \frac{k(\frac{m_g}{k}) + \mathcal{G}(\frac{1}{\mathcal{G}})}{k + \mathcal{G}}. \tag{19}$$

The above re-expression of the Laplace correction estimate can be interpreted as the weighted average of k points from the empirical distribution m_g/k for $g \in \{1, \ldots, \mathcal{G}\}$ and $\mathcal{G}$ points from the uniform distribution prior over the classes [11, pg 158].

Similarly, the MER estimate using a global likelihood weighted prior and uniform weights on the neighborhood points ($w_j = 1/k$ for all j) can be re-expressed:

$$\hat{\theta}_g = \frac{k(\frac{m_g}{k}) + v(\frac{r_g}{n}) + \mathcal{G}(\frac{1}{\mathcal{G}})}{k + v + \mathcal{G}}. \tag{20}$$

The above re-expression of the MER estimate can be interpreted as the weighted average of k points from the local empirical distribution m_g/k for $g \in \{1, \ldots, \mathcal{G}\}$, v points from the global empirical distribution r_g/n for $g \in \{1, \ldots, \mathcal{G}\}$, and $\mathcal{G}$ points from the uniform distribution over the classes.

6 Minimizing Expected Cost

Ideally, the estimated class would minimize

$$\underset{g}{\text{argmin}} \sum_{h=1}^{\mathcal{G}} C(g,h)\theta_h, \tag{21}$$

where $\theta = (\theta_1, \ldots, \theta_{\mathcal{G}})$ is the underlying probability density over the class labels $\{1, 2, \ldots, \mathcal{G}\}$. But in practice the true class probability distribution θ is not known, so it is estimated by maximum likelihood (or as proposed in this paper, minimum expected risk) based on the training set $\mathcal{T}$. Then the formula (2) is used to classify x as some $\hat{Y}$.

In this way, standard near-neighbor classification is a two-step process, where first a class pmf $\hat{\theta}$ is estimated, and then the class with the minimum expected cost is estimated as in (2). In this section, we propose classifying in one step. The one-step estimated class will minimize the expected misclassification cost, where the expectation is over the class pmf as well as over the misclassification cost conditioned on the class pmf. In this way, one directly minimizes the expected misclassification cost, and does not need to form an intermediate class pmf estimate with respect to some risk R on the pmf estimates. This also avoids the question of which risk function R is appropriate to use.

As analyzed by [25], classification is robust to large errors in the class pmf estimate if the errors are in the "right" direction. Thus one might expect different classification results by directly estimating the class that minimizes the expected misclassification cost, since the intermediate step of estimating the class pmf $\hat{\theta}$ is skipped.

We propose that in equation (21) the uncertainty about θ be modeled by a random vector Θ. Then the class is estimated by minimizing the expected value of $\sum_{h=1}^{\mathcal{G}} C(g,h)\Theta_h$ over the class labels, where the expectation is taken with respect to random vector Θ:

$$\underset{g}{\text{argmin}}\, E_{\Theta}[\sum_{h=1}^{\mathcal{G}} C(g,h)\Theta_h], \tag{22}$$

where $\Theta = (\Theta_1, \Theta_2, \ldots, \Theta_{\mathcal{G}}) \in [0,1]^{\mathcal{G}}$ is a random vector such that $\sum_{i=1}^{\mathcal{G}} \Theta_i = 1$.

The following corollary to Theorem 1 establishes that the result of this one-step classification is in fact equivalent to the two-step classification given in (2) using the MER estimate if $f(\theta)$ is defined the same in both classification approaches.

Corollary 1 *Suppose that $P\{\Theta = \theta\} = f(\theta) = f(\theta_1, \theta_2, \ldots, \theta_{\mathcal{G}}) = \gamma \prod_{g=1}^{\mathcal{G}} \theta_g^{\alpha_g}$ (likelihood function or a posteriori function). Let $C(g,h)$ be the cost of estimating class g when the truth is class h. Then choosing a class label g as per*

(22) is equivalent to

$$\operatorname*{argmin}_{g} \sum_{h=1}^{\mathcal{G}} C(g,h)\hat{\theta}_h$$

where

$$\hat{\theta}_h = \frac{\alpha_h + 1}{\sum_{h=1}^{\mathcal{G}} \alpha_h + \mathcal{G}}.$$

7 Is the Difference Significant?

The MER estimates theoretically minimize the expected risk (and classification error). In practice, is the difference between MER and ML estimation significant for near-neighbor learning? And how sensitive is the MER estimate to the assumed prior? We turn first to the first question. In the simulations of Subsection 7.2, we investigate the second question.

Given zero neighbors of class one out of two neighbors, or zero neighbors of class one out of one thousand neighbors, the ML estimate is the same. The ML estimate is only a function of the ratio of each class's neighbors to the total number of neighbors k. The MER estimate can be written

$$\hat{theta}_g = \frac{\frac{m_g}{k} + \frac{1}{k}}{1 + \frac{\mathcal{G}}{k}}.$$

From the above, it is seen that the MER estimate is a function of the ratio m_g/k, but is also a function of k. As k grows larger the MER estimate moves away from the uniform (or other prior), and closer to the neighbor's empirical class distribution. The smaller k is, the larger the difference between the MER and ML estimates. In the limit of $k \to \infty$, the ML and MER class probability estimates for a test point x converge. However, kNN algorithms are often run for small values of k, including $k = 1$. Also different from ML, the MER estimate is a function of the number of classes $\mathcal{G}$, and for larger numbers of classes and the same number of neighbors k, the estimate is less trusting of the empirical class distribution and closer to the prior.

For classification with two classes, the class label is determined by whether the class probability estimate for class one is above or below a threshold. For symmetric misclassification costs, the threshold is set at .5. For the two-class problem with class one and class two, the classification threshold t is theoretically optimally set [2] to minimize expected cost at

$$t = \frac{C(1,2)}{C(1,2) + C(2,1)}. \tag{23}$$

In practical learning problems such as computer-aided diagnostics of medical problems, the costs can be extremely asymmetric.

For symmetric classification costs, the classification decision will be unchanged given a MER or ML estimate of the class pmf, as stated in the following lemma for the two-class problem.

Lemma 4 *Let ϕ be a classifier that classifies a given test sample x, then*

$$\phi(x) = \begin{cases} 1 & : \text{ if } \hat{\theta}_{1,ML} > \frac{1}{2} \\ 0 & : \text{ otherwise} \end{cases} \Leftrightarrow \phi(x) = \begin{cases} 1 & : \text{ if } \hat{\theta}_{1,MER} > \frac{1}{2} \\ 0 & : \text{ otherwise.} \end{cases} \tag{24}$$

In the simulations of Subsection 7.2 we see that the farther the threshold is from .5 (due to asymmetric misclassification costs), the larger the difference between the two estimates.

7.1 Asymptotics

Near-neighbor classifiers have well-studied asymptotic behavior. We note that using MER estimation instead of ML estimation will not change the standard asymptotic near-neighbor results.

A supervised learning algorithm is L^r *consistent* if, when (X,Y), $(X_1,Y_1),(X_2,Y_2),\ldots,(X_n,Y_n)$ are iid, Y is real-valued, $r > 1$, and $E[|Y|^r] < \infty$, then $\hat{Y}(X) \to E[Y|X]$ in L^r. Using ML estimation, many near-neighbor classification methods are consistent under the standard assumptions that the near-neighbors are the k nearest, that the $k \to \infty$, while the total number of samples $n \to \infty$ and $k/n \to \infty$ [4, 6, 26].

Using uniform prior information, MER estimates as per (10) are trivially consistent for cases where maximum likelihood estimates are consistent, since the MER estimate converges to the ML estimate as $k \to \infty$.

For finite k and $n \to \infty$, asymptotic results relate the kNN error to the Bayes' error [27, 6]. By Lemma 4, the MER results will be the same as the standard ML results in these cases.

7.2 Simulations

To explore the difference between the MER and ML estimates, we present two simulations. Each simulation measures the misclassification costs in a two-class problem over a range of thresholds t where t is related to the misclassification costs as per (23), and $C(1,1) = C(2,2) = 0$. Over the range of $t = 0$ to $t = 1$, the cost matrix changes so that $C(1,2) + C(2,1) = 1$ is held constant.

There are many nonparametric neighborhood classifiers; see [6, 2, 28] for reviews and discussion. Results are given for kNN, for the symmetric tricube kernel [2, p. 168], and for a recent asymmetric kernel method that uses linear interpolation and the principle of maximum entropy (LIME) [29, 3, 26].

The tricube kernel is representative of the general class of positive, symmetric, smoothing kernels. Given a test sample x and k training samples $\{x_1, x_2, \ldots, x_k\}$, the tricube weight w_j is

$$w_j = \frac{(1 - \|x - x_j\|_2^3)^3}{\sum_{i=1}^{k}(1 - \|x - x_i\|_2^3)^3}.$$

The LIME weights solve a minimization problem. Let $\mathcal{W}$ be the collection of all probability mass functions w, that is, all n-tuples for which $w_i \geq 0$ if $i \leq k$ and $w_i = 0$ otherwise, and $\sum_i w_i = 1$, for all $i = 1, \ldots, n$. Then the LIME weights w^* solve

$$\underset{w \in \mathcal{W}}{\operatorname{argmin}} \left(D\left(\sum_{i=1}^{k} w_i x_i - x \right) - \lambda H(w) \right), \tag{25}$$

where D is some convex distortion function and $H(w)$ is the Shannon entropy. The first term of the LIME minimization attempts to find weights that solve the linear interpolation equations, making x the center of its weighted neighbors x_i. This is directly related to reducing the first-order bias of the estimate. The second term of the LIME minimization attempts to maximize entropy, which keeps the variance of the estimate low. The LIME weights are defined in terms of a trade-off parameter λ. Although λ can be trained using cross-validation, for these comparisons we set λ to a default low value ($\lambda = 10^{-6}$). Squared l_2 distance is used for D, and the optimization of (25) is done with a fast primal-dual log-barrier interior-point method.

7.3 Unit Square Simulation

For the unit square simulation, training samples and test samples are independently and identically drawn uniformly from a two-dimensional unit square. Each sample X_i has a probabilistic class label based on the sum of its components: $P(Y = 2) = .5X_i[1] + .5X_i[2]$ and $P(Y = 1) = 1 - P(Y = 2)$. The left side of Figure 1 shows an example of 1000 sample points.

Note that for this simulation the prior assumptions behind the MER estimate (10) hold because the true class probability θ is in fact uniform.

A set of 1000 test points was drawn, and then for each of 50 runs of the simulation, 100 different training points were drawn. The number of neighbors ranged from 1 to 10 nearest-neighbors in terms of Euclidean distance. Results were averaged over the 1000 test points and the 50 sets of different training samples. Figure 2 (left side) shows the performance of 1NN (which is the same for all near-neighbor methods). The cost curves are piecewise linear because with 1NN ML the class probability estimates are either $\hat{\theta}_1 = 0$ or $\hat{\theta}_1 = 1$, and thus the classification errors are the same for $t < 1/2$ and for $t \geq 1/2$. Since the costs go as t, the same number of classification errors appears as a linear cost segment in the figure. For 1NN MER, the class probability estimates are either $\hat{\theta}_1 = 1/3$ or $\hat{\theta}_1 = 2/3$, and thus the classification errors are the same for $t < 1/3$, for $1/3 \leq t < 2/3$, and for $t \geq 2/3$.

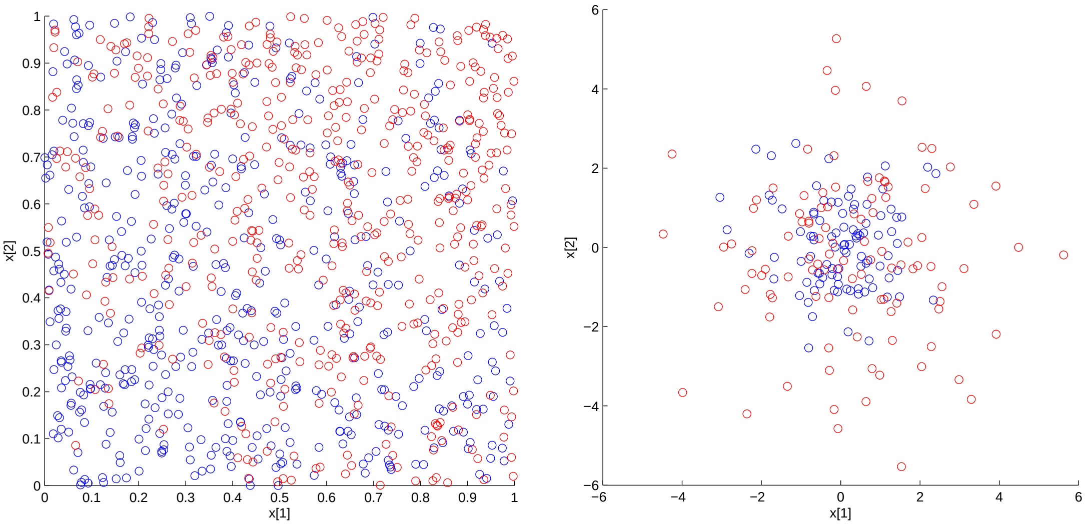

Figure 1: (left) Example of 1000 training points from the unit square simulation. (right) Example of 200 training points from the Gaussian simulation.

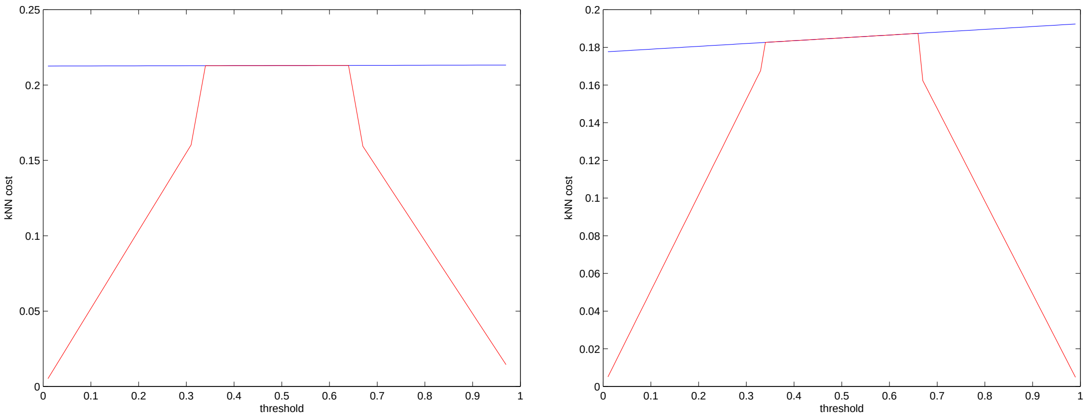

Figure 2: (left) Results from the unit square simulation with one nearest neighbor. (right) Results from the Gaussian simulation with one nearest neighbor. (blue = ML; red = MER)

Figure 3 shows the performance using five nearest neighbors, and Figure 4 shows the performance using ten nearest neighbors. As predicted theoretically, the MER estimates perform better, the performance difference is generally greater for more disparate misclassification costs, and the performance difference shrinks as k becomes larger. Results using $10,000$ training samples for each of 50 runs were very similar to the results presented here, which used 100 training samples for each of 50 runs.

A different perspective on the same results is given in Figure 5, where the ratio of the misclassification costs is plotted for one near-neighbor and five near-neighbors. Cost ratios for very asymmetric costs ($t < .2$ or $t > .8$) are striking.

7.4 Gaussian Simulation

A popular simulation example is used based on Gaussians with the same center [30, 31, 2, 3]. Training and test samples are drawn iid in a two-dimensional Euclidean space, and are equally likely to be from class one or class two. Class one points are distributed as a Gaussian, $\mathcal{N}(0, \Sigma)$, where the covariance matrix Σ is the 2×2 identity matrix. Class two points are similarly distributed as a Gaussian, $\mathcal{N}(0, 4\Sigma)$. A two-dimensional example of 200 sample points is given in the right side of Figure 1.

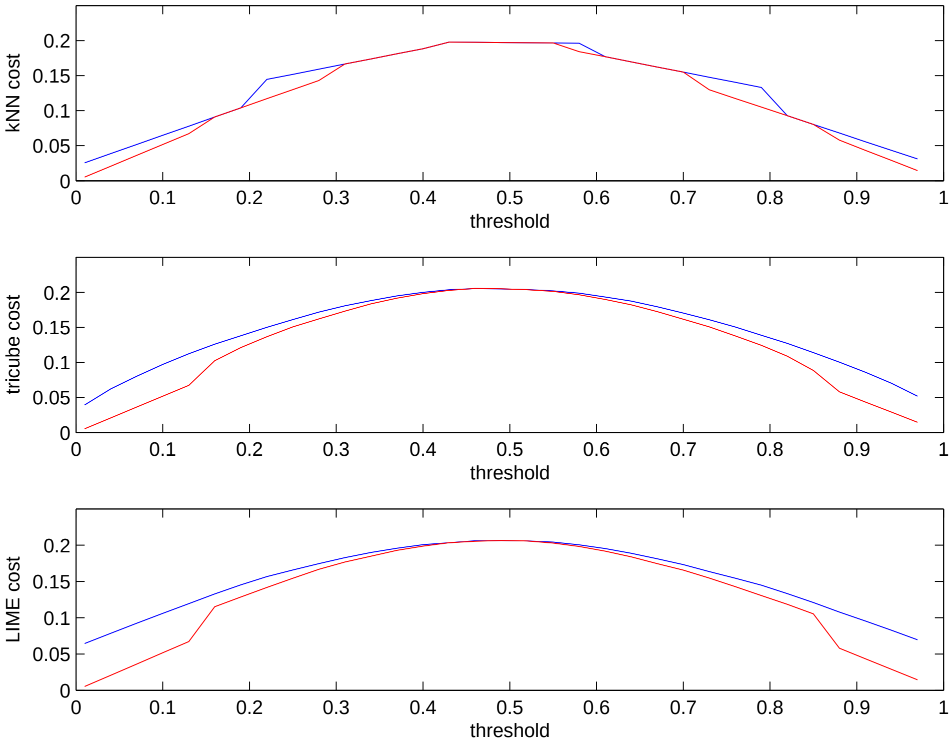

Figure 3: Results from the unit square simulation with five nearest neighbors. (blue = ML; red = MER)

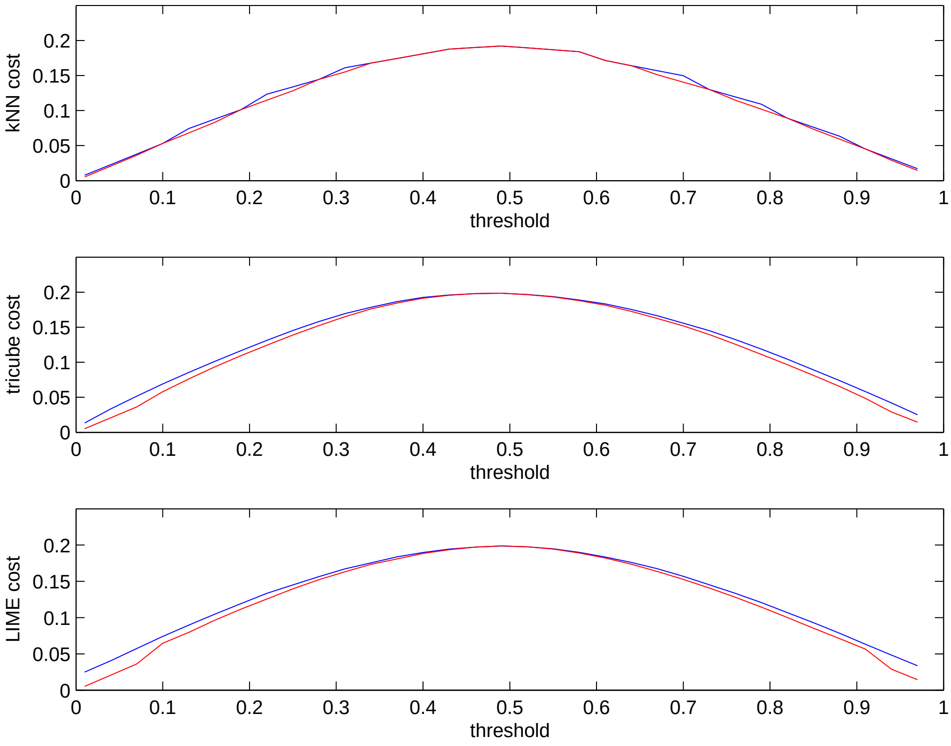

Figure 4: Results from the unit square simulation with ten nearest neighbors. (blue = ML; red = MER)

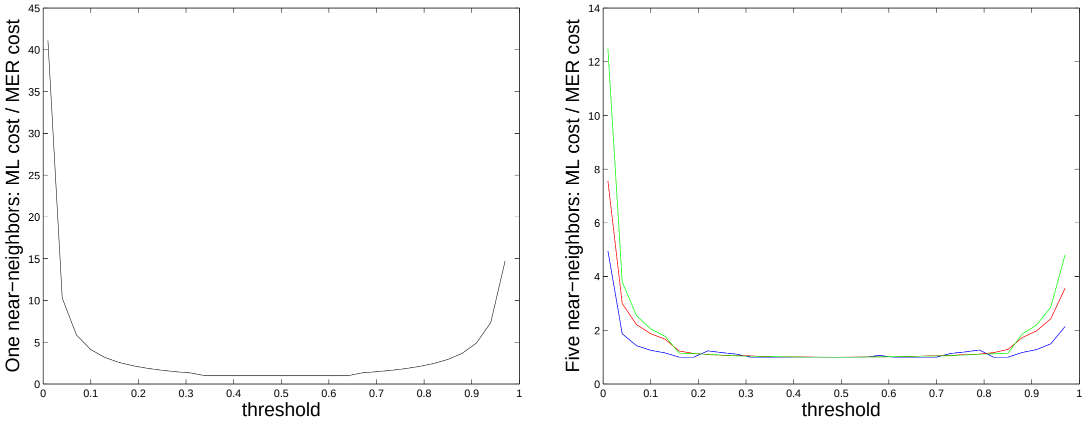

Figure 5: Left: Cost ratio for the unit square simulation for one nearest neighbor. Right: Cost ratios for the unit wquare simulation for five nearest neighbors. (blue = kNN; red = tricube; green = LIME)

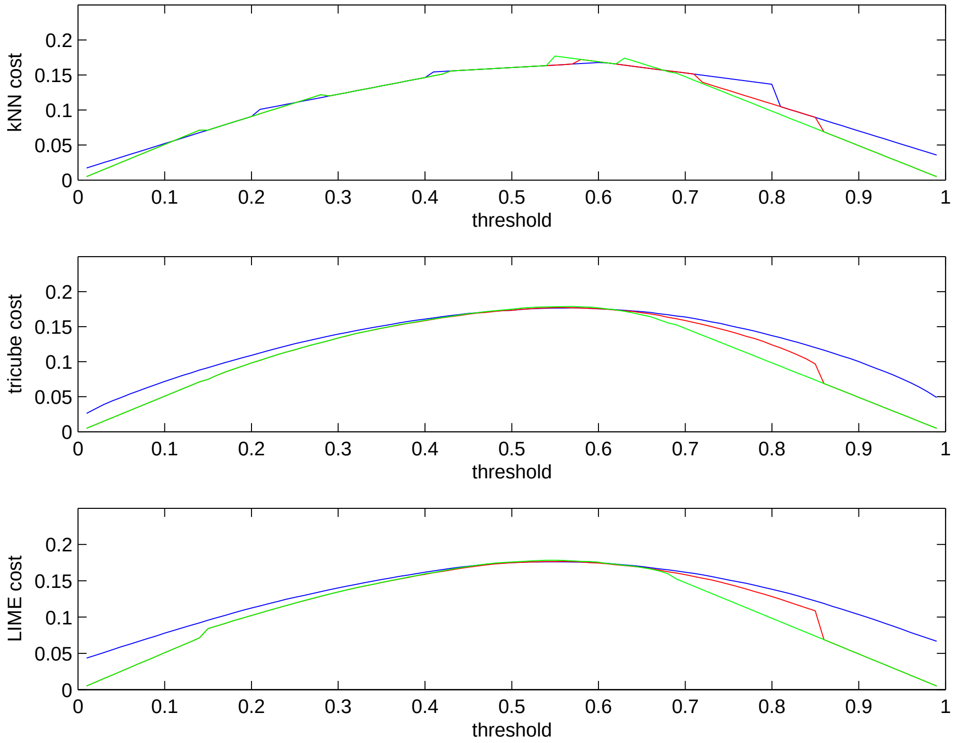

Figure 6: Results from the Gaussian simulation with five nearest neighbors. (blue = ML; red = MER; green = MER with prior; where the red line disappears, it overlaps the green line.)

For this simulation the prior assumption of uniformly likely class pmf θ used to derive the MER estimate (10) does *not* hold. For one, the probability of class one, θ_1, is never greater than $.8$ because the maximum of the ratio of the class one pdf to the sum of the class pdfs is .8:

$$\max_x \frac{\mathcal{N}(0,\Sigma)}{\mathcal{N}(0,\Sigma)+\mathcal{N}(0,4\Sigma)} = .8.$$

In practice, it might be difficult to know what an appropriate range for the prior likelihood of θ is. In order to investigate how much prior information matters, we compare ML, MER (with uniform prior on θ), and MER with a prior that restricts $\theta_1 \leq .8$, as per (11). Classifying based on one nearest-neighbor is shown in Figure 2 (right side).

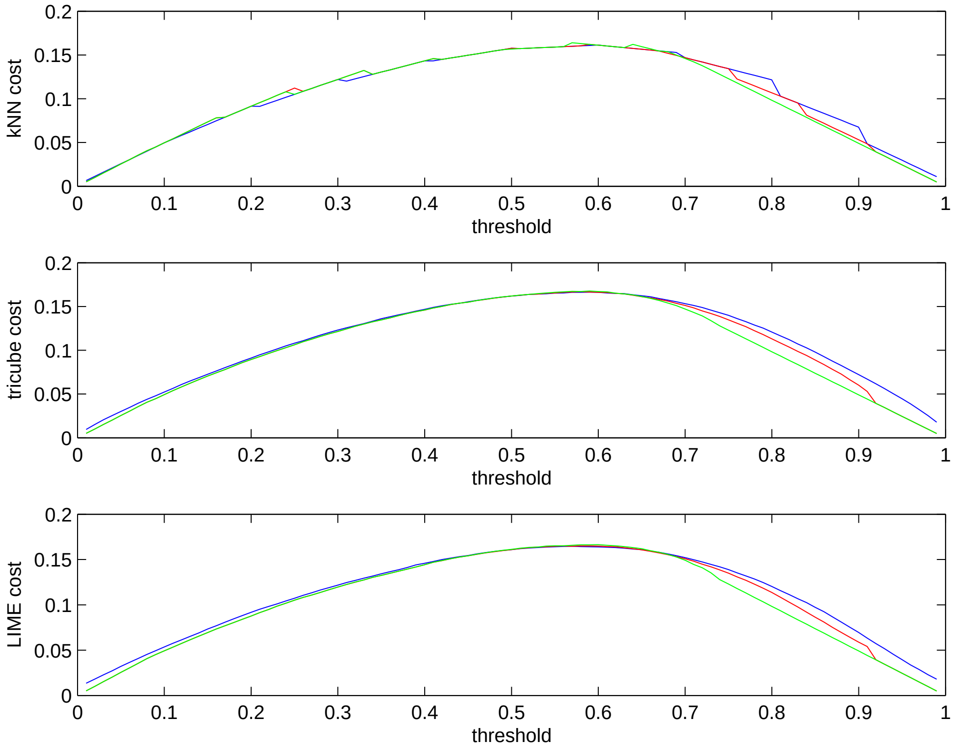

Figure 7: Results from the Gaussian simulation with ten nearest neighbors. (blue = ML; red = MER; green = MER with prior; where the red line disappears, it overlaps the green line.)

Classifying from five nearest neighbors and from ten nearest neighbors is shown in Figure 6 and Figure 7, respectively. The impact of the additional prior information is seen to improve performance.

8 Discussion

In this work we have investigated how minimum expected risk and minimum expected cost principles can improve average performance for near-neighbor classification methods. We have shown that for symmetric misclassification costs there is no difference to using prior information with maximum likelihood or minimum expected risk. However, the more asymmetric the misclassification costs, the greater benefit to be gained from the minimum expected risk principle. Even in a simple two-dimensional simulation, the classification cost is as much as forty times smaller using the minimum expected risk estimate.

The use of correct prior information can have a strong effect on the outcome, and we have discussed a few different cases for prior information. On the one hand, the practitioner with good information should be able to communicate that information to the learning algorithm, via the setting of this prior on the class pmf's. On the other hand, reasonably useful prior information may be difficult to obtain. The theoretical promise of the name "minimum expected risk estimation" delivers only to the extent that the prior information is useful.

This research leaves open some questions. One is how to derive useful prior information from the training data. In this paper we showed how to use the global likelihood, but it may be possible to estimate other prior information. A second question is how to analytically solve for estimates given other prior information. For the case of limited prior conditional probability of a class, such as $\theta_1 < .8$, we gave results for the two-class case by using the incomplete beta function. Solving for more classes would require a multiple incomplete beta function, of which the authors are not aware. Other prior information might not lead to analytic constraints, and the integration of risk might require Monte Carlo sampling.

Lastly, this paper has focused on minimizing expected misclassification cost for near-neighbor learning. Many other approaches to learning use ML estimation steps which could be replaced by MER estimation. Investigating the application of MER estimation to other learning paradigms may be an interesting avenue of future research.

9 Acknowledgements

This project received support from the Office of Naval Research, Code 321, Grant # N00014-05-1-0843, and from the National Science Foundation, Grant # SBE-0123552. The authors would like to thank Joydeep Ghosh and Inderjit Dhillon for helpful discussions.

Appendix

In this appendix we prove the results stated in the paper in the order the results appear.

Proof of Lemma 1. The proof of Lemma 1 is the same as the proof for Lemma 3, with the parameter v set to zero.

Proof of Theorem 1. The MER estimate $\hat{\theta}$ solves $\operatorname{argmin}_\phi p(\phi)$, where

$$p(\phi) = E_\Theta[R(\Theta,\phi)] = E_\Theta[d_\psi(\Theta,\phi)] = \int_\theta d_\psi(\theta,\phi)f(\theta)d\theta. \tag{26}$$

Substituting the definition of Bregman divergence from equation (7), we write

$$p(\phi) = \int_\theta [\psi(\theta) - \psi(\phi) - (\theta-\phi)^T\nabla\psi(\phi)] \prod_{g=1}^{\mathcal{G}} \theta_g^{\alpha_g} d\theta. \tag{27}$$

Differentiating both sides with respect to ϕ,

$$\begin{aligned}\nabla p(\phi) &= \int_\theta [-\nabla\psi(\phi) + \nabla\psi(\phi) - \nabla^2\psi(\phi)(\theta-\phi)] \prod_{g=1}^{\mathcal{G}} \theta_g^{\alpha_g} d\theta & (28)\\ &= -\int_\theta [\nabla^2\psi(\phi)(\theta-\phi)] \prod_{g=1}^{\mathcal{G}} \theta_g^{\alpha_g} d\theta. & (29)\end{aligned}$$

Setting the first-order optimality condition $\nabla p(\phi)$ equal to zero and solving for ϕ,

$$\nabla p(\phi) = 0 \Rightarrow \ \nabla^2\psi(\phi) \int_\theta (\theta-\phi) \prod_{g=1}^{\mathcal{G}} \theta_g^{\alpha_g} d\theta = 0. \tag{30}$$

Because of the strict convexity of ψ, $\nabla^2\psi(\phi)$ is a $\mathcal{G} \times \mathcal{G}$ positive definite Hessian matrix, and thus equation (30) implies that

$$\int_\theta (\theta-\phi) \prod_{g=1}^{\mathcal{G}} \theta_g^{\alpha_g} d\theta = 0 \Leftrightarrow \phi = \frac{\int_\theta \theta \prod_{g=1}^{\mathcal{G}} \theta_g^{\alpha_g} d\theta}{\int_\theta \prod_{g=1}^{\mathcal{G}} \theta_g^{\alpha_g} d\theta}. \tag{31}$$

Taking the gth element of vector equation (31),

$$\phi_g = \frac{\int_\theta \theta_g \prod_{g=1}^{\mathcal{G}} \theta_g^{\alpha_g} d\theta}{\int_\theta \prod_{g=1}^{\mathcal{G}} \theta_g^{\alpha_g} d\theta}. \tag{32}$$

Using Dirichlet's integral [32, pgs. 32-34] equation (32) can be written as

$$\phi_g = \frac{\{\Gamma(\alpha_1+1)\ldots\Gamma(\alpha_{g-1}+1)\Gamma(\alpha_g+2)\Gamma(\alpha_{g+1}+1)\ldots\Gamma(\alpha_{\mathcal{G}}+1)\}/\Gamma(\mathcal{G}+1+\sum_{j=1}^{\mathcal{G}}\alpha_j)}{[\Gamma(\alpha_1+1)\Gamma(\alpha_2+1)\ldots\Gamma(\alpha_{\mathcal{G}}+1)]/\Gamma(\mathcal{G}+\sum_{j=1}^{\mathcal{G}}\alpha_j)},$$

which simplifies to

$$\phi_g = \frac{(\alpha_g + 1)[\prod_{g=1}^{\mathcal{G}} \Gamma(\alpha_g + 1)]/\Gamma(\mathcal{G} + 1 + \sum_{j=1}^{\mathcal{G}} \alpha_j)}{[\prod_{g=1}^{\mathcal{G}} \Gamma(\alpha_g + 1)]/\Gamma(\mathcal{G} + \sum_{j=1}^{\mathcal{G}} \alpha_j)},$$

and thus,

$$\hat{\theta}_g = \frac{\alpha_g + 1}{\sum_{g=1}^{\mathcal{G}} \alpha_g + \mathcal{G}}, \qquad g = 1, 2, \ldots, \mathcal{G},$$

which is the estimate $\hat{\theta}_g$ given in (8) of Theorem 1.

An alternative proof uses a recent result [20, Theorem1], which establishes that the minimizer of $E[d_\psi(R, s)]$ is the mean of R. Recognizing that ϕ_g in (32) is the mean of the random variable Θ_g with respect to the normalized pdf

$$\frac{f(\theta)}{\int_\theta f(\theta) d\theta},$$

it follows that ϕ_g is the sought minimizer. More recent results on minimizing the expected Bregman divergence are given in [21].

Proof of Lemma 2. The first part of this proof follows the proof of Theorem 1, except that $f(\theta)$ in Theorem 1 is replaced by $f(\theta)q(\theta)$. Equation (32) becomes,

$$\phi_1 = \frac{\int_{\theta_1=0}^{a} \theta_1 \prod_{g=1}^{2} \theta_g^{\alpha_g} d\theta_1}{\int_{\theta_1=0}^{a} \prod_{g=1}^{2} \theta_g^{\alpha_g} d\theta_1}. \tag{33}$$

The numerator and denominator may be written as standard incomplete Beta functions $\mathcal{B}$. Thus (33) becomes

$$\phi_1 = \frac{\mathcal{B}(a, \alpha_1 + 2, \alpha_2 + 1)}{\mathcal{B}(a, \alpha_1 + 1, \alpha_2 + 1)}, \tag{34}$$

as stated in the lemma.

Proof of Lemma 3. The Lagrangian $l(\theta)$ of the problem is

$$l(\theta) = \prod_{g=1}^{\mathcal{G}} \theta_g^{[k \sum_{j=1}^{k} w_j I_{(Y_j = g)} + v \sum_{i=1}^{n} u_i I_{(Y_i = g)}]} - \lambda \left(\sum_{g=1}^{\mathcal{G}} \theta_g - 1 \right),$$

where λ is the Lagrange multiplier, with the conditions

$$\lambda(\sum_{g=1}^{\mathcal{G}} \theta_g - 1) = 0, \tag{35}$$

$$\sum_{j=1}^{k} w_j = 1 \sum_{i=1}^{n} u_i = 1. \tag{36}$$

Use the first order condition to solve for the optimal solution. When differentiating with respect to the variable θ_h and setting it equal to zero, we have for $h = 1, 2, \ldots, \mathcal{G}$,

$$\lambda = \frac{[k \sum_{j=1}^{k} w_j I_{(Y_j = h)} + v \sum_{i=1}^{n} u_i I_{(Y_i = h)}]}{\hat{\theta}_h} \prod_{g=1}^{\mathcal{G}} \hat{\theta}_g^{[k \sum_{j=1}^{k} w_j I_{(Y_j = g)} + v \sum_{i=1}^{n} u_i I_{(Y_i = g)}]}. \tag{37}$$

Taking any two of $\mathcal{G}$ equations of (37) and eliminating λ yields

$$\frac{k \sum_{j=1}^{k} w_j I_{(Y_j = h)} + v \sum_{i=1}^{n} u_i I_{(Y_i = h)}}{\hat{\theta}_h} = \frac{k \sum_{j=1}^{k} w_j I_{(Y_j = \bar{h})} + v \sum_{i=1}^{n} u_i I_{(Y_i = \bar{h})}}{\hat{\theta}_{\bar{h}}},$$

which, solving for $\hat{\theta}_h$, gives

$$\hat{\theta}_h = \frac{k\sum_{j=1}^{k} w_j I_{(Y_j=h)} + v\sum_{i=1}^{n} u_i I_{(Y_i=h)}}{k\sum_{j=1}^{k} w_j I_{(Y_j=\bar{h})} + v\sum_{i=1}^{n} u_i I_{(Y_i=\bar{h})}} \hat{\theta}_{\bar{h}} \quad h = 1,\ldots,\mathcal{G}.$$

Then, using the constraint (35) on $\hat{\theta}_h$ gives

$$\bar{\theta}_{\bar{h}} \sum_{h=1}^{\mathcal{G}} [k\sum_{j=1}^{k} w_j I_{(Y_j=h)} + v\sum_{i=1}^{n} u_i I_{(Y_i=h)}] = k\sum_{j=1}^{k} w_j I_{(Y_j=\bar{h})} + v\sum_{i=1}^{n} u_i I_{(Y_i=\bar{h})}.$$

Since $\sum_{g=1}^{\mathcal{G}}[k\sum_{j=1}^{k} w_j I_{(Y_j=g)} + v\sum_{i=1}^{n} u_i I_{(Y_i=g)}] = k + v$ by (36), the MAP estimate $\hat{\theta}_h$ is as stated in the lemma.

Proof of Corollary 1. The following are equivalent:

$$\begin{aligned} & \text{argmin}_g \quad E_\Theta[\sum_{h=1}^{\mathcal{G}} C(g,h)\Theta_h] && (38)\\ \equiv\ & \text{argmin}_g \quad \sum_{h=1}^{\mathcal{G}} C(g,h)E_\Theta[\Theta_h] && (39)\\ \equiv\ & \text{argmin}_g \quad \sum_{h=1}^{\mathcal{G}} C(g,h)\hat{\theta}_h, && (40)\end{aligned}$$

where (39) follows from (38) by the linearity of expectation, and (40) follows from (39) because the MER estimate $\hat{\theta}_h$ as stated in Theorem 1, Equation (8) is equal to $E_\Theta[\Theta_h]$ as shown in the proof of Theorem 1 (32).

Proof of Lemma 4. First we show that $\hat{\theta}_{1,ML} > \frac{1}{2}$ implies $\hat{\theta}_{1,MER} > \frac{1}{2}$. Rewriting $\hat{\theta}_{1,MER}$ in terms of $\hat{\theta}_{1,ML}$,

$$\hat{\theta}_{1,MER} = \frac{k(\hat{\theta}_{1,ML}) + 2(\frac{1}{2})}{k+2} > \frac{k(\frac{1}{2}) + 2(\frac{1}{2})}{k+2} = \frac{(k+2)(\frac{1}{2})}{k+2} = \frac{1}{2}.$$

To prove the reverse, assume that $\hat{\theta}_{1,MER} > 1/2$,

$$\hat{\theta}_{1,MER} = \frac{k(\hat{\theta}_{1,ML}) + 2(\frac{1}{2})}{k+2} > \frac{1}{2}.$$

Cross-multiplying and solving for $\hat{\theta}_{1,ML}$,

$$\begin{aligned} k(\hat{\theta}_{1,ML}) + 2(\frac{1}{2}) &> (k+2)\frac{1}{2} = \frac{k}{2} + 1\\ \Rightarrow \quad \hat{\theta}_{1,ML} &> \frac{1}{2}.\end{aligned}$$

References

[1] M. R. Gupta, L. Cazzanti, and S. Srivastava, “Minimum expected risk estimates for nonparametric neighborhood classifiers,” *Proceedings of the IEEE Workshop on Statistical Signal Processing*, 2005.

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer-Verlag, 2001.

[3] M. R. Gupta, R. M. Gray, and R. A. Olshen, “Nonparametric supervised learning with linear interpolation and maximum entropy,” *IEEE Trans. on Pattern Analysis and Machine Learning. Preprint available at www.ee.washington.edu/research/guptalab/publications.html*, forthcoming.

[4] C. J. Stone, "Consistent nonparametric regression," *The Annals of Statistics*, vol. 5, no. 4, pp. 595–645, 1977.

[5] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge: Cambridge University Press, 2003.

[6] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer-Verlag, 1996.

[7] E. L. Lehmann and G. Casella, *Theory of Point Estimation*. New York: Springer, 1998.

[8] L. Bregman, "The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, pp. 200–217, 1967.

[9] S. Censor and Y. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford, England: Oxford University Press, 1997.

[10] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*. Chicester, England: Wiley Series in Probability and Statistics, 2000.

[11] E. T. Jaynes and G. T. Bretthorst, *Probability Theory: the Logic of Science*. Cambridge: Cambridge University Press, 2003.

[12] F. Provost and P. Domingos, "Tree induction for probability-based rankings," *Machine Learning*, vol. 52, no. 3, pp. 199–216, 2003.

[13] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning*, vol. 36, pp. 105–139, 1999.

[14] J. P. Bradford, C. Kunz, R. Kohavi, C. Brunk, and C. E. Brodley, "Pruning decision trees with misclassification costs," *Proceedings of the Tenth European Conference on Machine Learning*, pp. 131–136, 1998.

[15] T. Niblett, "Constructing decision trees in noisy domains," *Proceedings of the Second European Working Session on Learning*, pp. 67–78, 1987.

[16] B. Cestnik, "Estimating probabilities: A crucial task in machine learning," *Proceedings of the European Conference on Artificial Intelligence*, pp. 147–149, 1990.

[17] P. Clark and R. Boswell, "Rule induction with CN2: Some recent improvements," *Proceedings of the Fifth European Working Session on Learning*, pp. 151–163, 1991.

[18] P. Domingos, "Unifying instance-based and rule-based induction," *Machine Learning*, vol. 24, pp. 141–168, 1996.

[19] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?" *Proceedings of the IEEE*, vol. 88, no. 8, 2000.

[20] A. Banerjee, X. Guo, and H. Wang, "On the optimality of conditional expectation as a Bregman predictor," *IEEE Trans. on Information Theory*, vol. 51, no. 7, pp. 2664–2669, 2005.

[21] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.

[22] G. E. P. Box and G. C. Tiao, *Bayesian inference in statistical analysis*. Reading, Massachusetts: Addison-Wesley, 1973.

[23] S. Džeroski, B. Cestnik, and I. Petrovski, "Using the m-estimate in rule induction," *Journal of Computing and Information Technology*, pp. 37–46, 1993.

[24] R. Carnap, *The Continuum of Inductive Methods*. Chicago: University of Chicago Press, 1952.

[25] J. H. Friedman, “On bias, variance, 0/1 loss, and the curse-of-dimensionality,” *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 55–77, 1997.

[26] M. P. Friedlander and M. R. Gupta, “On minimizing distortion and relative entropy,” *IEEE Trans. on Information Theory. Preprint available at ee.washington.edu/research/guptalab/publications/FriedlanderGupta.pdf*, forthcoming.

[27] T. M. Cover and P. E. Hart, “Nearest neighbor pattern classification,” *IEEE Trans. Information Theory*, vol. 13, pp. 21–27, 1967.

[28] J. Rice, “Boundary modification for kernel regression,” *Communications in Statistics, Theory and Methods*, vol. 13, pp. 893–900, 1984.

[29] M. R. Gupta and R. M. Gray, “Reducing bias in supervised learning,” *Proceedings of the IEEE Workshop on Statistical Signal Processing*, pp. 482–485, 2003.

[30] T. Kohonen, G. Barna, and R. Chrisley, “Statistical pattern recognition with neural networks: benchmarking studies,” *Proceedings of the IEEE Intl. Conf. on Neural Networks*, vol. 1, pp. 61–68, 1988.

[31] R. M. Gray and R. A. Olshen, “Vector quantization and density esimation,” *Proceedings of the Compression and Complexity of Sequences Conference*, pp. 172–193, 1997.

[32] G. E. Andrews, R. Askey, and R. Roy, *Special functions*. New York: Cambridge University Press, 2000.