

# Completely Lazy Classifiers: Bayesian Neighborhoods

**Eric K. Garcia**  
**Sergey Feldman**  
**Maya R. Gupta**

*Department of Electrical Engineering  
University of Washington  
Seattle, WA 98195, USA*

ERIC.GARCIA@U.WASHINGTON.EDU  
SERGEYF@EE.WASHINGTON.EDU  
GUPTA@EE.WASHINGTON.EDU

**Santosh Srivastava**

*Fred Hutchinson Cancer Research Center  
Seattle, WA 98109, USA*

SSRIVAST@FHCRC.ORG

**Editor:**

## Abstract

Local classifiers are sometimes called lazy learners because they do not process the training data until presented with a test sample. However, such methods are generally not completely lazy, because the neighborhood size  $k$  (or other locality parameter) is usually chosen by cross-validation on the training set, which can require significant preprocessing and risks overfitting. We propose a simple Bayesian alternative to cross-validation of the neighborhood size that requires no pre-processing. We motivate this Bayesian neighborhoods technique by showing it extends the standard Bayes decision rule to minimize expected misclassification costs by treating the neighborhood size as a random variable. Additionally, the proposed estimated posterior minimizes the expected Bregman loss. We analyze the effect of this approach on several standard and state-of-the-art local classifiers, including adaptive metric kNN, a local support vector machine, and a local Bayesian quadratic discriminant analysis. The empirical effectiveness of this technique vs. cross-validation is validated with experiments on several benchmark data sets. The results show that classification performance is generally the same as cross-validation despite using no training. Further, when the distribution of test data and training data are not identical, Bayesian neighborhoods can offer drastically better results, because there is no risk of overfitting the neighborhood size to the mismatched training data distribution.

**Keywords:** lazy learning, Bayesian estimation, cross-validation, local learning, Bayesian QDA

## 1. Introduction

It is inefficient or infeasible to train a statistical learning algorithm on all the training data for some applications. One solution is to use a local learning method, such as kNN, that only requires training a classifier on a subset of the training data that is deemed relevant to a particular test sample. Local classifiers can achieve competitive error rates on practical problems (Hastie et al., 2001; Lam et al., 2002; Gupta et al., 2006a; Zhang et al., 2006), can trivially adapt to evolving training data, and are suitable for problems where one cannot assume that training and test samples are drawn from the same distribution. Traditionally, local classifiers have been weighted nearest-neighbor voting methods, though recently

a number of model-based local classifiers have been shown to be promising. Because local methods defer processing of the training data, these methods are sometimes called lazy (Aha, 1997). In practice, however, they are not completely lazy because most local algorithms determine a neighborhood size parameter  $k$  that is appropriate for a particular data set by cross-validation. To enable just-in-time learning, we propose a Bayesian approach to the neighborhood selection problem that produces completely lazy classifiers by averaging local learning discriminants for different neighborhood sizes. This simple strategy requires no preprocessing, but we will show it produces similar errors to cross-validation for iid data sets, and can lead to large error reductions in some cases where the training and test are not iid.

Cross-validation is by far the most common way to choose neighborhood size for local classifiers, but it suffers from many drawbacks, including: a) prohibitively long training time on larger data sets, b) the unjustified assumption that there is a single optimal value of  $k$  for the entire feature space, and c) a lack of a rigorous method for choosing the set of cross-validated neighborhood sizes. The theory behind cross-validation assumes that the training and test data sets are drawn independently from identical distributions (iid). In practice, the iid assumption is often unrealistic. For some applications, such as speech processing, learning from the web, or trying to predict behavior of an evolving pathogen, the iid assumption fails because the distribution of the data evolves over time. Similarly, there may be significant biases in how the training data is collected versus the distribution of the test data; for instance the volunteers who choose to participate in a study may be statistically different from the larger population for which the study hopes to make predictions. By not cross-validating the neighborhood size or other learning parameters, one avoids overfitting to the training distribution in such cases. The proposed completely lazy learning only makes locally iid assumptions.

There have been previous efforts to define neighborhoods for local learning methods that do not require cross-validation. For example, a small set of experiments showed that using the relative-neighborhood-graph neighbors of the test point yields generally lower error than  $k$  nearest neighbors for classification for a small set of experiments (Sánchez et al., 1997). For local linear interpolation and local linear regression, good results have been achieved with the test point’s natural neighbors (Sibson, 1981; Gupta et al., 2008) and the enclosing kNN neighbors (Gupta et al., 2008), which attempt to enclose a test point in the convex hull of its neighbors. Although such spatially adaptive neighborhoods have produced significant error reductions for low-dimensional learning problems, they tend to be computationally challenging or ill-suited for general classification problems where the number of samples may be large, but not dense in the sample space.

Another set of methods are kNN committee classifiers. Classification by committee is a learning approach that attempts to alleviate the weaknesses of individual classifiers by combining them (Hastie et al., 2001, ch. 8). The proposed Bayesian neighborhoods is a committee method in that it takes an average of the predicted probabilities (or discriminants) corresponding to each neighborhood size. In the absence of prior knowledge, we give each log-sampled neighborhood size equal weight. Other researchers have also proposed taking weighted averages over neighborhoods of kNN classifier discriminants, but with the key difference that the proposed methods are not completely lazy. Most recently, Ghosh et al. (2005) take a weighted average of the predicted probabilities of kNN classifiers, but

Table 1: Key Notation

$d$	number of dimensions	$E_Z$	expectation w.r.t. $Z$
$X_j \in \mathbb{R}^d$	random training sample ( $d \times 1$ )	$p$	distribution
$x_j$	realization of $X_j$	$\hat{p}$	estimated distribution
$Y_j \in \{1, 2, \dots, G\}$	random class label for $X_j$	$P$	probability mass function
$G$	number of class labels	$N(x)$	random Gaussian evaluated at $x$
$X \in \mathbb{R}^d$	random test sample ( $d \times 1$ )	$B$	seed matrix for prior $p(N)$
$n$	number of training samples	$C(g, h)$	cost of classifying as class $g$
$k$	number of neighbors		if truth is class $h$

weights each neighborhood size by a function of the corresponding cross-validation error for that neighborhood size. Similarly, in work by Paik and Yang (2004) and Holmes and Adams (2002), weighted averages over neighborhoods are calculated with weights that depend on estimates of how useful each neighborhood size is for classifying, where the estimate is based on the training data. Other prior work in ensemble methods for kNN have used component classifiers that are not based on different neighborhoods, and are also not completely lazy. Bay (1998) took a committee of kNN classifiers where each acted on a random subsets of features. Hall and Samworth (2005) have analyzed bagged nearest neighbor classifiers, though at least one empirical study suggests bagging kNN has little effect (Speed, 2003). Other researchers built component kNN classifiers on different condensed subsets of the training samples (Skalak, 1996; Alpaydin, 1997). Masip and Vitrià (2006) consider kNN classifiers on different linear combinations of features and use boosting to find an optimal feature set.

The main contribution of this paper is demonstrating that Bayesian neighborhoods can work as well as using a cross-validated neighborhood size, without the pre-processing. A secondary contribution is developing a local version of the Bayesian quadratic discriminant analysis classifier (Srivastava et al., 2007; Frigyük et al., 2006) to form a Gaussian classifier that approximately minimizes expected misclassification error with respect to uncertainty in both the neighborhood and the fitted Gaussians; we term this *local Bayesian discriminant analysis* (local BDA). First, in Section 2, we introduce Bayesian neighborhoods and show that it minimizes expected misclassification costs for generative classifiers. We discuss the relationship of Bayesian neighborhoods to classifier fusion and Bayesian model averaging. In Section 3 we describe how Bayesian neighborhoods affects different classes of local classifiers. Experiments comparing Bayesian neighborhoods to cross-validation on benchmark data sets are detailed in Section 4 for seven local classifiers: kNN, a local linear SVM (SVM-KNN) (Zhang et al., 2006), nearest-hyperplane kNN (HKNN) (Vincent and Bengio, 2001), discriminant adaptive nearest-neighbor (DANN) (Hastie and Tibshirani, 1996), local ridge regression classification (ridge), local nearest means (local NM) (Mitani and Hamamoto, 2000), local BDA. The paper concludes in Section 5 with a summary of results and open questions. Key notation is summarized in Table 1.

## 2. Neighborhood Size as a Random Variable

Given the true joint probability distribution of features and labels  $p_{X,Y}$  and a prior distribution over class labels  $p_Y$ , a test sample  $x \in \mathbb{R}^d$  can be assigned the label  $y^* \in \{1, 2, \dots, G\}$  that minimizes the expected misclassification cost (with respect to  $p_{X,Y}$ ). That is,  $y^*$  solves,

$$\operatorname{argmin}_{g \in \{1, 2, \dots, G\}} \sum_{h=1}^G C(g, h) P_{Y|X}(h|x),$$

where  $C(g, h)$  is the cost of labeling a sample as class  $g$  when the truth is class  $h$ .

Estimating the class-conditional distribution works best when the training samples and test samples are drawn iid from the same distribution. It is less assumptive to suppose that only a subset of the  $n$  training samples are drawn iid from the same class-conditional distribution as the test sample  $x$ , and we use the  $k$  nearest training samples as an intuitive subset. Let  $\hat{P}_{Y|X,K}(h|x, k)$  denote the estimated local posterior distribution estimated from the  $k$  nearest neighbors of  $x$ . Then we propose to treat the neighborhood size as a random variable  $K$ , and minimize the expected misclassification costs with respect to  $K$ , so that the estimated class solves,

$$\begin{aligned} & \operatorname{argmin}_{g \in \{1, 2, \dots, G\}} E_K \left[ \sum_{h=1}^G C(g, h) \hat{P}_{Y|X,K}(h|x, K) \right] \\ \equiv & \operatorname{argmin}_{g \in \{1, 2, \dots, G\}} \sum_{h=1}^G C(g, h) E_K \left[ \hat{P}_{Y|X,K}(h|x, K) \right]. \end{aligned} \quad (1)$$

We refer to this approach as using *Bayesian neighborhoods*.

Treating the neighborhood size  $K$  as a random variable requires a prior distribution  $P_K$  over possible neighborhood sizes. We note that this is essentially the same issue one encounters with cross-validation, in that it requires the specification of a set of possible parameter choices. In the absence of prior knowledge, we propose a sampled log-uniform prior over the set  $\mathcal{K} = \{2^1, 2^2, \dots, 2^\gamma\}$ , such that  $P_K(k) = 1/|\mathcal{K}|$  for  $k \in \mathcal{K}$  and  $\gamma = \min(\lfloor \log_2(d \log_2 n) \rfloor, \lfloor \log_2 n \rfloor)$ . For example, for the Vowel data set with  $n = 528$  training samples and  $d = 10$  features,  $k \in \{2, 4, 8, 16, 32, 64\}$ . This choice of  $P_K$  was motivated in part by the computational simplicity of a sampled set of  $k$ , and in part by the curse of dimensionality, which suggests that higher-dimensional feature spaces require more neighbors to span the feature space. For some data sets, however,  $2^{\lfloor \log_2(d \log_2 n) \rfloor}$  is larger than the number of available training samples, in which case the largest neighborhood size is taken to be  $2^{\lfloor \log_2 n \rfloor}$ . Lastly, we make  $P_K$  depend on  $n$  for the sake of consistency (see Section 3.1). Some model-based local classifiers use  $k$  neighbors from each class. In that case, let  $\bar{n}$  be the average number of neighbors per class, and let the prior probability on the  $g$ th class's neighborhood size  $k_g$  be  $P_K(k_g) = 1/|\mathcal{K}|$  for  $k_g \in \mathcal{K}$  and  $\gamma = \min(\lfloor \log_2(d \log_2 \bar{n}) \rfloor, \lfloor \log_2 \bar{n} \rfloor, n_g)$ .

Many classifiers produce estimates  $\hat{p}_{X|Y}(x, h)$  of the likelihood for each class  $h$ . Given the standard decision rule  $\hat{y} = \operatorname{argmax}_h \hat{P}_{Y|X}(h|x)$  with estimated prior  $\hat{P}_Y(h)$ , the standard decision rule can be written in terms of the likelihood:  $\hat{y} = \operatorname{argmax}_h \hat{p}_{X|Y}(x|h) \hat{P}_Y(h)$  because  $p_X(x)$  does not change the decision. In contrast in the decision rule given by (1),  $p_X(x)$

cannot be ignored. By Bayes' rule, (1) can be written,

$$\hat{y} = \operatorname{argmin}_{g \in \{1, 2, \dots, G\}} \sum_{h=1}^G C(g, h) E_K \left[ \frac{\hat{p}_{X|Y, K}(x|h, K) \hat{P}_{Y|K}(h|K)}{\hat{p}_{X|K}(x|K)} \right]. \quad (2)$$

For classifiers that produce an estimate of the likelihood, we estimate each  $p_{X|K}(x|k)$  after estimating the class likelihoods to ensure its role as a normalizer such that (2) becomes

$$\operatorname{argmin}_{g \in \{1, 2, \dots, G\}} \sum_{h=1}^G C(g, h) E_K \left[ \frac{\hat{p}_{X|Y, K}(x|h, K) \hat{P}_{Y|K}(h|K)}{\sum_{j=1}^G \hat{p}_{Y, X|K}(j, x|K)} \right]. \quad (3)$$

Throughout this paper, we assume uniform class priors such that  $\hat{P}_{Y|K}(h) = \hat{P}_Y(h) = 1/G$ . Then (3) simplifies to,

$$\operatorname{argmin}_{g \in \{1, 2, \dots, G\}} \sum_{h=1}^G C(g, h) E_K \left[ \frac{\hat{p}_{X|Y, K}(x|h, K)}{\sum_{j=1}^G \hat{p}_{X|Y, K}(x|j, K)} \right]. \quad (4)$$

### 2.1 Minimum Expected Bregman-Loss Motivation

The decision rule in (1) can also be motivated by noting that its estimated class posterior distribution is the distribution that minimizes any expected Bregman error with respect to uncertainty in the neighborhood size. That is, the pmf over classes  $E_K \left[ \hat{P}_{Y|X, K}(h|x, K) \right]$  used in (1) solves

$$\operatorname{argmin}_{r \in [0, 1]^G, \sum_{h=1}^G r(h)=1} E_K \left[ d \left( r, \hat{P}_{Y|X, K}(h|x, K) \right) \right],$$

where  $d$  is any Bregman divergence (such as squared  $\ell_2$  distance). This property follows from Theorem 1 of Banerjee et al. (2005). (For more on minimizing expected Bregman divergences, see Ghosh et al. (2005); Gupta et al. (2006b), or Frigiyk et al. (2006)).

## 3. Local Classifiers and Bayesian Neighborhoods

In the next subsections we consider the effect of using Bayesian neighborhoods with a number of popular and state-of-the-art local classifiers, and propose a local Bayesian quadratic discriminant analysis (QDA) that is also motivated by minimizing expected Bregman loss.

### 3.1 Bayesian Neighborhoods and kNN

The posterior distribution estimate of the kNN classifier is given by

$$\hat{P}_{Y|X, K}(h|x, k) = \frac{1}{k} \sum_{j=1}^k I_{(y_j=h)},$$

where  $I$  is the indicator function, and  $y_j$  is the label of the  $j$ th neighbor of  $x$ .

The  $h$ th posterior distribution for kNN with Bayesian neighborhoods is given by,

$$\begin{aligned}
E_K[\hat{P}_{Y|X,K}(h|x,K)] &= \sum_{k=1}^n P_K(k) \left( \frac{1}{k} \sum_{j=1}^k I_{(y_j=h)} \right) \\
&= \sum_{k=1}^n P_K(k) \left( \sum_{j=1}^n \left( \frac{1}{k} I_{(j \leq k)} \right) I_{(y_j=h)} \right) \\
&= \sum_{j=1}^n \left( \sum_{k=1}^n \frac{P_K(k)}{k} I_{(j \leq k)} \right) I_{(y_j=h)}. \tag{5}
\end{aligned}$$

From (5), one sees that kNN with Bayesian neighborhoods is a weighted nearest neighbor classifier that applies weights that depend on the distance to the test point. For example, if we take a uniform prior  $P_K(k) = \frac{1}{M}$  for  $k \in \{1, 2, \dots, M\}$ , then the weight for the  $(M+1)$ th nearest neighbor is  $w_{M+1} = 0$ ,  $w_M = \frac{1}{M^2}$ ,  $w_{M-1} = \frac{1}{M^2} + \frac{1}{M(M-1)}$ , and so on. This effectively creates a weighting function (kernel) that adapts to the spread of the data and decreases with distance (as long as  $P_K$  is decreasing faster than linearly in  $k$ ). This kernel effect also applies to standard weighted kNN classifiers that employ a fixed weighting kernel.

The Bayesian neighborhoods kNN classifier is consistent if the prior  $P_K$  produces a weight vector  $w$  on the training samples that satisfies Stone's conditions (Stone, 1977). The conditions can be met by a prior  $P_K$  that is non-increasing in  $k$ , and that has support on  $\{1, \dots, M(n)\}$  such that  $M(n) \rightarrow \infty$  as  $n \rightarrow \infty$  but slowly such that  $M(n)/n \rightarrow 0$ . Lastly,  $P_K$  must have high enough entropy that  $\max_j(w_j) \rightarrow 0$  as  $M(n) \rightarrow \infty$ . The default sampled log-uniform prior described in Section 2 meets Stone's conditions, and thus forms a consistent classifier when used with kNN.

### 3.2 Locally Linear Classifiers

Locally linear classifiers fit a hyperplane to the neighboring training samples, then classify the test point based on the resulting linear discriminant(s). Two state-of-the-art locally linear classifiers are SVM-KNN (Zhang et al., 2006) and local ridge regression. The SVM-KNN applies a linear kernel SVM to the test samples'  $k$  nearest neighbors, and has a regularization parameter  $C$ ; both  $k$  and  $C$  are recommended to be chosen by cross-validation (Zhang et al., 2006). Local linear regression has been used for classification since at least 1977 (Stone, 1977). To ensure numerical stability, we use least-squares fits with a ridge regularization penalty on the hyperplane slope coefficients with fixed regularization parameter  $\kappa = 1$  (Hoerl and Kennard, 1970; Hastie et al., 2001).

Let  $f_{k,g}(x)$  be a local discriminant for class  $g$  learned from the  $k$  nearest neighbors of  $x$ . Then for the Bayesian neighborhood approach we classify based on the expected discriminants  $\{E_K[f_{K,g}(x)]\}$  for  $g = 1, \dots, G$ . Like kNN, using a locally linear classifier with a Bayesian neighborhood results in the nearer-samples having a greater contribution. However, because the contributions can be positive, zero, or negative, the precise effect of using Bayesian neighborhoods is less predictable for these methods than for kNN.

### 3.3 Locally Gaussian Classifiers

In this section, we discuss how two recent local classifiers can be interpreted as locally modeling the class posterior as Gaussian, and propose a local Bayesian QDA classifier. We consider how Bayesian neighborhoods affects such classifiers. Each of these locally Gaussian classifiers uses the  $k$  nearest neighbors from each class for a total of  $k \times G$  neighbors.

#### 3.3.1 LOCAL NEAREST MEANS (LOCAL NM)

The local nearest means classifier calculates the mean of each class in a neighborhood of the test sample, and classifies the test sample depending on which local class mean is nearest (Mitani and Hamamoto, 2000, 2006). Compared to the standard nearest-means classifier, local nearest-means drastically reduces the potentially large bias inherent in modeling each class as being characterized by its mean feature vector. Compared to standard kNN, the classification variance due to outlying samples is reduced. Local nearest-means can be equivalently expressed as a generative classifier where each class is locally modeled as being drawn from a Gaussian distribution with identity covariance. We use the resulting likelihood estimates in (3).

Local nearest means can still have a significant model bias problem for large  $k$  relative to the total number of samples  $n$  and the dimension of the feature space  $d$ . We have found that cross-validated neighborhood sizes tend to be very small with local nearest means, effectively keeping the local means close to the test point. Thus, when used with Bayesian neighborhoods, the prior  $P_K$  should have more support for small  $k$  than local classifiers with less model bias.

#### 3.3.2 K-LOCAL HYPERPLANE DISTANCE NEAREST NEIGHBOR ALGORITHM

Vincent and Bengio (2001) proposed a local classifier they termed  $k$ -local hyperplane distance nearest neighbor algorithm (HKNN), which we show is equivalent to a local Mahalanobis nearest-means classifier. Vincent and Bengio (2001) motivated HKNN as classifying a test point  $x$  by drawing  $k$  nearest neighbors from each class, projecting  $x$  to the linear span of each set of  $k$  points (the affine hull) and choosing the class with minimal projection distance. Given the standard assumption that the training samples are in general position, such a classifier would be indeterminate if  $k_h$  exceeds the dimensionality of the data for each  $h$ , because each class's neighbors' affine hull would span the entire feature space. They mitigate this problem by regularizing the projection weights, such that the HKNN classification rule is to classify  $x$  as the class  $h$  that has minimum discriminant  $d_k(x, \mu_h)$ , where

$$d_k^2(x, \mu_h) = \min_{\alpha \in \mathbb{R}^k} \|(x - \mu_h) - X_h \alpha\|_2^2 + \lambda \|\alpha\|_2^2, \tag{6}$$

where  $X_h$  is a  $d \times k$  matrix of the  $k$  nearest training samples of class  $h$  demeaned by the class mean  $\mu_h$ , and the regularization parameter  $\lambda$  is trained by cross-validation.

We show that HKNN is equivalent to a local Mahalanobis nearest-means classifier:

**Lemma 1** *The HKNN class  $h$  discriminant (6) can be equivalently expressed as,*

$$d_k(x, \mu_h) = (x - \mu_h)^T (I + \lambda^{-1} X_h X_h^T)^{-1} (x - \mu_h). \tag{7}$$

The proof is given in the appendix. From (7), one sees that the HKNN discriminant is the log-likelihood of  $x$  given a Gaussian distribution with regularized covariance  $I + \lambda^{-1} X_h X_h^T$ , just as it appears in regularized QDA (Friedman, 1989). However, the HKNN discriminant does not include the Gaussian’s normalization term, which would add an additional factor of  $\ln |I + \lambda^{-1} X_h X_h^T|$ . In regularized QDA, this normalization term penalizes classes that are less predictable in terms of their preferred feature vectors.

### 3.3.3 LOCAL BAYESIAN QDA (LOCAL BDA)

Next, we propose another locally Gaussian classifier, where the regularization is enacted by a Bayesian estimate of the generating Gaussian. We apply Bayesian QDA locally to the  $k$  nearest neighbors of each class, and term this *local BDA*. For local BDA the estimated class-conditional likelihood is  $\hat{p}_{X|Y,K}(x|h,k) = E_{N_{h,k}}[N_{h,k}(x)]$ , where the  $N_{h,k}$  are independent random Gaussians drawn from  $p_{N_{h,k}|\mathcal{T}_x(h,k)}$ , and  $\mathcal{T}_x(h,k)$  are the  $k$  training sample pairs from class  $h$  nearest to the test sample  $x$ . As in HKNN and nearest means, applying QDA locally reduces its model bias. But here we have the added advantage of estimating the Gaussians with a data-dependent Bayesian approach which reduces the estimation variance as well.

Bayesian QDA classifiers were first proposed in the 1960’s (Geisser, 1964; Keehn, 1965), but were not found to perform well; in particular, they were found to have too much bias Ripley (2001). Recently, Srivastava et al. (2007) demonstrated that using a data-dependent prior and the Fisher information measure leads to a Bayesian QDA classifier that performs as well or better than other state-of-the-art approaches to QDA, including regularized QDA (Friedman, 1989) and eigenvalue-decomposition discriminant analysis (Bensmail and Celeux, 1996). It has been shown that the mean Gaussian with respect to the posterior minimizes the expected functional Bregman risk between the estimated pdf and the possible Gaussians that could have generated the training samples (Srivastava et al., 2007; Frigyyik et al., 2006).

The data-dependent Bayesian QDA classifier of Srivastava et al. (2007) requires cross-validating hyperparameters of the inverted Wishart prior: a scale parameter  $q$  and a seed matrix  $B_h$  for the  $h$ th class for  $h = 1, \dots, G$ . For our local BDA classifier, we fix the scale-parameter of the inverted Wishart distribution prior to be  $q = d + 3$ , which makes the prior the standard inverted gamma distribution if  $d = 1$  (Srivastava and Gupta, 2006). Srivastava et al. (2007) fix  $B_h$  so that  $B_h/q$  is a coarse estimate of the empirical covariance. Then because the prior has maximum probability for Gaussians with covariance matrix equal to  $B_h/q$ , the prior produces a data-dependent bias that regularizes the likelihood towards the coarse estimate of the empirical covariance. Following this tactic, we fix the prior’s seed matrix to be

$$B_h = 0.95q \text{diag} \left( \hat{\Sigma}_{ML,h} \right) + 0.05I,$$

where  $\text{diag} \left( \hat{\Sigma}_{ML,h} \right)$  is the diagonal of the maximum likelihood estimate of the local class  $h$  covariance matrix. For very small  $k$ , the diagonal may be ill-posed, and so we regularize it with the identity matrix; the choice of 5% regularization was chosen without experimentation to be small enough that the emphasis is on the data-dependent diagonal, but large enough to ensure that the inverse of  $B_h$  is not ill-posed. We do not expect that changing the amount of  $I$  in  $B_h$  to 1% or 10% would have much effect.

Then following from Theorem 1 of Srivastava et al. (2007), the proposed local Bayesian quadratic discriminant analysis (local BDA) classifier estimates the  $h$ th local class-conditional likelihood to be

$$\begin{aligned} \hat{p}_{X|Y,K}(x|h,k) &= E_{N_{h,k}}[N_{h,k}(x)] \\ &= \frac{\left(\frac{2k}{k+1}\right)^{\frac{d}{2}} \Gamma\left(\frac{k+d+4}{2}\right) \left| \sum_{i=1}^n (x_i - \bar{x}_h)(x_i - \bar{x}_h)^T I_{(y_i=h)} + B_h \right|^{\frac{k+d+3}{2}}}{\Gamma\left(\frac{k+d}{2}\right) \left| \sum_{i=1}^n (x_i - \bar{x}_h)(x_i - \bar{x}_h)^T I_{(y_i=h)} + \frac{k(x-\bar{x}_h)(x-\bar{x}_h)^T}{k+1} + B_h \right|^{\frac{k+d+4}{2}}}, \end{aligned} \tag{8}$$

where  $\Gamma(\cdot)$  is the standard gamma function, and  $\bar{x}_h$  is the average of the  $k$  nearest training feature vectors from class  $h$ .

Local BDA with Bayesian neighborhoods produces the decision rule:

$$\operatorname{argmin}_{g \in \{1,2,\dots,G\}} \sum_{h=1}^G C(g,h) E_K \left[ \frac{E_{N_{h,K}}[N_{h,K}(x)]}{\sum_{j=1}^G E_{N_{j,K}}[N_{j,K}(x)]} \right].$$

The estimated class-conditional distributions  $E_{N_{h,k}}[N_{h,k}(x)]$  given in (8) are not Gaussians, but in fact the local BDA decision boundary is locally quadratic:

**Lemma 2** *For fixed  $k$ , the local BDA decision boundary is piecewise quadratic.*

The proof is given in the appendix.

### 3.3.4 DISCRIMINANT ADAPTIVE NEAREST NEIGHBORS

The discriminant adaptive nearest neighbors (DANN) method (Hastie and Tibshirani, 1996) also uses a locally Gaussian assumption. DANN models each local class likelihood as a Gaussian with the same covariance matrix. Then, DANN uses the Gaussian assumption to imply a local distance metric, which is then used to find the  $k$  nearest neighbors to the test point  $x$ . It is dissimilar to the previous three algorithms in that the final classifier used is a weighted kNN classifier. The Gaussian assumption is used merely to adapt the metric, and not to classify  $x$ .

## 4. Experiments and Results

In the introduction we discussed some potential shortcomings of cross-validation. We proposed instead averaging the class posteriors or discriminants with respect to the neighborhood choice. In the following sections we provide experimental results to show that the proposed Bayesian neighborhood works as well or better in practice as cross-validation, and does so without the required training.

First we motivate using Bayesian neighborhoods rather than cross-validation on a case study of the Vowel data set. Then we compare kNN and the proposed Bayesian neighborhood on seven classifiers and seven benchmark data sets. The experimental details are given in Sec. 4.2, followed by results using standard train/test partitions in Sec. 4.3 and random train/test partitions in Sec. 4.4.

## 4.1 A Case Study of the Vowel Data Set

To demonstrate the sub-optimality of the choosing a fixed  $k$ , as done in cross-validation, Fig. 1 shows how 45 randomly chosen test samples from Vowel are classified using HKNN for neighborhood sizes  $k = 2, \dots, 32$ . In this figure, white indicates a correct classification and black indicates an incorrect classification. One sees that for many test points the classification is quite sensitive to the choice of  $k$ . There is no range of values for  $k$  which produce correct classifications consistently across the samples (i.e. no broad white columns). The Vowel data set is of practical interest because the training and test sets consist of vocalizations generated by two separate sets of individuals, thus the training and test samples are not identically distributed. We found that Fig. 1 is representative, regardless of the classifier used.

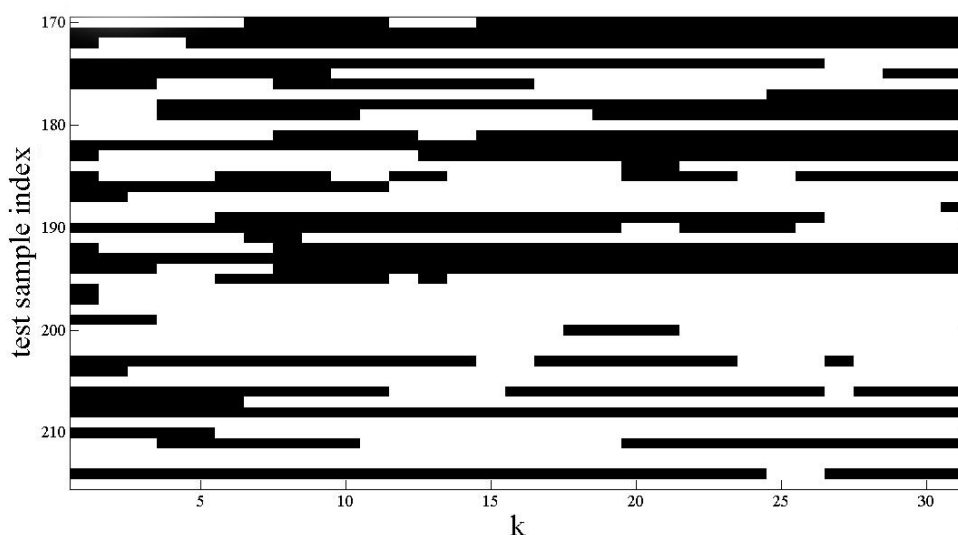


Figure 1: Rows are test samples; columns are neighborhood sizes. White indicates a correct classification and black indicates a misclassification using HKNN

To illustrate why Bayesian neighborhoods can work better than a fixed choice of  $k$ , we plot in Figures 2 and 3 the eleven class discriminants for four local classifiers, and for two test samples from the Vowel data set. Depending on the classifier,  $k$  either denotes the total number of neighbors (kNN, SVM-KNN), or the number of neighbors from each class (local BDA and HKNN). In the plots, the thick line corresponds to the discriminant value for the correct class, and thin lines correspond to discriminant values for each of the other classes. One can see that the classifiers each have a different sensitivity to the choice of neighborhood, with the most sensitive classifier being SVM-KNN. In general, the discriminant values appear to be a “noisy” function of  $k$ , which motivates averaging over the discriminant values as done with Bayesian neighborhoods. Note that although in this example the discriminants are being plotted for a dense range of  $k$  values, in practice, the discriminants are computed only for a sparse sampling of  $k$ . For our experiments there were never more than 12 values of  $k$  for any data set.

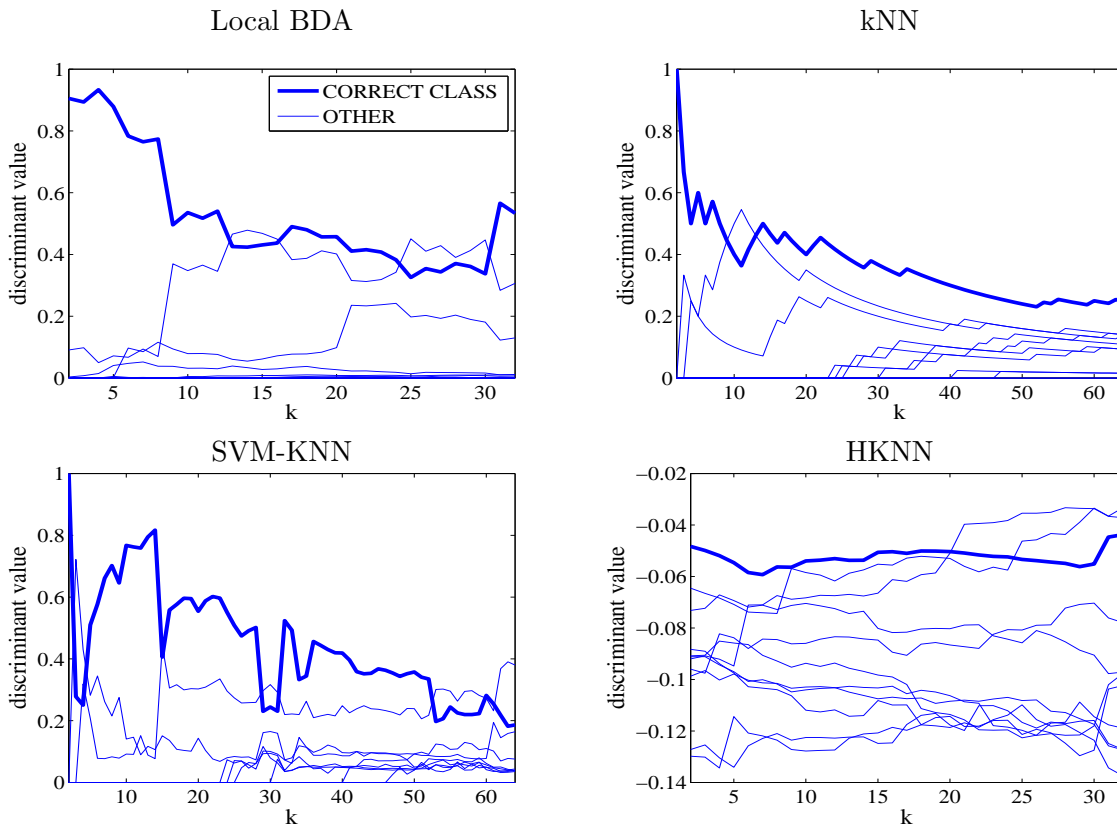


Figure 2: Discriminant vs neighborhood size for example test point 284 from Vowel.

To establish that Bayesian neighborhoods can perform better than using a fixed value of  $k$ , we performed a small experiment allowing a cross-validated classifier to train on the test data. It is important to note that here each cross-validated classifier chooses the fixed  $k$  that achieves the lowest possible error on the test set (not possible in practice). The results are shown in Table 2. We used the sampled log-uniform prior proposed in section 2 for the Bayesian neighborhood to generate  $k \in \{2,4,8,16,32\}$  neighbors from each class for HKNN, local NM, and local BDA, and  $k \in \{2,4,8,16,32,64\}$  total neighbors for kNN, DANN, ridge, and SVM-KNN. The cross-validated  $k$  were chosen from  $k \in \{2,3, \dots, 32\}$  neighbors from each class for HKNN, NM, and local BDA, and  $k \in \{2,3, \dots, 64\}$  total neighbors for kNN, DANN, ridge, and SVM-KNN. Table 2 shows that the Bayesian neighborhood for Vowel actually performs better than the best possible fixed  $k$  for local BDA, local NM, and kNN, and only slightly worse for HKNN and DANN. Thus even when cross-validation is allowed to “cheat” by cross-validating on the test data, Bayesian neighborhoods can still be competitive.

#### 4.2 Training/Test Experimental Details

We performed experiments comparing cross-validation and the proposed Bayesian neighborhoods on seven standard benchmark data sets; information about the data sets is given

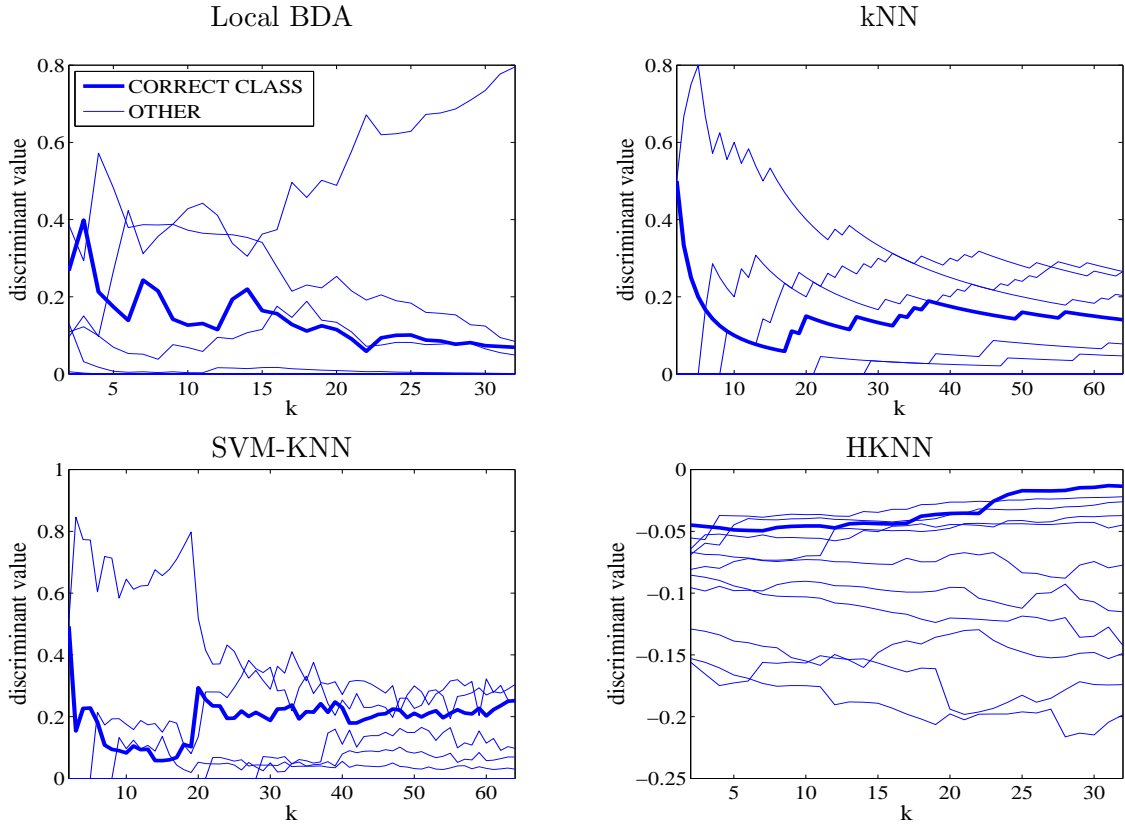


Figure 3: Discriminant vs neighborhood size for example test point 55 from Vowel.

Table 2: % Test Error for Vowel

	Local BDA	Local NM	HKNN	KNN	DANN	Pinv	SVM-KNN
<i>BN</i>	<b>34.0</b>	<b>38.5</b>	40.3	<b>48.0</b>	39.8	42.6	44.1
Use best <i>k</i> for test	35.9	42.0	<b>39.0</b>	48.1	<b>38.5</b>	<b>38.5</b>	<b>37.7</b>

in Table 3. Each data set was used twice, once with standard train/test partitions for reproducibility and once with random train/test partitions for statistical significance. These data sets were chosen because all features are real-valued, there are no missing data, and there are standard training/test partitions. For each standard/random partition, the training data were normalized to have a mean of 0 and standard deviation of 1, and the test data were then normalized by the same values. Features constant over all training samples were removed.<sup>1</sup>

For the experiments on random partitions detailed in Sec. 4.4, the test and training data were combined, and then randomly divided ten times into 50-50 splits to form ten iid

1. MATLAB code is available at <http://idl.ee.washington.edu>.

test and training sets. Due to the large size of the USPS and Isolet data sets, they were not included in the randomized-partition experiments.

All classifier parameters were set by 10-fold cross-validation and in the case of cross-validation error ties, the smallest tied-parameter was chosen. The HKNN algorithm requires a regularization parameter  $\lambda$  as well as a neighborhood size  $k$ . For the cross-validation runs, the HKNN  $\lambda$  was cross-validated as  $\lambda \in \{1, 5, 10, 20, 30, 50\}$  as recommended by the HKNN authors. For the completely lazy learning approach using the Bayesian neighborhood, we fixed the HKNN regularization parameter to be  $\lambda = 1$  as the minimal amount of regularization recommended by the HKNN authors.

The SVM-KNN algorithm also requires a regularization parameter  $C$  in addition to the neighborhood size  $k$ . For the cross-validation runs, the SVM-KNN  $C$  was cross-validated as  $C \in \{.001, .01, .1, 1, 10, 100, 1000\}$ , based on standard practice. For the completely lazy learning approach using the Bayesian neighborhood, we fixed the SVM-KNN regularization parameter to be the default value of  $C = 1$ .<sup>2</sup> For multi-class data sets we implemented  $\binom{n}{2}$  one-against-one classifiers.

In all cases, the support of the prior  $P_K$  and the set of choices of  $k$  for cross-validation were the same. For kNN, DANN, SVM-KNN, and ridge:  $k \in \{2^1, \dots, 2^\gamma\}$  with  $\gamma = \min(\lfloor \log_2(d \log_2 n) \rfloor, \lfloor \log_2 n \rfloor)$ , where  $n$  denotes the number of training samples available in the cross-validation (for 10-fold cross-validation,  $n$  is 90% of the total training samples). For HKNN, local NM and local BDA, and for each class  $g$ , we set

$$\gamma(g) = \min(\lfloor \log_2(d \log_2 \bar{n}) \rfloor, \lfloor \log_2 \bar{n} \rfloor, n_g),$$

where  $\bar{n}$  denotes the average number of training samples for each class available in the cross-validation and  $n_g$  is the number of training samples available in the cross-validation for that class. The prior  $P_K$  was taken as in Section 2 to be uniform over the sampled log-uniform set of possible  $k$ 's.

Throughout the experiments, we assume the class prior probabilities  $P_Y$  are uniform over the set of possible classes.

Table 3: Information About the Benchmark data sets

	# of Classes	# of Features	# of Total Samples	# of Standard Train/Test
Vowel	11	10	990	528/462
Image Seg.	7	19	2,310	210/2,100
Opt. Digits	10	64	5,620	3,823/1,797
Letter Rec.	26	16	20,000	16,000/4,000
Pen Digits	10	16	10,992	7,494/3,498
USPS	10	256	9,298	7,291/2,007
Isolet	26	617	7,797	6,238/1,559

2. SVM was implemented with LIBSVM (Chang and Lin, 2001).

### 4.3 Standard-Partition Benchmark Data Set Results

Table 4 shows the classification errors using the standard partitions of the benchmark data sets. For each data set, the best performance is marked in bold, and for six of the seven data sets is achieved with the Bayesian neighborhood method. We note that some of the algorithms experience a drastic reduction in error with Bayesian neighborhoods, for example the 34% error achieved with local BDA is, to the best of our knowledge, the lowest recorded. Ridge error increases slightly with Bayesian neighborhoods on Vowel, ImageSeg and USPS, but decreases by 32% for OptDigits, 33% for LetterRec, 20% for PenDigits, and 32% for Isolet.

Table 4: % Test Error Given Standard Training/Test Partitions

	Vowel		ImageSeg		OptDigits		LetterRec		PenDigits		USPS		Isolet	
	CV	BN	CV	BN	CV	BN	CV	BN	CV	BN	CV	BN	CV	BN
kNN	52.4	48.1	12.9	12.1	3.5	3.5	5.3	5.2	2.7	3.1	5.7	7.6	8.7	6.9
DANN	39.0	39.8	7.5	7.5	4.0	4.3	5.1	4.6	2.8	3.1	8.2	8.1	8.6	8.4
Local NM	43.3	38.5	10.0	12.1	3.3	3.3	4.0	5.9	2.3	2.9	<b>4.4</b>	5.4	4.3	3.8
Local BDA	44.6	<b>34.0</b>	6.9	<b>6.3</b>	2.2	2.0	2.9	3.1	2.1	2.2	5.4	5.4	3.4	3.3
HKNN	43.9	40.3	9.4	9.1	2.5	2.9	4.2	4.4	2.3	2.3	4.6	5.9	3.7	3.7
ridge	40.9	42.6	8.2	8.4	2.5	<b>1.7</b>	4.2	<b>2.8</b>	2.1	<b>1.7</b>	5.4	6.2	7.4	5.0
SVM-KNN	49.4	44.2	7.9	9.7	2.3	2.2	3.9	3.4	2.1	2.1	4.7	4.6	3.7	<b>3.1</b>

**Bold:** Best result on the data set.

Over the seven data sets, local BDA performs consistently well and has the lowest mean error, with ridge, HKNN and SVM-KNN the next top performers. Because Bayesian neighborhoods is designed to require no training, we used fixed regularization parameters for HKNN ( $\lambda = 1$ ) and SVM-KNN ( $C = 1$ ) for the results in the column marked BN. Surprisingly, SVM-KNN only does better with its two parameters cross-validated on one of the seven data sets (ImageSeg), and HKNN does better with its parameters cross-validated on three of the seven data sets. Table 5 shows the cross-validation choices.

Table 5: Cross-validated Parameter Choices Given Standard Training/Test Partitions

	Vowel	ImageSeg	OptDigits	LetterRec	PenDigits	USPS	Isolet
kNN $k$	2	4	4	4	2	4	16
DANN $k$	2	2	4	4	2	4	4
Local NM $k$	2	2	8	4	8	4	16
Local BDA $k$	2	8	16	8	32	32	32
HKNN $k$	8	4	64	8	4	32	128
HKNN $\lambda$	1	1	50	1	1	50	50
ridge $k$	8	8	256	8	32	32	128
SVM-KNN $k$	2	128	128	64	32	256	256
SVM-KNN $C$	0.001	10	0.1	10	10	0.01	0.1

#### 4.4 Random-Partition Benchmark Data Set Results

Table 6 shows the mean misclassification rate averaged over the 10 randomized train/test splits. For each dataset the lowest mean score is in bold, as well as any results for which the lowest mean score classifier was not statistically significantly better, according to a two-sided Wilcoxon nonparametric signed rank tests with a significance value of  $p = .05$ . With these iid partitions and averaged over ten randomizations, one sees less dramatic differences between the cross-validation and Bayesian neighborhood error rates.

Table 7 shows for each dataset and classifier whether the classification results were statistically significantly better using cross-validation or Bayesian neighborhoods according to two-sided Wilcoxon nonparametric signed rank tests with a significance value of  $p = .05$ ; the mark – denotes that neither was significantly better than the other. The results vary by algorithm. Local NM prefers cross-validation; we believe this is because it has a larger model bias than the other classifiers, and performs particularly poorly with large  $k$ . Thus, averaging the discriminants over the given range of  $k$  results in higher error. Given a smaller range of  $k$ , we hypothesize local NM would not favor cross-validation. On the other hand, HKNN is statistically significantly better with Bayesian neighborhoods in four out of five cases. The other classifiers do not show consistent statistically significant differences. However, in terms of the different metric of average errors shown in Table 6, we note that local BDA and ridge never performed better with cross-validation than with Bayesian neighborhoods.

Because the training and test are iid, one would expect the fact that we do not cross-validate the regularization parameters for HKNN and SVM-KNN for Bayesian neighborhoods would put it a greater disadvantage than in the standard partition experiments. This appears to have mattered only a little, with HKNN’s average error better on three of five data sets without any cross-validation, and SVM-KNN’s average error better on three of five data sets with cross-validation.

Comparing the different local classifiers, local BDA again performs consistently well and achieves the lowest total average error for Table 6, with HKNN the second best performer in terms of total average error, and SVM-KNN and ridge the next top performers.

Table 6: % Test Error Averaged Over 10 Random Training/Test Partitions

	Vowel		ImageSeg		OptDigits		LetterRec		PenDigits	
	CV	BN	CV	BN	CV	BN	CV	BN	CV	BN
kNN	12.2	13.4	6.1	6.8	2.9	3.0	7.6	6.9	0.9	1.4
DANN	8.5	9.3	4.3	4.2	2.3	2.6	6.7	6.3	1.2	1.5
Local NM	6.5	12.8	5.0	8.6	2.0	2.8	5.3	7.8	0.6	1.3
Local BDA	6.4	6.3	4.0	<b>3.9</b>	1.4	<b>1.3</b>	4.2	4.1	0.6	0.6
HKNN	<b>5.1</b>	<b>4.3</b>	5.1	4.2	1.5	1.7	5.0	5.4	0.6	<b>0.5</b>
ridge	7.3	6.5	4.6	4.2	1.8	1.4	5.8	<b>3.8</b>	<b>0.5</b>	<b>0.5</b>
SVM-KNN	6.3	8.2	<b>3.9</b>	4.6	<b>1.3</b>	1.4	4.9	4.9	0.6	0.6

**Bold:** Best mean result for each dataset, and results that are not statistically significantly worse.

Table 7: Statistically Significantly Better Performance: Bayesian Neighborhoods vs. Cross-validation

	Local BDA	Local NM	HKNN	KNN	DANN	Ridge	SVM-KNN
Vowel	-	CV	<i>BN</i>	CV	-	-	CV
Image Seg.	-	CV	<i>BN</i>	CV	-	-	CV
Opt. Digits	<i>BN</i>	CV	CV	-	CV	<i>BN</i>	-
Letter Rec.	-	CV	<i>BN</i>	<i>BN</i>	<i>BN</i>	<i>BN</i>	-
Pen Digits	-	CV	<i>BN</i>	CV	CV	-	-

## 5. Conclusions and Open Questions

In this paper, we have proposed a Bayesian alternative to neighborhood selection for local classifiers that is optimal in the sense that the recovered posterior minimizes the expected Bregman divergence to the true posterior distribution. We showed that Bayesian neighborhoods achieves error rates that are competitive to those given by a cross-validated neighborhood size with seven different local classifiers, but requires no pre-processing. Bayesian neighborhoods coupled with the proposed local BDA classifier takes expectations with respect to both the uncertain posterior and neighborhood size, and performed strongly across the set of experiments compared to the six other local classifiers.

While not suitable for all applications, lazy learning is an effective approach for a wide variety of tasks, in particular those characterized by an evolving training distribution where frequent re-training is impractical, and those tasks where the training set is too large to train global classifiers. More generally, because lazy learning makes strictly looser assumptions than globally-trained classifiers about training and test samples being iid, we hypothesize that effective completely lazy learning methods can perform better than globally-trained classifiers for applications where the iid assumption is not valid, though this remains an open question.

Although we used a sampled log-uniform prior  $P_K$  that we believe is a reasonable choice given no other information, prior probabilities can have a strong effect on Bayesian estimation, and how to choose an optimal or effective data-dependent prior over the neighborhood sizes is an open question.

## Acknowledgments

We thank Hyrum Anderson for helpful discussions. This research was supported by the United States Office of Naval Research.

## Appendix

### Proof of Lemma 1:

For notational simplicity, we denote  $X_h$  and  $\mu_h$  by  $X$  and  $\mu$  in this proof. The  $\alpha$  which solves the minimization in (6) has the closed-form solution,

$$\alpha = (X^T X + \lambda I)^{-1} X^T (x - \mu).$$

With this  $\alpha$ , the HKNN discriminant becomes,

$$\begin{aligned} d(x, \mu)^2 &= \|(x - \mu) - X(X^T X + \lambda I)^{-1} X^T (x - \mu)\|_2^2 + \lambda \|(X^T X + \lambda I)^{-1} X^T (x - \mu)\|_2^2 \\ &= \|(I - X(X^T X + \lambda I)^{-1} X^T)(x - \mu)\|_2^2 + \lambda \|(X^T X + \lambda I)^{-1} X^T (x - \mu)\|_2^2 \\ &\stackrel{(a)}{=} \|\lambda(\lambda I + X X^T)^{-1} (x - \mu)\|_2^2 + \lambda \|X^T (\lambda I + X X^T)^{-1} (x - \mu)\|_2^2 \\ &= (x - \mu)^T (\lambda I + X X^T)^{-1} \lambda (\lambda I + X X^T) (\lambda I + X X^T)^{-1} (x - \mu) \\ &= \lambda (x - \mu)^T (\lambda I + X X^T)^{-1} (x - \mu), \end{aligned}$$

where (a) follows by the matrix identities  $I - A(I + BA)^{-1} B = (I + AB)^{-1}$  and  $(I + AB)^{-1} A = A(I + BA)^{-1}$  (Petersen and Pedersen, 2005).

### Proof of Lemma 2:

The decision boundary between class 1 and 2 is defined by the set of  $x$  such that  $E_{N_{1,k}}[N_{1,k}(x)] = E_{N_{2,k}}[N_{2,k}(x)]$ . From (8), without loss of generality the decision boundary between class 1 and 2 is,

$$\left[1 + \frac{k}{k+1} (x - \bar{x}_1)^T D_1^{-1} (x - \bar{x}_1)\right]^{\frac{k+d+4}{2}} = \gamma_{db} \left[1 + \frac{k}{k+1} (x - \bar{x}_2)^T D_2^{-1} (x - \bar{x}_2)\right]^{\frac{k+d+4}{2}}, \quad (9)$$

where,

$$D_{h,k} = B_{h,k} + \sum_{i=1}^k (x_i - \bar{x}_h)(x_i - \bar{x}_h)^T I_{(y_i=h)},$$

and  $\gamma_{db}$  is a constant that depends on the training samples and the number of training samples, but does not depend on the test sample  $x$ . Because the exponentiated terms must always be real and positive, raising both sides of (9) to the power  $2/(k+d+4)$ , gives the following quadratic decision boundary:

$$\left(1 + \frac{k}{k+1} (x - \bar{x}_1)^T D_1^{-1} (x - \bar{x}_1)\right) = \tilde{\gamma}_{db} \left(1 + \frac{k}{k+1} (x - \bar{x}_2)^T D_2^{-1} (x - \bar{x}_2)\right),$$

where  $\tilde{\gamma}_{db} = \gamma_{db}^{\frac{2}{k+d+4}}$ .

## References

D. Aha. *Lazy Learning*. Springer, 1997.

- E. Alpaydin. Voting over multiple condensed nearest neighbors. *Artificial Intelligence Research*, (11):115–132, 1997.
- A. Banerjee, X. Guo, and H. Wang. On the optimality of conditional expectation as a Bregman predictor. *IEEE Trans. on Information Theory*, 51(7):2664–2669, 2005.
- S. Bay. Combining nearest neighbor classifiers through multiple feature subsets. *Proc. Intl. Conf. Machine Learning (ICML)*, pages 37–45, 1998.
- H. Bensmail and G. Celeux. Regularized Gaussian discriminant analysis through eigenvalue decomposition. *Journal American Statistical Association*, 91:1743–1748, 1996.
- Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- J. H. Friedman. Regularized discriminant analysis. *Journal American Statistical Association*, 84(405):165–175, 1989.
- B. A. Frigiyik, S. Srivastava, and M. R. Gupta. Functional Bregman divergence and Bayesian estimation of distributions. *arXiv preprint cs.IT/0611123*, 2006.
- S. Geisser. Posterior odds for multivariate normal distributions. *Journal Royal Society Series B Methodological*, 26:69–76, 1964.
- A. K. Ghosh, P. Chaudhuri, and C. A. Murthy. On visualization and aggregation of nearest neighbor classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1592–1602, October 2005.
- M. R. Gupta, R. Gray, and R. Olshen. Nonparametric supervised learning by linear interpolation with maximum entropy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(5):766–781, May 2006a.
- M. R. Gupta, S. Srivastava, and L. Cazzanti. Minimum expected risk estimation for near-neighbor classification. *UWEE Tech Report Series*, (2006-0006), 2006b.
- M. R. Gupta, E. K. Garcia, and E. M. Chin. Adaptive local linear regression with application to printer color management. *IEEE Trans. Image Processing (to appear)*, 2008.
- P. Hall and R. J. Samworth. Properties of bagged nearest neighbour classifiers. *Journal Royal Statistical Society B*, 67:363–379, 2005.
- T. Hastie and R. Tibshirani. Discriminative adaptive nearest neighbour classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(6):607–615, 1996.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, New York, 2001.
- A. E. Hoerl and R. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

- C. C. Holmes and N. M. Adams. A probabilistic nearest neighbour method for statistical pattern recognition. *Journal Royal Statistical Society B*, 64:295–306, 2002.
- D. G. Keehn. A note on learning for Gaussian properties. *IEEE Trans. on Information Theory*, 11:126–132, 1965.
- W. Lam, C. Keung, and D. Liu. Discovering useful concept prototypes for classification based on filtering and abstraction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(8):1075–1090, August 2002.
- D. Masip and J. Vitrià. Boosted discriminant projections for nearest neighbor classifiers. *Pattern Recognition*, 39:164–170, 2006.
- Y. Mitani and Y. Hamamoto. Classifier design based on the use of nearest neighbor samples. *Proc. Intl. Conf. on Pattern Recognition*, pages 769–772, 2000.
- Y. Mitani and Y. Hamamoto. A local mean-based nonparametric classifier. *Pattern Recognition Letters*, 27:1151–1159, 2006.
- M. Paik and Y. Yang. Combining nearest neighbor classifiers versus cross-validation selection. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004.
- K. B. Petersen and M. S. Pedersen. The matrix cookbook. Technical report, Technical University of Denmark, 2005.
- B. Ripley. *Pattern recognition and neural nets*. Cambridge University Press, Cambridge, 2001.
- J. S. Sánchez, F. Pla, and F. J. Ferri. On the use of neighbourhood-based non-parametric classifiers. *Pattern Recognition Letters*, pages 1179–1186, 1997.
- R. Sibson. *Interpreting multivariate data*, chapter : A brief description of natural neighbour interpolation, pages 21–36. John Wiley, 1981.
- D. B. Skalak. *Prototype Selection for Composite Nearest Neighbor Classification*. PhD thesis, Univ. of Massachusetts, 1996.
- T. G. Speed. *Statistical Analysis of Gene Expression Microarray Data*. CRC Press, 2003.
- S. Srivastava and M. R. Gupta. Distribution-based Bayesian minimum expected risk for discriminant analysis. *Proc. IEEE Intl. Symposium on Information Theory*, pages 2294–2298, 2006.
- S. Srivastava, M. R. Gupta, and B. A. Frigyik. Bayesian quadratic discriminant analysis. *Journal of Machine Learning Research*, 8:1287–1314, 2007.
- C. J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4):595–645, 1977.
- P. Vincent and Y. Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. *NIPS*, pages 985–992, 2001.

H. Zhang, A. C. Berg, M. Maire, and J. Malik. SVM-KNN: discriminative nearest neighbor classification for visual category recognition. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2:2126–2136, 2006.