Classification via Minimum Incremental Coding Length (MICL)

John Wright, *Student Member, IEEE,* Yangyu Tao, Zhouchen Lin, *Member, IEEE*
Yi Ma, *Senior Member, IEEE* and Heung-Yeung Shum, *Fellow, IEEE*

Abstract

We present a simple new criterion for classification, based on principles from lossy data compression. The criterion assigns a test sample to the class that uses the minimum number of additional bits to code the test sample, subject to an allowable distortion. We rigorously prove asymptotic optimality of this criterion for Gaussian (normal) distributions and analyze its relationships to classical classifiers. The theoretical results provide new insights into the relationships among a variety of popular classifiers such as MAP, RDA, k-NN, and SVM, as well as unsupervised methods based on lossy coding [18]. Our formulation induces several good effects on the resulting classifier. First, minimizing the lossy coding length induces a regularization effect which stabilizes the (implicit) density estimate in a small sample setting. Second, compression provides a uniform means of handling classes of varying dimension. The new criterion and its kernel and local versions perform competitively on synthetic examples, as well as on real imagery data such as handwritten digits and face images. On these problems, the performance of our simple classifier approaches the best reported results, without using domain-specific information. All MATLAB code and classification results are publicly available for peer evaluation at `http://perception.csl.uiuc.edu/jnwright/coding`.

Index Terms

Classification, Lossy Data Coding, Regularization, MAP, RDA, k-NN, SVM.

I. INTRODUCTION

The quintessential problem in statistical learning [11], [26] is to construct a classifier from labeled training data $(\boldsymbol{x}_i, y_i) \overset{iid}{\sim} p_{X,Y}(\boldsymbol{x}, y)$. Here, $\boldsymbol{x}_i \in \mathbb{R}^n$ is the observation, and $y_i \in \{1, \ldots, K\}$ its associated class label. The goal is to construct a classifier $g : \mathbb{R}^n \to \{1, \ldots, K\}$ which minimizes the expected risk (or probability of error):

$$g^* = \arg\min \mathbb{E}[I_{g(X) \neq Y}], \tag{1}$$

where the expectation is taken with respect to $p_{X,Y}$. When the conditional class distributions $p_{X|Y}(\boldsymbol{x}|y)$ and the class priors $p_Y(y)$ are known, then the *maximum a posterior* (MAP) assignment

$$\hat{y}(\boldsymbol{x}) = \arg\max_{y \in \{1,\ldots,K\}} \ln p_{X|Y}(\boldsymbol{x}|y) + \ln p_Y(y) \tag{2}$$

gives the optimal classifier.

A. Issues with Learning the Distributions from Training Samples

In the typical classification setting, the distributions $p_{X|Y}(\boldsymbol{x}|y)$ and $p_Y(y)$ need to be learned in advance from a set of training data whose class labels are given. Conventional approaches to model estimation (implicitly) assume that the distributions are nondegenerate and the samples are sufficiently dense. However, these assumptions fail in many classification problems which are vital for applications in computer vision [13], [15], [16], [27]. For instance, in the case of face recognition, the set of images of a person's face taken from different angles and under different lighting conditions often lie in a low-dimensional subspace or submanifold of the ambient space [12]. As a result, the associated distributions are degenerate or nearly degenerate. Moreover, due to the high dimensionality of imagery data, the set of training images is typically sparse.

Inferring the generating probability distribution $p_{X,Y}$ from a sparse set of samples is an inherently ill-conditioned problem [26]. Furthermore, in the case of degenerate distributions, the classical likelihood function (2) does not have a well-defined maximum [26]. Thus, to infer the distribution from the training data or to use it to classify new observations, the distribution or its likelihood function needs to be properly "regularized." Typically, this is accomplished either explicitly via smoothness constraints, or implicitly via parametric assumptions on the distribution [3]. However, even if the distributions are assumed to be generic Gaussians, explicit regularization is still necessary to achieve good small-sample performance [7]. This effect is particularly insidious in the high-dimensional data spaces common in computer vision, pattern recognition and bioinformatics. For example, naive covariance estimators are inconsistent when the number of samples is proportional to the dimension of the space [2], as are estimators of subspace structure such as principal components [14].

In many real problems in computer vision, the distributions associated with different classes of data have different model complexity. For instance, when detecting a face in an image, features associated with the face often have a low-dimensional structure which is "embedded" as a sub-manifold in a cloud of essentially random features from the background. Model selection criteria such as the *minimum description length* (MDL) [17], [23] serve as important modifications to MAP for estimating a model across classes of different complexity. It selects the optimal model as the one that minimizes the overall coding length of the given (training) data, hence the name "minimum description length" or "minimum coding length" [1]. However, notice that MDL does

not specify how the model complexity should be properly accounted for when classifying new test data among models that have different dimensions.[1]

B. Minimum Coding Length Principle for Classification

Once the distributions $p_{X|Y}$ and p_Y are estimated from the training data, the classifier is usually obtained by substituting the estimated distributions $\hat{p}_{X|Y}$ and $\hat{p}_Y$ into the MAP classifier (2). Notice that the MAP classifier (2) is equivalent to

$$\hat{y}(\boldsymbol{x}) = \arg\min_{y\in\{1,\ldots,K\}} -\ln p_{X|Y}(\boldsymbol{x}|y) - \ln p_Y(y). \tag{3}$$

This gives the MAP classifier another interpretation. The optimal classifier should minimize Shannon' optimal (lossless) coding length of the test data $\boldsymbol{x}$ with respect to the distribution of the true class: The first term is the number of bits needed to code $\boldsymbol{x}$ w.r.t. the distribution of class y, and the second term is the number of bits needed to code the label y for $\boldsymbol{x}$. In this paper, we essentially follow this minimum coding length principle for classification.

However, as we have contended in the previous subsection, the (potentially degenerate) distributions $p_{X|Y}(\boldsymbol{x}|y)$ and $p_Y(y)$ can be very difficult to learn from a few samples in a high-dimensional space. It therefore makes more sense to look for other good surrogates for implementing the above minimum coding length principle. Our idea is to measure how efficiently a new observation can be encoded by each class of the training data subject to an allowable distortion, and to assign the new observation to the class that requires the minimum number of additional bits. We dub this criterion "*minimum incremental coding length*" *(MICL)* for classification, as a counterpart of the MDL principle for model estimation and as a surrogate for the minimum coding length principle for classification.

We will see that the proposed criterion naturally addresses the issues of regularization and model complexity. Regularization is introduced through the use of *lossy coding*, i.e. coding the test data $\boldsymbol{x}$ up to an allowable distortion. This contrasts with Shannon's optimal coding length which requires the precise knowledge of the true distributions, and thus places our approach more along the lines of lossy MDL [20]. As we will also see, the lossy coding length naturally accounts

[1]Note that model estimation is about inferring a model from the training data whereas classification is about inferring a decision on a new test sample *given* the models.

for model complexity by directly measuring the difference in the volume (hence dimension) of the training data with and without the new observation.

In [18], we have investigated minimum lossy coding length in the context of *unsupervised* data clustering. There, the coding length subject to an allowable distortion was used to measure the goodness of a clustering, and a simple agglomerative method was proposed to segment data from mixtures of Gaussians or linear subspaces. In a sense, this paper extends these results to the supervised domain, inducing a simple new classifier and studying its properties. Moreover, the new theoretical results described here further explain for the surprising efficacy of the simple clustering algorithm of [18]. For example, Theorem 1 implies that the agglomerative method of [18] makes a decision at each step based on a regularized version of (Gaussian) maximum likelihood or maximum a posterior.

C. Relationships to Existing Classifiers

While MICL and MDL both operate by minimizing a coding-theoretic objective, MICL differs strongly from traditional MDL approaches to classification such as those examined in [9]. Those methods choose an optimal *decision boundary* from an allowable set by minimizing the following coding length:

$$g^* = \arg\min_{g \in \mathcal{G}} \; L(g) + \log \binom{m}{\sum_i I_{g(\boldsymbol{x}_i) \neq y_i}}, \tag{4}$$

where $L(g)$ is the number of bits needed to code the classifying boundary g within certain class $\mathcal{G}$, and the second term counts the cost of coding training samples misclassified by g. This approach has been proven *inconsistent* in [9]. In contrast, MICL uses coding length *directly* as a measure of how well the training data represent the new sample. The inconsistency result of [9] does not apply in this modified context. In fact, MICL will have more in common with the classical ML/MAP decision criteria, since maximizing the likelihood also minimizes the number of bits needed to code the sample according to Shannon's optimal *lossless* coding scheme. However, the use of *lossy coding* distinguishes MICL from these approaches. Within the lossy data coding framework, we establish in this paper that the MICL criterion leads to a family of classifiers that generalize the conventional MAP classifier (2). We rigorously show that for Gaussian distributions, the MICL criterion asymptotically converges to a regularized version

of MAP[2] (see Theorem 1) and we also gives a precise estimate on the convergence rate (see Theorem 2). In the Gaussian case, one effect of lossy coding is to induce a regularization effect similar to Friedman's Regularized Discriminant Analysis (RDA) [7][3], with similar gains in finite sample performance with respect to MAP/QDA.

The fully Bayesian approach to model estimation, in which posterior distributions over model parameters are estimated also claims finite sample gains over ML/MAP [19], [21]. However, these methods generally require that the number of samples be larger than the dimension of the space. When this condition is not satisfied (as for high-dimensional or degenerate data), the result becomes strongly dependent on the choice of prior[4]. MICL requires no such assumptions, and in fact sees its greatest advantage when the sample size is small. Notice, however, that Theorem 1 also ensures asymptotic equivalence to the Bayesian approach, since it too converges to ML/MAP asymptotically.

When the distributions involved are not Gaussian, the MICL criterion can be easily extended via a nonlinear kernel or can be applied in a local neighborhood of the test sample, similar to the popular k-Nearest Neighbor (k-NN) classifier [6], [22]. However, the local MICL classifier significantly improves the k-NN classifier as it accounts for both the number of samples and the distribution of the samples within the neighborhood. When dealing with almost degenerate distributions or sparse samples, the distribution of the neighboring samples typically contains more information than the majority label about the correct class of the new observation (see Figure 4 for a comparison).

Work on Support Vector Machines (SVM) [26] has shown that not all samples in the training data are equally important for the resulting classifier. In this framework, the final decision hypersurface is represented in terms of a small portion of nearby samples, called "support vectors." Thus, for generic distributions, the SVM may significantly compress the training data

[2]MAP subject to a Gaussian assumption is also known in the learning literature as Quadratic Discriminant Analysis (QDA) [11].

[3]Throughout this paper, we only consider the version of RDA which regularizes the covariance by a multiple of the identity: $\tilde{\Sigma} = \Sigma + \alpha I$. Regularizing by the pooled data covariance as in [7] is less appropriate if we wish to consider groups with significantly different and anisotropic covariances.

[4]In the Gaussian case, Jeffery's prior no longer suffices in this regime, and stronger assumptions on the parameters of the distribution are required to regularize the problem.

for classification purposes. However, if the training data have degenerate structure such that most samples lie on low-dimensional subspaces or submanifolds, almost all the samples help determine the global shape of the optimal separating hyperplane or hypersurface. In this case, learning the separating hyperplane or hypersurface via SVM may no longer be more generalizable than directly harnessing the low-dimensional structures of the training data via MICL for classification (see Figure 4 for a comparison). Moreover, the kernelized version of MICL provides a simpler alternative to the SVM approach of constructing a linear decision boundary in the embedded (kernel) space, potentially exploiting details of the structure of the embedded data (see Figure 5 for an example).

D. Contributions of this Paper

The main contribution of this paper is to establish for the first time a formal and rigorous connection between classification and lossy data compression. The theoretical results provide new insights into the relationships among a variety of popular classifiers such as MAP, RDA, k-NN, and SVM, unsupervised methods such as [18], as well as the relationship of classification to important statistical concepts such as regularization, model complexity, and rate distortion. As a result, the proposed MICL classifier, though very simple, performs competitively under a wider range of conditions than many conventional classifiers. Extensive simulations and experiments on real imagery data show that MICL often approaches the best reported results from more sophisticated classifiers or systems, *without* using any domain-specific information (Section III).

II. Classification Criteria and Analysis

A. Minimum Incremental Coding Length

We formulate the problem of classification from the perspective of lossy data coding and compression [5]. A *lossy coding scheme* maps vectors $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) \in \mathbb{R}^{n \times m}$ to a sequence of binary bits, from which the original vectors can be recovered upto an allowable distortion $\mathbb{E}[\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2] \leq \varepsilon^2$. The length of the bit sequence is then a function: $L_\varepsilon(\mathcal{X}) : \mathbb{R}^{n \times m} \to \mathbb{Z}_+$. Given a lossy coding scheme and its associated coding length function $L_\varepsilon(\cdot)$, we can encode each class of training data $\mathcal{X}_j \doteq \{\boldsymbol{x}_i : y_i = j\}$ using $L_\varepsilon(\mathcal{X}_j)$ bits. The entire training dataset can

be represented by a two-part code using

$$\text{Length}\big(\mathcal{X},\mathcal{Y}\big) = \sum_{j=1}^{K} L_\varepsilon(\mathcal{X}_j) - |\mathcal{X}_j| \log_2 p_Y(j) \text{ (bits)}. \tag{5}$$

Here, the second term is the minimum number of bits needed to (losslessly) code the class labels y_i.

Now, suppose we are given a new (test) sample $\boldsymbol{x} \in \mathbb{R}^n$, whose associated class label is $y(\boldsymbol{x}) = j$. If we code $\boldsymbol{x}$ jointly with the training data $\mathcal{X}_j$ of the jth class, the number of additional bits needed to code the pair $(\boldsymbol{x}, y)$ is:

$$\delta L_\varepsilon(\boldsymbol{x}, j) = L_\varepsilon(\mathcal{X}_j \cup \{\boldsymbol{x}\}) - L_\varepsilon(\mathcal{X}_j) + L(j). \tag{6}$$

Here, the first two terms measure the excess bits needed to code $(\boldsymbol{x}, \mathcal{X}_j)$ upto distortion ε^2, while the last term $L(j)$ is the cost of losslessly coding the label $y(\boldsymbol{x}) = j$. One may view these as "finite-sample lossy" surrogates for the Shannon coding lengths in the ideal classifier (3). This interpretation naturally lends it to the following classifier:

Criterion 1 (Minimum Incremental Coding Length): Assign $\boldsymbol{x}$ to the class which minimizes the number of additional bits needed to code $(\boldsymbol{x}, \hat{y})$, subject to the distortion ε:

$$\hat{y}(\boldsymbol{x}) \doteq \arg\min_{j=1,\ldots,K} \delta L_\varepsilon(\boldsymbol{x}, j). \tag{7}$$

The above criterion (7) can be taken as a general principle for classification, in the sense that it can be applied using any lossy coding scheme and its associated coding length function. Nevertheless, in order for the classification to be effective, the coding scheme should be such that the associated coding length is the shortest possible for the given data. More specifically, if the data are from some family of distributions, the asymptotically optimal coding length is given by the rate-distortion of the distribution[5] [5]; Or if we consider the data as a discrete set of points, the coding length should be approximately[6] the minimum among all possible coding schemes subject to the given distortion. For the rest of this subsection, we discuss how to choose the function $L_\varepsilon(\cdot)$ for the data $\boldsymbol{x}$ and $L(\cdot)$ for the label $y(\boldsymbol{x})$ in the formula (6) of δL_ε.

[5]The construction of optimal coding schemes (achieving the lower bound given by the rate-distortion of the data distribution) is a difficult problem, even in the Gaussian case (see e.g. [10]). Note however, that for the purposes of classification, it is only necessary for there to exist *in principle* a coding scheme whose length function is L_ε.

[6]Approximation is necessary even if the given data are binary numbers instead of real-valued vectors, since the universal minimum coding length, or Kolmogorov complexity, of the data is non-computable [5].

1) Lossy Coding Length of Gaussian Data: We will first consider a coding length function L_ε which is approximately (asymptotically) optimal for Gaussian distributions. The (implicit) use of a coding scheme which is optimal for Gaussian sources is equivalent to assume that the conditional class distributions $p_{X|Y}$ are unimodal, and can be well-approximated by Gaussians.[7] We will rigorously analyze the performance of the MICL in this (admittedly restrictive) scenario, and demonstrate its relationships with classical classifiers such as MAP and RDA. In Section III we will show using the same L_ε function, how the MICL can be extended to arbitrary, multimodal distributions via an effective local Gaussian approximation.

For a multivariate Gaussian source $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, the average number of bits needed to code a vector subject to a distortion ε^2 is approximately:

$$R_\varepsilon(\Sigma) \doteq \frac{1}{2}\log_2 \det\left(I + \frac{n}{\varepsilon^2}\Sigma\right) \quad \text{(bits/vector)}. \tag{8}$$

Then given the data $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$ with sample mean $\hat{\boldsymbol{\mu}} = \frac{1}{m}\sum_i \boldsymbol{x}_i$, we can represent them upto expected distortion ε^2 using on average $R_\varepsilon(\hat{\Sigma})$ bits, where $\hat{\Sigma}(\mathcal{X}) = \frac{1}{m-1}\sum_i (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T$ is the sample covariance, and so the number of bits needed for the m vectors is $mR_\varepsilon(\hat{\Sigma})$. Since the optimal codebook is adaptive to the data, we need additional $nR_\varepsilon(\hat{\Sigma})$ bits to represent the principal axes of the covariance matrix. In addition, we need an extra $\frac{n}{2}\log_2\left(1 + \frac{\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}}}{\varepsilon^2}\right)$ bits to code the mean vector $\hat{\boldsymbol{\mu}}$. Thus, the total number of bits required to code $\mathcal{X}$ becomes:

$$L_\varepsilon(\mathcal{X}) \doteq \frac{m+n}{2}\log_2 \det\left(I + \frac{n}{\varepsilon^2}\hat{\Sigma}(\mathcal{X})\right) + \frac{n}{2}\log_2\left(1 + \frac{\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}}}{\varepsilon^2}\right). \tag{9}$$

The first term, therefore, gives the number of bits needed to code the distribution of the vectors $\boldsymbol{x}_i$ about their mean, $\hat{\boldsymbol{\mu}}$, while the second gives the number of bits needed to code the mean.

In addition to well-approximating the optimal coding length for Gaussian data, one can show that this function gives a good upper bound on the number of bits needed to code finitely many samples lying on a linear subspace (or equivalently, a degenerate Gaussian distribution). See [18] for a more detailed derivation and justification.

2) Coding of the Class Label: Since the label Y is discrete, it can be coded losslessly. The form of the final term $L(j)$ in (6) depends on one's prior assumptions about the nature of the

[7]This assumption can be significantly relaxed. The same analysis and results can be easily generalized to a mixture of Gaussians.

test data. If the test class labels Y are known to have the marginal distribution $P[Y = j] = \pi_j$, then the optimal coding lengths are (within one bit):

$$L(j) = -\log_2 \pi_j. \tag{10}$$

If the testing data are also iid samples from the same distribution $p_{X,Y}$ as the training data, then we may estimate $\hat{\pi}_j = \frac{|\mathcal{X}_j|}{m}$. Conversely, if we have no prior knowledge regarding the distribution of the class labels, it may be more appropriate to set $\pi_j \equiv \frac{1}{K}$, in which case the excess coding length depends only on the number of additional bits needed to encode $\boldsymbol{x}$. Similar to the MAP classifier (2), the choice of π_j effectively gives a prior on class labels.

3) The Overall Algorithm: Given the coding length function (86) for the observations and the coding length (10) for the class label, we summarize the MICL criterion (7) as Algorithm 1 below.

Algorithm 1 (The MICL Classifier).

1: **Input:** a set of m training samples partitioned into K classes $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_K$ and a test sample $\boldsymbol{x}$.

2: Compute prior distribution of class labels $\pi_j = |\mathcal{X}_j|/m$.

3: Compute incremental coding length of $\boldsymbol{x}$ for each class:

$$\delta L_\varepsilon(\boldsymbol{x}, j) = L_\varepsilon(\mathcal{X}_j \cup \{\boldsymbol{x}\}) - L_\varepsilon(\mathcal{X}_j) - \log_2 \pi_j,$$

where

$$L_\varepsilon(\mathcal{X}) \doteq \frac{m+n}{2} \log_2 \det\left(I + \frac{n}{\varepsilon^2}\hat{\Sigma}(\mathcal{X})\right) + \frac{n}{2}\log_2\left(1 + \frac{\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}}}{\varepsilon^2}\right).$$

4: Let $\hat{y}(\boldsymbol{x}) = \arg\min_{j=1,\ldots,K} \delta L_\varepsilon(\boldsymbol{x}, j)$.

5: **Output:** $\hat{y}(\boldsymbol{x})$.

Figure 1 shows the performance of the MICL classifier on two simple but informative toy problems in $\mathbb{R}^2$. In both cases, the MICL criterion harnesses the covariance structure of the data to achieve good classification results, even in sparsely sampled regions. In the left example, the criterion *interpolates* the data structure to achieve correct classification, even near the origin where the samples are sparse. In the right example, the criterion *extrapolates* the horizontal line to the other side of the plane. In both cases, methods such as k-NN and support vector machine

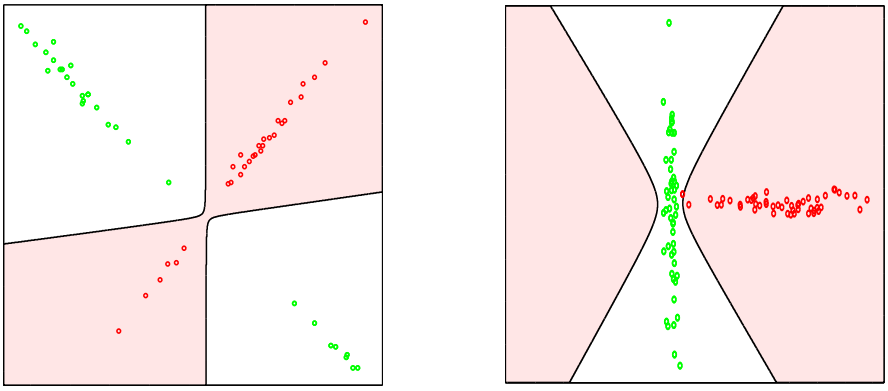


Fig. 1. MICL harnesses the covariance structure of the data to *interpolate* (left) and *extrapolate* (right) in regions where the training samples are sparse.

(SVM) fail to give correct classification in these regions (see Figure 4 for a comparison). The astute reader may notice, however, that these decision boundaries are very similar to what MAP/QDA would give. This raises an important question: what is the precise relationship between MICL and MAP, and under what circumstances is MICL superior?

B. Asymptotic Behavior and Relationship to MAP

In this section, we analyze the asymptotic behavior of the MICL criterion (7) using coding length function (86), as the number of training samples, m, goes to infinity. We will see that asymptotically, classification based on the incremental coding length is equivalent to a regularized version of MAP (or ML), subject to a reward on the dimension of the classes. The precise correspondence is given by the following theorem, proven in Appendix I:

Theorem 1 (Asymptotic MICL): Let the training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m \overset{iid}{\sim} p_{X,Y}(\boldsymbol{x}, y)$, with[8] $\boldsymbol{\mu}_j \doteq \mathbb{E}[X|Y=j]$, $\Sigma_j \doteq Cov(X|Y=j)$. Then as $m \to \infty$, the MICL criterion coincides (asymptotically, with probability one) with the decision rule

$$\hat{y}(\boldsymbol{x}) = \underset{j=1,\ldots,K}{\operatorname{argmax}} \mathcal{L}_G\Big(\boldsymbol{x} \,\Big|\, \boldsymbol{\mu}_j,\, \Sigma_j + \frac{\varepsilon^2}{n} I\Big) + \ln \pi_j + \frac{1}{2} D_\varepsilon(\Sigma_j), \tag{11}$$

[8]We assume that the first and second moments of the conditional distributions exist.

where $\mathcal{L}_G(\cdot|\,\boldsymbol{\mu},\Sigma)$ is the log-likelihood function for a $\mathcal{N}(\boldsymbol{\mu},\Sigma)$ distribution[9], and $D_\varepsilon(\Sigma_j) \doteq \mathrm{tr}\left(\Sigma_j(\Sigma_j+\frac{\varepsilon^2}{n}I)^{-1}\right)$ is the effective dimension of the j-th model, relative to the distortion ε^2.

This result shows that asymptotically, MICL generates a family of MAP-like classifiers parametrized by the distortion ε^2. Notice that if all of the distributions are non-degenerate (i.e.their covariance matrices Σ_j are nonsingular), then $\lim_{\varepsilon\to 0}\left(\Sigma_j+\frac{\varepsilon^2}{n}I\right)=\Sigma_j$, and $\lim_{\varepsilon\to 0} D_\varepsilon(\Sigma_j)=n$, a constant across the various classes. Thus, for nondegenerate data, the family of classifiers induced by MICL contains the conventional MAP classifier (2) at $\varepsilon=0$. Any reasonable rule for choosing the distortion ε^2 given a finite number, m, of samples should therefore ensure that $\varepsilon\to 0$ as $m\to\infty$. This guarantees that for non-degenerate distributions, MICL converges to the asymptotically optimal MAP criterion.

Simulations (e.g. Figure 1) suggest that the limiting behavior does provide useful information about the performance of the classifier on finite training data. Yet Theorem 1 is only strictly valid as $m\to\infty$, giving no indication as to whether one should expect to observe such behavior in practical scenarios. The following result, proven in Appendix II shows that the MICL discriminant functions, $\delta L_\varepsilon(\boldsymbol{x},j)$ converge quickly to their limiting form, $\delta L_\varepsilon^\infty(\boldsymbol{x},j)$:

Theorem 2 (MICL Convergence Rate): As the number of samples, $m\to\infty$, the MICL criterion (7) converges to its asymptotic form, (25) at a rate of $m^{-\frac{1}{2}}$. More specifically[10], with probability at least $1-\alpha$, $\left|\delta L_\varepsilon(\boldsymbol{z},j)-\delta L_\varepsilon^\infty(\boldsymbol{z},j)\right|\le c(\alpha)\cdot m^{-\frac{1}{2}}$ for some constant $c(\alpha)>0$. From the proof of the theorem, one may further notice that the constant c becomes smaller when the covariance tends to singular, which suggests that the convergence speed is higher when the distributions are closer to being degenerate.

C. Improvements over MAP

In the above, we have established the fact that asymptotically, the MICL criterion (25) is just as good as the MAP criterion. Nevertheless, in the cases of finite samples or degenerate

[9]Notice that although the *form* of the criterion involves a Gaussian log-likelihood, the result holds for arbitrary second-order $p_{X,Y}$, and makes no Gaussian assumption. However, directly applying the MICL with coding length (86) to complicated multimodal distributions will often result in poor classification performance, and is therefore not advisable. Section III discusses how MICL can be modified to handle arbitrary data distributions.

[10]Assuming that the fourth moments $E[\|\boldsymbol{x}-\boldsymbol{\mu}\|^4]$ of the conditional distributions exist.

distributions, the MICL criterion makes several important modifications to the MAP criterion, which may significantly improve its performance.

1) Regularization and Finite-Sample Behavior: Notice that the first two terms of the asymptotic MICL criterion (25) have the form of a MAP criterion, based on an $\mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j + \frac{\varepsilon^2}{n} I)$ distribution, with prior π_j. This is in some sense equivalent to softening or regularizing the distribution by $\frac{\varepsilon^2}{n}$ along each dimension, and has two important effects. First, it renders the associated MAP decision rule well-defined, even when the true data distribution might be (almost) degenerate. Even for non-degenerate distributions, there is empirical evidence showing that for appropriately chosen ε, $\hat{\Sigma} + \frac{\varepsilon^2}{n} I$ gives a more stable finite-sample estimate of the covariance [7], leading to lower misclassification rates.

Figure 2 demonstrates this effect on two simple examples in $\mathbb{R}^2$. In each example, we vary the number of training samples, m, and the distortion ε. For each (m, ε) combination, we draw m training samples from two Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i), i = 1, 2$, and estimate the Bayes risk of the resulting MICL and MAP classifiers. This procedure is repeated 500 times, to estimate the overall Bayes risk with respect to variations in the training data. In Figure 2 we visualize the (estimated) difference in risks, $R_{MAP} - R_{MICL}$. Positive values, then, indicate that MICL is outperforming MAP. The red line approximates the zero level-set of the difference in risks, where the two methods perform equally well.

The generating distributions are parameterized as (at left) $\boldsymbol{\mu}_1 = [-\frac{1}{2}, 0]$, $\boldsymbol{\mu}_2 = [\frac{1}{2}, 0]$, $\Sigma_1 = \Sigma_2 = I$, and (at right) $\boldsymbol{\mu}_1 = [-\frac{3}{4}, 0]$, $\boldsymbol{\mu}_2 = [\frac{3}{4}, 0]$, $\Sigma_1 = diag(1, 4)$, $\Sigma_2 = diag(4, 1)$. At left, in the isotropic case, MICL outperforms MAP for all sufficiently large ε. with a larger performance gain when the number of samples is small. In the anisotropic case (right), for a good range of ε, MICL dramatically outperforms MAP for small sample sizes. We will see in the next example that this effect becomes more pronounced as the dimension, n, increases.

2) Dimension Reward: The effective dimension term $D_\varepsilon(\Sigma_j)$ in the asymptotic MICL criterion (25) can be rewritten as $D_\varepsilon(\Sigma_j) = \sum_{i=1}^{n} \frac{\lambda_i}{\frac{\varepsilon^2}{n} + \lambda_i}$, where λ_i is the ith eigenvalue of Σ_j. Notice that if the data distribution lies on a perfect subspace of dimension d (i.e.. $\lambda_1, \ldots, \lambda_d \gg \frac{\varepsilon^2}{n}$ and $\lambda_{d+1}, \ldots, \lambda_n \ll \frac{\varepsilon^2}{n}$), $D^\perp$ will be exactly d, the dimension of the subspace. In general, D can be viewed as "softened" estimate of the dimension, relative to the distortion ε^2. This quantity has been dubbed the "effective number of parameters" in the context of ridge regression [11]. Thus,

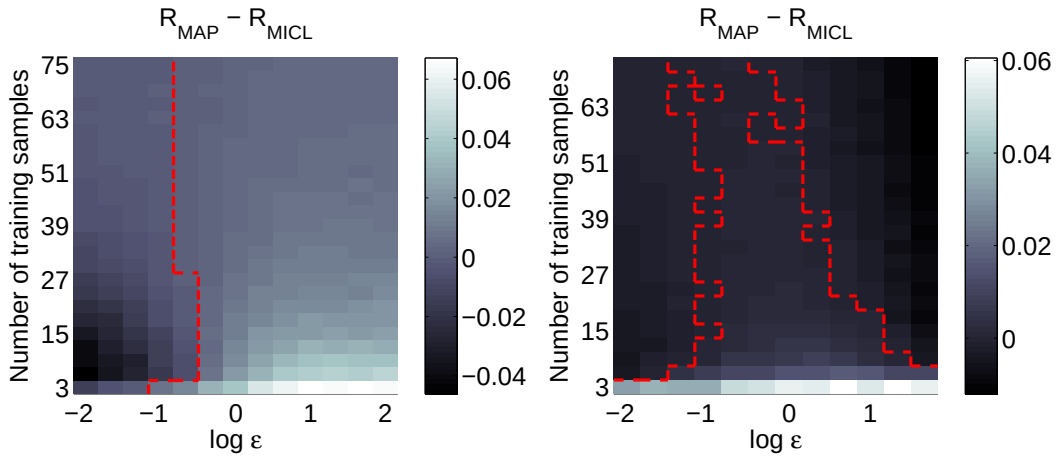


Fig. 2. Excess misclassification risk incurred by using MAP rather than MICL, as a function of ε and m. MICL outperforms MAP in most settings, with the largest gain when m is relatively small. Left: two isotropic Gaussians in $\mathbb{R}^2$. Right: anisotropic Gaussians in $\mathbb{R}^2$.

minimizing the MICL criterion rewards distributions that have relatively higher dimension.[11] Note however, that this effect is somewhat countered by the regularization induced by ε, which has a larger "reward" effect on lower dimensional distributions.

Figure 3 empirically compares MICL to the conventional MAP and the *regularized* MAP (or RDA [7]). In this example, we draw m samples from three nested Gaussian distributions: one has a full rank n, one has rank $n/2$, and one has rank 1. All samples are corrupted by 4% Gaussian noise. We estimate the Bayes risk for each (m, n) combination by averaging over 500 independent trials. For fairness of comparision, the regularization parameter in RDA, and the distortion ε for MICL are chosen independently for each trial to minimize the cross-validation error over the training data. Plotted are the (estimated) differences in risk, $R_{MAP} - R_{MICL}$ (left) and $R_{RDA} - R_{MICL}$ (right). The red lines again correspond to the zero level-set of the difference. Notice that with little surprise, MICL outperforms MAP for most (m, n), and that the effect is most pronounced when n is large and m is small. Interestingly, when m is much smaller than n (e.g. the bottom row of Figure 3 right), MICL demonstrates a significant performance gain with respect to RDA. As the number of samples increases, though, there is a region where RDA

[11]Notice that here dimension assumes an "opposite" role to that in model estimation where we typically penalize models with higher dimension.

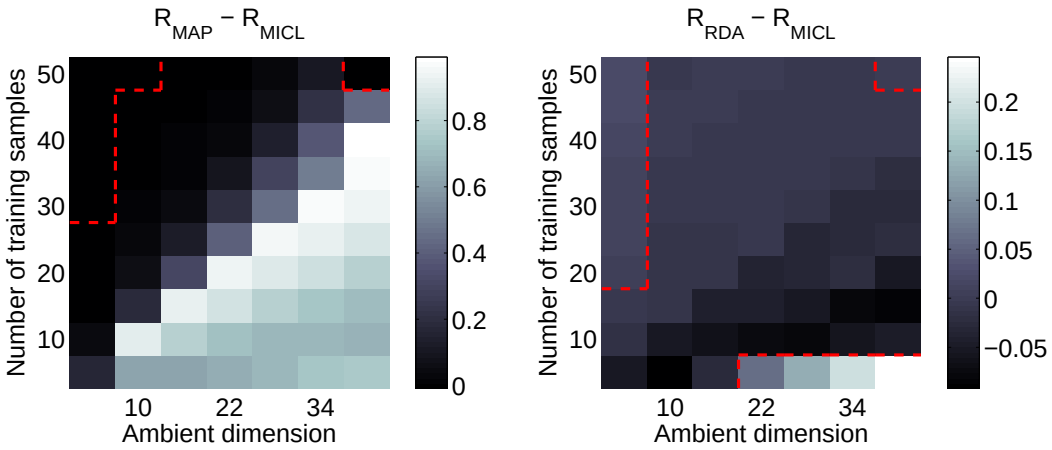


Fig. 3. Excess risk incurred by using MAP and RDA rather than MICL, as a function of number of samples m and dimension n.

is slightly better. However, for most (m, n) considered here, MICL and RDA have rather close performance.[12]

D. Implementation Issues

The rigorous analysis of the Gaussian case in the previous subsections reveals many good properties of the proposed MICL criterion. In reality, the distribution(s) of the data of interest may not be Gaussian. If the rate-distortion function of such distribution(s) is known, one could, in principle, carry out similar analysis as for the Gaussian case. Nevertheless, in this subsection, we discuss some practical ways of modifying the MICL criterion that are applicable to arbitrary distributions, without losing some of the desirable properties of MICL.

1) Kernel MICL Criterion: Since $\mathcal{X}\mathcal{X}^T$ and $\mathcal{X}^T\mathcal{X}$ have the same non-zero eigenvalues, we have the following identity

$$\log_2 \det\big(I + \frac{n}{\varepsilon^2 m}\mathcal{X}\mathcal{X}^T\big) = \log_2 \det\big(I + \frac{n}{\varepsilon^2 m}\mathcal{X}^T\mathcal{X}\big). \tag{12}$$

From this identity, we notice that to evaluate the coding length function (86), we only need to compute the inner products between the data points. Thus, if the data $\boldsymbol{x}$ (of each class) are not Gaussian but there exists a nonlinear map $\psi : \mathbb{R}^n \to \mathcal{H}$ such that the transformed data

[12]Note that RDA [7] is designed to be nearly optimal for finite samples of Gaussians.

$\phi(\boldsymbol{x})$ are (approximately) Gaussian, we can replace the inner product $\boldsymbol{x}_1^T\boldsymbol{x}_2$ with a new one $k(\boldsymbol{x}_1,\boldsymbol{x}_2) \doteq \psi(\boldsymbol{x}_1)^T\psi(\boldsymbol{x}_2)$. The so-defined symmetric positive definite function $k(\boldsymbol{x}_1,\boldsymbol{x}_2)$ is known in statistical learning as a "kernel function"[13]. Thus, by a proper choice of the kernel function, one may achieve better classification performance for certain classes of non-Gaussian distributions. In practice, some popular choices of the kernel functions include the polynomial kernel $k(\boldsymbol{x}_1,\boldsymbol{x}_2) = (\boldsymbol{x}_1^T\boldsymbol{x}_2+1)^d$, the radial basis function (RBF) kernel $k(\boldsymbol{x}_1,\boldsymbol{x}_2) = \exp(-\gamma\|\boldsymbol{x}_1-\boldsymbol{x}_2\|^2)$ and their variants. Notice that by replacing $\boldsymbol{x}_1^T\boldsymbol{x}_2$ with $k(\boldsymbol{x}_1,\boldsymbol{x}_2)$, we are now classifying the test sample $\boldsymbol{x}$ by assigning it to the class which minimizes the additional bits to code $\psi(\boldsymbol{x})$ jointly with $\psi(\boldsymbol{x}_1)\ldots\psi(\boldsymbol{x}_m)$. Appendix IV describes how to properly account for the mean and dimension of the lifted data, so that the discriminant functions are well-defined, and correspond to a proper coding length.

The transformation described above is similar to that used in generalizing SVM to nonlinear decision boundaries. Notice, however, that whereas SVM constructs a linear decision boundary in the lifted space $\mathcal{H}$, kernel MICL exploits the covariance structure of the lifted data, generating decision boundaries that are (asymptotically) quadratic when the $\psi(\boldsymbol{x})$ *are* Gaussian in $\mathcal{H}$. Thus, the theoretical advantage of kernel MICL over kernel SVM is clear. In Section III-B we will see that even for real data whose statistical nature is unclear, kernel MICL outperforms SVM when applied with the same kernel function.

2) Local MICL Criterion: For data drawn from complicated multi-modal distributions, it may be difficult or impossible to find a kernel function that converts the data into Gaussians. In this case, we can apply the MICL criterion locally, in a neighborhood of the test sample $\boldsymbol{x}$. For instance, we may consider the k-nearest[14] neighbors of $\boldsymbol{x}$ in the training set $\mathcal{X}$, which we denote as $N^k(\boldsymbol{x})$. Training data in this neighborhood that belong to each class are $N_j^k(\boldsymbol{x}) \doteq \mathcal{X}_j \cap N^k(\boldsymbol{x}), j = 1,\ldots,K$. Then in the MICL classifier (Algorithm 1), we replace the incremental coding length $\delta L_\varepsilon(\boldsymbol{x},j)$ by its local version:

$$\delta L_\varepsilon(\boldsymbol{x},j) = L_\varepsilon(N_j^k(\boldsymbol{x}) \cup \{\boldsymbol{x}\}) - L_\varepsilon(N_j^k(\boldsymbol{x})) + L(j), \tag{13}$$

where $L(j)$ is replaced with its local version: $L(j) = -\log_2 \frac{|N_j^k(\boldsymbol{x})|}{|N^k(\boldsymbol{x})|}$.

[13]The necessary and sufficient conditions for $k(\cdot,\cdot)$ to be a kernel function are given by Mercer's Theorem [26].

[14]In terms of the Euclidean distance.

The local MICL criterion gives a universal classifier that is applicable to arbitrary distributions. As a corollary to Theorem 1, we have

Corollary 3 (Asymptotic Local MICL): If the probabilistic density function $p_j(\boldsymbol{x}) = p(\boldsymbol{x}|y = j)$ of each class is non-degenerate, as k and m go to ∞ the local MICL criterion converges to the MAP criterion:

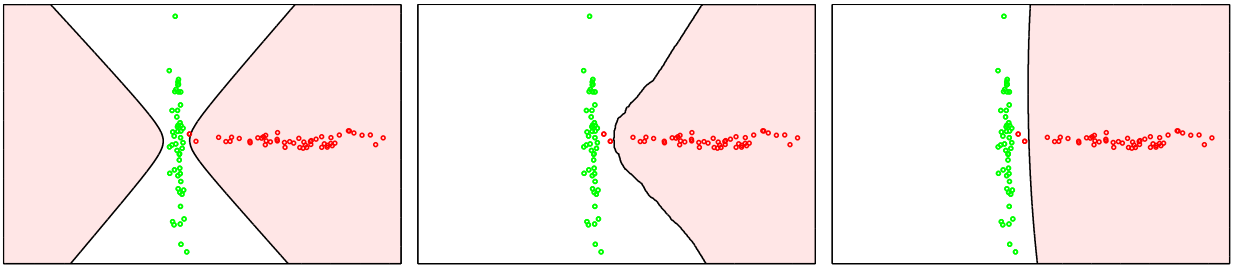
$$\hat{y}(\boldsymbol{x}) = \underset{j=1,\ldots,K}{\operatorname{argmax}} \ln p_j(\boldsymbol{x}) + \ln \pi_j.$$

Proof: [Proof (sketch)] For any fixed k, when the sample size m goes to infinity, the radius of the neighborhood goes to zero. Hence $\boldsymbol{\mu}_j \to \boldsymbol{x}$ and $\Sigma_j \to 0$ and the first term in the asymptotic MICL (25) is the same for all classes. Also the third term D goes to n as ε goes to zero. The only remaining effective term in the classifier is the coding length $L(j)$ for the class label. Since $\frac{|N_j^k(\boldsymbol{x})|}{|N^k(\boldsymbol{x})|} \to \pi_j \cdot p_j(\boldsymbol{x})$ as $k \to \infty$, we have the conclusion of the corollary. ∎

Thus, when the sample size is large or more precisely when the density of samples around the query point is high, the local MICL criterion behaves more like a k-Nearest Neighbor (k-NN) criterion since the effect of the first and third term in (25) diminishes. The criterion, just like the k-NN criterion, approximates the MAP criterion when the sample size goes to infinity and k is large.

However, the finite-sample behavior of the local MICL criterion can be drastically different from that of k-NN, especially when the samples are sparse and the distributions involved are almost degenerate because in those cases, the first and third term in (25) become significant. The first term approximates the local shape of the distribution $p_j(\boldsymbol{x})$ from the handful neighboring samples $N_j^k(\boldsymbol{x})$ by a (regularized) Gaussian;[15] and the third term accounts for the dimension of the subspace spanned by these samples in case $p_j(\boldsymbol{x})$ is close to degenerate around $\boldsymbol{x}$. These two terms together provide a more comprehensive measure of how well the test sample $\boldsymbol{x}$ can be interpolated or extrapolated by its neighboring training samples, in terms of their frequency as well as their shape. As we will demonstrate in the next section with extensive simulations and experiments, the local MICL criterion consistently has superior finite-sample performance over the conventional k-NN criterion.

[15]In the same spirit as using a Gaussian kernel in the Parzen's density estimator [26].



(a) **MICL** (b) k-NN (c) SVM-RBF

Fig. 4. Extrapolation of data structure. Left: MICL. Center: 5-NN. Right SVM-RBF.

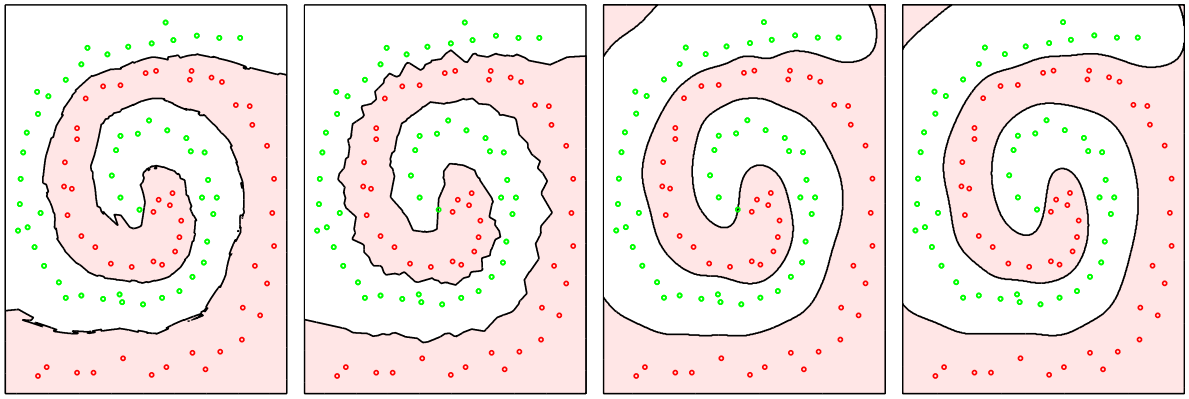
III. Simulations and Experiments

In this Section, we conduct extensive simulations and experiments on real imagery data. Our results show that MICL and its kernel and local variants approach the best reported results from more sophisticated classifiers or systems, *without* using any domain-specific information. In our implementation, the complexity of the global MICL (Algorithm 1) is quadratic in the dimension of the data; the complexity of the local MICL is similar to that of k-NN.

A. Simulations on Synthetic Data

a) Extrapolation of Data Structure.: We compare the decision boundary given by MICL in Figure 1 (right) to that of k-NN and SVM. For MICL we choose $\varepsilon = 1$, for k-NN $k = 5$, and SVM is run with a RBF kernel with $\gamma = \frac{1}{2}$. All three methods give plausible decision boundaries on the right side of the vertical line. However, both k-NN and SVM assign everything on the left side of the vertical line to that line, whereas MICL *extrapolates* the data structure to this side. Note that while MICL is certainly not the only classifier capable of such extrapolation, it does provide a very simple and effective means of harnessing data structure that is ignored by methods such as k-NN and SVM-RBF.

b) Local MICL and Kernel MICL.: Figure 5 compares the nonlinear extensions to MICL discussed in Section III on a two-spiral decision problem. Here we choose $K = 5$, $\varepsilon = 2.5$ for local MICL (LMICL), $k = 5$ for k-NN, an RBF kernel with $\gamma = 1000$ and $\varepsilon = 1$ for kernel MICL (KMICL), and the same kernel for SVM. The local version of MICL exploits the approximately-locally-linear structure of the data to produce a smoother decision boundary than k-NN. Also, notice that both kernel MICL and kernel SVM produce smooth decision boundaries



(a) **LMICL** (b) 5-NN (c) **KMICL** (d) SVM

Fig. 5. Comparison of nonlinear extensions to MICL against SVM and k-NN. Notice that local MICL improves upon k-NN, producing a smoother and more intuitive decision boundary. Kernel MICL and SVM produce similar boundaries, that are smoother and better respect the data structure than those given by either of the local methods.

that extrapolate the spiral structure of the data in the upper left corner. However, the improved performance of these kernelized methods comes at the price of having to select a proper kernel, a non-trivial problem for this dataset, since certain popular kernels (e.g. polynomial) do not work for this dataset.

B. Tests on Real Imagery Data

c) Handwritten Digit Recognition.: We first test the MICL classifier on two standard datasets for handwritten digit recognition (Table I top). The MNIST handwritten digit dataset [15] consists of 60,000 training images and 10,000 test images. We achieved better results using the local version of MICL, due to non-Gaussian distribution of the data. With $k = 20$ and $\varepsilon = 150$, local MICL achieves a test error 1.59%, outperforming simple methods such as k-NN as well as many more complicated neural network approaches (e.g. LeNet-1 [15]). MICL's error rate approaches the best result for a generic learning machine (1.1% error for SVM with a degree-4 polynomial kernel). Problem specific approaches, such as generating synthetic training samples, have resulted in lower error rates, however, with the best reported result achieved using a specially engineered neural network [25].

We also test on the challenging USPS digits database (Table I bottom). Here, even humans have considerable difficulties (about 2.5% error). With $k = 35$ and $\varepsilon = 0.03$, local MICL achieves an

error rate of 4.88%, again outperforming k-NN (best error rate achieved with $k = 4$). We further compare the performance of kernel MICL to SVM[16] on this dataset using the same homogeneous, degree 3 polynomial kernel, and identical preprocessing (normalization and centering). This allows us to compare pure classification performace, independent of the various engineering improvements. Here, SVM achieves a 5.3% error, while kernel-MICL achieves an error rate of 4.7% with distortion $\varepsilon = 0.0067$. This ε was chosen fully automatically, via leave-one-out cross validation within the training set. It is optimal for the range $\log \varepsilon \in \{-10, -9, \ldots, 9, 10\}$.

Using domain-specific information, one can achieve better results. For instance [24] (best reported in [26]) achieves 2.7% error using tangent distance to a large number of prototypes. Other preprocessing steps, for example using many synthetic training images or more advanced skew-correction and normalization techniques have been applied to lower the error rate for SVM-poly to 4.1 % in [26]. While we have avoided extensive preprocessing here, so as to isolate the effect of the classifier, such preprocessing can be readily incorporated into our framework.

Method	Error (%)	Method	Error (%)
LMICL	**1.59**	k-NN	3.09
SVM-Poly [26]	1.1	Best [25]	0.4

Method	Error (%)	Method	Error (%)
LMICL	**4.88**	k-NN	5.28
k-MICL-Poly	**4.7**	SVM-Poly [4]	5.3

TABLE I

RESULTS FOR HANDWRITTEN DIGIT RECOGNITION ON TWO STANDARD DATASETS. TOP: MNIST DATASET. BOTTOM: USPS DATASET. THE RESULTS IN THE RIGHTMOST COLUMN ARE WITH IDENTICAL PREPROCESSING AND KERNEL FUNCTION. KERNEL-MICL OUTPERFORMS SVM ON THIS LEVEL PLAYING FIELD.

d) Face Recognition.: We further verify MICL's appropriateness for vision problems by testing its performance on the Yale Face Database B [8][17], which tests illumination- and pose-sensitivity of face recognition algorithms. The dataset is divided into four subsets, corresponding

[16]For this experiment, we use the LIB-SVM implementation of SVM [4]

[17]We use the normalized and cropped version of this dataset, as in [16].

to different illumination angles. Following [8], [16], we use subsets 1 and 2 for training, and report the average test error across the four subsets. We apply Algorithm 1, *not* the local or kernel version, directly to the raw imagery data, with $\varepsilon = 75$. Table II shows the comparison with popular face recognition techniques such as Eigenfaces. We see that MICL significantly outperforms classical subspace techniques on this problem, with error 0.88% near the best reported results given in [8], [16] that were obtained using a domain-specific model of illumination for face images. We suggest that the source of this improved performance is precisely the regularization induced by lossy coding. In this problem the number of training vectors per class, 19, is very small compared to the dimension, $n = 32,256$.[18] Our simulations (e.g. the lower right corner of Figure 3) show that this is exactly the circumstance in which MICL is superior to MAP and even RDA. Interestingly, this suggests that if we have a criterion that directly exploits the degenerate or low-dimensional structures of the data, performing dimensionality reduction before classifying becomes unnecessary or even undesirable.[19]

Method	Error (%)	Method	Error (%)
MICL	**0.88**	Eigenface [8]	25.8
Subspace [8]	4.6	Best [16]	0

TABLE II

FACE RECOGNITION UNDER WIDELY VARYING ILLUMINATION. MICL OUTPERFORMS A VARIETY OF CLASSICAL FACE RECOGNITION METHODS SUCH AS EIGENFACES ON YALE FACE DATABASE B [8].

IV. CONCLUSION

In this paper, we propose and study a new classification criterion based on the principle of lossy data compression, called the minimum incremental coding length (MICL) criterion. We formally establish its theoretical optimality. It generates a family of classifiers which give us more insights to classical techniques such as MAP, RDA, and k-NN. This family of classifiers

[18]We apply our method to the raw 168×192 images without additional preprocessing.

[19]Working directly in the high-dimensional space is computationally feasible thanks to the kernel property (12), and can be further accelerated via block determinant identites (see Appendix III for details).

extends the working conditions of these classical techniques to situations where the sample set is *sparse* or *degenerate* in a high-dimensional space.

On real vision problems, the MICL criterion and its kernel and local versions perform competitively (nearly optimally for the face recognition problem) *without* utilizing any domain-specific engineering. We believe that its good performance mainly comes from the fact that MICL can automatically exploit low-dimensional structure in high-dimensional imagery data for classification purposes. This ability allows MICL to be applied in practice with little preprocessing and engineering of the data, thus avoiding the risk of over-fitting the data. Due to its simplicity and flexibility, we believe it can be successfully applied to even wider range of real-world data and classification problems.

APPENDIX I
PROOF OF THEOREM 1

In this section, we prove Theorem 1 of Section 2.2. We will require the following two lemmas, the first of which is useful for computing higher order derivatives of the coding length function:

Lemma 4: Let δ_{kl}^{ij} be the matrix whose k,l entry is one and whose other entries are all zero. Let $\Lambda(m) \doteq I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\Sigma$, and $\Psi \doteq (\Lambda(m)+\Gamma)^{-T}$. Then for $k \geq 1$,

$$\frac{\partial^k \ln\det(\Lambda(m)+\Gamma)}{\partial\Gamma_{i_1,j_1}\partial\Gamma_{i_2,j_2}\ldots\partial\Gamma_{i_k,j_k}} = (-1)^{k+1}\left(\sum_{\sigma\in\mathrm{Sym}(k-1)}\Psi\prod_{l=1}^{k-1}\left[\delta^{ij}_{j_{\sigma(l)}i_{\sigma(l)}}\Psi\right]\right)_{i_k j_k}, \tag{14}$$

where $\mathrm{Sym}(p)$ is the symmetric group on p letters. Thus, the k-th partials of $\log_2\det(\Lambda(m)+\Gamma)$ are all $\Theta(1)$ with respect to increasing m.

Proof: Induction on k. For $k=1$, the standard result that $\frac{\partial \ln\det W}{\partial W} = W^{-T}$ gives

$$\frac{\partial\ln\det(\Lambda(m)+\Gamma)}{\partial\Gamma_{i_1,j_1}} = \Big((\Lambda(m)+\Gamma)^{-T}\Big)_{i_1,j_1} = (\Psi)_{i_1,j_1}. \tag{15}$$

Suppose that (14) holds for $1\ldots k-1$. Then

$$\frac{\partial^{k-1}\ln\det(\Lambda(m)+\Gamma)}{\partial\Gamma_{i_1,j_1}\partial\Gamma_{i_2,j_2}\ldots\partial\Gamma_{i_{k-1},j_{k-1}}} = (-1)^{k}\left(\sum_{\sigma\in\mathrm{Sym}(k-2)}\Psi\prod_{l=1}^{k-2}\left[\delta^{ij}_{j_{\sigma(l)}i_{\sigma(l)}}\Psi\right]\right)_{i_{k-1}j_{k-1}} \tag{16}$$

and so the k-th partial is given by

$$(-1)^k\left(\frac{\partial}{\partial\Gamma_{i_k,j_k}}\sum_{\sigma\in\mathrm{Sym}(k-2)}\Psi\delta^{ij}_{j_{\sigma(1)}i_{\sigma(1)}}\Psi\ldots\Psi\delta^{ij}_{j_{\sigma(k-2)}i_{\sigma(k-2)}}\Psi\right)_{i_{k-1}j_{k-1}} =$$

$$(-1)^k \left(\sum_\sigma \quad \frac{\partial \Psi}{\partial \Gamma_{i_k,j_k}} \delta^{ij}_{j_{\sigma(1)} i_{\sigma(1)}} \Psi \ldots \Psi \delta^{ij}_{j_{\sigma(k-2)} i_{\sigma(k-2)}} \Psi \quad + \quad \ldots \right.$$

$$\left. + \quad \Psi \delta^{ij}_{j_{\sigma(1)} i_{\sigma(1)}} \Psi \ldots \Psi \delta^{ij}_{j_{\sigma(k-2)} i_{\sigma(k-2)}} \frac{\partial \Psi}{\partial \Gamma_{i_k,j_k}} \quad \right)_{i_{k-1} j_{k-1}} \tag{17}$$

Notice that $\frac{\partial \Psi}{\partial \Gamma_{i_k,j_k}} = -\Psi \delta^{ij}_{j_k,i_k} \Psi$. Plugging this quantity into (17), changing the order of the partials wrt Γ_{i_k,j_k} and $\Gamma_{i_{k-1},j_{k-1}}$, and recognizing that the sum is now over all permutations of $\{1 \ldots k{-}1\}$ gives the desired formula. ∎

Our main use of this Lemma is to establish that partials of $\ln\det(\Lambda(m)+\Gamma)$ are all $O(1)$.

Now, let $R_\varepsilon(\mathcal{Q}) \doteq \frac{1}{2}\log_2\det(I + \frac{n}{\varepsilon^2}\hat{\Sigma}(\mathcal{Q}))$ denote the *coding rate* associated with a set of samples $\mathcal{Q}$, and let $\delta R_\varepsilon(\mathcal{Q}, \boldsymbol{z}) \doteq R_\varepsilon(\mathcal{Q} \cup \{\boldsymbol{z}\}) - R_\varepsilon(\mathcal{Q})$ denote the change in rate due to introducing a new sample, $\boldsymbol{z}$. The following lemma shows that δR_ε is asymptotically quadratic in $\boldsymbol{z}$:

Lemma 5: Let $\boldsymbol{q}_1 \ldots \boldsymbol{q}_m \ldots \overset{iid}{\sim} p_Q(\boldsymbol{q})$, and let $\mathbb{E}[Q] = \boldsymbol{\mu}$ and $Cov(Q) = \Sigma$. Let $\mathcal{Q}^{(m)} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m] \in \mathbb{R}^{n \times m}$. Then $\forall\, \boldsymbol{z} \in \mathbb{R}^n$,

$$\lim_{m\to\infty} \quad 2m \ln 2 \; \delta R_\varepsilon(\mathcal{Q}^{(m)}, \boldsymbol{z}) \;=\; (\boldsymbol{z}-\boldsymbol{\mu})^T \left(\Sigma + \frac{\varepsilon^2}{n} I\right)^{-1} (\boldsymbol{z}-\boldsymbol{\mu}) \;-\; \mathrm{tr}\big(\Sigma(\Sigma + \frac{\varepsilon^2}{n} I)^{-1}\big) \quad a.s. \tag{18}$$

Proof: Let $\Gamma \doteq \frac{n}{\varepsilon^2}\frac{m}{(m+1)^2}(\boldsymbol{z}-\hat{\boldsymbol{\mu}})(\boldsymbol{z}-\hat{\boldsymbol{\mu}})^T$. Then,

$$\begin{aligned} 2\ln 2 \; \delta R_\varepsilon &= \ln\det\left(I + \frac{n}{\varepsilon^2}\hat{\Sigma}(\mathcal{Q}^{(m)} \cup \{\boldsymbol{z}\})\right) - \ln\det\left(I + \frac{n}{\varepsilon^2}\hat{\Sigma}(\mathcal{Q}^{(m)})\right) \\ &= \ln\det\left(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma}(\mathcal{Q}^{(m)}) + \Gamma\right) - \ln\det\left(I + \frac{n}{\varepsilon^2}\hat{\Sigma}(\mathcal{Q}^{(m)})\right). \end{aligned}$$

Since $\ln\det(\Lambda)$ is analytic in the entries of the matrix Λ, we may Taylor expand the first term in Γ, about $\Gamma = 0$. The above becomes

$$\ln\det\left(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma}\right) + \sum_{i,j}\left[\left(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma}\right)^{-1}\right]_{ij}\Gamma_{ij} + O(m^{-2}) \;-\; \ln\det\left(I + \frac{n}{\varepsilon^2}\hat{\Sigma}\right). \tag{19}$$

Here, we have used that $\frac{\partial \ln\det\Lambda}{\partial \Lambda_{ij}} = (\Lambda^{-T})_{ij}$. The fact that the higher order terms go as m^{-2} follows from Lemma 4. Applying the definition of Γ and rearranging gives

$$\frac{1}{m+1}(\boldsymbol{z}-\hat{\boldsymbol{\mu}})^T\left(\frac{\varepsilon^2}{n}\frac{m+1}{m}I + \hat{\Sigma}\right)^{-1}(\boldsymbol{z}-\hat{\boldsymbol{\mu}}) \;-\; \ln\left[\frac{\det(I + \frac{n}{\varepsilon^2}\hat{\Sigma})}{\det(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma})}\right] \;+\; O(m^{-2}). \tag{20}$$

So, $\lim_{m\to\infty} 2m \ln 2 \; \delta R_\varepsilon(\mathcal{Q}^{(m)}, \boldsymbol{z})$ is equal to

$$\lim_{m\to\infty} \frac{m}{m+1}(\boldsymbol{z}-\hat{\boldsymbol{\mu}})^T \left(\frac{\varepsilon^2}{n}\frac{m+1}{m} I + \hat{\Sigma}(\mathcal{Q}^{(m)})\right)^{-1}(\boldsymbol{z}-\hat{\boldsymbol{\mu}}) - \ln \left[\frac{\det(I + \frac{n}{\varepsilon^2}\hat{\Sigma}(\mathcal{Q}^{(m)})}{\det(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma}(\mathcal{Q}^{(m)}))}\right]^m + O(m^{-1}). \tag{21}$$

The first term goes to $(\boldsymbol{z} - \boldsymbol{\mu})^T \left(\Sigma + \frac{\varepsilon^2}{n} I\right)^{-1} (\boldsymbol{z} - \boldsymbol{\mu})$ almost surely. Let $\hat{\lambda}_1 \ldots \hat{\lambda}_n$ be the eigenvalues of the sample covariance, $\hat{\Sigma}$. Then the limit of the middle term is:

$$\begin{aligned}
\lim_{m\to\infty} \ln \left[\frac{\det(I + \frac{n}{\varepsilon^2}\hat{\Sigma})}{\det(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma})}\right]^m &= \ln \prod_{i=1}^n \lim_{m\to\infty} \left[\frac{1 + \frac{n}{\varepsilon^2}\hat{\lambda}_i}{1 + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\lambda}_i}\right]^m & (22)\\
&= \ln \prod_{i=1}^n \exp\left(\frac{\lambda_i}{\frac{\varepsilon^2}{n} + \lambda_i}\right) & (23)\\
&= \operatorname{tr}\big(\Sigma(\Sigma + \frac{\varepsilon^2}{n} I)^{-1}\big). & (24)
\end{aligned}$$

Here, in (22) we have used that $\lim_{m\to\infty} \left[\frac{\alpha+\beta}{\alpha+\frac{m}{m+1}\beta}\right]^m = \exp(\frac{\beta}{\beta+\alpha})$, in conjunction with the almost sure convergence of the sample eigenvalues $\hat{\lambda}_i$ to the true covariance's eigenvalues λ_i. This establishes the lemma. ∎

Theorem 1, restated below, is a straightforward consequence of this analysis.

Theorem 1 (Asymptotic MICL) *Let the training samples* $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m \overset{iid}{\sim} p_{X,Y}(\boldsymbol{x}, y)$*, with*[20] $\boldsymbol{\mu}_j \doteq \mathbb{E}[X|Y = j]$*,* $\Sigma_j \doteq Cov(X|Y = j)$*. Then as* $m \to \infty$*, the MICL criterion coincides (eventually, with probability one) with the decision rule*

$$\hat{y}(\boldsymbol{x}) = \operatorname*{argmax}_{j=1,\ldots,K} \mathcal{L}_G\Big(\boldsymbol{x} \,\Big|\, \boldsymbol{\mu}_j,\, \Sigma_j + \frac{\varepsilon^2}{n} I\Big) + \ln \pi_j + \frac{1}{2} D_\varepsilon(\Sigma_j), \tag{25}$$

where $\mathcal{L}_G(\cdot|\,\boldsymbol{\mu}, \Sigma)$ *is the log-likelihood function for a* $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ *distribution, and*

$$D_\varepsilon(\Sigma_j) \doteq \operatorname{tr}\left(\Sigma_j\big(\Sigma_j + \frac{\varepsilon^2}{n} I\big)^{-1}\right) \tag{26}$$

is the effective codimension of the j*-th model, relative to* ε*.*

Proof: We first consider the decision boundary between two classes whose means and covariances are $\boldsymbol{\mu}_1, \Sigma_1$ and $\boldsymbol{\mu}_2, \Sigma_2$ respectively. Let $\mathcal{X}^{(m)} \doteq [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{n\times m}$ be the first m training vectors, $\mathcal{X}_j^{(m)} \doteq \{\boldsymbol{x}_i \in \mathcal{X}^{(m)} : y_i = j\}$ the subset of the first m training vectors belonging to the j-th class, and $m_j \doteq |\mathcal{X}_j^{(m)}|$. Let $M_\varepsilon(\mathcal{X}) \doteq \frac{n}{2}\log_2(1 + \frac{\|\hat{\boldsymbol{\mu}}(\mathcal{X})\|^2}{\varepsilon^2})$ be the number of bits

[20] We assume that the first and second moments of the conditional distributions exist.

needed to code the mean, and $\delta M_\varepsilon(\mathcal{X}, \boldsymbol{z})$ the change due to introducing sample $\boldsymbol{z}$. Applying the definition of L_ε and rearranging, we have that $\delta L_\varepsilon(\boldsymbol{z}, 1) < \delta L_\varepsilon(\boldsymbol{z}, 2)$ iff

$$\begin{aligned}(m_1+n)\,\delta R_\varepsilon\Big(\mathcal{X}_1^{(m)}, \boldsymbol{z}\Big) \;+\; R_\varepsilon\Big(\mathcal{X}_1^{(m)} \cup \{\boldsymbol{z}\}\Big) \;+\; \delta M_\varepsilon(\mathcal{X}_1^{(m)}, \boldsymbol{z}) \;-\; \log_2 \hat{\pi}_1 \\ <\quad (m_2+n)\delta R_\varepsilon\Big(\mathcal{X}_2^{(m)}, \boldsymbol{z}\Big) \;+\; R_\varepsilon\Big(\mathcal{X}_2^{(m)} \cup \{\boldsymbol{z}\}\Big) \;+\; \delta M_\varepsilon(\mathcal{X}_2^{(m)}, \boldsymbol{z}) \;-\; \log_2 \hat{\pi}_2, \quad (27)\end{aligned}$$

Now, w.p.1., $\forall\, \boldsymbol{z} \in \mathbb{R}^n$, $R_\varepsilon\Big(\mathcal{X}_j^{(m)} \cup \{\boldsymbol{z}\}\Big) \to R_\varepsilon(\Sigma_j)$, $\delta M_\varepsilon(\mathcal{X}_j^{(m)}, \boldsymbol{z}) \to 0$ and $\hat{\pi}_j \to \pi_j$.
Let us multiply (27) by $\ln 2$ and let $m \to \infty$. Using Lemma 5 to evaluate the limit of the first term, we have that w.p.1., $\hat{y}(\boldsymbol{z}) = 1$ iff

$$\begin{aligned}&\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu}_1)^T \left(\Sigma_1 + \frac{\varepsilon^2}{n} I\right)^{-1} (\boldsymbol{z}-\boldsymbol{\mu}_1) \;-\; \frac{1}{2} D_\varepsilon(\Sigma_1) \;+\; \frac{1}{2}\ln\det(I + \frac{n}{\varepsilon^2}\Sigma_1) \;-\; \ln \pi_1 \\ <\quad &\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu}_2)^T \left(\Sigma_2 + \frac{\varepsilon^2}{n} I\right)^{-1} (\boldsymbol{z}-\boldsymbol{\mu}_2) \;-\; \frac{1}{2} D_\varepsilon(\Sigma_2) \;+\; \frac{1}{2}\ln\det(I + \frac{n}{\varepsilon^2}\Sigma_2) \;-\; \ln \pi_2. \quad (28)\end{aligned}$$

Notice that the first and third terms on each side sum to $-\mathcal{L}_G(\boldsymbol{z}|\boldsymbol{\mu}_j, \Sigma_j + \frac{\varepsilon^2}{n} I)$. Multiplying by -1 converts the minimization to a maximization, and extending to K classes by considering the decision boundaries between each pair of classes establishes the result, (25). ∎

APPENDIX II
PROOF OF THEOREM 2

In this section, we analyze the convergence rate of the MICL discriminant functions to their limiting form (25), proving Theorem 2 of the paper. Throughout this section we consider the discriminant function $\delta L_\varepsilon(\boldsymbol{z}, j)$ associated with a single group with mean $\boldsymbol{\mu}_j$ and covariance Σ_j, and so for compactness of notation we will drop the subscript j. In the course of proving Theorem 1, we showed that the incremental coding length can be written as

$$\begin{aligned}\delta L_\varepsilon(\boldsymbol{z}) \;&=\; (m+n)\,\delta R_\varepsilon(\mathcal{X}, \boldsymbol{z}) \;+\; R_\varepsilon(\mathcal{X} \cup \{\boldsymbol{z}\}) \;+\; \delta M_\varepsilon(\mathcal{X}, \boldsymbol{z}) \;-\; \log_2 \hat{\pi} \quad (29)\\ &=\; \frac{1}{2\ln 2}(\boldsymbol{z}-\hat{\boldsymbol{\mu}})^T \left(\hat{\Sigma} + \frac{\varepsilon^2}{n}\frac{m+1}{m} I\right)^{-1} (\boldsymbol{z}-\hat{\boldsymbol{\mu}}) - \frac{m}{2\ln 2}\ln\left[\frac{\det(I + \frac{n}{\varepsilon^2}\hat{\Sigma})}{\det(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma})}\right] \\ &\quad + R_\varepsilon(\mathcal{X} \cup \{\boldsymbol{z}\}) \;+\; \delta M_\varepsilon(\mathcal{X}, \boldsymbol{z}) \;-\; \log_2 \hat{\pi} \;+\; O(m^{-1}) \quad (30)\end{aligned}$$

with limiting form

$$\delta L_\varepsilon^\infty(\boldsymbol{z}) = \frac{1}{2\ln 2}(\boldsymbol{z}-\boldsymbol{\mu})^T \left(\Sigma + \frac{\varepsilon^2}{n} I\right)^{-1} (\boldsymbol{z}-\boldsymbol{\mu}) \;-\; \frac{D_\varepsilon(\Sigma)}{2\ln 2} \;+\; R_\varepsilon(\Sigma) \;-\; \log_2 \pi. \quad (31)$$

We will need the following deviation bounds on the empirical class probability, $\hat{\pi} = \frac{1}{m}\sum_i I_{y_i=j}$, the sample mean, $\hat{\boldsymbol{\mu}} = \frac{1}{m}\sum_i \boldsymbol{x}_i$ and sample covariance $\hat{\Sigma} = \frac{1}{m-1}\sum_i (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T$.

Lemma 6: Suppose the fourth moment $E[\,\|\boldsymbol{x} - \boldsymbol{\mu}\|^4\,]$ exists. The following three equations then hold simultaneously with probability at least $1 - 3\alpha$:

$$|\hat{\pi} - \pi| \;\leq\; \sqrt{\frac{\pi(1-\pi)}{m\alpha}}, \tag{32}$$

$$\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\| \;\leq\; \sqrt{\frac{\mathrm{tr}\,(\Sigma)}{m\alpha}}, \qquad \text{and} \tag{33}$$

$$\|\hat{\Sigma} - \Sigma\|_F \;\leq\; g(m,\alpha) + o(m^{-\frac{1}{2}}). \tag{34}$$

where

$$g(m,\alpha) \doteq \sqrt{\frac{\mathbb{E}[\|\boldsymbol{z} - \boldsymbol{\mu}\|^4] - \|\Sigma\|_F^2}{m\alpha}} + 2\|\boldsymbol{\mu}\|\sqrt{\frac{\mathrm{tr}\,(\Sigma)}{m\alpha}} \tag{35}$$

and the residual $o(m^{-\frac{1}{2}})$ in (34) is independent of α.

Proof: Notice that $\mathbb{E}[\hat{\pi}] = \pi$ and $var(\hat{\pi}) = \pi(1-\pi)/m$. By Chebyschev's inequality,

$$P\left[|\hat{\pi} - \pi| \geq \sqrt{\frac{\pi(1-\pi)}{m\alpha}}\right] \leq \alpha. \tag{36}$$

Similarly,

$$P[\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_F \geq \eta] \leq \frac{\mathbb{E}[\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2]}{\eta^2} = \frac{\mathrm{tr}\,(cov(\hat{\boldsymbol{\mu}}))}{\eta^2} = \frac{\mathrm{tr}\,(\Sigma)}{m\eta^2}, \tag{37}$$

so that $P\left[\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\| \geq \sqrt{\frac{\mathrm{tr}(\Sigma)}{m\alpha}}\right] \leq \alpha$.

Let $\tilde{\Sigma} \doteq \frac{1}{m}\sum_i (\boldsymbol{x}_i - \mu)(\boldsymbol{x}_i - \mu)^T$. Then

$$\|\hat{\Sigma} - \tilde{\Sigma}\|_F \;=\; \left\|\frac{1}{m-1}\tilde{\Sigma} + \boldsymbol{\mu}(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T + \hat{\boldsymbol{\mu}}(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})^T\right\|_F \tag{38}$$

$$\leq\; \frac{1}{m-1}\|\tilde{\Sigma}\|_F + (\|\boldsymbol{\mu}\| + \|\hat{\boldsymbol{\mu}}\|)\,\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\| \tag{39}$$

$$\leq\; 2\|\boldsymbol{\mu}\|\sqrt{\frac{\mathrm{tr}\,(\Sigma)}{m\alpha}} + o(m^{-\frac{1}{2}}) \tag{40}$$

on the event (33). We will next bound $\|\tilde{\Sigma} - \Sigma\|_F$. Let $\boldsymbol{\xi} \doteq \mathrm{vec}((\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T)$. Then $\mathbb{E}[\boldsymbol{\xi}] = \mathrm{vec}(\Sigma)$ and $cov(\boldsymbol{\xi}) = E[\boldsymbol{\xi}\boldsymbol{\xi}^T] - \mathrm{vec}(\Sigma)\mathrm{vec}(\Sigma)^T$. Then,

$$P[\|\tilde{\Sigma} - \Sigma\|_F \geq \gamma] \;\leq\; \frac{\mathbb{E}[\|\tilde{\Sigma} - \Sigma\|_F^2]}{\gamma^2} = \frac{\mathrm{tr}\left(cov(\mathrm{vec}(\tilde{\Sigma}))\right)}{\gamma^2} \tag{41}$$

$$=\; \frac{\mathbb{E}[\|\boldsymbol{\xi}\|^2] - \|\mathrm{vec}(\Sigma)\|^2}{m\gamma^2} = \frac{\mathbb{E}[\|\boldsymbol{x} - \boldsymbol{\mu}\|^4] - \|\Sigma\|_F^2}{m\gamma^2}. \tag{42}$$

Setting the left hand side of (42) equal to α and solving for the upper bound γ gives

$$P\left[\|\hat{\Sigma}-\Sigma\|_F \geq \sqrt{\frac{\mathbb{E}[\|\boldsymbol{x}-\boldsymbol{\mu}\|^4]-\|\Sigma\|_F^2}{m\alpha}}\right] \leq \alpha. \tag{43}$$

$\|\hat{\Sigma}-\Sigma\|_F \leq \|\tilde{\Sigma}-\Sigma\|_F + \|\hat{\Sigma}-\tilde{\Sigma}\|_F$, so (40) and (43) give (34). Applying a union bound, Equations (32), (33) and (34) hold simultaneously with probability at least $1-3\alpha$. ∎

We will analyze, term by term, the convergence of (30) to (31), proving the following theorem:

Theorem 2 (MICL Convergence Rate) *Suppose the fourth moment $E[\,\|\boldsymbol{x}-\boldsymbol{\mu}\|^4\,]$ exists. As $m\to\infty$, the MICL discriminant functions converge to their asymptotic form at a rate of $m^{-\frac{1}{2}}$. More specifically, with probability at least $1-3\alpha$,*

$$\begin{aligned}|\delta L_\varepsilon(\boldsymbol{z}) - \delta L_\varepsilon^\infty(\boldsymbol{z})| \;\leq\; & \frac{g(m,\alpha)}{2\ln 2}\left(\|\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\|^2 + \|\Psi^{-1}\Sigma\Psi^{-1}\|_F + \sqrt{n}\|\Psi^{-1/2}\|_F^2\right) \\ & + \frac{1}{\ln 2}\sqrt{\frac{\mathrm{tr}\,(\Sigma)}{m\alpha}}\,\|\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\| + \frac{1}{\ln 2}\sqrt{\frac{1-\pi}{m\pi\alpha}} + o(m^{-\frac{1}{2}}).\end{aligned} \tag{44}$$

where $\Psi \doteq \Sigma + \frac{\varepsilon^2}{n}I$, and $g(m,\alpha)$ is defined in (35).

Proof: For compactness of notation, let $\hat{\Psi}(m) \doteq \hat{\Sigma} + \frac{\varepsilon^2}{n}\frac{m+1}{m}I$. Fix $\alpha>0$ and let E be the event that the three conditions in Lemma 6 are satisfied. From Lemma 6, $P[E]\geq 1-3\alpha$.

a) Quadratic term.: We first analyze the difference between the quadratic term in (30) and its limiting form:

$$\left|(\boldsymbol{z}-\hat{\boldsymbol{\mu}})^T\hat{\Psi}(m)^{-1}(\boldsymbol{z}-\hat{\boldsymbol{\mu}}) \;-\; (\boldsymbol{z}-\boldsymbol{\mu})^T\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\right| \tag{45}$$

Writing $\boldsymbol{z}-\hat{\boldsymbol{\mu}} = (\boldsymbol{z}-\boldsymbol{\mu}) + (\boldsymbol{\mu}-\hat{\boldsymbol{\mu}})$ and expanding gives

$$\begin{aligned}(\boldsymbol{z}-\hat{\boldsymbol{\mu}})^T\hat{\Psi}(m)^{-1}(\boldsymbol{z}-\hat{\boldsymbol{\mu}}) \;=\; & (\boldsymbol{z}-\boldsymbol{\mu})^T\hat{\Psi}(m)^{-1}(\boldsymbol{z}-\boldsymbol{\mu}) \\ & + 2(\boldsymbol{z}-\boldsymbol{\mu})^T[\hat{\Psi}(m)^{-1}-\Psi^{-1}+\Psi^{-1}](\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}) + o(m^{-\frac{1}{2}}) \qquad (46)\\ =\; & (\boldsymbol{z}-\boldsymbol{\mu})^T\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu}) + (\boldsymbol{z}-\boldsymbol{\mu})^T\Psi^{-1}(\Sigma-\hat{\Sigma})\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu}) \\ & + 2(\boldsymbol{z}-\boldsymbol{\mu})^T\Psi^{-1}(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}) + o(m^{-\frac{1}{2}}). \qquad (47)\end{aligned}$$

In (46) we have used that $\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}\|^2 = o(m^{-\frac{1}{2}})$, and in (47) that $\hat{\Psi}(m)^{-1} = \Psi^{-1} + \Psi^{-1}(\Sigma-\hat{\Sigma})\Psi^{-1} + o(m^{-\frac{1}{2}})$. On event E, (45) is bounded above by

$$\|\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\|^2\|\Sigma-\hat{\Sigma}\|_F + 2\|\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\|\,\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}\| + o(m^{-\frac{1}{2}}) \tag{48}$$

$$\leq g(m,\alpha)\|\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\|^2 + 2\sqrt{\frac{\mathrm{tr}\,(\Sigma)}{m\alpha}}\|\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\| + o(m^{-\frac{1}{2}}) \tag{49}$$

b) Dimension term.: We next consider the convergence of the dimension term, D_ε:

$$\left| m \ln \left[\frac{\det(I + \frac{n}{\varepsilon^2}\hat{\Sigma})}{\det(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma})} \right] - \mathrm{tr}\big(\Sigma(\Sigma + \frac{\varepsilon^2}{n} I)^{-1}\big) \right| \tag{50}$$

Let $B \doteq \Sigma - \hat{\Sigma}$. Then

$$\ln\det(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma}) - \ln\det(I + \frac{n}{\varepsilon^2}\hat{\Sigma}) \tag{51}$$

$$= \ln\det(\Psi - B - \frac{1}{m+1}\hat{\Sigma}) - \ln\det(\Psi - B) \tag{52}$$

$$= \ln\det(I - \Psi^{-1}(B + \frac{1}{m+1}\hat{\Sigma})) - \ln\det(I - \Psi^{-1}B) \tag{53}$$

$$= \ln\det(I - (I - \Psi^{-1}B)^{-1}\Psi^{-1}\frac{1}{m+1}\hat{\Sigma}) \tag{54}$$

$$= \ln\det(I - (I + \Psi^{-1}B)\Psi^{-1}\frac{1}{m+1}\Sigma + o(m^{-\frac{3}{2}})) \tag{55}$$

$$= \ln\det(I - \Psi^{-1}\frac{1}{m+1}\Sigma) + \ln\det(I - \Psi^{-1}B\Psi^{-1}\frac{1}{m+1}\Sigma + o(m^{-\frac{3}{2}})). \tag{56}$$

where in (55) we have used that $(I - \Psi^{-1}B)^{-1} = I + \Psi^{-1}B + o(m^{-\frac{1}{2}})$.

Let the ζ_i be the eigenvalues of $\Psi^{-1}\Sigma$, and ω_i the eigenvalues of $\Psi^{-1}B\Psi^{-1}\Sigma$. Then,

$$m \ln \left[\frac{\det(I + \frac{n}{\varepsilon^2}\hat{\Sigma})}{\det(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma})} \right] = \ln\prod_{i=1}^{n}(1 - \frac{\zeta_i}{m+1})^m + \ln\prod_{i=1}^{n}(1 - \frac{\omega_i}{m+1})^m \tag{57}$$

$$= \ln\prod_{i=1}^{n} e^{-\zeta_i}(1 + \frac{\zeta_i}{m} + o(m^{-1})) + \ln\prod_{i=1}^{n} e^{-\omega_i}(1 + \frac{\omega_i}{m} + o(m^{-1})) \tag{58}$$

$$= \mathrm{tr}\left(\Psi^{-1}\Sigma\right) + \sum_{i=1}^{n}\ln(1 + \frac{\zeta_i}{m} + o(m^{-1})) + \mathrm{tr}\left(\Psi^{-1}B\Psi^{-1}\Sigma\right) + \sum_{i=1}^{n}\ln(1 + \frac{\omega_i}{m} + o(m^{-1})) \tag{59}$$

$$= \mathrm{tr}\big(\Sigma(\Sigma + \frac{\varepsilon^2}{n} I)^{-1}\big) + \mathrm{tr}\left(\Psi^{-1}\Sigma\Psi^{-1}(\Sigma - \hat{\Sigma})\right) + o(m^{-1}). \tag{60}$$

On E, (50) is bounded above by

$$\left|\mathrm{tr}\left(\Psi^{-1}\Sigma\Psi^{-1}(\Sigma - \hat{\Sigma})\right)\right| + o(m^{-1}) \leq \|\Psi^{-1}\Sigma\Psi^{-1}\|_F \|\Sigma - \hat{\Sigma}\|_F + o(m^{-1}) \tag{61}$$

$$\leq g(m,\alpha)\|\Psi^{-1}\Sigma\Psi^{-1}\|_F + o(m^{-1}). \tag{62}$$

c) Rate, mean and class label.: We now consider the convergence of $R_\varepsilon(\mathcal{X} \cup \{\boldsymbol{z}\})$ to $R_\varepsilon(\Sigma)$. Let $\Gamma \doteq \frac{m}{(m+1)^2}(\boldsymbol{z}-\hat{\boldsymbol{\mu}})(\boldsymbol{z}-\hat{\boldsymbol{\mu}})^T$.

$$
\begin{aligned}
|R_\varepsilon(\mathcal{X} \cup \{\boldsymbol{z}\}) - R_\varepsilon(\Sigma)| &= \left| \frac{1}{2}\log_2 \det(\frac{\varepsilon^2}{n} I + \frac{m}{m+1}\hat{\Sigma} + \Gamma) - \frac{1}{2}\log_2 \det(\frac{\varepsilon^2}{n} I + \Sigma) \right| \\
&= \frac{1}{2}\left| \log_2 \det\left(I + \Psi^{-1/2}\left[(\hat{\Sigma}-\Sigma) - \frac{1}{m+1}\hat{\Sigma} + \Gamma \right] \Psi^{-1/2} \right) \right| \quad (63) \\
&\leq \frac{n}{2}\log_2\left(1 + \frac{1}{\sqrt{n}} \left\| \Psi^{-1/2}\left[(\hat{\Sigma}-\Sigma) - \frac{1}{m+1}\hat{\Sigma} + \Gamma \right] \Psi^{-1/2} \right\|_F \right) \quad (64) \\
&\leq \frac{\sqrt{n}}{2\ln 2}\|\Psi^{-1/2}(\hat{\Sigma}-\Sigma)\Psi^{-1/2}\|_F + o(m^{-\frac{1}{2}}) \quad (65) \\
&\leq \frac{\sqrt{n}}{2\ln 2}\|\Psi^{-1/2}\|_F^2 \|\hat{\Sigma}-\Sigma\|_F + o(m^{-\frac{1}{2}}). \quad (66)
\end{aligned}
$$

In going from (63) to (64), we have used that for symmetric $A \in \mathbb{R}^{n\times n}$ with eigenvalues $\{\lambda_i\}$,

$$
\begin{aligned}
|\det(I+A)| &\leq \prod_i (1+|\lambda_i|) \leq \left(1 + \frac{\sum_i |\lambda_i|}{n} \right)^n \leq \left(1 + \frac{1}{\sqrt{n}} \left(\sum_i \lambda_i^2 \right)^{1/2} \right)^n \quad (67) \\
&= \left(1 + \frac{1}{\sqrt{n}} \operatorname{tr}\left(A^T A\right)^{1/2} \right)^n = \left(1 + \frac{1}{\sqrt{n}} \|A\|_F \right)^n. \quad (68)
\end{aligned}
$$

On E, the first term of (66) is is bounded above by

$$
\frac{\sqrt{n}}{2\ln 2} g(m,\alpha) \|\Psi^{-1/2}\|_F^2. \quad (69)
$$

Next, consider the excess cost to code the sample mean, and let $\nu \doteq \frac{m}{m+1}$, $\bar{\nu} \doteq \frac{1}{m+1}$. Then

$$
\begin{aligned}
|\delta M_\varepsilon(\mathcal{X}, \boldsymbol{z})| &= \left| \frac{n}{2}\log_2(1 + \frac{\|\nu\hat{\boldsymbol{\mu}} + \bar{\nu}\boldsymbol{z}\|^2}{\varepsilon^2}) - \frac{n}{2}\log_2(1 + \frac{\|\hat{\boldsymbol{\mu}}\|^2}{\varepsilon^2}) \right| \quad (70) \\
&\leq \frac{n}{2}\log_2\left(1 + \left| \frac{\|\nu\hat{\boldsymbol{\mu}} + \bar{\nu}\boldsymbol{z}\|^2 - \|\hat{\boldsymbol{\mu}}\|^2}{\varepsilon^2} \right| \right) \quad (71) \\
&= \frac{n}{2}\log_2(1 + O(m^{-1})) \quad (72) \\
&= o(m^{-\frac{1}{2}}). \quad (73)
\end{aligned}
$$

Finally, we consider the convergence of the cost of coding the class label, Y. On E, $|\hat{\pi} - \pi| \leq \sqrt{\frac{\pi(1-\pi)}{m\alpha}}$. Then,

$$
\begin{aligned}
|\log_2\hat{\pi} - \log_2\pi| &= \log_2\left(1 + \frac{|\hat{\pi}-\pi|}{\min(\hat{\pi},\pi)} \right) \leq \log_2\left(1 + \frac{|\hat{\pi}-\pi|}{\pi - |\hat{\pi}-\pi|} \right) \quad (74) \\
&\leq \frac{1}{\ln 2}\frac{\sqrt{1-\pi}}{\sqrt{m\pi\alpha} - \sqrt{1-\pi}} = \frac{1}{\ln 2}\sqrt{\frac{1-\pi}{m\pi\alpha}} + o(m^{-\frac{1}{2}}). \quad (75)
\end{aligned}
$$

Combining (49), (62), (69), (73) and (74) gives the result, (44). ∎

APPENDIX III
EFFICIENT IMPLEMENTATION IN HIGH DIMENSIONAL SPACES

Given training samples $\mathcal{X} \in \mathbb{R}^{n\times m}$, and a test sample $\boldsymbol{z} \in \mathbb{R}^n$, the MICL decision rule requires us to compute the following discriminant function:

$$\delta L_\varepsilon(\boldsymbol{x}, j) = L_\varepsilon(\mathcal{X}_j \cup \{\boldsymbol{x}\}) - L_\varepsilon(\mathcal{X}_j) - \log_2 \pi_j \tag{76}$$

where

$$L_\varepsilon(\mathcal{X}) \doteq \frac{m+n}{2}\log_2 \det\left(I + \frac{n}{\varepsilon^2}\hat{\Sigma}(\mathcal{X})\right) + \frac{n}{2}\log_2\left(1 + \frac{\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}}}{\varepsilon^2}\right) \tag{77}$$

In high dimensional spaces, i.e. when $n \gg m$, it is generally advantageous to work with the kernelized version of the rate function, in which the sample covariance $\hat{\Sigma}$ is replaced by the mean-centered matrix of inner products $\frac{1}{m-1}\Phi_m\mathcal{X}^T\mathcal{X}\Phi_m$, where $\Phi_m \doteq I - \frac{1}{m}\mathbf{1}\mathbf{1}^T$ is the mean-centering matrix. Notice that the second and third terms of (76) can be precomputed offline, during the training stage. However, the first term depends on the new sample, $\boldsymbol{z}$, and requires computing the log-determinant of a $n \times n$ or $m \times m$ matrix. Straightforward numerically stable implementations require $\Theta(m^3)$ time (computing $\log\det$ either via Cholesky decomposition or singular value decomposition). In this section we show how the online computation required to evaluate (76) can be reduced to $\Theta(m^2)$, with a corresponding practical speedup of several orders of magnitude for the datasets considered in this paper.

We will work with the kernelized version of the rate function:

$$R_\varepsilon(\mathcal{X}) = \frac{1}{2}\log_2\det\left(I + \frac{n}{\varepsilon^2}\frac{1}{m-1}\bar{\mathcal{X}}^T\bar{\mathcal{X}}\right) = \frac{1}{2}\log_2\det\left(I + \frac{n}{\varepsilon^2}\frac{1}{m-1}\Phi_m\mathcal{X}^T\mathcal{X}\Phi_m\right), \tag{78}$$

where $\Phi_m \doteq I - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T \in \mathbb{R}^{m\times m}$.

The quantity of interest, then, is the coding rate when test sample $\boldsymbol{z}$ is introduced:

$$R_\varepsilon(\mathcal{X}\cup\{\boldsymbol{z}\}) = \frac{1}{2}\log_2\det\left(I + \frac{n}{\varepsilon^2 m}\Phi_{m+1}\begin{bmatrix} K & \boldsymbol{b} \\ \boldsymbol{b}^T & c\end{bmatrix}\Phi_{m+1}\right). \tag{79}$$

Here $K_{ij} = \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle$, $\boldsymbol{b}_i = \langle \boldsymbol{x}_i, \boldsymbol{z}\rangle$ and $c = \langle \boldsymbol{z}, \boldsymbol{z}\rangle$, where the inner product $\langle\cdot,\cdot\rangle$ can be the standard Euclidean inner product (global MICL), or some nonlinear kernel function (kernel MICL). (79) can be written as

$$\frac{1}{2}\log_2\det\begin{bmatrix} I + Q + \mathbf{1}\boldsymbol{p}^T + \boldsymbol{p}\mathbf{1}^T + \lambda\mathbf{1}\mathbf{1}^T & \boldsymbol{q} \\ \boldsymbol{q}^T & \xi\end{bmatrix}, \tag{80}$$

where, letting $\Upsilon \doteq I_m - \frac{1}{m+1}\mathbf{1}_m\mathbf{1}_m^T$ denote the upper left block of the mean-centering matrix, Φ_{m+1},

$$\begin{aligned} Q &\doteq \frac{n}{\varepsilon^2 m}\Upsilon K \Upsilon, \qquad \boldsymbol{p} \doteq -\frac{n}{\varepsilon^2 m}\frac{1}{m+1}\Upsilon \boldsymbol{b}, \qquad \lambda \doteq \frac{n}{\varepsilon^2 m}\frac{c}{(m+1)^2}, \\ \xi &\doteq 1 + \frac{n}{\varepsilon^2 m}\frac{1}{(m+1)^2}\left(\mathbf{1}^T K \mathbf{1} - 2m\mathbf{1}^T\boldsymbol{b} + cm^2\right) \\ \boldsymbol{q} &\doteq \frac{n}{\varepsilon^2 m}\frac{1}{m+1}\left(-\Upsilon K\mathbf{1} + m\Upsilon\boldsymbol{b} + \frac{\mathbf{1}^T\boldsymbol{b}}{m+1}\mathbf{1} - \frac{mc}{m+1}\mathbf{1}\right). \end{aligned} \tag{81}$$

Here, Q is constant for each class, and can be precomputed during the training phase. Notice that the total time to compute $\boldsymbol{p}, \boldsymbol{q}, \lambda, \xi$ is quadratic in the dimension n.

We will apply the following identities regarding small-rank-adujstments of matrix quantites (the third of which is the Sherman-Woodbury-Morrison matrix inversion lemma):

$$\det\begin{bmatrix} A & \boldsymbol{b} \\ \boldsymbol{b}^T & c \end{bmatrix} = \det(A)(c - \boldsymbol{b}^T A^{-1}\boldsymbol{b}). \tag{82}$$

$$\det(A + BCB^T) = \det(A)\det(C)\det(C^{-1} + B^T A^{-1} B). \tag{83}$$

$$(A + BCB^T)^{-1} = A^{-1} - A^{-1}B(C^{-1} + B^T A^{-1}B)^{-1}B^T A^{-1}. \tag{84}$$

Let $\Gamma \doteq I + Q + \mathbf{1}\boldsymbol{p}^T + \boldsymbol{p}\mathbf{1}^T + \lambda\mathbf{1}\mathbf{1}^T \doteq I + Q + \begin{bmatrix} \mathbf{1} & \boldsymbol{p} \end{bmatrix} \Lambda \begin{bmatrix} \mathbf{1}^T \\ \boldsymbol{p}^T \end{bmatrix}$. The determinant in (80) becomes

$$\begin{aligned} \det\begin{bmatrix} \Gamma & \boldsymbol{q} \\ \boldsymbol{q}^T & \xi \end{bmatrix} &= (\det\Gamma)(\xi - \boldsymbol{q}^T\Gamma^{-1}\boldsymbol{q}) \\ &= \det(I+Q)\det(\Lambda)\det\left(\Lambda^{-1} + \begin{bmatrix} \mathbf{1}^T \\ \boldsymbol{p}^T \end{bmatrix}(I+Q)^{-1}\begin{bmatrix} \mathbf{1} & \boldsymbol{p} \end{bmatrix}\right)(\xi - \boldsymbol{q}^T\Gamma^{-1}\boldsymbol{q}). \end{aligned}$$

Here, the first follows from (82), and the second from (83). $\det(I+Q)$ and $(I+Q)^{-1}$ can be precomputed offline. A straightforward application of (84) gives that

$$\Gamma^{-1} = (I+Q)^{-1} - (I+Q)^{-1}\begin{bmatrix} \mathbf{1} & \boldsymbol{p} \end{bmatrix}\left(\Lambda^{-1} + \begin{bmatrix} \mathbf{1}^T \\ \boldsymbol{p}^T \end{bmatrix}(I+Q)^{-1}\begin{bmatrix} \mathbf{1} & \boldsymbol{p} \end{bmatrix}\right)^{-1}\begin{bmatrix} \mathbf{1}^T \\ \boldsymbol{p}^T \end{bmatrix}(I+Q)^{-1}.$$

Then, for $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^m$, let $s_{\boldsymbol{uv}} \doteq \boldsymbol{u}^T(I+Q)^{-1}\boldsymbol{v}$. We can write the above in terms of quadratic products involving $\mathbf{1}$, $\boldsymbol{q}$ and $\boldsymbol{p}$:

$$\det \begin{bmatrix} \Gamma & \boldsymbol{q} \\ \boldsymbol{q}^T & \xi \end{bmatrix} = \det(I+Q)\, \det(\Lambda)\, \det\left(\Lambda^{-1} + \begin{bmatrix} s_{\mathbf{11}} & s_{\mathbf{1}\boldsymbol{p}} \\ s_{\mathbf{1}\boldsymbol{p}} & s_{\boldsymbol{pp}} \end{bmatrix}\right) \times \left(\xi - s_{\boldsymbol{qq}} + \begin{bmatrix} s_{\boldsymbol{q}\mathbf{1}} \\ s_{\boldsymbol{qp}} \end{bmatrix}^T \left(\Lambda^{-1} + \begin{bmatrix} s_{\mathbf{11}} & s_{\mathbf{1}\boldsymbol{p}} \\ s_{\mathbf{1}\boldsymbol{p}} & s_{\boldsymbol{pp}} \end{bmatrix}\right)^{-1} \begin{bmatrix} s_{\boldsymbol{q}\mathbf{1}} \\ s_{\boldsymbol{qp}} \end{bmatrix}\right) \tag{85}$$

The $s_{\boldsymbol{u}\,\boldsymbol{v}}$ can be computed in quadratic time, and given these the remaining operations are constant time.

APPENDIX IV
IMPLEMENTATION OF KERNEL MICL

We start with the coding length function

$$\begin{aligned} L_\varepsilon(\mathcal{X}) &\doteq \frac{m+n}{2}\log_2\det\left(I + \frac{n}{\varepsilon^2}\hat{\Sigma}(\mathcal{X})\right) + \frac{n}{2}\log_2\left(1 + \frac{\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}}}{\varepsilon^2}\right) && (86)\\ &= \frac{m+n}{2}\log_2\det\left(I + \frac{n}{\varepsilon^2}\frac{1}{m-1}(\mathcal{X} - \hat{\boldsymbol{\mu}}\mathbf{1}^T)(\mathcal{X} - \hat{\boldsymbol{\mu}}\mathbf{1}^T)^T\right) + \frac{n}{2}\log_2\left(1 + \frac{\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}}}{\varepsilon^2}\right) \\ &= \frac{m+n}{2}\log_2\det\left(I + \frac{n}{\varepsilon^2}\frac{1}{m-1}\mathcal{X}\Phi_m\Phi_m^T\mathcal{X}^T\right) + \frac{n}{2}\log_2\left(1 + \frac{\mathbf{1}^T\mathcal{X}^T\mathcal{X}\mathbf{1}}{m^2\varepsilon^2}\right). && (87)\end{aligned}$$

Here, $\Phi_m \doteq I - \frac{1}{m}\mathbf{1}\mathbf{1}^T \in \mathbb{R}^{m\times m}$ is the *mean-centering matrix*. Noticing that the nonzero eigenvalues of $(\mathcal{X}\Phi_m)(\mathcal{X}\Phi_m)^T$ and $(\mathcal{X}\Phi_m)^T(\mathcal{X}\Phi_m)$ are equal, the above is equal to

$$\frac{m+n}{2}\log_2\det\left(I + \frac{n}{\varepsilon^2}\frac{1}{m-1}\Phi_m^T K\Phi_m\right) + \frac{n}{2}\log_2\left(1 + \frac{\mathbf{1}^T K\mathbf{1}}{m^2\varepsilon^2}\right), \tag{88}$$

where $K = \mathcal{X}^T\mathcal{X} \in \mathbb{R}^{m\times m}$ is the kernel matrix, or Grammian: $K_{ij} = \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle$.

As discussed in Section III, when the data $\mathcal{X}$ are nonlinear or non-Gaussian, MICL can still be applied if we know a map $\psi : \mathbb{R}^n \to \mathcal{H}$ such that $\psi(\boldsymbol{x})$ is approximately linear or Gaussian. Suppose we are given such a map from the data space to a Hilbert space $\mathcal{H}$ of finite dimension N, and suppose that we know a kernel function $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \langle \psi(\boldsymbol{x}_1), \psi(\boldsymbol{x}_2)\rangle_{\mathcal{H}}$. Often, $\mathcal{H}$ is very high-dimensional and it is computationally costly to actually compute $\psi(\boldsymbol{x})$. However, since $k(\cdot,\cdot)$ is known, we can still efficiently compute the coding length in the high dimensional space $\mathcal{H}$ by replacing n with N in (88) and replacing $K_{ij} = \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle$ with $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Notice that $\Phi_m K\Phi_m$ still corresponds to the mean-centered matrix of inner products (of the vectors $\psi(\boldsymbol{x}_i)$), and $\frac{1}{m^2}\mathbf{1}^T K\mathbf{1}$ corresponds to the norm-squared of the sample mean of the $\psi(\boldsymbol{x}_i)$.

Example 7 (Homogeneous Polynomial): Setting $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \left(\gamma\, \boldsymbol{x}_1^T \boldsymbol{x}_2\right)^d$ gives the homogeneous polynomial kernel used in Section III-B for handwritten digit recognition. In this case,

$$\psi : \boldsymbol{x} = [x_1, \ldots, x_n] \mapsto \gamma^{d/2} \left[x_1^d,\, \sqrt{d} x_1^{d-1} x_2,\, \ldots, \sqrt{d} x_{n-1} x_n^{d-1},\, x_n^d\right] \in \mathbb{R}^N, \tag{89}$$

where $N = M_n^{[d]} = \binom{n+d-1}{d-1}$.

Example 8 (Radial Basis Function): Another popular choice is

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\left(-\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2}{2\sigma^2}\right). \tag{90}$$

In this case, $\mathcal{H}$ is infinite-dimensional, and (88) is not valid (i.e.the coding length is infinite). However, we can instead consider the renormalized discriminant functions

$$\overline{\delta L}_\varepsilon(\boldsymbol{x}, i) = \frac{2\delta L_\varepsilon(\boldsymbol{x}, i) - n \log_2 n}{n}. \tag{91}$$

For every finite n, $\overline{\delta L}_\varepsilon(\boldsymbol{x}, i)$ gives the same classification as $\delta L_\varepsilon(\boldsymbol{x}, i)$, but as $n \to \infty$,

$$\begin{aligned} \overline{\delta L}_\varepsilon(\boldsymbol{x}, i) \;\; \to \;\; & \log_2 \det{}^+\left(\frac{1}{\varepsilon^2 m} \Phi_{m+1} K' \Phi_{m+1}\right) + \log_2\left(1 + \frac{\mathbf{1}^T K' \mathbf{1}}{\varepsilon^2 (m+1)^2}\right) \\ & - \log_2 \det{}^+\left(\frac{1}{\varepsilon^2 (m-1)} \Phi_m K \Phi_m\right) - \log_2\left(1 + \frac{\mathbf{1}^T K \mathbf{1}}{\varepsilon^2 m^2}\right), \end{aligned} \tag{92}$$

where K and K' are the kernel matrices before and after introducing the test sample $\boldsymbol{x}$ and $\det{}^+(A)$ denotes the product of the positive eigenvalues of $A \succeq 0$. It is interesting to notice that if $\text{rank}(K') = \text{rank}(K) + 1$ for each group,

$$\begin{aligned} \overline{\delta L}_\varepsilon(\boldsymbol{x}, i) + 2\log_2 \varepsilon \;\; \to \;\; & \log_2 \det{}^+\left(\frac{1}{m} \Phi_{m+1} K' \Phi_{m+1}\right) + \log_2\left(1 + \frac{\mathbf{1}^T K' \mathbf{1}}{\varepsilon^2 (m+1)^2}\right) \\ & - \log_2 \det{}^+\left(\frac{1}{(m-1)} \Phi_m K \Phi_m\right) - \log_2\left(1 + \frac{\mathbf{1}^T K \mathbf{1}}{\varepsilon^2 m^2}\right). \end{aligned} \tag{93}$$

The "covariance" portion of the discriminant function becomes independent of the choice of distortion! Only the cost of encoding the $\hat{\boldsymbol{\mu}}$ still depends on ε.

REFERENCES

[1] A. Barron, J. Rissanen, and B. Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, 1998.

[2] P. Bickel and E. Levina. Regularized estimation of large covariance matrices. *to appear in Annals of Statistics*, 2007.

[3] P. Bickel and B. Li. Regularization in statistics. *TEST*, 15(2):271–344, 2006.

[4] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[5] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications, 1991.

[6] B. Dasarathy. *Nearest Neighbor Pattern Classification Techniques*. IEEE Computer Society Press, 1991.

[7] J. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84:165–175, 1989.

[8] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *PAMI*, 23(6):643–660, 2001.

[9] P. Grunwald and J. Langford. Suboptimal behaviour of Bayes and MDL in classification under misspecification. In *Proceedings of Conference on Learning Theory*, 2004.

[10] J. Hamkins and K. Zeger. Gaussian source coding with spherical codes. *IEEE Transactions on Information Theory*, 48(11):2980–2989, 2002.

[11] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[12] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *Proceedings of CVPR*, 2003.

[13] A. Holub, M. Welling, and P. Perona. Combining generative models and fisher kernels for object recognition. In *ICCV*, pages 136–143, 2005.

[14] I. Johnstone and A. Lu. Sparse principal component analysis. *preprint,* `http://www-stat.stanford.edu/~imj/WEBLIST/AsYetUnpub/sparse.pdf`, 2006.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[16] K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *PAMI*, 27(5):684–698, 2005.

[17] J. Li. A source coding approach to classification by vector quantization and the principle of minimum description length. In *IEEE DCC*, pages 382–391, 2002.

[18] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[19] D. MacKay. Developments in probabilistic modelling with neural networks – ensemble learning. In *Proc. 3rd Annual Symposium on Neural Networks*, pages 191–198, 1995.

[20] M. Madiman, M. Harrison, and I. Kontoyiannis. Minimum description length vs. maximum likelihood in lossy data compression. In *IEEE International Symposium on Information Theory*, 2004.

[21] T. Minka. Inferring a gaussian distribution. *MIT Media Lab Note*, 1998.

[22] S.A. Nene and S.K. Nayar. A Simple Algorithm for Nearest Neighbour Search in High Dimensions. *PAMI*, 19(9):989–1003, 1997.

[23] J.J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[24] P. Simard, Y. LeCun, and J. Denker. Efficient pattern recognition using a new transformation distance. In *Proceedings of NIPS*, volume 5, 1993.

[25] P. Simard, D. Steinkraus, and J. Platt. Best practice for convolutional neural networks applied to visual document analysis. In *ICDAR*, pages 958–962, 2003.

[26] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2000.

[27] X. Wang and X. Tang. A unified framework for subspace face recognition. *PAMI*, 26:1222–1228, 2004.