CLASSIFICATION VIA MINIMUM INCREMENTAL CODING LENGTH (MICL)*

JOHN WRIGHT†, YI MA† AND YANGYU TAO‡ ZHOUCHEN LIN‡, HEUNG-YEUNG SHUM‡

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 demonstrate the asymptotic optimality of this criterion for Gaussian distributions and analyze its relationships to classical classifiers. The theoretical results clarify the connections between our approach and popular classifiers such as MAP, RDA, k-NN, and SVM, as well as unsupervised methods based on lossy coding. 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/coding/home.htm`.

Key words. Classification, Lossy Data Coding, Regularization, MAP, RDA.

1. Introduction.

One quintessential problem in statistical learning [15, 32] 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):

$$g^* = \arg\min \mathbb{E}[I_{g(X)\neq Y}], \tag{1.1}$$

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, the *maximum a posterior* (MAP) assignment

$$\hat{y}(\boldsymbol{x}) = \arg\max_{y\in\{1,\ldots,K\}} p_{X|Y}(\boldsymbol{x}|y)\, p_Y(y) \tag{1.2}$$

gives the optimal classifier.

1.1. 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 nonsingular and the samples are sufficiently dense. However, these assumptions fail in many classification problems that are vital for applications in computer vision [17, 20, 21, 33]. 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 [16]. As a result, the associated distributions are singular or nearly singular. 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 [32]. Furthermore, in the case of singular distributions, the classical likelihood function (1.2) does not have a well-defined maximum [32]. Thus, to infer

*This work was partially supported by NSF CAREER IIS-0347456, NSF CRS-EHS-0509151, NSF CCF-TF-0514955, and ONR YIP N00014-05-1-0633. A preliminary version of this work has appeared in Neural Information Processing Systems (NIPS) 2007.

†Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign, Illinois 61801 ({`jnwright,yima`}`@uiuc.edu`).

‡Microsoft Research in Asia, Beijing, China, 100084 ({`f-yyutao,zhoulin,hshum`}`@microsoft.com`).

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 [5]. However, even if the distributions are assumed to be generic Gaussians, explicit regularization is still necessary to achieve good small-sample performance [11]. 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 [4], as are estimators of subspace structure such as principal components [18].

In many real problems in computer vision, the distributions associated with different classes of data have different intrinsic complexity, lying on subspaces or manifolds of different dimension. 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 submanifold in a cloud of essentially random features from the background. Model selection criteria such as the *minimum description length* (MDL) [22, 28] serve as important modifications to MAP for estimating a model across classes of different complexity. MDL selects the model 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]

1.2. 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 (1.2). Notice that the MAP classifier (1.2) is equivalent to

$$\hat{y}(\boldsymbol{x}) = \arg\min_{y \in \{1,\ldots,K\}} -\ln p_{X|Y}(\boldsymbol{x}|y) - \ln p_Y(y). \tag{1.3}$$

This gives the MAP classifier another interpretation. The optimal classifier should minimize Shannon's optimal (lossless) coding length of the test data $\boldsymbol{x}$ with respect to the distribution of the true class, together with the class assignment: 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 singular) 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 the "*minimum incremental coding length*" *(MICL)* criterion 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 [25]. We will further see that the lossy

[1] Whereas model estimation involves inferring a model from the training data, classification involves inferring a decision about a new test sample given the models.

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

1.3. Contributions of this Paper. The main contribution of this paper is to introduce a new approach to classification based on lossy data compression. We thoroughly analyze its performance in the Gaussian case, and demonstrate its optimality. We then extend it to arbitrary data distributions via local and kernel implementations. The theoretical results clarify the relationship between this new approach and popular classifiers such as MAP, Regularized Discriminant Analysis (RDA) [11], k-Nearest Neighbor (k-NN) [10, 27], and Support Vector Machine (SVM) [8, 32], and also provide a new interpretation to (unsupervised) clustering methods based on lossy coding [23]. The proposed MICL classifier, though very simple, performs competitively compared to conventional classifiers, under a wide range of conditions. 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.

1.4. Organization of this Paper. This paper is organized as follows: in Section 2, we introduce the general MICL criterion, and discuss how it can be applied to unimodal or Gaussian distributions. Section 3 contains the main theoretical results of the paper, analyzing the large-sample behavior of MICL. Section 4 discusses local and kernel implementations that are valid for arbitrary data distributions. Finally, In Section 5 we perform numerous simulations and experiments to verify the properties of the algorithm and demonstrate its performance on real imagery data. Additional mathematical and implementation details are given in the appendix.

We delay a more thorough discussion of the relationship between MICL and existing classifiers until after we have formally introduced our approach. We will discuss its relationship to, and advantages over, a number of popular techniques in machine learning, including other MDL/MAP variants (Section 2.5), RDA (Section 2.5), k-NN (Section 4.2), and SVM (Section 4.1). For a more complete review of the vast literature on supervised learning, we refer the reader to [6, 15, 32].

2. Basic Ideas and Algorithm.

2.1. Minimum Incremental Coding Length. We formulate the problem of classification from the perspective of lossy data coding and compression [9]. 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 up to 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

$$\text{Length}\big(\mathcal{X}, \mathcal{Y}\big) = \sum_{j=1}^{K} L_\varepsilon(\mathcal{X}_j) - |\mathcal{X}_j| \log_2 p_Y(j) \text{ (bits)}. \tag{2.1}$$

Here, the second term is the number of bits needed to (losslessly) code the class labels y_i using the optimal scheme for the empirical distribution of y.

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:

$$\delta L_\varepsilon(\boldsymbol{x}, j) = L_\varepsilon(\mathcal{X}_j \cup \{\boldsymbol{x}\}) - L_\varepsilon(\mathcal{X}_j) + L(j). \tag{2.2}$$

Here, the first two terms measure the excess bits needed to code $(\boldsymbol{x}, \mathcal{X}_j)$ up to 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 (1.3). This interpretation naturally leads to the following classification criterion:

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:*

$$\hat{y}(\boldsymbol{x}) \doteq \arg\min_{j=1,\ldots,K} \delta L_\varepsilon(\boldsymbol{x}, j). \tag{2.3}$$

The above criterion (2.3) 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. For example, if the data distribution is known a-priori, the optimal coding length is given by the its rate-distortion [9]; Or if we consider the data as a discrete set of points, the coding length should be approximately[2] minimal among all possible coding schemes subject to the given distortion. In either case, however, for classification purposes we only require that L_ε correspond *in principle* to some coding scheme; we do not need to explicitly encode the data.[3]

2.2. Lossy Coding Length of Gaussian Data. We will first consider a coding length function L_ε, derived in [23], that is approximately (asymptotically) optimal for Gaussian distributions. The implicit use of a coding scheme which is optimal for Gaussian sources is equivalent to assuming that the conditional class distributions $p_{X|Y}$ are unimodal, and can be well-approximated by Gaussians. 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 4 we will show how using the same L_ε function, MICL can be extended to arbitrary, multimodal distributions via an effective local Gaussian approximation.

For a multivariate Gaussian source $\mathcal{N}(\mathbf{0}, \Sigma)$, the average number of bits needed to code a vector subject to a distortion ε^2 is approximately:

$$R_\varepsilon(\Sigma) \doteq \frac{1}{2} \log_2 \det\left(I + \frac{n}{\varepsilon^2}\Sigma\right) \quad \text{(bits/vector)}. \tag{2.4}$$

This approximation can be motivated from the perspective of sphere packing, as the ratio of the volume of an equi-probability ellipsoid defined by Σ to that of an n-dimensional ε-ball. It can also be viewed as a a upper bound for Gaussian rate-distortion that is valid for all ε [9].[4]

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 their deviations about the mean up to 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. The number of bits required to encode the m differences $\boldsymbol{x}_1 - \hat{\boldsymbol{\mu}}, \ldots, \boldsymbol{x}_m - \hat{\boldsymbol{\mu}}$ is therefore $mR_\varepsilon(\hat{\Sigma})$. However,

[2] 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 [9].

[3] Constructing optimal coding schemes that achieve the lower bound given by the rate-distortion is a difficult problem even in the Gaussian case (see e.g. [14]).

[4] Strictly speaking, the rate-distortion function for a $\mathcal{N}(0, \Sigma)$ source is $R = \frac{1}{2}\log_2 \det(\frac{n}{\varepsilon^2}\Sigma)$ when $\frac{\varepsilon^2}{n}$ is smaller than the smallest eigenvalue of Σ. The above approximation is tightest when the distortion ε is relatively small. When $\frac{\varepsilon^2}{n}$ is larger than some of the eigenvalues of Σ, the rate distortion function becomes more complicated [9]. Nevertheless, the approximate formula $R = \frac{1}{2}\log_2 \det(I + \frac{n}{\varepsilon^2}\Sigma)$ can be viewed as the rate distortion of a "regularized" source, that works for all $\varepsilon > 0$. More details on this approximation can be found in [23].

the optimal encoder/decoder pair requires prior knowledge of $\hat{\Sigma}$, i.e., its principal axes. These principal axes can be specified using an additional $nR_\varepsilon(\hat{\Sigma})$ bits. Finally, the expected number of bits required to encode the sample mean $\hat{\boldsymbol{\mu}}$ can be bounded by $\frac{n}{2}\log_2\left(1+\frac{\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}}}{\varepsilon^2}\right)$. This bound was derived in [23] starting from the assumption that, on average, the number of bits required to encode $t \in \mathbb{R}$ up to distortion ε^2 is $\frac{1}{2}\log_2(1+t^2/\varepsilon^2)$. Since this is again an upper bound to the (scalar) Gaussian rate-distortion, the bound on the number of bits needed to encode the mean is tightest when $\hat{\boldsymbol{\mu}}$ is Gaussian, but remains valid for general $\hat{\boldsymbol{\mu}}$. Combining these quantities, the total number of bits required to encode $\mathcal{X}$ becomes:

$$L_\varepsilon(\mathcal{X}) \doteq \frac{m+n}{2}\log_2\det\left(I+\frac{n}{\varepsilon^2}\hat{\Sigma}(\mathcal{X})\right)+\frac{n}{2}\log_2\left(1+\frac{\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}}}{\varepsilon^2}\right). \tag{2.5}$$

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 bounds the expected number of bits needed to code finitely many samples lying on a linear subspace. The proof, given in [23], suggests a coding scheme in which the principal axes of the distribution and the coordinates with respect to those axes are encoded separately, so that the resulting distortion is less than ε^2.

2.3. 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 (2.2) depends on one's prior assumptions about the nature of the 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):

$$L(j) = -\log_2\pi_j. \tag{2.6}$$

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 (1.2), the choice of π_j effectively gives a prior on class labels.

2.4. The Overall Algorithm. Given the coding length function (2.5) for the observations and the coding length (2.6) for the class label, we summarize the MICL criterion (2.3) as Algorithm 1 below.

2.5. Relationship to Existing Classifiers. Although MICL and MDL [28] both operate by minimizing a coding-theoretic objective, MICL differs significantly from traditional MDL approaches to classification such as those examined in [13]. Those methods choose an optimal *decision boundary* from an allowable set by minimizing the following coding length:

$$g^* = \arg\min_{g\in\mathcal{G}} L(g) + \log_2\binom{m}{\sum_i I_{g(\boldsymbol{x}_i)\neq y_i}}, \tag{2.7}$$

where $L(g)$ is the number of bits needed to code the classifying boundary g within certain class $\mathcal{G}$, and the second term[5] counts the cost of coding training samples misclassified by g. This approach has been proven inconsistent in [13]. In contrast, MICL uses coding length *directly* as a measure of how well the training data represent the new sample. The inconsistency result of [13] does not apply in this modified context. In fact, MICL has more in common

[5]This is the logarithm of the number of subsets of size m_{miss} out of a set of m elements, where m_{miss} is the number of misclassifications.

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

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. In the next section we will see that the MICL criterion leads to a family of classifiers, parameterized by the distortion ε, that generalize the conventional MAP classifier (1.2).

In the Gaussian case, we will see that lossy coding leads to a regularization effect similar to Friedman's Regularized Discriminant Analysis (RDA) [11], which replaces the sample covariance $\hat{\Sigma}$ with a regularized version[6] $\hat{\Sigma} + \alpha I$ in the likelihood function. The main motivation for RDA is improved small-sample performance, and MICL exhibits similar gains in finite sample performance with respect to MAP. We will, however, see a significant difference between MICL and RDA in how they handle classes of varying intrinsic dimension.

The fully Bayesian approach to model estimation, in which posterior distributions over model parameters are estimated also claims finite sample gains over ML/MAP [24,26]. However, when the number of samples is smaller than the number of free parameters in the model (as for high-dimensional data), the result becomes strongly dependent on the choice of prior[7]. MICL does not require the number of samples to be larger than the ambient dimension, and in fact sees its greatest advantage when the sample size is small. As we will see, MICL is asymptotically equivalent to the Bayesian approach, since it too converges to ML/MAP.

3. Analysis of MICL with L_ε for Gaussian Data. We begin this section with a motivating example. Figure 3.1 shows the performance of the MICL classifier on two toy problems in $\mathbb{R}^2$. In both cases, the MICL criterion classifies observations in sparsely sampled regions based on the covariance structure of the data. In the left example, the criterion *interpolates* the data structure 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. On the other hand, k-NN and support vector machine (SVM) do not extrapolate the linear structure of the data (see Figure 5.1 for a comparison). The reader may notice, however, that the MICL 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

[6] Throughout this paper, we only consider the version of RDA which regularizes the covariance by a multiple of the identity. Regularizing by the pooled data covariance as in [11] is less appropriate if we wish to consider groups with significantly different and anisotropic covariances.

[7] In the Gaussian case, when the number of samples is smaller than the dimension of the space, Jeffreys' prior no longer suffices, and stronger assumptions on the parameters of the distribution are required to regularize the problem.

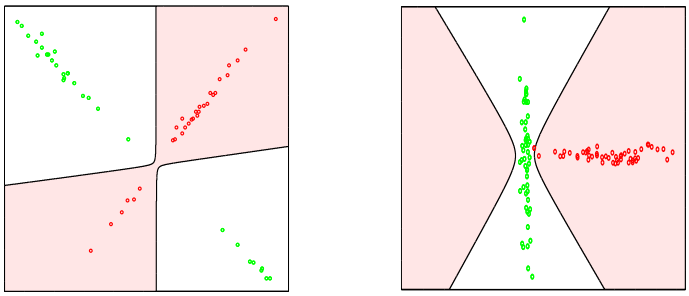


FIG. 3.1. *MICL harnesses the covariance structure of the data to* interpolate *(left) and* extrapolate *(right) in regions where the training samples are sparse.*

is MICL superior?

3.1. Asymptotic Behavior and Relationship to MAP. In this section, we analyze the asymptotic behavior of the MICL criterion (2.3) using coding length function (2.5), 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, whose proof we delay to Appendix A:

THEOREM 3.1 (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*

$$\hat{y}(\boldsymbol{x}) = \underset{j=1,\ldots,K}{\operatorname{argmax}} \mathcal{L}_G\Big(\boldsymbol{x} \,\Big|\, \boldsymbol{\mu}_j,\, \Sigma_j + \frac{\varepsilon^2}{n} I\Big) + \ln \pi_j + \frac{1}{2} D_\varepsilon(\Sigma_j), \tag{3.1}$$

where $\mathcal{L}_G(\cdot|\,\boldsymbol{\mu}, \Sigma)$ *is the log-likelihood function for a* $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ *distribution*[9], *and* $D_\varepsilon(\Sigma_j) \doteq \operatorname{tr}\Big(\Sigma_j(\Sigma_j + \frac{\varepsilon^2}{n} I)^{-1}\Big)$ *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 nonsingular (i.e. their covariance matrices Σ_j are nonsingular), then $\lim_{\varepsilon\to 0}\Big(\Sigma_j + \frac{\varepsilon^2}{n} I\Big) = \Sigma_j$, and $\lim_{\varepsilon\to 0} D_\varepsilon(\Sigma_j) = n$, a constant across the various classes. Thus, for nonsingular data, (the closure of) the family of asymptotic decision boundaries induced by MICL contains the conventional MAP classifier (1.2) 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 $m \to \infty$. As long as $\varepsilon(m)$ does not decrease too quickly, the limiting behavior in (3.1) dominates, and $\hat{y}(\boldsymbol{x})$ converges to the asymptotically optimal MAP criterion. By examining the residuals in the proof of Theorem 3.1 (especially the $O(m^{-1})$ term in Equation (A.8) of the appendix), one can show that if $\varepsilon(m) = \omega(m^{-1/4})$, $\hat{y}(\boldsymbol{x}) \to \arg\max_j \mathcal{L}_G(\boldsymbol{x}|\boldsymbol{\mu}_j, \Sigma_j) + \ln \pi_j$.

Simulations (e.g. Figure 3.1) suggest that the limiting behavior does provide useful information about the performance of the classifier on finite training data. Yet Theorem 3.1

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

[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 (2.5) to complicated multimodal distributions will often result in poor classification performance, and is therefore not advisable. Section 4 discusses how MICL can be modified to handle arbitrary data distributions.

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 B 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 3.2 (MICL Convergence Rate). *As the number of samples,* $m \to \infty$*, the MICL criterion* (2.3) *converges to its asymptotic form,* (3.1) *at a rate of* $m^{-\frac{1}{2}}$*. More specifically*[10]*, with probability at least* $1 - \alpha$*,* $\left|\delta L_\varepsilon(\boldsymbol{x}, j) - \delta L_\varepsilon^\infty(\boldsymbol{x}, j)\right| \leq c(\alpha) \cdot m^{-\frac{1}{2}}$ *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 nearly singular.

3.2. Improvements over MAP. In the above, we have established that asymptotically, the MICL criterion (3.1) is equivalent to the MAP criterion. Nevertheless, in the cases of finite samples or singular distributions, the MICL criterion makes several important modifications to the MAP criterion, which may significantly improve its performance.

3.2.1. Regularization and Finite-Sample Behavior. Notice that the first two terms of the asymptotic MICL criterion (3.1) 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) singular. Even for nonsingular 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 [11], leading to lower misclassification rates.

Figure 3.2 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 3.2 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.

3.2.2. Dimension Reward. The effective dimension term $D_\varepsilon(\Sigma_j)$ in the asymptotic MICL criterion (3.1) 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 very close to 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 [15]. Thus, minimizing the MICL criterion rewards distributions

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

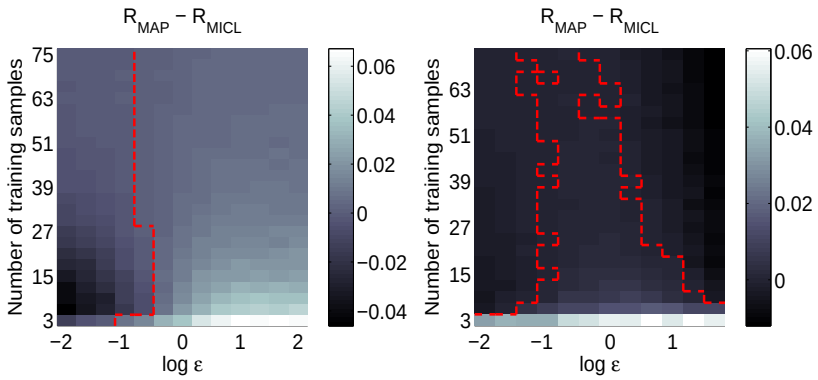


FIG. 3.2. *Excess misclassification risk incurred by using MAP rather than MICL, as a function of ε and m. MICL outperforms MAP in most settings, with the largest gain when m is relatively small. Left: two isotropic Gaussians in $\mathbb{R}^2$. Right: anisotropic Gaussians in $\mathbb{R}^2$.*

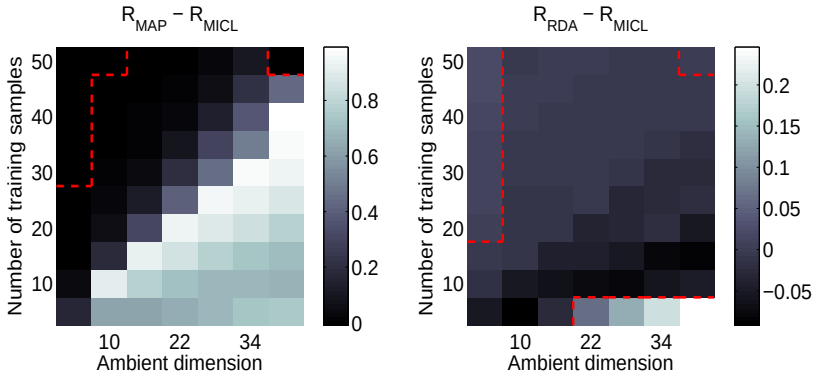


FIG. 3.3. *Excess risk incurred by using MAP and RDA rather than MICL, as a function of number of samples m and dimension n.*

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 3.3 empirically compares MICL to the conventional MAP and the *regularized* MAP (or RDA [11]). 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. For a rank-d distribution, we sample data iid $\mathcal{N}\left(0, \begin{bmatrix} I_{d\times d} & 0 \\ 0 & I_{n-d\times n-d} \end{bmatrix}\right)$, and add iid $\mathcal{N}(0, .04)$ noise to each sample, to simulate real data that may be nearly degenerate, but are not perfectly so. 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 pro-

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

nounced when n is large and m is small. Interestingly, when m is much smaller than n (e.g. the bottom row of Figure 3.3 right), MICL demonstrates a significant performance gain with respect to RDA. As the number of samples increases, though, there is a region where RDA is slightly better. However, for most (m, n) considered here, MICL and RDA have rather close performance.[12]

4. Implementation Issues. The rigorous analysis of the Gaussian case in the previous section reveals many good properties of the proposed MICL criterion. In reality, however, the distribution(s) of the data of interest may not be Gaussian. If the rate-distortion function of the distribution(s) is known, in principle, one could 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 discussed above.

4.1. 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

$$\log_2 \det\big(I+\frac{n}{\varepsilon^2 m}\mathcal{X}\mathcal{X}^T\big) = \log_2 \det\big(I+\frac{n}{\varepsilon^2 m}\mathcal{X}^T\mathcal{X}\big). \tag{4.1}$$

Thus, one can evaluate the coding length function (2.5) using only the inner products between the data points. 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 $\psi(\boldsymbol{x})$ 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]. By choosing a proper kernel function, one may achieve better classification performance for certain classes of non-Gaussian distributions. In practice, some popular choices 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 D 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 Support Vector Machines (SVM) [8, 32] to nonlinear decision boundaries. In fact, SVM can be loosely interpreted as a lossy compression approach to classification, since it represents the final decision hypersurface is represented in terms of a small portion of nearby samples, called "support vectors." However, for degenerate data lying on low-dimensional subspaces or submanifolds, almost all the training 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 5.1 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 5.2 for an example). In Section 5.2 we will see that even for real data whose statistical nature is unclear, kernel MICL outperforms SVM when applied with the same kernel function.

[12] Note that RDA [11] is designed to be nearly optimal for finite samples of Gaussians.

[13] The necessary and sufficient conditions for $k(\cdot,\cdot)$ to be a kernel function are given by Mercer's Theorem [32].

4.2. Local MICL Criterion. For data drawn from complicated multimodal distributions, it may be difficult or impossible to find a kernel function that renders the data approximately Gaussian. 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 $N_j^k(\boldsymbol{x}) \doteq \mathcal{X}_j \cap N^k(\boldsymbol{x}), j = 1, \ldots, K$. In the MICL classifier (Algorithm 1), we can replace the incremental coding length $\delta L_\varepsilon(\boldsymbol{x}, j)$ by its local version:

$$\delta L_\varepsilon(\boldsymbol{x}, j) = L_\varepsilon(N_j^k(\boldsymbol{x}) \cup \{\boldsymbol{x}\}) - L_\varepsilon(N_j^k(\boldsymbol{x})) + L(j), \tag{4.2}$$

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

The local MICL criterion gives a universal classifier that is applicable to arbitrary distributions:

PROPOSITION 4.1 (Asymptotic Local MICL). *Suppose the probability density function $p_j(\boldsymbol{x}) = p(\boldsymbol{x}|y = j)$ of each class is nonsingular. Then if $\varepsilon > 0$ is held constant while $m, k \to \infty$ with $k(m) = o(m)$, 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)] With $k = o(m)$, for any fixed $\boldsymbol{x}$ the radius of $N^k(\boldsymbol{x})$ shrinks to zero. Hence $\hat{\boldsymbol{\mu}}_j \to \boldsymbol{x}$ and $\hat{\Sigma}_j \to 0$. Hence, $\left|L_\varepsilon(N_j^k(\boldsymbol{x}) \cup \{\boldsymbol{x}\}) - L_\varepsilon(N_j^k(\boldsymbol{x}))\right| \to 0$ for each j. 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 desired conclusion. □

Thus, when the sample size is large, or more precisely when the density of samples around the query point is high, local MICL behaves like k-Nearest Neighbor (k-NN), since the effect of the first and third term in (3.1) diminishes. Similar to k-NN, local MICL approximates the MAP criterion when the sample size goes to infinity and k is large.

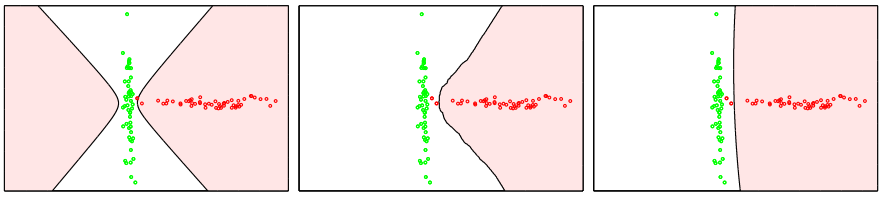
However, the finite-sample behavior of the local MICL criterion can be dramatically different from that of k-NN, especially when the samples are sparse and the distributions involved are almost singular. In those cases, the first and third term in (3.1) become significant. The first term approximates 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 singular 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 shape as well as their frequency. 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.

5. 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 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.

5.1. Simulations on Synthetic Data.

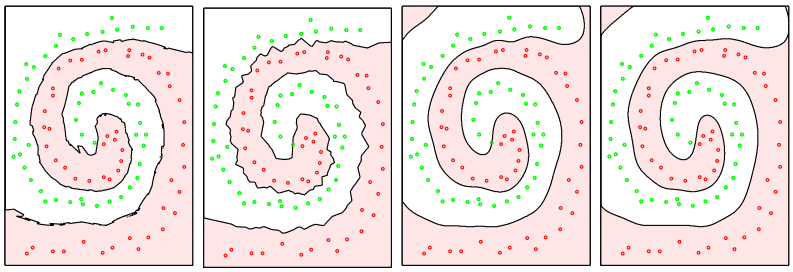
[14] In terms of the Euclidean distance.

[15] In the same spirit as using a Gaussian kernel in Parzen's density estimator [32].



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

FIG. 5.1. *Extrapolation of data structure. Left: MICL. Center: 5-NN. Right SVM-RBF.*



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

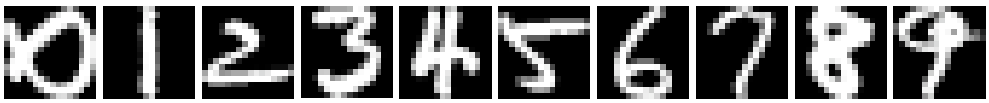
FIG. 5.2. *Comparison of nonlinear extensions to MICL against SVM and k-NN. Notice that local MICL improves upon k-NN, producing a smoother and more intuitive decision boundary. Kernel MICL and SVM produce similar boundaries, that are smoother and better respect the data structure than those given by either of the local methods.*

Extrapolation of Data Structure. We compare the decision boundary given by MICL in Figure 3.1 (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.

Local MICL and Kernel MICL. Figure 5.2 compares the nonlinear extensions to MICL discussed in Section 4 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 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.

5.2. Tests on Real Imagery Data. Real imagery data encountered in applications of learning and vision are often characterized by complicated distributions that may not satisfy the Gaussian assumption underlying MICL. In fact, this difficulty in characterizing the distribution of imagery data has played a major role in the popularity of flexible, empirical classifiers such as k-NN and SVM for vision tasks. In this section, we compare MICL to other generic classifiers, and demonstrate the applicability and advantages of MICL even in

Modified National Institute of Standards and Technology (MNIST) Database



United States Postal Service (USPS) Database

FIG. 5.3. *Digit databases. The MNIST database (top) consists of* $70,000$ *images of resolution* 28×28*. The USPS database (bottom) consists of* $9,298$ *images of resolution* 16×16*. A randomly chosen example of each digit is displayed.*

this nonparametric setting.

Handwritten Digit Recognition. We first test the MICL classifier on two standard datasets for handwritten digit recognition (Table 5.1 top). The MNIST handwritten digit dataset [20] consists of 60,000 training images and 10,000 test images (see Figure 5.3 top for a visualization). We achieved better results using the local version of MICL, due to non-Gaussian distribution of the data. We select the free parameters k and ε by leave-one-out cross-validation, over the range $k \in \{5, 10, 15, \ldots, 75\}$ and $\log(\varepsilon) \in \{-10, -9, \ldots, 9, 10\}$. The cross-validation error is minimized at $k = 50$ and $\log(\varepsilon) = -10$. This very small value of ε suggests that for this dataset, once we have restricted our attention to the k nearest neighbors, the local affine structure of the data is more relevant for classification than the class labels themselves. With these automatically chosen parameters, our algorithm achieves a test error of 1.61%, outperforming simple methods such as k-NN as well as many more complicated neural network approaches (e.g. LeNet-1 [20]). 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 [30].

We also test on the challenging USPS digits database, visualized in Figure 5.3 bottom. Here, even humans have considerable difficulties (about 2.5% error). We first apply local MICL with $k = 20$ and $\log(\varepsilon) = -6$ chosen by leave-one-out cross-validation from the range $k \in \{5, 10, \ldots, 75\}$ and $\log(\varepsilon) \in \{-10, -9, \ldots, 9, 10\}$. With these parameters, local MICL achieves an error rate of 4.78% (see Table 5.1 bottom), 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 $\log(\varepsilon) = -5$. 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 example, using many synthetic training images or more advanced skew-correction and normalization techniques lowers the error rate for SVM-poly to 4.1% in [32]. Non-Euclidean distance metrics are also useful: [29] (best reported in [32]) achieves 2.7% error using tangent distance to a

[16]For this experiment, we use the LIB-SVM implementation of SVM [7]

large number of prototypes. Further performance improvements have been achieved by applying matching techniques with local deformation prior to classification [19]. All of these techniques aim at eliminating some of the variation due to nonrigid deformation and misalignment, variations that are not well-handled by holistic methods that treat the entire image as a vector. While we have avoided extensive preprocessing here, so as to isolate the effect of the classifier, such preprocessing can also be incorporated into our framework.

TABLE 5.1

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 in this comparison.

Method	Error (%)	Method	Error (%)
LMICL	**1.61**	k-NN	3.09
SVM-Poly [32]	1.1	Best [30]	0.4

Method	Error (%)	Method	Error (%)
LMICL	**4.78**	k-NN	5.28
k-MICL-Poly	**4.7**	SVM-Poly [7]	5.3

Face Recognition. We further verify MICL's appropriateness for real vision problems using face recognition under varying illumination as an example. Researchers in face recognition have observed both empirically and theoretically that images of the same face under varying lighting conditions lie near a low-dimensional linear subspace [2]. This simple structure of the data suggests that good performance could be obtained directly using Algorithm 1, without resort to local or kernel methods. However, these linear subspaces are embedded in a very high-dimensional image space, and we generally have very few training samples per class with which to infer them.

For this experiment, we use the Extended Yale Face Database B [12][17], which tests the illumination-sensitivity of face recognition algorithms. We work with a standard set of $1,694$ frontal images of 38 subjects. Each image has resolution 168×192. The database is divided into four subsets, corresponding to increasingly extreme illumination angles, visualized in Figure 5.4. We use Subset 1 for training, and test the algorithm's ability to extrapolate to Subsets 2-4. We apply Algorithm 1, not the local or kernel version directly to the raw imagery data. For this dataset, the leave-one-out cross validation error is essentially flat (varying by only a single image) across $\log \varepsilon \in \{-8, \ldots, 6.5\}$. We report the recognition rate at $\log \varepsilon = -0.75$, the midpoint of this range.

We compare MICL with two standard face recognition techniques based on PCA [31] and LDA [3]. For PCA, we choose the projected dimension to minimize the *test* error across $\{5, 10, \ldots, 100\}$, while for LDA, we choose the maximum possible dimension, 37, which also minimizes the test error. We also compare to the Nearest Subspace [21] classifier, which assigns the test image to the class that minimizes the distance between it and the linear span of the training samples from that class. Finally, we compare to RDA, again choosing the regularization parameter α (recall that RDA replaces $\hat{\Sigma}$ with $\hat{\Sigma} + \alpha I$ in the Gaussian likelihood) by cross-validation from the range $\log \alpha \in \{-8, \ldots, 8\}$. The leave-one-out cross-validation error is flat (and minimal) across $\log \alpha = -8, \ldots, 2$. As for ε above, we choose $\log \alpha = -3$, the midpoint of this range.

Table 5.2 reports the recognition error rate for each of the 5 algorithms across the three training subsets. MICL outperforms the two classical techniques significantly, suggesting that if we have a criterion that directly exploits the singular or low-dimensional structures

[17] We use the normalized and cropped version of this dataset, as in [21].

FIG. 5.4. *The Extended Yale B Face Database includes 1,694 images of 38 subjects under varying illumination. The database is divided into four subsets taken under increasingly extreme lighting conditions. In our experiments, we train using Subset 1, and test with each of the remaining subsets.*

of the data, performing dimensionality reduction before classifying becomes unnecessary or even undesirable.[18] For all illuminations, it performs similarly to RDA, and for moderate illuminations, its performance approaches that of the nearest subspace technique, again suggesting that MICL can automatically exploit degenerate structure that may be present in the high-dimensional data.

TABLE 5.2
Face recognition under widely varying illumination. Recognition error rates for various test sets, with Subset 1 as training. MICL outperforms classical techniques such as PCA, and performs competitively with other subspace-based techniques.

	Subset 2	Subset 3	Subset 4
PCA + NN [31]	0.7%	19.9%	85.3%
LDA + NN [3]	0%	1.3%	59.4%
RDA [11]	0%	0.4%	21.8%
MICL	0%	0.4%	20.6%
Nearest Subspace [21]	0%	0%	11.8%

High dimensional data spaces pose challenges to any learning algorithm, in the form of dramatically under-sampled distributions. However, they also open the door to new geometric tools for recognition. In related work, we have shown how sparse representation and compressed sensing [34] can be applied to achieve robust and accurate face recognition despite occlusion and variations in illumination. These techniques rely on geometric phenomena[19] that do not occur in low-dimensional spaces, again suggesting that if the proper tools are available, it may be best to treat the data as-is, rather than performing dimensionality reduction.

6. Discussion. 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 establish its asymptotic optimality for Gaussian data. It generates a family of classifiers, which we connect to classical techniques such as MAP, RDA, and k-NN. This family of classifiers extends the working conditions of these classical techniques to situations where the sample set is sparse or the distribution is singular in a high-dimensional space.

[18] Working directly in the high-dimensional space is computationally feasible thanks to the kernel property (4.1), and can be further accelerated via block determinant identites (see Appendix C for details).

[19] For example, the existence of centrally neighborly polytopes.

Our results also have implications for unsupervised learning. In [23], lossy coding length was used as an objective function for clustering, and a simple agglomerative method was proposed to segment data from mixtures of Gaussians or linear subspaces. The new theoretical results described here further explain for the surprising efficacy of the simple clustering algorithm of [23]. For example, Theorem 3.1 implies that the agglomerative method of [23] makes a decision at each step based on a regularized version of (Gaussian) maximum likelihood or maximum a posterior.

On real vision problems, the MICL criterion and its kernel and local versions perform competitively (nearly optimally for the face recognition problem) without 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, reducing the risk of overfitting. Due to its simplicity and flexibility, we believe it can be successfully applied to even wider range of real-world data and classification problems.

REFERENCES

[1] A. BARRON, J. RISSANEN, AND B. YU, *The minimum description length principle in coding and modeling*, IEEE Transactions on Information Theory, 44 (1998), pp. 2743–2760.

[2] R. BASRI AND D. JACOBS, *Lambertian reflection and linear subspaces*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 25 (2003).

[3] P. BELHUMEUR, J. HESPANDA, AND D. KRIEGMAN, *Eigenfaces vs. Fisherfaces: recognition using class specific linear projection*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 19 (1997), pp. 711–720.

[4] P. BICKEL AND E. LEVINA, *Regularized estimation of large covariance matrices*, to appear in Annals of Statistics, (2007).

[5] P. BICKEL AND B. LI, *Regularization in statistics*, TEST, 15 (2006), pp. 271–344.

[6] C. BISHOP, *Pattern Recognition and Machine Learning*, Springer, 2006.

[7] C. CHANG AND C. LIN, *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[8] T. COVER, *Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition*, IEEE Transactions on Electronic Computers, 14 (1965), pp. 326–334.

[9] T. COVER AND J. THOMAS, *Elements of Information Theory*, Wiley Series in Telecommunications, 1991.

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

[11] J. FRIEDMAN, *Regularized discriminant analysis*, Journal of the American Statistical Association, 84 (1989), pp. 165–175.

[12] A. GEORGHIADES, P. BELHUMEUR, AND D. KRIEGMAN, *From few to many: Illumination cone models for face recognition under variable lighting and pose*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23 (2001), pp. 643–660.

[13] P. GRUNWALD AND J. LANGFORD, *Suboptimal behaviour of Bayes and MDL in classification under misspecification*, in Proceedings of Conference on Learning Theory, 2004.

[14] J. HAMKINS AND K. ZEGER, *Gaussian source coding with spherical codes*, IEEE Transactions on Information Theory, 48 (2002), pp. 2980–2989.

[15] T. HASTIE, R. TIBSHIRANI, AND J. FRIEDMAN, *The Elements of Statistical Learning*, Springer, 2001.

[16] J. HO, M. YANG, J. LIM, K. LEE, AND D. KRIEGMAN, *Clustering appearances of objects under varying illumination conditions*, in Proceedings of CVPR, 2003.

[17] A. HOLUB, M. WELLING, AND P. PERONA, *Combining generative models and fisher kernels for object recognition*, in Proceedings of International Conference on Computer Vision, 2005, pp. 136–143.

[18] I. JOHNSTONE AND A. LU, *Sparse principal component analysis*, preprint, `http://www-stat.stanford.edu/~imj/WEBLIST/AsYetUnpub/sparse.pdf`, (2006).

[19] D. KEYSERS, T. DESELAERS, C. GOLIAN, AND H. NEY, *Deformation models for image recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29 (2007), pp. 1422–1435.

[20] Y. LECUN, L. BOTTOU, Y. BENGIO, AND P. HAFFNER, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.

[21] K. LEE, J. HO, AND D. KRIEGMAN, *Acquiring linear subspaces for face recognition under variable lighting*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27 (2005), pp. 684–698.

[22] J. LI, *A source coding approach to classification by vector quantization and the principle of minimum description length*, in IEEE Data Compression Conference, 2002, pp. 382–391.
[23] 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).
[24] D. MACKAY, *Developments in probabilistic modelling with neural networks – ensemble learning*, in Proc. 3rd Annual Symposium on Neural Networks, 1995, pp. 191–198.
[25] 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.
[26] T. MINKA, *Inferring a gaussian distribution*, MIT Media Lab Note, (1998).
[27] S.A. NENE AND S.K. NAYAR, *A Simple Algorithm for Nearest Neighbour Search in High Dimensions*, PAMI, 19 (1997), pp. 989–1003.
[28] J.J. RISSANEN, *Modeling by shortest data description*, Automatica, 14 (1978), pp. 465–471.
[29] P. SIMARD, Y. LECUN, AND J. DENKER, *Efficient pattern recognition using a new transformation distance*, in Proceedings of Neural Information Processing Systems, vol. 5, 1993.
[30] P. SIMARD, D. STEINKRAUS, AND J. PLATT, *Best practice for convolutional neural networks applied to visual document analysis*, in ICDAR, 2003, pp. 958–962.
[31] M. TURK AND A. PENTLAND, *Eigenfaces for recognition*, in IEEE Conference on Computer Vision and Pattern Recognition, 1991.
[32] V. VAPNIK, *The Nature of Statistical Learning Theory*, Springer, 2000.
[33] X. WANG AND X. TANG, *A unified framework for subspace face recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (2004), pp. 1222–1228.
[34] J. WRIGHT, A. GANESH, A. YANG, S. SASTRY, AND Y. MA, *Robust face recognition via sparse representation*, To appear in IEEE Transactions on Pattern Analysis and Machine Intelligence preprint: http://percetion.csl.uiuc.edu/recognition, (2008).

Appendix A. 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 A.1. *Let δ_{kl} 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 $k \geq 1$,*

$$\frac{\partial^k \ln\det(\Lambda(m)+\Gamma)}{\partial\Gamma_{i_1,j_1}\partial\Gamma_{i_2,j_2}\dots\partial\Gamma_{i_k,j_k}} = (-1)^{k+1}\left(\sum_{\sigma\in Sym(k-1)}\Psi\prod_{l=1}^{k-1}\left[\delta_{j_{\sigma(l)}i_{\sigma(l)}}\Psi\right]\right)_{i_k j_k}, \qquad \text{(A.1)}$$

where 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

$$\frac{\partial \ln\det(\Lambda(m)+\Gamma)}{\partial\Gamma_{i_1,j_1}} = \left((\Lambda(m)+\Gamma)^{-T}\right)_{i_1,j_1} = (\Psi)_{i_1,j_1}. \qquad \text{(A.2)}$$

Suppose that (A.1) holds for $1\dots k-1$. Then

$$\frac{\partial^{k-1} \ln\det(\Lambda(m)+\Gamma)}{\partial\Gamma_{i_1,j_1}\partial\Gamma_{i_2,j_2}\dots\partial\Gamma_{i_{k-1},j_{k-1}}} = (-1)^{k}\left(\sum_{\sigma\in \mathrm{Sym}(k-2)}\Psi\prod_{l=1}^{k-2}\left[\delta_{j_{\sigma(l)}i_{\sigma(l)}}\Psi\right]\right)_{i_{k-1}j_{k-1}} \qquad \text{(A.3)}$$

and so the k-th partial is given by

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

$$(-1)^k \left(\sum_{\sigma} \quad \frac{\partial \Psi}{\partial \Gamma_{i_k,j_k}} \delta_{j_{\sigma(1)} i_{\sigma(1)}} \Psi \dots \Psi \delta_{j_{\sigma(k-2)} i_{\sigma(k-2)}} \Psi \quad + \quad \dots \right.$$
$$\left. + \quad \Psi \delta_{j_{\sigma(1)} i_{\sigma(1)}} \Psi \dots \Psi \delta_{j_{\sigma(k-2)} i_{\sigma(k-2)}} \frac{\partial \Psi}{\partial \Gamma_{i_k,j_k}} \quad \right)_{i_{k-1} j_{k-1}} \tag{A.4}$$

Notice that $\frac{\partial \Psi}{\partial \Gamma_{i_k,j_k}} = -\Psi \delta_{j_k,i_k} \Psi$. Plugging this quantity into (A.4), 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 $\{1 \dots k{-}1\}$ 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 A.2. *Let* $\boldsymbol{q}_1 \dots \boldsymbol{q}_m \dots \overset{iid}{\sim} p_Q(\boldsymbol{q})$, *and let* $\mathbb{E}[Q] = \boldsymbol{\mu}$ *and* $Cov(Q) = \Sigma$. *Let* $\mathcal{Q}^{(m)} = [\boldsymbol{q}_1, \dots, \boldsymbol{q}_m] \in \mathbb{R}^{n \times m}$. *Then* $\forall\, \boldsymbol{z} \in \mathbb{R}^n$,

$$\lim_{m\to\infty} \quad 2m \ln 2 \; \delta R_\varepsilon(\mathcal{Q}^{(m)}, \boldsymbol{z}) \;=\; (\boldsymbol{z}{-}\boldsymbol{\mu})^T \left(\Sigma + \frac{\varepsilon^2}{n} I\right)^{-1} (\boldsymbol{z}{-}\boldsymbol{\mu}) - tr\big(\Sigma(\Sigma{+}\frac{\varepsilon^2}{n} I)^{-1}\big) \quad a.s. \tag{A.5}$$

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

$$\ln\det\left(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma}\right) + \sum_{i,j}\left[\left(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma}\right)^{-1}\right]_{ij}\Gamma_{ij}$$
$$+ O(m^{-2}) \;-\; \ln\det\left(I + \frac{n}{\varepsilon^2}\hat{\Sigma}\right). \tag{A.6}$$

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 A.1. Applying the definition of Γ and rearranging gives

$$\frac{1}{m+1}(\boldsymbol{z}{-}\hat{\boldsymbol{\mu}})^T \left(\frac{\varepsilon^2}{n}\frac{m+1}{m} I + \hat{\Sigma}\right)^{-1} (\boldsymbol{z}{-}\hat{\boldsymbol{\mu}}) \;-\; \ln\left[\frac{\det(I + \frac{n}{\varepsilon^2}\hat{\Sigma})}{\det(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma})}\right] \;+\; O(\varepsilon^{-4}m^{-2}). \tag{A.7}$$

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

$$\lim_{m\to\infty} \quad \left\{ \frac{m}{m+1}(\boldsymbol{z} - \hat{\boldsymbol{\mu}})^T \left(\frac{\varepsilon^2}{n}\frac{m+1}{m} I + \hat{\Sigma}(\mathcal{Q}^{(m)})\right)^{-1} (\boldsymbol{z} - \hat{\boldsymbol{\mu}}) \right.$$
$$\left. - \ln\left[\frac{\det(I + \frac{n}{\varepsilon^2}\hat{\Sigma}(\mathcal{Q}^{(m)})}{\det(I + \frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma}(\mathcal{Q}^{(m)}))}\right]^m + O(\varepsilon^{-4}m^{-1}) \right\}. \tag{A.8}$$

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:

$$\lim_{m\to\infty} \ln\left[\frac{\det(I+\frac{n}{\varepsilon^2}\hat{\Sigma})}{\det(I+\frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma})}\right]^m = \ln\prod_{i=1}^{n}\lim_{m\to\infty}\left[\frac{1+\frac{n}{\varepsilon^2}\hat{\lambda}_i}{1+\frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\lambda}_i}\right]^m \tag{A.9}$$

$$= \ln\prod_{i=1}^{n}\exp\left(\frac{\lambda_i}{\frac{\varepsilon^2}{n}+\lambda_i}\right) \tag{A.10}$$

$$= \mathrm{tr}\big(\Sigma(\Sigma+\frac{\varepsilon^2}{n}I)^{-1}\big). \tag{A.11}$$

Here, in (A.9) 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*

$$\hat{y}(\boldsymbol{x}) = \underset{j=1,\ldots,K}{\mathrm{argmax}}\ \mathcal{L}_G\Big(\boldsymbol{x}\,\big|\,\boldsymbol{\mu}_j,\ \Sigma_j+\frac{\varepsilon^2}{n}I\Big) + \ln\pi_j + \frac{1}{2}D_\varepsilon(\Sigma_j), \tag{A.12}$$

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

$$D_\varepsilon(\Sigma_j) \doteq \mathrm{tr}\left(\Sigma_j\big(\Sigma_j+\frac{\varepsilon^2}{n}I\big)^{-1}\right) \tag{A.13}$$

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 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

$$\begin{aligned}&(m_1+n)\,\delta R_\varepsilon\Big(\mathcal{X}_1^{(m)},\boldsymbol{z}\Big) + R_\varepsilon\Big(\mathcal{X}_1^{(m)}\cup\{\boldsymbol{z}\}\Big) + \delta M_\varepsilon(\mathcal{X}_1^{(m)},\boldsymbol{z}) - \log_2\hat{\pi}_1\\ <\quad &(m_2+n)\delta R_\varepsilon\Big(\mathcal{X}_2^{(m)},\boldsymbol{z}\Big) + R_\varepsilon\Big(\mathcal{X}_2^{(m)}\cup\{\boldsymbol{z}\}\Big) + \delta M_\varepsilon(\mathcal{X}_2^{(m)},\boldsymbol{z}) - \log_2\hat{\pi}_2,\end{aligned} \tag{A.14}$$

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 (A.14) by $\ln 2$ and let $m\to\infty$. Using Lemma A.2 to evaluate the limit of the

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

first term, we have that w.p.1., $\hat{y}(\boldsymbol{z}) = 1$ iff

$$\begin{aligned} &\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu}_1)^T\left(\Sigma_1+\frac{\varepsilon^2}{n}I\right)^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_1) \;-\; \frac{1}{2}D_\varepsilon(\Sigma_1) \;+\; \frac{1}{2}\ln\det(I+\frac{n}{\varepsilon^2}\Sigma_1) \;-\; \ln\pi_1 \\ <\quad &\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu}_2)^T\left(\Sigma_2+\frac{\varepsilon^2}{n}I\right)^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_2) - \frac{1}{2}D_\varepsilon(\Sigma_2) \;+\; \frac{1}{2}\ln\det(I+\frac{n}{\varepsilon^2}\Sigma_2) \;-\; \ln\pi_2. \end{aligned} \tag{A.15}$$

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, (A.12). □

Appendix B. Proof of Theorem 2.

In this section, we analyze the convergence rate of the MICL discriminant functions to their limiting form (A.12), 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

$$\delta L_\varepsilon(\boldsymbol{z}) = (m+n)\,\delta R_\varepsilon(\mathcal{X}, \boldsymbol{z}) \;+\; R_\varepsilon(\mathcal{X}\cup\{\boldsymbol{z}\}) \;+\; \delta M_\varepsilon(\mathcal{X}, \boldsymbol{z}) \;-\; \log_2\hat{\pi} \tag{B.1}$$

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

with limiting form

$$\delta L^\infty_\varepsilon(\boldsymbol{z}) = \frac{1}{2\ln 2}(\boldsymbol{z}-\boldsymbol{\mu})^T\left(\Sigma+\frac{\varepsilon^2}{n}I\right)^{-1}(\boldsymbol{z}-\boldsymbol{\mu}) \;-\; \frac{D_\varepsilon(\Sigma)}{2\ln 2} \;+\; R_\varepsilon(\Sigma) \;-\; \log_2\pi. \tag{B.3}$$

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

$$|\hat{\pi}-\pi| \le \sqrt{\frac{\pi(1-\pi)}{m\alpha}}, \tag{B.4}$$

$$\|\hat{\boldsymbol{\mu}}-\boldsymbol{\mu}\| \le \sqrt{\frac{\operatorname{tr}(\Sigma)}{m\alpha}}, \qquad \textit{and} \tag{B.5}$$

$$\|\hat{\Sigma}-\Sigma\|_F \le g(m,\alpha) + o(m^{-\frac{1}{2}}). \tag{B.6}$$

where

$$g(m,\alpha) \doteq \sqrt{\frac{\mathbb{E}[\|\boldsymbol{z}-\boldsymbol{\mu}\|^4] - \|\Sigma\|_F^2}{m\alpha}} + 2\|\boldsymbol{\mu}\|\sqrt{\frac{\operatorname{tr}(\Sigma)}{m\alpha}} \tag{B.7}$$

and the residual $o(m^{-\frac{1}{2}})$ in (B.6) *is independent of α.*

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

$$P\left[|\hat{\pi} - \pi| \geq \sqrt{\frac{\pi(1-\pi)}{m\alpha}}\right] \leq \alpha. \tag{B.8}$$

Similarly,

$$P[\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_F \geq \eta] \leq \frac{\mathbb{E}[\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2]}{\eta^2} = \frac{\operatorname{tr}(cov(\hat{\boldsymbol{\mu}}))}{\eta^2} = \frac{\operatorname{tr}(\Sigma)}{m\eta^2}, \tag{B.9}$$

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

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

$$\|\hat{\Sigma} - \tilde{\Sigma}\|_F = \left\| \frac{1}{m-1}\tilde{\Sigma} + \boldsymbol{\mu}(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T + \hat{\boldsymbol{\mu}}(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})^T \right\|_F \tag{B.10}$$

$$\leq \frac{1}{m-1}\|\tilde{\Sigma}\|_F + (\|\boldsymbol{\mu}\| + \|\hat{\boldsymbol{\mu}}\|)\,\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\| \tag{B.11}$$

$$\leq 2\|\boldsymbol{\mu}\|\sqrt{\frac{\operatorname{tr}(\Sigma)}{m\alpha}} + o(m^{-\frac{1}{2}}) \tag{B.12}$$

on the event (B.5). We will next bound $\|\tilde{\Sigma} - \Sigma\|_F$. Let $\boldsymbol{\xi} \doteq \operatorname{vec}((\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T)$. Then $\mathbb{E}[\boldsymbol{\xi}] = \operatorname{vec}(\Sigma)$ and $cov(\boldsymbol{\xi}) = E[\boldsymbol{\xi}\boldsymbol{\xi}^T] - \operatorname{vec}(\Sigma)\operatorname{vec}(\Sigma)^T$. Then,

$$P[\|\tilde{\Sigma} - \Sigma\|_F \geq \gamma] \leq \frac{\mathbb{E}[\|\tilde{\Sigma} - \Sigma\|_F^2]}{\gamma^2} = \frac{\operatorname{tr}\left(cov(\operatorname{vec}(\tilde{\Sigma}))\right)}{\gamma^2} \tag{B.13}$$

$$= \frac{\mathbb{E}[\|\boldsymbol{\xi}\|^2] - \|\operatorname{vec}(\Sigma)\|^2}{m\gamma^2} = \frac{\mathbb{E}[\|\boldsymbol{x} - \boldsymbol{\mu}\|^4] - \|\Sigma\|_F^2}{m\gamma^2}. \tag{B.14}$$

Setting the left hand side of (B.14) equal to α and solving for the upper bound γ gives

$$P\left[\|\hat{\Sigma} - \Sigma\|_F \geq \sqrt{\frac{\mathbb{E}[\|\boldsymbol{x} - \boldsymbol{\mu}\|^4] - \|\Sigma\|_F^2}{m\alpha}}\right] \leq \alpha. \tag{B.15}$$

$\|\hat{\Sigma} - \Sigma\|_F \leq \|\tilde{\Sigma} - \Sigma\|_F + \|\hat{\Sigma} - \tilde{\Sigma}\|_F$, so (B.12) and (B.15) give (B.6). Applying a union bound, Equations (B.4), (B.5) and (B.6) hold simultaneously with probability at least $1 - 3\alpha$. □

We will analyze, term by term, the convergence of (B.2) to (B.3), 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 $1 - 3\alpha$,*

$$|\delta L_\varepsilon(\boldsymbol{z}) - \delta L_\varepsilon^\infty(\boldsymbol{z})| \leq \frac{g(m,\alpha)}{2\ln 2}\left(\|\Psi^{-1}(\boldsymbol{z} - \boldsymbol{\mu})\|^2 + \|\Psi^{-1}\Sigma\Psi^{-1}\|_F + \sqrt{n}\|\Psi^{-1/2}\|_F^2\right)$$
$$+ \frac{1}{\ln 2}\sqrt{\frac{\operatorname{tr}(\Sigma)}{m\alpha}}\,\|\Psi^{-1}(\boldsymbol{z} - \boldsymbol{\mu})\| + \frac{1}{\ln 2}\sqrt{\frac{1-\pi}{m\pi\alpha}} + o(m^{-\frac{1}{2}}) \tag{B.16}$$

where $\Psi \doteq \Sigma + \frac{\varepsilon^2}{n}I$, and $g(m,\alpha)$ is defined in (B.7).

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 B.1 are satisfied. From Lemma B.1, $P[E] \geq 1 - 3\alpha$.

Quadratic term. We first analyze the difference between the quadratic term in (B.2) and its limiting form:

$$\left|(\boldsymbol{z}-\hat{\boldsymbol{\mu}})^T\hat{\Psi}(m)^{-1}(\boldsymbol{z}-\hat{\boldsymbol{\mu}}) \;-\; (\boldsymbol{z}-\boldsymbol{\mu})^T\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\right| \tag{B.17}$$

Writing $\boldsymbol{z}-\hat{\boldsymbol{\mu}} = (\boldsymbol{z}-\boldsymbol{\mu}) + (\boldsymbol{\mu}-\hat{\boldsymbol{\mu}})$ and expanding $(\boldsymbol{z}-\hat{\boldsymbol{\mu}})^T\hat{\Psi}(m)^{-1}(\boldsymbol{z}-\hat{\boldsymbol{\mu}})$ gives

$$(\boldsymbol{z}-\boldsymbol{\mu})^T\hat{\Psi}(m)^{-1}(\boldsymbol{z}-\boldsymbol{\mu}) + 2(\boldsymbol{z}-\boldsymbol{\mu})^T[\hat{\Psi}(m)^{-1} - \Psi^{-1} + \Psi^{-1}](\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}) + o(m^{-\frac{1}{2}}) \tag{B.18}$$

$$\begin{aligned} =\; & (\boldsymbol{z}-\boldsymbol{\mu})^T\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu}) + (\boldsymbol{z}-\boldsymbol{\mu})^T\Psi^{-1}(\Sigma-\hat{\Sigma})\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu}) \\ & + 2(\boldsymbol{z}-\boldsymbol{\mu})^T\Psi^{-1}(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}) + o(m^{-\frac{1}{2}}). \end{aligned} \tag{B.19}$$

In (B.18) we have used that $\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}\|^2 = o(m^{-\frac{1}{2}})$, and in (B.19) that $\hat{\Psi}(m)^{-1} = \Psi^{-1} + \Psi^{-1}(\Sigma-\hat{\Sigma})\Psi^{-1} + o(m^{-\frac{1}{2}})$. On event E, (B.17) is bounded above by

$$\|\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\|^2\|\Sigma-\hat{\Sigma}\|_F + 2\|\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\|\,\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}\| + o(m^{-\frac{1}{2}}) \tag{B.20}$$

$$\leq g(m,\alpha)\|\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\|^2 + 2\sqrt{\frac{\operatorname{tr}(\Sigma)}{m\alpha}}\|\Psi^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\| + o(m^{-\frac{1}{2}}) \tag{B.21}$$

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

$$\left| m\ln\left[\frac{\det(I+\frac{n}{\varepsilon^2}\hat{\Sigma})}{\det(I+\frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma})}\right] - \operatorname{tr}\big(\Sigma(\Sigma+\frac{\varepsilon^2}{n}I)^{-1}\big)\right| \tag{B.22}$$

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

$$\ln\det(I+\frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma}) - \ln\det(I+\frac{n}{\varepsilon^2}\hat{\Sigma}) \tag{B.23}$$

$$= \ln\det(\Psi - B - \frac{1}{m+1}\hat{\Sigma}) - \ln\det(\Psi - B) \tag{B.24}$$

$$= \ln\det(I - \Psi^{-1}(B+\frac{1}{m+1}\hat{\Sigma})) - \ln\det(I-\Psi^{-1}B) \tag{B.25}$$

$$= \ln\det(I - (I-\Psi^{-1}B)^{-1}\Psi^{-1}\frac{1}{m+1}\hat{\Sigma}) \tag{B.26}$$

$$= \ln\det(I - (I+\Psi^{-1}B)\Psi^{-1}\frac{1}{m+1}\Sigma + o(m^{-\frac{3}{2}})) \tag{B.27}$$

$$= \ln\det(I - \Psi^{-1}\frac{1}{m+1}\Sigma) + \ln\det(I - \Psi^{-1}B\Psi^{-1}\frac{1}{m+1}\Sigma + o(m^{-\frac{3}{2}})). \tag{B.28}$$

where in (B.27) 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,

$$m\ln\left[\frac{\det(I+\frac{n}{\varepsilon^2}\hat{\Sigma})}{\det(I+\frac{n}{\varepsilon^2}\frac{m}{m+1}\hat{\Sigma})}\right] = \ln\prod_{i=1}^{n}(1-\frac{\zeta_i}{m+1})^m + \ln\prod_{i=1}^{n}(1-\frac{\omega_i}{m+1})^m \tag{B.29}$$

$$= \ln\prod_{i=1}^{n}e^{-\zeta_i}(1+\frac{\zeta_i}{m}+o(m^{-1})) + \ln\prod_{i=1}^{n}e^{-\omega_i}(1+\frac{\omega_i}{m}+o(m^{-1})) \tag{B.30}$$

$$= \operatorname{tr}\left(\Psi^{-1}\Sigma\right) + \sum_{i=1}^{n}\ln(1+\frac{\zeta_i}{m}+o(m^{-1})) + \operatorname{tr}\left(\Psi^{-1}B\Psi^{-1}\Sigma\right) + \sum_{i=1}^{n}\ln(1+\frac{\omega_i}{m}+o(m^{-1})) \tag{B.31}$$

$$= \operatorname{tr}\big(\Sigma(\Sigma+\frac{\varepsilon^2}{n}I)^{-1}\big) + \operatorname{tr}\left(\Psi^{-1}\Sigma\Psi^{-1}(\Sigma-\hat{\Sigma})\right) + o(m^{-1}). \tag{B.32}$$

On E, (B.22) is bounded above by

$$\left|\operatorname{tr}\left(\Psi^{-1}\Sigma\Psi^{-1}(\Sigma-\hat{\Sigma})\right)\right| + o(m^{-1}) \le \|\Psi^{-1}\Sigma\Psi^{-1}\|_F\|\Sigma-\hat{\Sigma}\|_F + o(m^{-1}) \tag{B.33}$$

$$\le g(m,\alpha)\|\Psi^{-1}\Sigma\Psi^{-1}\|_F + o(m^{-1}). \tag{B.34}$$

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

$$\left|\frac{1}{2}\log_2\det(\frac{\varepsilon^2}{n}I+\frac{m}{m+1}\hat{\Sigma}+\Gamma) - \frac{1}{2}\log_2\det(\frac{\varepsilon^2}{n}I+\Sigma)\right|$$

$$= \frac{1}{2}\left|\log_2\det\left(I+\Psi^{-1/2}\left[(\hat{\Sigma}-\Sigma)-\frac{1}{m+1}\hat{\Sigma}+\Gamma\right]\Psi^{-1/2}\right)\right| \tag{B.35}$$

$$\le \frac{n}{2}\log_2\left(1+\frac{1}{\sqrt{n}}\left\|\Psi^{-1/2}\left[(\hat{\Sigma}-\Sigma)-\frac{1}{m+1}\hat{\Sigma}+\Gamma\right]\Psi^{-1/2}\right\|_F\right) \tag{B.36}$$

$$\le \frac{\sqrt{n}}{2\ln 2}\|\Psi^{-1/2}(\hat{\Sigma}-\Sigma)\Psi^{-1/2}\|_F + o(m^{-\frac{1}{2}}) \tag{B.37}$$

$$\le \frac{\sqrt{n}}{2\ln 2}\|\Psi^{-1/2}\|_F^2\|\hat{\Sigma}-\Sigma\|_F + o(m^{-\frac{1}{2}}). \tag{B.38}$$

In going from (B.35) to (B.36), we have used that for symmetric $A\in\mathbb{R}^{n\times n}$ with eigenvalues $\{\lambda_i\}$,

$$|\det(I+A)| \le \prod_i(1+|\lambda_i|) \le \left(1+\frac{\sum_i|\lambda_i|}{n}\right)^n \le \left(1+\frac{1}{\sqrt{n}}\left(\sum_i\lambda_i^2\right)^{1/2}\right)^n \tag{B.39}$$

$$= \left(1+\frac{1}{\sqrt{n}}\operatorname{tr}\left(A^TA\right)^{1/2}\right)^n = \left(1+\frac{1}{\sqrt{n}}\|A\|_F\right)^n. \tag{B.40}$$

On E, the first term of (B.38) is is bounded above by

$$\frac{\sqrt{n}}{2\ln 2}g(m,\alpha)\|\Psi^{-1/2}\|_F^2. \tag{B.41}$$

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

$$|\delta M_\varepsilon(\mathcal{X}, \boldsymbol{z})| = \left| \frac{n}{2} \log_2(1 + \frac{\|\nu\hat{\boldsymbol{\mu}} + \bar{\nu}\boldsymbol{z}\|^2}{\varepsilon^2}) - \frac{n}{2} \log_2(1 + \frac{\|\hat{\boldsymbol{\mu}}\|^2}{\varepsilon^2}) \right| \tag{B.42}$$

$$\leq \frac{n}{2} \log_2 \left(1 + \left| \frac{\|\nu\hat{\boldsymbol{\mu}} + \bar{\nu}\boldsymbol{z}\|^2 - \|\hat{\boldsymbol{\mu}}\|^2}{\varepsilon^2} \right| \right) \tag{B.43}$$

$$= \frac{n}{2} \log_2(1 + O(m^{-1})) \tag{B.44}$$

$$= o(m^{-\frac{1}{2}}). \tag{B.45}$$

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,

$$|\log_2 \hat{\pi} - \log_2 \pi| \;=\; \log_2 \left(1 + \frac{|\hat{\pi} - \pi|}{\min(\hat{\pi}, \pi)} \right) \;\leq\; \log_2 \left(1 + \frac{|\hat{\pi} - \pi|}{\pi - |\hat{\pi} - \pi|} \right) \tag{B.46}$$

$$\leq\; \frac{1}{\ln 2} \frac{\sqrt{1-\pi}}{\sqrt{m\pi\alpha} - \sqrt{1-\pi}} \;=\; \frac{1}{\ln 2} \sqrt{\frac{1-\pi}{m\pi\alpha}} \;+\; o(m^{-\frac{1}{2}}). \tag{B.47}$$

Combining (B.21), (B.34), (B.41), (B.45) and (B.46) gives the result, (B.16). □

Appendix C. 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:

$$\delta L_\varepsilon(\boldsymbol{x}, j) = L_\varepsilon(\mathcal{X}_j \cup \{\boldsymbol{x}\}) - L_\varepsilon(\mathcal{X}_j) - \log_2 \pi_j \tag{C.1}$$

where

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

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 (C.1) 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 (C.1) 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:

$$R_\varepsilon(\mathcal{X}) = \frac{1}{2} \log_2 \det \left(I + \frac{n}{\varepsilon^2} \frac{1}{m-1} \bar{\mathcal{X}}^T \bar{\mathcal{X}} \right) = \frac{1}{2} \log_2 \det \left(I + \frac{n}{\varepsilon^2} \frac{1}{m-1} \Phi_m \mathcal{X}^T \mathcal{X} \Phi_m \right), \tag{C.3}$$

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:

$$R_\varepsilon(\mathcal{X} \cup \{\boldsymbol{z}\}) = \frac{1}{2} \log_2 \det \left(I + \frac{n}{\varepsilon^2 m} \Phi_{m+1} \begin{bmatrix} K & \boldsymbol{b} \\ \boldsymbol{b}^T & c \end{bmatrix} \Phi_{m+1} \right). \tag{C.4}$$

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). (C.4) can be written as

$$\frac{1}{2} \log_2 \det \begin{bmatrix} I + Q + \mathbf{1}\boldsymbol{p}^T + \boldsymbol{p}\mathbf{1}^T + \lambda \mathbf{1}\mathbf{1}^T & \boldsymbol{q} \\ \boldsymbol{q}^T & \xi \end{bmatrix}, \tag{C.5}$$

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},

$$\begin{aligned} Q &\doteq \frac{n}{\varepsilon^2 m} \Upsilon K \Upsilon, \qquad \boldsymbol{p} \doteq -\frac{n}{\varepsilon^2 m} \frac{1}{m+1} \Upsilon \boldsymbol{b}, \qquad \lambda \doteq \frac{n}{\varepsilon^2 m} \frac{c}{(m+1)^2}, \\ \xi &\doteq 1 + \frac{n}{\varepsilon^2 m} \frac{1}{(m+1)^2} \left(\mathbf{1}^T K \mathbf{1} - 2m \mathbf{1}^T \boldsymbol{b} + cm^2 \right) \\ \boldsymbol{q} &\doteq \frac{n}{\varepsilon^2 m} \frac{1}{m+1} \left(-\Upsilon K \mathbf{1} + m \Upsilon \boldsymbol{b} + \frac{\mathbf{1}^T \boldsymbol{b}}{m+1} \mathbf{1} - \frac{mc}{m+1} \mathbf{1} \right). \end{aligned} \tag{C.6}$$

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-adjustments of matrix quantites (the third of which is the Sherman-Woodbury-Morrison matrix inversion lemma):

$$\det \begin{bmatrix} A & \boldsymbol{b} \\ \boldsymbol{b}^T & c \end{bmatrix} = \det(A)(c - \boldsymbol{b}^T A^{-1} \boldsymbol{b}). \tag{C.7}$$

$$\det(A + BCB^T) = \det(A) \det(C) \det(C^{-1} + B^T A^{-1} B). \tag{C.8}$$

$$(A + BCB^T)^{-1} = A^{-1} - A^{-1} B (C^{-1} + B^T A^{-1} B)^{-1} B^T A^{-1}. \tag{C.9}$$

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 (C.5) 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 (C.7), and the second from (C.8). $\det(I + Q)$ and $(I + Q)^{-1}$ can be precomputed offline. A straightforward application of (C.9) 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 $\boldsymbol{p}$:

$$\begin{aligned} \det \begin{bmatrix} \Gamma & \boldsymbol{q} \\ \boldsymbol{q}^T & \xi \end{bmatrix} = \det(I + Q) \, \det(\Lambda) \, \det \left(\Lambda^{-1} + \begin{bmatrix} s_{\mathbf{11}} & s_{\mathbf{1}\boldsymbol{p}} \\ s_{\mathbf{1}\boldsymbol{p}} & s_{\boldsymbol{pp}} \end{bmatrix} \right) \times \\ \left(\xi - s_{\boldsymbol{qq}} + \begin{bmatrix} s_{\boldsymbol{q}\mathbf{1}} \\ s_{\boldsymbol{qp}} \end{bmatrix}^T \left(\Lambda^{-1} + \begin{bmatrix} s_{\mathbf{11}} & s_{\mathbf{1}\boldsymbol{p}} \\ s_{\mathbf{1}\boldsymbol{p}} & s_{\boldsymbol{pp}} \end{bmatrix} \right)^{-1} \begin{bmatrix} s_{\boldsymbol{q}\mathbf{1}} \\ s_{\boldsymbol{qp}} \end{bmatrix} \right) \end{aligned} \tag{C.10}$$

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

Appendix D. Implementation of Kernel MICL.

We start with the coding length function

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

$$= \frac{m+n}{2}\log_2\det\left(I+\frac{n}{\varepsilon^2}\frac{1}{m-1}(\mathcal{X}-\hat{\boldsymbol{\mu}}\mathbf{1}^T)(\mathcal{X}-\hat{\boldsymbol{\mu}}\mathbf{1}^T)^T\right)+\frac{n}{2}\log_2\left(1+\frac{\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}}}{\varepsilon^2}\right)$$

$$= \frac{m+n}{2}\log_2\det\left(I+\frac{n}{\varepsilon^2}\frac{1}{m-1}\mathcal{X}\Phi_m\Phi_m^T\mathcal{X}^T\right)+\frac{n}{2}\log_2\left(1+\frac{\mathbf{1}^T\mathcal{X}^T\mathcal{X}\mathbf{1}}{m^2\varepsilon^2}\right). \tag{D.2}$$

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

$$\frac{m+n}{2}\log_2\det\left(I+\frac{n}{\varepsilon^2}\frac{1}{m-1}\Phi_m^TK\Phi_m\right)+\frac{n}{2}\log_2\left(1+\frac{\mathbf{1}^TK\mathbf{1}}{m^2\varepsilon^2}\right), \tag{D.3}$$

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 4, 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 (D.3) and replacing $K_{ij}=\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle$ with $K_{ij}=k(\boldsymbol{x}_i,\boldsymbol{x}_j)$. Notice that $\Phi_mK\Phi_m$ still corresponds to the mean-centered matrix of inner products (of the vectors $\psi(\boldsymbol{x}_i)$), and $\frac{1}{m^2}\mathbf{1}^TK\mathbf{1}$ corresponds to the norm-squared of the sample mean of the $\psi(\boldsymbol{x}_i)$.

EXAMPLE D.1 (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 5.2 for handwritten digit recognition. In this case,*

$$\psi:\boldsymbol{x}=[x_1,\ldots,x_n]\mapsto\gamma^{d/2}\left[x_1^d,\,\sqrt{d}x_1^{d-1}x_2,\,\ldots,\sqrt{d}x_{n-1}x_n^{d-1},\,x_n^d\right]\in\mathbb{R}^N, \tag{D.4}$$

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

EXAMPLE D.2 (Radial Basis Function). *Another popular choice is*

$$k(\boldsymbol{x}_1,\boldsymbol{x}_2)=\exp\left(-\frac{\|\boldsymbol{x}_1-\boldsymbol{x}_2\|_2^2}{2\sigma^2}\right). \tag{D.5}$$

In this case, $\mathcal{H}$ is infinite-dimensional, and (D.3) *is not valid (i.e.the coding length is infinite). However, we can instead consider the normalized discriminant functions*

$$\overline{\delta L}_\varepsilon(\boldsymbol{x},i)=\frac{2\delta L_\varepsilon(\boldsymbol{x},i)-n\log_2 n}{n}. \tag{D.6}$$

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

$$\overline{\delta L}_\varepsilon(\boldsymbol{x},i)\to\log_2\det{}^+\left(\frac{1}{\varepsilon^2m}\Phi_{m+1}K'\Phi_{m+1}\right)+\log_2\left(1+\frac{\mathbf{1}^TK'\mathbf{1}}{\varepsilon^2(m+1)^2}\right)$$
$$-\log_2\det{}^+\left(\frac{1}{\varepsilon^2(m-1)}\Phi_mK\Phi_m\right)-\log_2\left(1+\frac{\mathbf{1}^TK\mathbf{1}}{\varepsilon^2m^2}\right), \tag{D.7}$$

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 rank$(K') =$ rank$(K) + 1$ for each group,

$$\overline{\delta L}_\varepsilon(\boldsymbol{x}, i) + 2\log_2 \varepsilon \to \log_2 \det^+\left(\frac{1}{m}\Phi_{m+1}K'\Phi_{m+1}\right) + \log_2\left(1 + \frac{\mathbf{1}^T K'\mathbf{1}}{\varepsilon^2(m+1)^2}\right) - \log_2 \det^+\left(\frac{1}{(m-1)}\Phi_m K \Phi_m\right) - \log_2\left(1 + \frac{\mathbf{1}^T K\mathbf{1}}{\varepsilon^2 m^2}\right). \quad \text{(D.8)}$$

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.