Journal of Machine Learning Research 0 (00) 0-00 Submitted 0/00; Published 00/00

Extrapolating expected accuracies for recognition tasks

Charles Zheng CHARLES.Y.ZHANG@GMAIL.COM
Department of Statistics
Stanford University
Palo Alto, CA

Rakesh Achanta
Department of Statistics
Stanford University
Palo Alto, CA

Yuval Benjamini YUVAL.BENJAMINI@MAIL.HUJI.AC.IL
Department of Statistics
The Hebrew University of Jerusalem,
Jerusalem, Israel

Editor: No One Yet

Abstract

The difficulty of multi-class classification generally increases with the number of classes. Using data from a subset of the classes, can we predict how well a classifier will scale with an increased number of classes? Under the assumption that the classes are sampled identically and independently from a population, and under the assumption that the classifier is based on independently learned scoring functions, we show that the expected accuracy when the classifier is trained on k classes is the $k-1$st moment of a certain distribution that can be estimated from data. We present an unbiased estimation method based on the theory, and demonstrate its application on a facial recognition example.

Keywords: Multiclass classification

1. Introduction

Multi-class classification is often applied to problems with large and complex label sets. [[These large multi-class problems are encountered when there exists a large domain of objects or categories to be recognized from data.]] Leading examples include detecting the speaker from his voice patterns (Togneri and Pullella, 2011), identifying the author from her written text (Stamatatos et al., 2014), or labeling the object category from its image (Duygulu et al., 2002; Deng et al., 2010; Oquab et al., 2014). In all these examples, the algorithm observes an input x, and uses the classifier function h to guess the label y from a large label set $\mathcal{S}$.

There are multiple practical challenges in developing recognition systems for large label sets. Collecting high quality training data is perhaps the main obstacle: for a specific system, it is more affordable to collect data on a small set of classes first, even if the long-term goal is to generalize to large sets. Furthermore, the classifier development stage can also be accelerated by training first on small sets of classes, because each training

cycle may require substantially less resources. Indeed, comparisons studying how small-set performance generalizes to larger-set performance can found in the literature (Oquab et al., 2014; Griffin et al., 2007). [[A natural question, then, is how does changing the size of the label set affects the classification accuracy?]]

Technically, we consider a pair of classification problems on finite label sets: a source task with label set $\mathcal{S}_{k_1}$ of size k_1, and a target task with a larger label set $\mathcal{S}_{k_2}$ of size $k_2 > k_1$. For each label set $\mathcal{S}_k$, one constructs the classification rule $h^{(k)} : \mathcal{X} \to \mathcal{S}_k$. Supposing that in each task, the test example (X^*, Y^*) has a joint distribution, define the generalization accuracy for label set $\mathcal{S}_k$ as

$$\mathrm{GA}_k = \Pr[h^{(k)}(X^*) = Y^*]. \tag{1}$$

The problem of *performance extrapolation* is the following: using data from only the source task $\mathcal{S}_{k_1}$, can one predict the accuracy for a target task with a larger label set $\mathcal{S}_{k_2}$?

A natural use case for this accuracy prediction would be in the deployment of a facial recognition system. Suppose a system was developed in the lab on a database of k_1 individuals. Clients would like to deploy this system on a new larger set of k_2 individuals. Performance extrapolation could allow the lab to predict how well the algorithm will perform on the client's problem, accounting for the difference in label set size.

Extrapolation should be possible when the source and target classifications are two instances of the same recognition problem. In many cases, the set of categories $\mathcal{S}$ is to some degree a random or arbitrary selection out of a larger, perhaps infinite, set of potential categories $\mathcal{Y}$. Yet any specific experiment uses a fixed finite set. For example, categories in the classical Caltech-256 image recognition dataset (Griffin et al., 2007) were assembled by aggregating keywords proposed by students and then collecting matching images from the web. The arbitrary nature of the label set is even more apparent in biometric applications (face recognition, authorship, fingerprint identification) where the labels correspond to human individuals (Togneri and Pullella, 2011; Stamatatos et al., 2014). In all these cases, the number of the labels used to define a concrete dataset is therefore an experimental choice rather than a property of the domain. Despite the arbitrary nature of these choices, such datasets are viewed as representing the larger problem of recognition within the given domain, in the sense that success on such a dataset should inform performance on similar problems.

In this paper, we model the label sets as randomly sampled from some population. Not only does the assumption of randomness capture the ambiguity of actual label sets, but it also provides a powerful formalism for answering the question of how to extrapolate. Furthermore, we assume that both $\mathcal{S}_{k_1}$ and $\mathcal{S}_{k_2}$ are i.i.d. samples from a population (or distribution) of labels π, which is defined on the label space $\mathcal{Y}$. Since the label set is random, the generalization accuracy of a given classifier becomes a random variable. We can formalize the problem of performance extrapolation as the problem of estimating the average generalization accuracy AGA_k of an i.i.d. label set $\mathcal{S}_k$ of size k. The condition of i.i.d. sampling of labels ensures that the separation of labels in a random set $\mathcal{S}_{k_2}$ can be inferred by looking at the empirical separation in $\mathcal{S}_{k_1}$, and therefore that some estimate of the average accuracy on $\mathcal{S}_{k_2}$ can be obtained. We also make the assumption that the classifiers train a separate model for each class. This convenient property allows us to characterize the accuracy of the classifier by selectively conditioning on one class at a time.

Our paper presents two main contributions. The first is a formula for calculating the k-class average accuracy of a marginal classifier. The only unknown quantity in the formula is a function $\bar{D}$ which is determined by properties of the data distribution and the classifier. The second is a method for extrapolating the average accuracy curve from k_1-class data to a larger number of classes. The method is based on estimating the unknown function $\bar{D}$, and under certain conditions it has the property of being an unbiased estimator of the average accuracy.

The paper is organized as follows. In the rest of this section, we discuss related work. The framework of randomized classification is introduced in Section 2, and there we also introduce a toy example which is revisited throughout the paper. Section 3 contains our theory of extrapolation, and section 4 introduces our proposed estimation method. In Section 5, we demonstrate our method on a facial recognition problem, making use of the OpenFace feature extraction network Amos et al. (2016).

1.1 Related work

Existing work on classification for large label sets deals with the computational challenges of jointly optimizing the many parameters required for these models (Crammer and Singer, 2001; Lee et al., 2004; Weston et al., 1999): these works are tied to a specific algorithm for classification. Gupta et al. (2014) present a method for estimating the accuracy of a classifier which can be used to improve performance for general classifiers.

Our work deals more with the question of how to link performance between two different but related classification tasks, which falls under the subject of transfer learning (Pan and Yang, 2010). The transfer learning problem we study is one where the source task has labeled data for label set $\mathcal{S}_{k_1}$, and where we want to use the first task to predict our performance on a target task which has labeled data for $\mathcal{S}_{k_2}$. Under Pan and Wang's terminology, our setup is an example of multi-task learning. Examples of transfer learning from one label set to another include Oquab et al. (2014), Donahue et al. (2014), Sharif Razavian et al. (2014).

There do exist instances in the literature of the extrapolation of classification error for a larger number of classes, but we are aware of no methods that are theoretically justified. Kay et al. (2008) tested a model for classifying natural images based on a subject's fMRI brain scan, achieving over 0.75 accuracy of classification on a set of $k_1 = 1750$ natural stimuli and fit an exponential model to the curve in order to determine that it would take a hypothetical set of size $k_2 = 10^{9.5}$ before the accuracy of their model drops below 0.10. However, we are aware of no theoretical backing for an exponential extrapolation.

The theoretical framework we adopt is one where there exists a family of classification problems with increasing number of classes. We are far from the first to consider such a framework: Shannon's foundational paper on information theory (Shannon, 1948) considered the error rate of a random codebook, which is a special case of the randomized classification setup that we study. More recently, a number of authors have considered the problem of high-dimensional feature selection for multiclass classification with a large number of classes (Pan et al., 2016; Abramovich and Pensky, 2015; Davis et al., 2011). All of these works assume specific distributional models for classification. In contrast, the setting we study is more general; however, we do not deal with the problem of feature selection.

2. Randomized classification

2.1 Setup

The randomized classification model we study has the following features. We assume that there exists an infinite, perhaps continuous, label space $\mathcal{Y}$ and a example space $\mathcal{X} \in \mathbb{R}^p$. We assume there exists a prior distribution π on the label space $\mathcal{Y}$. And for each label $y \in \mathcal{Y}$, there exists a distribution of examples F_y. In other words, for an example-label pair (X, Y), the conditional distribution of X given $Y = y$ is given by F_y.

A random classification task can be generated as follows. The label set $\mathcal{S} = \{Y^{(1)}, \ldots, Y^{(k)}\}$ is generated by drawing labels $Y^{(1)}, \ldots, Y^{(k)}$ i.i.d. from π. For each label, we sample a training set and a test set. The training set is obtained by sampling r_{train} observations $X^{(i)}_{j,train}$ i.i.d. from $F_{Y^{(i)}}$ for $j = 1, \ldots, r_{train}$ and $i = 1, \ldots, k$. The test set is likewise obtained by sampling r observations $X^{(i)}_j$ i.i.d. from $F_{Y^{(i)}}$ for $j = 1, \ldots, r$.

We assume that the classifier $h(x)$ works by assigning a score to each label $y^{(i)} \in \mathcal{S}$, then choosing the label with the highest score. That is, there exist real-valued *score functions* $m_{y^{(i)}}(x)$ for each label $y^{(i)} \in \mathcal{S}$. Since the classifier is allowed to depend on the training data, it is convenient to view it (and its associated score functions) as random. We write $H(x)$ when we wish to work with the classifier as a random function, and likewise $M_y(x)$ to denote the score functions when they are considered as random.

For a fixed instance of the classification task with labels $\mathcal{S} = \{y^{(i)}\}_{i=1}^k$ and associated score functions $\{m_{y^{(i)}}\}_{i=1}^k$, recall the definition of the k-class generalization error (1). Assuming that there are no ties, it can be written in terms of score functions as

$$\mathrm{GA}_k(h) = \frac{1}{k} \sum_{i=1}^{k} \Pr[m_{y^{(i)}}(X^{(i)}) = \max_j m_{y^{(j)}}(X^{(i)})],$$

where $X^{(i)} \sim F_{y^{(i)}}$ for $i = 1, \ldots, k$. However, when we consider the labels $\{Y^{(i)}\}_{i=1}^k$ and associated score functions to be random, the generalization accuracy also becomes a random variable.

Suppose we specify k but do not fix any of the random quantities in the classification task. Then the k-class *average generalization accuracy* of a classifier is the expected value of the generalization accuracy $\mathrm{GA}_k(H)$ resulting from a random set of k labels, $Y^{(1)}, \ldots, Y^{(k)} \overset{iid}{\sim} \pi$, and their associated score functions:

$$\begin{aligned} \mathrm{AGA}_k &= \frac{1}{k} \sum_{i=1}^{k} \Pr[M_{Y^{(i)}}(X^{(i)}) = \max_j M_{Y^{(j)}}(X^{(i)})] \\ &= \Pr[M_{Y^{(1)}}(X^{(1)}) = \max_j M_{Y^{(j)}}(X^{(1)})]. \end{aligned}$$

The last line follows from noting that all k summands in the previous line are identical. [The definition of average generalization accuracy is illustrated in Figure 1.]

2.1.1 Marginal classifier

In our analysis, we do not want the classifier to rely too strongly on complicated interactions between the labels in the set. We therefore propose the following property of marginal separability for classification models:

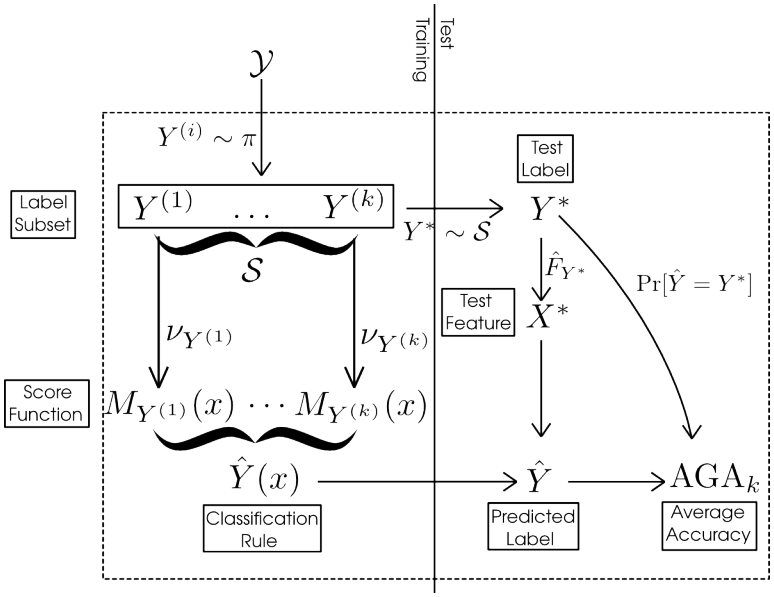


Figure 1: Average generalization accuracy

Definition 1 *The classifier $H(x)$ is called a* marginal classifier *if the score function $M_{y^{(i)}}(x)$ only depends on the label $y^{(i)}$ and the class training set $X^{(i)}_{j,train}$.*

$$M_{y^{(i)}}(x) = g(x; y^{(i)}, X^{(i)}_{1,train}, ..., X^{(i)}_{r_{train},train})$$

This means that the score function for $y^{(i)}$ does not depend on other labels $y^{(j)}$ or their training samples. Therefore, each M_y can be considered to have been drawn from a distribution ν_y. Classes "compete" only through selecting the highest score, but not in constructing the score functions. The operation of a marginal classifier is illustrated in figure 2.

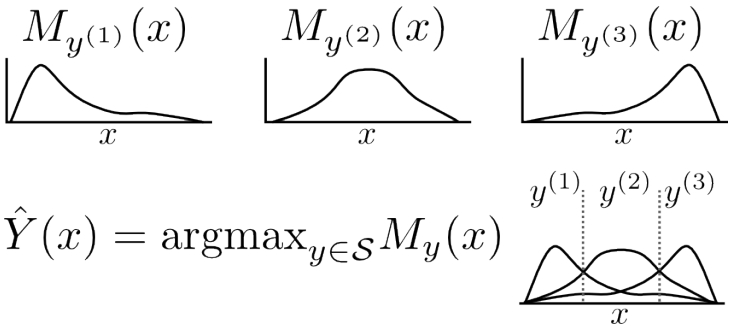


Figure 2: Classification rule

The *marginal* property allows us to prove strong results about the accuracy of the classifier under i.i.d. sampling assumptions.

Comments:

1. If H is a marginal classifier then $M_{Y^{(i)}}$ is independent of $Y^{(j)}$ and $M_{Y^{(j)}}$ for $i \neq j$.

2. Estimated Bayes classifier are a primary example of a marginal classifier. Let $\hat{f}_y$ be a density estimate of the example distribution under label y obtained from the empirical distribution $\hat{F}_y$. Then, we can use the estimated density to produce the score functions:

$$m_y^{EB}(x) = \log(\hat{f}_y(x)).$$

The resulting empirical approximation for the Bayes classifier would be

$$h^{EB}(x) = \text{argmax}_{y \in \mathcal{S}}(m_y^{EB}(x)).$$

3. Both the Quadratic Discriminant Analysis and the naive Bayes classifiers can be seen as specific instances of an estimated Bayes classifier [1]. For QDA, the score function is given by

$$m_y^{QDA}(x) = -(x - \mu(\hat{F}_y))^T \Sigma(\hat{F}_y)^{-1}(x - \mu(\hat{F}_y)) - \log\det(\Sigma(\hat{F}_y)),$$

where $\mu(F) = \int y dF(y)$ and $\Sigma(F) = \int (y - \mu(F))(y - \mu(F))^T dF(y)$. In Naive Bayes, the score function is

$$m_y^{NB}(x) = \sum_{j=1}^{p} \log \hat{f}_{y,j}(x),$$

where $\hat{f}_{y,j}$ is a density estimate for the j-th component of $\hat{F}_y$.

4. For some classifiers, M_y is a deterministic function of y (and therefore ν_y is degenerate). A prime example is when exists fixed or pre-trained embeddings $g, \tilde{g}$ that map labels y and examples x into R^p. Then

$$M_y^{embed} = -\|g(y) - \tilde{g}(x)\|_2. \tag{2}$$

This describes, for example, a 1-nearest neighbor classifier.

5. There are many classifiers which do not satisfy the marginal property, such as multinomial logistic regression, multilayer neural networks, decision trees, and k-nearest neighbors.

Notational remark. Henceforth, we shall relax the assumption that the classifier $H(x)$ is based on a training set. Instead, we assume that there exist score functions $\{M_{Y^{(i)}}\}_{i=1}^k$ associated with the random label set $\{Y^{(i)}\}_{i=1}^k$, and that the score functions $M_{Y^{(i)}}$ are independent of the test set. The classifier $H(x)$ is marginal if and only if $M_{Y^{(i)}}$ are independent of both $Y^{(j)}$ and $M_{Y^{(j)}}$ for $j \neq i$.

1. QDA is the special case of the estimated Bayes classifier when $\hat{f}_y$ is obtained as the multivariate Gaussian density with mean and covariance parameters estimated from the data. Naive Bayes is the estimated Bayes classifier when $\hat{f}_y$ is obtained as the product of estimated componentwise marginal distributions of $p(x_i|y)$

2.2 Estimation of average accuracy

Suppose we have test data for a classification task with k_1 classes. That is, we have a label set $\mathcal{S}_{k_1} = \{y^{(i)}\}_{i=1}^{k_1}$ and its associated set of score functions $M_{y^{(i)}}$, as well as test observations $(x_1^{(i)}, \ldots, x_r^{(i)})$ for $i = 1, \ldots, k_1$. What would be the predicted accuracy for a new randomly sampled set of $k_2 \leq k_1$ labels?

Note that AGA_{k_2} is the expected value of the accuracy on the new set of k_2 labels. Therefore, any unbiased estimator of AGA_{k_2} will be an unbiased predictor for the accuracy on the new set.

Let us start with the case $k_2 = k_1 = k$. For each test observation $x_j^{(i)}$, define the ranks of the candidate classes $\ell = 1, \ldots, k$ by

$$R_j^{i,\ell} = \sum_{s=1}^{k} I\{m_{y^{(\ell)}}(x_j^{(i)}) \geq m_{y^{(s)}}(x_j^{(i)})\}.$$

The test accuracy is the fraction of observations for which the correct class also has the highest rank

$$\mathrm{TA}_k = \frac{1}{rk} \sum_{i=1}^{k} \sum_{j=1}^{r} I\{R_j^{i,i} = k\}. \tag{3}$$

Taking expectations over both the test set and the random labels, the expected value of the test accuracy is AGA_k; hence, TA_k provides the desired estimator.

Next, let us consider the case where $k_2 < k_1$. Consider label set $\mathcal{S}_{k_2}$ obtained by sampling k_2 labels uniformly without replacement from $\mathcal{S}_{k_1}$. Since $\mathcal{S}_{k_2}$ is unconditionally an i.i.d. sample from the population of labels π, the test accuracy of $\mathcal{S}_{k_2}$ is an unbiased estimator of AGA_{k_2}. However, we can get a better unbiased estimate of AGA_{k_2} by averaging over all the possible subsamples $\mathcal{S}_{k_2} \subset \mathcal{S}_{k_1}$. This defines the average test accuracy over subsampled tasks, ATA_{k_2}.

Remark. Naïvely, computing ATA_{k_2} requires us to train and evaluate $\binom{k_1}{k_2}$ classification rules. However, for marginal classifiers, retraining the classifier is not necessary. Looking at the rank $R_j^{i,i}$ of the correct label i for $x_j^{(i)}$, allows us to determine how many subsets $\mathcal{S}_2$ will result in a correct classification. Specifically, there are $R_j^{i,i} - 1$ labels with a lower score than the correct label i. Therefore, as long as one of the classes in $\mathcal{S}_2$ is i, and the other $k_2 - 1$ labels are from the set of $R_j^{i,i} - 1$ labels with lower score than i, the classification of $x_j^{(i)}$ will be correct. This implies that there are $\binom{R_j^{i,i}-1}{k_2-1}$ such subsets $\mathcal{S}_2$ where $x_j^{(i)}$ is classified correctly, and therefore the average test risk for all $\binom{k_1}{k_2}$ subsets $\mathcal{S}_2$ is

$$\mathrm{ATA}_{k_2} = \frac{1}{\binom{k_1}{k_2}} \frac{1}{rk_2} \sum_{i=1}^{k_1} \sum_{j=1}^{r} \binom{R_j^{i,i} - 1}{k_2 - 1}. \tag{4}$$

2.3 Toy Example: Bivariate normals

Let us illustrate these ideas using a toy example. Let (Y, X) have a bivariate normal joint distribution,

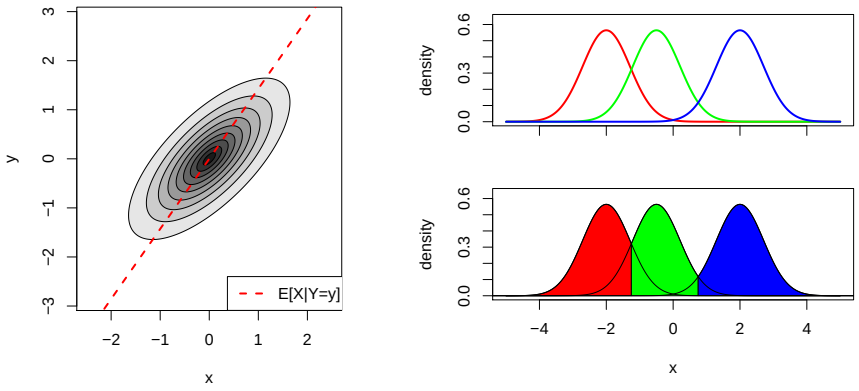
$$(Y, X) \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right),$$

as illustrated in figure 3(a). Therefore, for a given randomly drawn label Y, the conditional distribution of X for that label is univariate normal with mean ρY and variance $1-\rho^2$:

$$X|Y = y \sim N(\rho Y, 1-\rho^2).$$

Supposing we draw $k = 3$ labels y_1, y_2, y_3, the classification problem will be to assign a test instance X^* to the correct label. The test instance X^* would be drawn with equal probability from one of three conditional distributions $X|Y = y^{(i)}$, as illustrated in figure 3(b, top). The Bayes rule assigns X^* to the class with the highest density $p(x|y_i)$, as illustrated by figure 3(b, bottom): it is therefore a marginal classifier, with score function

$$M_{y^{(i)}}(x) = \log(p(x|y^{(i)})) = -\frac{(x-\rho y)^2}{2(1-\rho^2)} + \text{const.}$$



(a) (b)

Figure 3: (a) The joint distribution of (X, Y) is bivariate normal with correlation $\rho = 0.7$. (b) A typical classification problem instance from the bivariate normal model with $k = 3$ classes. Top: the conditional density of X given label Y, for $Y = \{y_1, y_2, y_3\}$. Bottom: the Bayes classification regions for the three classes.

For this model, the generalization accuracy of the Bayes rule for any label set $\{y^{(1)}, \ldots, y^{(k)}\}$ is given by

$$\begin{aligned}\text{GA}_k(y_1, \ldots, y_k) &= \frac{1}{k}\sum_{i=1}^{k} \Pr_{X\sim p(x|y_i)}[p(X|y_i) = \max_{j=1}^{k} p(X|y_j)] \\ &= \frac{1}{k}\sum_{i=1}^{k} \Phi\left(\frac{y^{[i+1]} - y^{[i]}}{2\sqrt{1-\rho^2}}\right) - \Phi\left(\frac{y^{[i-1]} - y^{[i]}}{2\sqrt{1-\rho^2}}\right)\end{aligned}$$

where Φ is the standard normal cdf, $y^{[1]} < \cdots < y^{[k]}$ are the sorted labels, and $y^{[0]} = -\infty$ and $y^{[k+1]} = \infty$. We numerically computed $\text{GA}_k(y_1, \ldots, y_k)$ for randomly drawn labels $Y_1, \ldots, Y_k \overset{iid}{\sim} N(0,1)$; the distributions of GA_k for $k = 2, \ldots, 10$ are illustrated in figure 4. The mean of the distribution of GA_k is the k-class average risk, AGA_k. The theory presented in the next section deals with how to analyze the average risk AGA_k as a function of k.

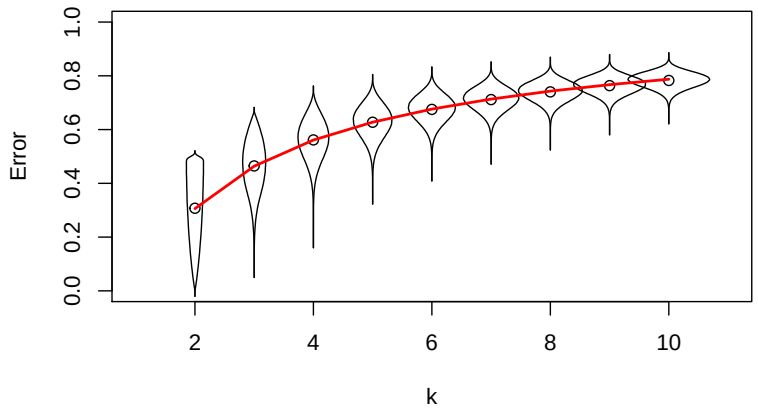


Figure 4: [change figure]The distribution of the classification risk for $k = 2, 3, \ldots, 10$ for the bivariate normal model with $\rho = 0.7$. Circles indicate the average classificatin risk; the red curve is the theoretically computed average risk.

3. Extrapolation

The section is organized as follows. We begin by introducing an explicit formula for the average accuracy AGA_k. The formula reveals that AGA_k is determined by moments of a one-dimensional function $\bar{D}(u)$. Through this formula, therefore, we can infer through subsample accuracies estimates of $\bar{D}(u)$. These estimates allow us to extrapolate the average generalization accuracy to an arbitrary number of labels.

The result of our analysis is to expose the average accuracy AGA_k as the weighted average of a function $\bar{D}(u)$, where $\bar{D}(u)$ is independent of k, and where k only changes the weighting. The result is stated as follows.

Theorem 2 *Suppose π, $\{F_y\}_{y\in\mathcal{Y}}$ and score functions M_y satisfy the tie-breaking condition. Then, there exists a cumulative distribution function $\bar{D}(u)$ defined on the interval $[0, 1]$ such that*

$$AGA_k = 1 - (k-1)\int \bar{D}(u)u^{k-2}du. \tag{5}$$

The tie-breaking allows us to neglect specifying the case when margins are tied.

Definition 3 Tie-breaking condition*: for all $x \in \mathcal{X}$, $M_Y(x) \neq M_{Y'}(x)$ with probability one for Y, Y' independently drawn from π.*

In practice, one can simply break ties randomly, which is mathematically equivalent to adding a small amount of random noise ϵ to the function $\mathcal{M}$.

3.1 Analysis of average accuracy

For the following discussion, we often consider a random label with its associated score function and example vector. Explicitly, this sampling can be written:

$$Y \sim \pi,\ M_Y|Y \sim \nu_Y,\ X|Y \sim F_Y.$$

Similarly we use $(Y', M_{Y'}, X')$ and (Y^*, M_{Y^*}, X^*) for two more triplets with independent and identical distributions. Specifically, X^* will typically note the test example, and therefore Y^* the true label and M_{Y^*} its score function.

The function $\bar{D}$ is related to a favorability function. Favorability measures the probability that the score for the example x^* is going to be maximized by the score function m_y, compared to a random competitor $M_{Y'}$. Formally, we write:

$$U_{x^*}(m_y) = \Pr[m_y(x^*) > M_{Y'}(x^*)]. \tag{6}$$

Note that for fixed example x^*, favorability is monotonically increasing in $m_y(x^*)$. If $m_y(x^*) > m_{y^\dagger}(x^*)$, then $U_{x^*}(y) > U_{x^*}(y^\dagger)$, because the event $\{m_y(x^*) > M_{Y'}(x^*)\}$ contains the event $\{m_{y^\dagger}(x^*) > M_{Y'}(x^*)\}$.

Therefore, given labels $y^{(1)}, \ldots, y^{(k)}$ and test instance x^*, we can think of the classifier as choosing the label with the greatest favorability:

$$\hat{y} = \text{argmax}_{y^{(i)} \in \mathcal{S}} m_{y^{(i)}}(x^*) = \text{argmax}_{y^{(i)} \in \mathcal{S}} U_{x^*}(m_{y^{(i)}}).$$

Furthermore, via a conditioning argument, we see that this is still the case even when the test instance and labels are random:

$$\hat{Y} = \text{argmax}_{Y^{(i)} \in \mathcal{S}} M_{Y^{(i)}}(X^*) = \text{argmax}_{Y^{(i)} \in \mathcal{S}} U_{X^*}(M_{Y^{(i)}}).$$

The favorability takes values between 0 and 1, and when any of its arguments are random, it becomes a random variable with a distribution supported on $[0, 1]$. In particular, we consider the following two random variables:

a. the *incorrect-label* favorability $U_{x^*}(M_Y)$ between a given fixed test instance x^*, and the score function of a random incorrect label M_Y, and

b. the *correct-label* favorability $U_{X^*}(M_{Y^*})$ between a random test instance X^*, and the score function of the correct label, M_{Y^*}.

3.1.1 Incorrect-label favorability

The incorrect-label favorability can be written explicitly as

$$U_{x^*}(M_Y) = \Pr[M_Y(x^*) > M_{Y'}(x^*)|M_Y]. \tag{7}$$

Note that M_Y and $M_{Y'}$ are identically distributed, and are both are unrelated to x^* that is fixed. This leads to the following result:

Lemma 4 *Under the tie-breaking condition, the incorrect-label favorability $U_{x^*}(M_Y)$ is uniformly distributed for any $x^* \in \mathcal{X}$, and*

$$\Pr[U_{x^*}(M_Y) \leq u] = u. \tag{8}$$

Proof is in the appendix.

3.1.2 Correct-label favorability

The correct-label favorability is

$$U^* = U_{X^*}(M_{Y^*}) = \Pr[M_{Y^*}(X^*) > M'_{Y'}(X^*)|Y^*, M_{Y^*}, X^*]. \tag{9}$$

The distribution of U^* will depend on π, F_y and ν_y, and generally cannot be written in a closed form. However, this distribution is central to our analysis–indeed, we will see that the function $\bar{D}$ appearing in theorem 2 is defined as the cumulative distribution function of U^*.

The special case of $k = 2$ shows the relation between the distribution of U^* and the average generalization accuracy, AGA_2. In the two-class case, the average generalization accuracy is the probability that a random correct label score function gives a larger value than a random distractor:

$$\mathrm{AGA}_2 = \Pr[M_{Y^*}(X^*) > M_{Y'}(X^*)].$$

where Y^* is the correct label, and Y' is a random incorrect label. If we condition on $Y^* = y^*$, $M_{Y^*} = m_{y^*}$ and $X^* = x^*$, we get

$$\mathrm{AGA}_2 = \mathbf{E}[\Pr[M_{Y^*}(X^*) > M_{Y'}(X^*)|Y^*, M_{Y^*}, X^*]].$$

Here, the conditional probability inside the expectation is the correct-label favorability. Therefore,

$$\mathrm{AGA}_2 = \mathbf{E}[U^*] = \int \bar{D}(u)du,$$

where $\bar{D}(u)$ is the cumulative distribution function of U^*, $\bar{D}(u) = \Pr[U^* \leq u]$. Theorem 2 extends this to general k; we now give the proof.

Proof of Theorem 2.

Without loss of generality, suppose that the true label is Y^* and the incorrect labels are $Y^{(1)}, \ldots, Y^{(k-1)}$. We have

$$\mathrm{AGA}_k = \Pr[M_{Y^*}(X^*) > \max_{i=1}^{k-1} M_{Y^{(i)}}(X^*)] = \Pr[U^* > \max_{i=1}^{k-1} U_{X^*}(M_{Y^{(i)}})]$$

recalling that $U^* = U_{X^*}(M_{Y^*})$. Now, if we condition on $X^* = x^*$, $Y^* = y^*$ and $M_{Y^*} = m_{y^*}$, then the random variable U^* becomes fixed, with value

$$u^* = U_{x^*}(m_{y^*}).$$

Therefore,

$$\begin{aligned}\mathrm{AGA}_k &= \mathbf{E}[\Pr[U^* > \max_{i=1}^{k-1} U_{X^*}(M_{Y^{(i)}})|X^* = x^*, Y^* = y^*, M_{Y^*} = y^*]] \\ &= \mathbf{E}[\Pr[U^* > \max_{i=1}^{k-1} U_{X^*}(M_{Y^{(i)}})|X^* = x^*, U^* = u^*]]\end{aligned}$$

Now define $U_{max,k-1} = \max_{i=1}^{k-1} U_{X^*}(M_{Y^{(i)}})$. Since by Lemma 4, $U_{X^*}(M_{Y^{(i)}})$ are i.i.d. uniform conditional on $X^* = x^*$, we know that

$$U_{max,k-1}|X^* = x^* \sim \mathrm{Beta}(k-1, 1). \tag{10}$$

Furthermore, $U_{max,k-1}$ is independent of U^* conditional on X^*. Therefore, the conditional probability can be computed as

$$\Pr[U^* > U_{max,k-1} | X^* = x^*, U^* = u^*] = \int_{u^*}^{1} (k-1)u^{k-2} du.$$

Consequently,

$$\begin{aligned}
\mathrm{AGA}_k &= \mathbf{E}[\Pr[U^* > \max_{i=1}^{k-1} U_{x^*}(M_{Y^{(i)}}) | X^* = x^*, U^* = u^*]] \\
&= \mathbf{E}[\int_0^{U^*} (k-1)u^{k-2} du | U^* = u^*] \\
&= \mathbf{E}[\int_0^1 I\{u \le U^*\}(k-1)u^{k-2} du] \\
&= (k-1)\int_0^1 \Pr[U^* \ge u] u^{k-2} du.
\end{aligned}$$

Or equivalently,

$$\mathrm{AGA}_k = 1 - (k-1)\int \bar{D}(u) u^{k-2} du.$$

where $\bar{D}(u)$ denotes the cumulative distribution function of U^* on $[0,1]$:

$$\bar{D}(u) = \Pr[U_{X^*}(M_{Y^*}) \le u]. \tag{11}$$

□.

Theorem 2 expresses the average accuracy as a weighted integral of the function $\bar{D}(u)$. Having this theoretical result allows us to understand how the expected k-class risk scales with k in problems where all the relevant densities are known. However, applying this result in practice to estimate AGA_k requires some means of estimating the unknown function $\bar{D}$– which we discuss in the section ?.

3.2 Favorability and average accuracy for the toy example

Recall that for the toy example from Section 2.3, the score function M_y was a non-random function of y that measures the distance between x and ρy

$$M_y(x^*) = \log(p(x^*|y)) = -\frac{(x^* - \rho y)^2}{2(1-\rho^2)}$$

For this model, the favorability function $U_{x^*}(m_y)$ compares the distance between x^* and ρy to the distance between x^* and $\rho Y'$ for a randomly chosen distractor $Y' \sim N(0,1)$:

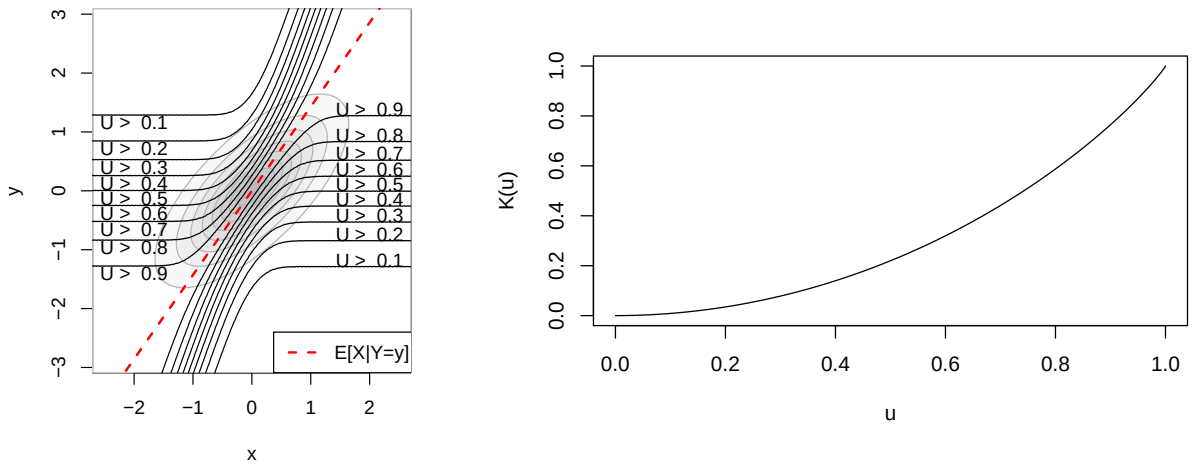
$$\begin{aligned}
U_{x^*}(m_y) &= \Pr[|\rho y - x^*| > |\rho Y' - x^*|] \\
&= \Phi\left(\frac{x^* + |\rho y - x^*|}{\rho}\right) - \Phi\left(\frac{x^* - |\rho y - x^*|}{\rho}\right),
\end{aligned}$$

where Φ is the standard normal cumulative distribution function. Figure 5(a) illustrates the level sets of the function $U_{x^*}(m_y)$. The highest values of $U_{x^*}(m_y)$ are near the line

$x^* = \rho y$ corresponding the to conditional mean of $X|Y$: as one moves farther from the line, $U_{x^*}(m_y)$ decays. Note however that large values of x^* and y (with the same sign) result in larger values of $U_{x^*}(m_y)$ since it becomes unlikely for $Y' \sim N(0,1)$ to exceed $Y = y$.

Using the formula above, we can calculate the correct-label favorability $U^* = U_{X^*}(M_{Y^*})$ and its cumulative distribution function $\bar{D}(u)$. The function $\bar{D}$ is illustrated in figure 5(b) for the current example with $\rho = 0.7$. The red curve in figure 4 was computed using the formula

$$\mathrm{AGA}_k = 1 - (k-1)\int \bar{D}(u)u^{k-2}du.$$



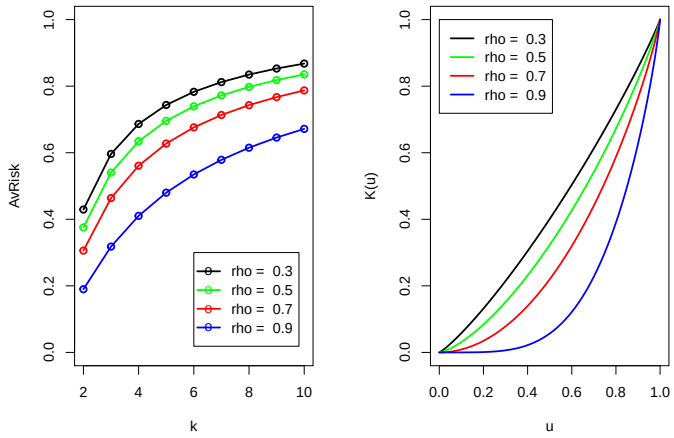
(a) (b)

Figure 5: (a) The level curves of the function $U_{x^*}(m_y)$ in the bivariate normal model with $\rho = 0.7$. (b) The function $\bar{D}(u)$, which gives the cumulative distribution function of the random variable $U_{X^*}(M_Y)$.

It is illuminating to consider how the average accuracy curves and the $\bar{D}(u)$ functions vary as we change the parameter ρ. Higher correlations ρ lead to higher accuracy, as seen in figure 6(a), where the accuracy curves are shifted upward as ρ increases from 0.3 to 0.9. The favorability $U_{x^*}(m_y)$ tends to be higher on average as well, which leads to lower values of the cumulative distribution function–as we see in figure 6(b), where the function $\bar{D}(u)$ becomes smaller as ρ increases.

4. Estimation

Next, we discuss how to use data from smaller classification tasks to extrapolate average accuracy. Assume that we have data from a k_1-class random classification task, and would like to estimate the average accuracy AGA_{k_2} for $k_2 > k_1$ classes. Our estimation method will use the k-class average test accuracies, $\mathrm{ATA}_2, ..., \mathrm{ATA}_{k_1}$ (see Eq 4), for its inputs.



(a) (b)

Figure 6: [figure change]The (a) average risk, (b) $\bar{D}(u)$ function for $k = 2, \ldots, 7$ for the bivariate normal model with $\rho \in \{0.3, 0.5, 0.7, 0.9\}$.

The key to understanding the behavior of the average accuracy AGA_k is the function $\bar{D}$. We adopt a linear model

$$\bar{D}(u) = \sum_{\ell=1}^{m} \beta_\ell h_\ell(u), \tag{12}$$

where $h_\ell(u)$ are known basis functions, and β_ℓ are the linear coefficients to be estimated.

Conveniently, the AGA_k can also be expressed in terms of the β_ℓ coefficients. If we plug in the assumed linear model (12) into the identity (5), then we get

$$1 - \text{AGA}_k = (k-1) \int \bar{D}(u) u^{k-2} du \tag{13}$$

$$= (k-1) \int_0^1 \sum_{\ell=1}^{m} \beta_\ell h_\ell(u) u^{k-2} du \tag{14}$$

$$= \sum_{\ell=1}^{m} \beta_\ell H_{\ell,k} \tag{15}$$

where

$$H_{\ell,k} = (k-1) \int_0^1 h_\ell(u) u^{k-2} du. \tag{16}$$

The constants $H_{\ell,k}$ are moments of the basis function h_ℓ: hence we call this method the *moment method.* Note that $H_{\ell,k}$ can be precomputed numerically for any $k \geq 2$.

Now, since the test accuracies ATA_k are unbiased estimates of AGA_k, this implies that the regression estimate

$$\hat{\beta} = \text{argmin}_\beta \sum_{k=2}^{k_1} \left((1 - \text{ATA}_k) - \sum_{\ell=1}^{m} \beta_\ell H_{\ell,k} \right)^2$$

Label	Training			Test
$y^{(1)}$=Amelia	$x_1^{(1)}$ = 	$x_2^{(1)}$ = 	$x_3^{(1)}$ = 	$x_*^{(1)}$ = 
$y^{(2)}$=Jean-Pierre	$x_1^{(2)}$ = 	$x_2^{(2)}$ = 	$x_3^{(2)}$ = 	$x_*^{(2)}$ = 
$y^{(3)}$=Liza	$x_1^{(3)}$ = 	$x_2^{(3)}$ = 	$x_3^{(3)}$ = 	$x_4^{(3)}$ = 
$y^{(4)}$=Patricia	$x_1^{(4)}$ = 	$x_2^{(4)}$ = 	$x_3^{(4)}$ = 	$x_4^{(4)}$ = 

Figure 7: Face recognition problem

is unbiased for β. The estimate of AGA_{k_2} is similarly obtained from (15), via

$$\widehat{\mathrm{AGA}_{k_2}} = 1 - \sum_{\ell=1}^{m} \hat{\beta}_\ell H_{\ell,k_2}. \tag{17}$$

5. Face recognition example

We demonstrate the extrapolation of average accuracy by predicting the accuracy of a face recognition on a large set of labels from the system's accuracy on a smaller subset.

5.1 Data

From the "Labeled Faces in the Wild" dataset (Huang et al. (2007)), we selected the 1672 individuals with at least 2 face photos. We form a dataset consisting of photo-label pairs $(x_j^{(i)}, y^{(i)})$ for $i = 1, \ldots, 1672$ and $j = 1, 2$ by randomly selecting 2 face photos for each individual.

5.2 Classifier

We implement a two-step face recognition system based on a precomputed neural-network based embedding followed by a one nearest neighbor classifier.

We used the OpenFace (Amos et al. (2016)) embedding for feature extraction. For each photo x, a 128-dimensional feature vector $g(x)$ is obtained as follows. The computer vision library DLLib is used to detect landmarks in x, and to apply a nonlinear transformation to align x to a template. The aligned photograph is then downsampled to a 96×96 image. The downsampled image is fed into a pre-trained deep convolutional neural network to obtain the 128-dimensional feature vector $g(x)$. More details are found in (Amos et al. (2016)).

The recognition system then works as follows. Suppose we want to perform facial recognition on a subset of the individuals, $I \subset \{1, \ldots, 1672\}$. Then, for all $i \in I$, we load one example-label pair, into the system, $(x_1^{(i)}, y^{(i)})$. In order to identify a new photo $\vec{z}^*$, we obtain the feature vector $g(x^*)$, and guess the label $\hat{y}$ with the minimal Euclidean distance between $g(y^{(i)})$ and $g(x^*)$, which implies a score function

$$M_{y^{(i)}}(x^*) = -||g(x_1^{(i)}) - g(x^*)||^2.$$

k_1	100	200	300	400
MSE	0.0844	0.0356	0.00465	0.000415

Table 1: MSE on predicting TA_{1672} from $\widehat{\mathrm{AGA}_{k_1}}$

The test accuracy is assessed on the unused repeat for all individuals in I. Note that the assumptions of our estimation method are met in this example because one-nearest neighbor is a marginal classifier. We note that m-nearest neighbor for $m > 1$ is not marginal.

5.3 Experimental Details

We treat the full data-set of 1672 as our population of labels. In each experiment, we choose a random subset of size $k_1 < 1672$, for which we observe both training and test data. From this data, we will estimate AGA_{k_2} for k_2 between $k_2 = k_1 + 1, ..., 1672$. We use $k_1 = 100, 200, 400, 800$, with $B = 300$ repeats for each subset size.

For the selected subset of size k_1, we can estimate the test accuracy at every $k \leq k_1$, getting the vector $(\mathrm{ATA}_2, ..., \mathrm{ATA}_{k_1})$. Specifically, we use a linear spline basis,

$$h_\ell(u) = \left[u - \frac{\ell - 1}{m}\right]_+$$

for $\ell = 1, \ldots, m$. Here we take $m = 10000$. We fit the β coefficient vector using non-negative least squares:

$$\hat{\beta} = \mathrm{argmin}_{\beta:\beta_\ell \geq 0} \sum_{k=2}^{k_1} \left((1 - \mathrm{ATA}_k) - \sum_{\ell=1}^{m} \beta_\ell H_{\ell,k} \right)^2$$

The non-negativity constraint on the coefficients β, combined with the linear spline basis, ensures that $\bar{D}(u)$ is a monotonic and convex function on $[0, 1]$.

Based on the β coefficients, we can compute the prediction for the full dataset $\hat{AGA}_{1672}$. We compare the prediction to the observed accuracy on full dataset TA_{1672}, which approximates the ground truth.

5.4 Results

The extrapolation results can be seen in Figures 8 and 9. In the first, we show a extrapolation estimate for $k_1 = 400$. In the second, we see multiple instances for different values of k_1. As can be seen, both the accuracy as well as the variances decrease rapidly as k_1 increases. In general, for $k_1 > 400$ the paths of the different extrapolation curves are not very different. The root-mean-square errors between at $k_2 = 1672$ can be seen in Table 1.

6. Discussion

In this work, we did not develop the theory needed to understand how the choice of basis functions $\{h_\ell\}$, or also how the adoption of constraints on the fitted function $\bar{D}(u)$ may affect estimation. Based on our experience with simulated and real data examples, we can offer a couple of practical suggestions.

a b

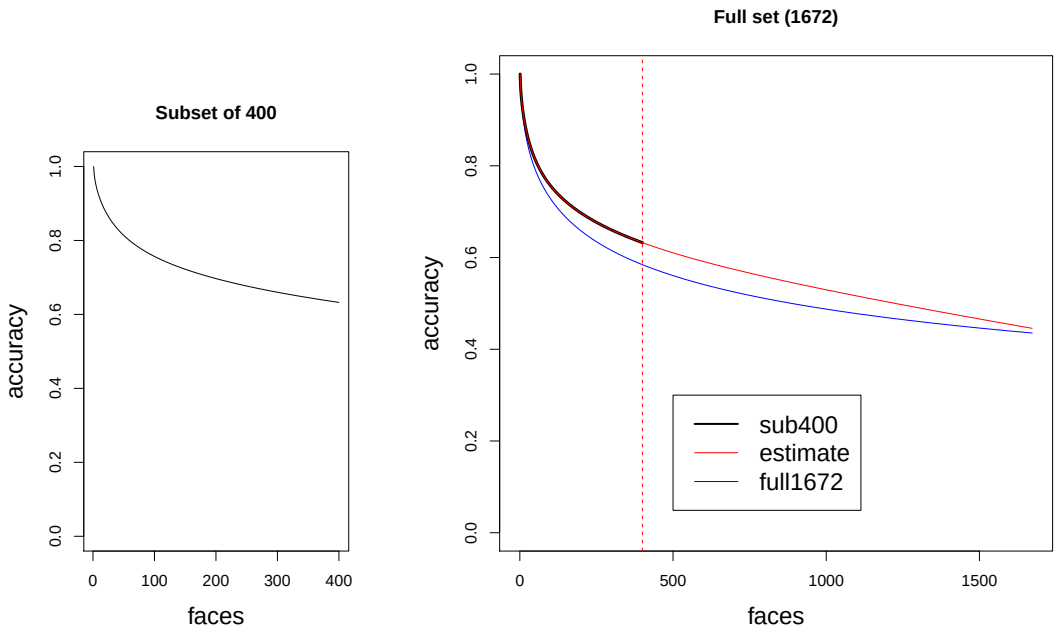


Figure 8: (a) The estimated average accuracy for $k = 2, \ldots, 400$ given a dataset of 400 faces subsampled from Labeled Faces in the Wild. (b) Estimated average accuracy for $k > 400$ on the same dataset, compared to the ground truth (average k-class test accuracy using all 1672 classes).

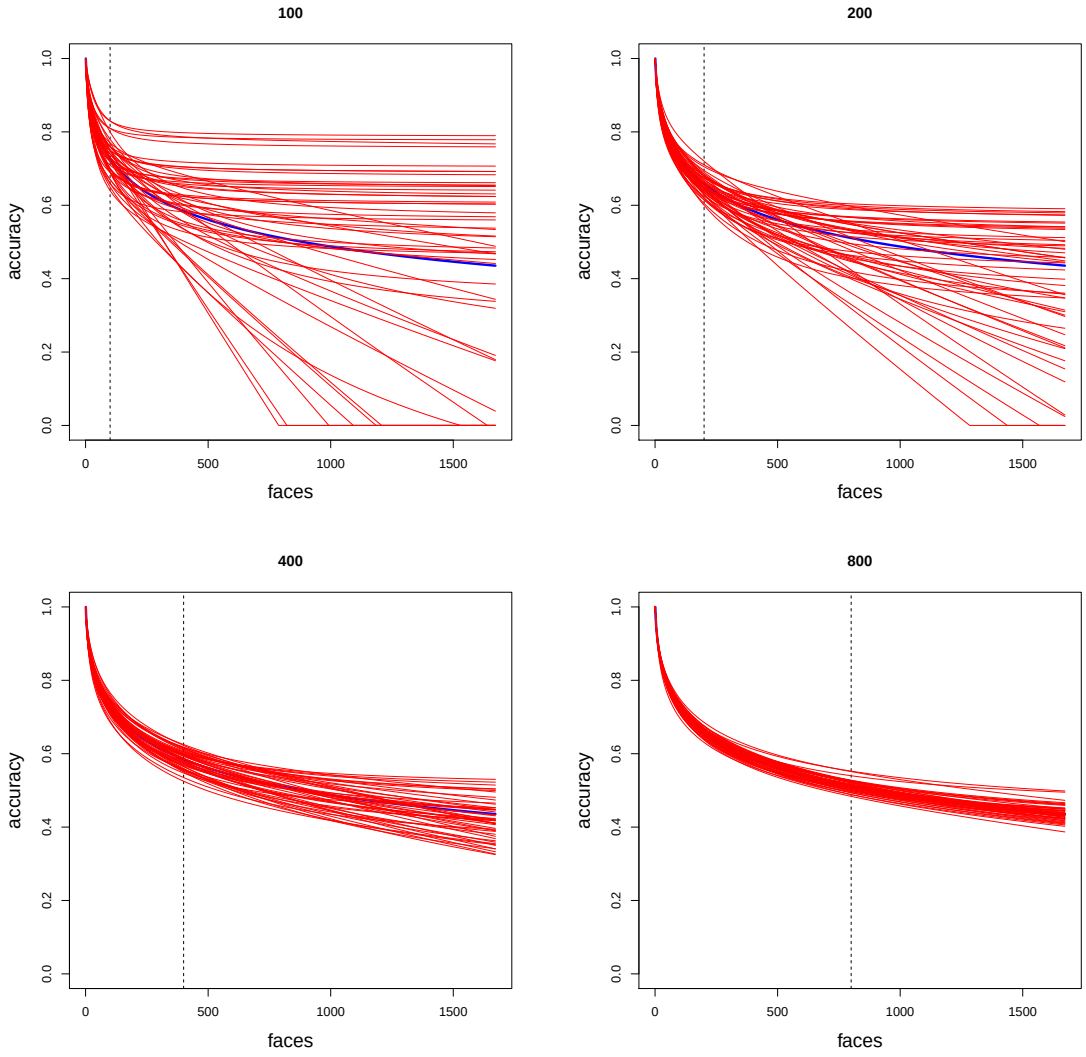


Figure 9: Estimated average accuracy using subsampled datasets of size k, compared to the ground truth (average k-class test accuracy using all 1672 classes).

We know that $\bar{D}(u)$ has to be monotone and bounded by one, because it is a cumulative distribution function. Furthermore, in many real and simulated examples, $\bar{D}(u)$ also has the property of convexity. We have also found that including the monotonicity and convexity constraint in the estimation procedure improves estimation performance.

A convenient way to impose both the monotonicity and the convexity constraint on $\bar{D}(u)$ is to choose $\{h_\ell\}$ to be a family of linear splines, $h_\ell(u) = [u - u_\ell]_+$, and then to require that the spline coefficients β_ℓ are non-negative, as we described in section 5.3. We expect that higher-order spline bases should also do well, although it would be slightly less straightforward to impose the convexity and monotonicity constraints.

The number and location of knots for the spline basis is also quite important. In the totally unconstrained linear model, controlling the number of basis functions is crucial for preventing overfitting. When at least one of either the monotonicity or convexity constraint is adopted, however, the fit is less sensitive to the number of basis functions, as long as the knots form a sufficiently fine discretization of the unit interval. Yet, even then, for computational reasons, it can be helpful to choose the location of the knots carefully. When attempting to extrapolate to a large number of classes, the behavior of $\bar{D}(u)$ for u close to 1 is far more important than the shape of $\bar{D}(u)$ in the rest of the interval. This is because the weighting density $\text{Beta}(k-1, 1)$ used to compute the higher moments is concentrated near 1 for large k. A good heuristic is to consider that the mean of the weighting density is $\frac{k-1}{k}$, which suggests that the scale of discretization near 1 should be smaller than $1/k_2$ when the goal is to estimate the average accuracy for k_2 classes. Meanwhile, the standard deviation of the weighing density is approximately $1/k$, meaning that the behavior of $\bar{D}(u)$ for u smaller than $1 - \frac{5}{k_2}$ (that is, u which are more than 4 standard deviations away from the mean of the weighting density) has negligible effect on the extrapolation. Therefore, a coarser discretization can be used for knots while are smaller than $1 - \frac{5}{k_2}$. Still, the overall discretization should not be coarser than order $1/k_1$, so that the model has sufficient degrees of freedom to explain the subsampled accuracy curve.

An alternative to the linear model (or variations thereof) is to consider parametric families for the function $\bar{D}(u)$. When the family is a good fit to the the data, this can considerably improve extrapolation power, especially when the size of the dataset is limited. We discuss some parametric models for $\bar{D}(u)$ in forthcoming work.

A limitation of our analysis of average accuracy is that it applies only to marginal classifiers. Theoretically, it may be possible to extend the analysis to certain non-marginal classifiers, for instance, by showing that in the large-k limit, such classifiers, become "asymptotically" marginal, in some sense. However, practically speaking, there is no obstacle to applying our extrapolation method to a non-marginal classifier. It would be interesting to see how well our proposed method still works for some non-marginal classifiers.

Another limitation of our analysis is that it applies only to randomized classification problems with i.i.d. sampling of classes. Extension to other sampling mechanisms, such as cluster sampling, may be possible if one can work out how the dependence between classes translates into dependence between the favorabilities $U_{X^*}(m_{Y^{(k)}})$.

More broadly, the assumption that the labels in $\mathcal{S}_k$ are a random sample from a distribution π may be inappropriate. Many natural classification problems arise from hierarchically partitioning a space of instances into a set of labels. Therefore, rather than modeling $\mathcal{S}_k$ as

a random sample, it may be more suitable to model it as a random hierarchical partition of $\mathcal{Y}$ (for instance, arising from an optional Pólya tree process (Wong et al., 2010).)

In the current paper, we only discussed extrapolating the classification accuracy–or equivalently, the risk for the zero-one cost function. However, it is possible to extend our analysis to risk functions with arbitrary cost functions, which is the subject of forthcoming work.

In our analysis, we also limited the training set sizes to be the same in both the source and target tasks. It would be very nice to be able to relax this assumption, since in most practical applications, the training set sizes are varied within label sets as well as between tasks. The idea of extrapolating from learning curves (Cortes et al., 1994) may be one way to adjust for varying training set sizes.

Many other aspects of transfer learning could also be considered in conjunction with the issue of transferring from one label set to another, such as transfer to a different domain, or transfer to a target task without training data (Pan and Yang, 2010).

Acknowledgments

We thank Jonathan Taylor, Trevor Hastie, John Duchi, Steve Mussmann, Qingyun Sun, Robert Tibshirani for useful discussion. CZ is supported by an NSF graduate research fellowship, and would also like to thank the Hebrew University (??? grant) for travel support.

References

Felix Abramovich and Marianna Pensky. Feature selection and classification of high-dimensional normal vectors with possibly large number of classes. *arXiv preprint arXiv:1506.01567*, 2015.

Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

Corinna Cortes, Lawrence D Jackel, Sara A Solla, Vladimir Vapnik, and John S Denker. Learning curves: Asymptotic values and rate of convergence. In *Advances in Neural Information Processing Systems*, pages 327–334, 1994.

Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.

Justin Davis, Marianna Pensky, and William Crampton. Bayesian feature selection for classification with possibly large number of classes. *Journal of Statistical Planning and Inference*, 141(9):3256–3266, 2011.

Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *European conference on computer vision*, pages 71–84. Springer, 2010.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

Pinar Duygulu, Kobus Barnard, Joao FG de Freitas, and David A Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European conference on computer vision*, pages 97–112. Springer, 2002.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

Maya R Gupta, Samy Bengio, and Jason Weston. Training highly multiclass classifiers. *Journal of Machine Learning Research*, 15(1):1461–1492, 2014.

Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

Kendrick N Kay, Thomas Naselaris, Ryan J Prenger, and Jack L Gallant. Identifying natural images from human brain activity. *Nature*, 452(March):352–355, 2008. ISSN 0028-0836. doi: 10.1038/nature06713.

Yoonkyung Lee, Yi Lin, and Grace Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81, 2004.

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

Rui Pan, Hansheng Wang, and Runze Li. Ultrahigh-dimensional multiclass linear discriminant analysis by pairwise sure independence screening. *Journal of the American Statistical Association*, 111(513):169–179, 2016.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, 1948. ISSN 00058580. doi: 10.1002/j.1538-7305.1948.tb01338.x. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6773024`.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

Efstathios Stamatatos, Walter Daelemans, Ben Verhoeven, Patrick Juola, Aurelio López-López, Martin Potthast, and Benno Stein. Overview of the author identification task at pan 2014. In *CLEF (Working Notes)*, pages 877–897, 2014.

Roberto Togneri and Daniel Pullella. An overview of speaker identification: Accuracy and robustness issues. *IEEE Circuits and Systems Magazine*, 11(2):23–61, 2011.

Jason Weston, Chris Watkins, et al. Support vector machines for multi-class pattern recognition. In *ESANN*, volume 99, pages 219–224, 1999.

Wing H Wong, Li Ma, et al. Optional pólya tree and bayesian inference. *The Annals of Statistics*, 38(3):1433–1459, 2010.