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 exchangeably, and under the assumption that the classifier is generative (e.g. QDA or Naive Bayes), we show that the expected accuracy when the classifier is trained on k classes is the $k-1$st moment of a *conditional accuracy distribution*, which can be estimated from data. This provides the theoretical foundation for performance extrapolation based on pseudolikelihood, unbiased estimation, and high-dimensional asymptotics. We investigate the robustness of our methods to non-generative classifiers in simulations and one optical character recognition example.

Keywords: Multiclass classification

1. Introduction

An algorithm that can use sensory information to automatically distinguish between multiple scenarios has increasingly many applications in modern life. Examples include detecting the speaker from his voice patterns, identifying the author from her written text, or labeling the object category from its image. All these examples can be described as recognition problems: the algorithm observes an input x, and uses the classifier function h to guess the label y from a discrete set $\mathcal{Y}$ of possible labels. In all applications described above, the space of potential labels is practically infinite. But in any particular experiment, the number of different labels k used would be finite. A natural question, then, is how changing the number of possible labels affects the classification accuracy.

More technically, we consider a sequence of classification problems on finite label subsets $|\mathcal{S}_k| < \cdots < |\mathcal{S}_K|, \mathcal{S}_i \subset \mathcal{Y}$, where in the k-th task, one constructs the classification rule $h^{(k)} : \mathcal{X} \to \mathcal{S}_k$. Supposing that (X, Y) have a joint distribution, define the generalization

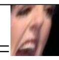
Label	Training			Test
$y^{(1)}$=Amelia	$x_1^{(1)}$ = 	$x_2^{(1)}$ = 	$x_3^{(1)}$ = 	$x_*^{(1)}$ = 
$y^{(2)}$=Jean-Pierre	$x_1^{(2)}$ = 	$x_2^{(2)}$ = 	$x_3^{(2)}$ = 	$x_*^{(2)}$ = 
$y^{(3)}$=Liza	$x_1^{(3)}$ = 	$x_2^{(3)}$ = 	$x_3^{(3)}$ = 	$x_4^{(3)}$ = 
$y^{(4)}$=Patricia	$x_1^{(4)}$ = 	$x_2^{(4)}$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

Figure 1: Face recognition problem

accuracy for the k-th problem as

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

The problem of *performance extrapolation* is the following: using data from only $\mathcal{S}_k$, can one predict the accuracy for the larger label set $\mathcal{S}_K$? Note that unlike other extrapolations from a smaller sample to a larger population, the classification problem becomes harder as the number of distractor classes increases.

Posing this question rigorously requires some effort. Within the statistics and machine learning literature, the usual formalism for studying a recognition task is to pose it as a *multi-class classification* problem. One delineates a finite set of distinct entities which are to be recognized and distinguished, which is the *label set* $\mathcal{S}$.

However, under this setup, it is usually assumed that the label set $\mathcal{S}$ is a finite and specific set of categories, which is known in advance. [[Citations YB]] For human recognition systems, a fixed and finite label set is not always appropriate: humans can easily learn to recognize new objects. Similarly, we think about success of recognition systems as a property that is more general than accuracy on any individual set of classes. Therefore we need to extend the notion of accuracy to deal with multiple but related classification tasks. [[Some citations of people doing mildly similar things YB: e.g. one-shot learning, transfer learning]]

Accurate answers to this problem are not only of theoretical interest, but also have practical implications:

Example 1:

Facial recognition is an important technology with applications in security and in social media, such as automatic tagging of photographs on Facebook. The basic problem is illustrated in Figure 1: given a collection of tagged and cropped photographs $\{(x_j^{(i)}), y^{(i)}\}$, where $y^{(i)}$ is the label, and $x_j^{(i)}$ is a vector containing the pixel intensities, assign labels y to untagged photographs x_*. Here, the notation $x_j^{(i)}$ indicates the jth labelled photograph in the database belonging to the ith individual.

A client might want to run a facial recognition system developed by lab A on their own database of individuals. In this case, there might be no overlap between the people in lab A's database and the people in the client's database. And yet, the client might still expect the performance figures reported by lab A to be informative of how well the recognition system will do on their own database!

Example 2: A neuroscientist is interested in how well the brain activity in various regions of the brain can discriminate between different classes of stimuli. Kay et al. (2008) obtained fMRI brain scans which record how a single subject's visual cortex responds to natural images. They wanted to know how well the brain signals could discriminate between different images. For a set of 1750 photographs, they constructed a classifier which achieved over 0.75 accuracy of classification. Based on exponential extrapolation, they estimate that it would take on the order of $10^{9.5}$ classes before the accuracy of the model drops below 0.10! A theory of performance extrapolation could be useful for the purpose of making such extrapolations in a more principled way.

Both examples refer to the typical development of a recognition system. Recognition systems are first tested on small datasets, even if the longer-term goal is to deploy them on large datasets with many more classes. The question is how much decay in the performance should we expect for the larger problem compared to the smaller problem. [Cite What Does Classifying...] Theory for performance extrapolation may therefore reveal models with bad scaling properties in the pilot stages of development.

Our primary goal in this paper is to formulate this question, and identify scenarios where answers are possible. The most important condition is that the smaller problem would be representative of the larger one. For simplicity, we assume that both $\mathcal{S}_K$ and $\mathcal{S}_k$ are iid samples from a population (or distribution) of labels. (Other sampling mechanisms would require some modification). The condition of i.i.d. sampling of labels ensures that the separation of labels in a random set $\mathcal{S}_K$ can be inferred by looking at the empirical separation in $\mathcal{S}_k$, and therefore that some estimate of the achievable accuracy on $\mathcal{S}_K$ can be obtained.

Our analysis considers a restricted set of classifiers, *marginal classifiers*, which 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. In section **??**, we use this technique to reveal that the expected risk for classifying on the label set $\mathcal{Y}_k$, for all k, is governed by a specific function - the *conditional risk* - that depends on the true distributions and the classifier. As long as one can recover the conditional risk function $\bar{D}(u)$, one can compute the average risk for any number of classes. In section 5, we empirically study the performance curves of classifiers on sequences of classification tasks.

The central theme of this thesis is the study of *randomized classification*, which can be motivated as an extension of the classical multi-class classification framework to accommodate the possibility of growing or infinite label sets $\mathcal{Y}$. The basic approach taken is to assume an infinite or even continuous label space $\mathcal{Y}$, and then to study the problem of classification on finite label sets S which are randomly sampled from $\mathcal{Y}$. This, therefore defines a *randomized classification* problem where the label set is finite but may vary from instance to instance. One can then proceed to answer questions about the variability of the performance due to randomness in the labels, or how performance changes depending on the size of the random label set.

1.1 Motivation

The formalism of classification is inadequate for studying many practical questions related to the generalizability of the facial recognition system. Using test data, we estimate the generalization accuracy of a recognition system. However, these estimated accuracies apply only to the particular collection of individuals $\{y^{(1)}, \dots, y^{(k)}\}$. If we were to add a new individual $y^{(k+1)}$ to the dataset, for instance, when photographs are uploaded on Facebook containing a new user, this defines a totally new classification problem because the expanded set of labels $\{y^{(1)}, \dots, y^{(k+1)}\}$ defines a different response space than the old set of labels $\{y^{(1)}, \dots, y^{(k)}\}$. Yet, these two classification problems are clearly linked.

The question of how to link performance between two different but related classification tasks is an active area of research, known as *transfer learning.* But while the two examples we just listed might be considered as examples of transfer learning problems, the current literature on transfer learning, as far as we know, does not study the problem of *mutating label sets.* Therefore, to address this new class of questions about the generalizability of the recognition system, we need to formalize our notions of (a) what constitutes a 'recognition system' which can be applied to different classification problems, and (b) what assumptions about the problem, and what assumptions about the classifiers used, allow one to infer performance in one classification problem based on performance in another classification problem.

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 $\mathcal{S} = \{Y^{(1)}, \dots, Y^{(k)}\}$ is generated by drawing labels $Y^{(1)}, \dots, Y^{(k)}$ 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 $j = 1, \dots, r_{train}$ and $i = 1, \dots, k$. The test set is likewise obtained by sampling r observations $X^{(i)}_j$ i.i.d. from $F_{Y^{(i)}}$ for $j = 1, \dots, r$.

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, $Y^{(1)}, \ldots, Y^{(k)} \stackrel{iid}{\sim} \pi$, 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 2.]

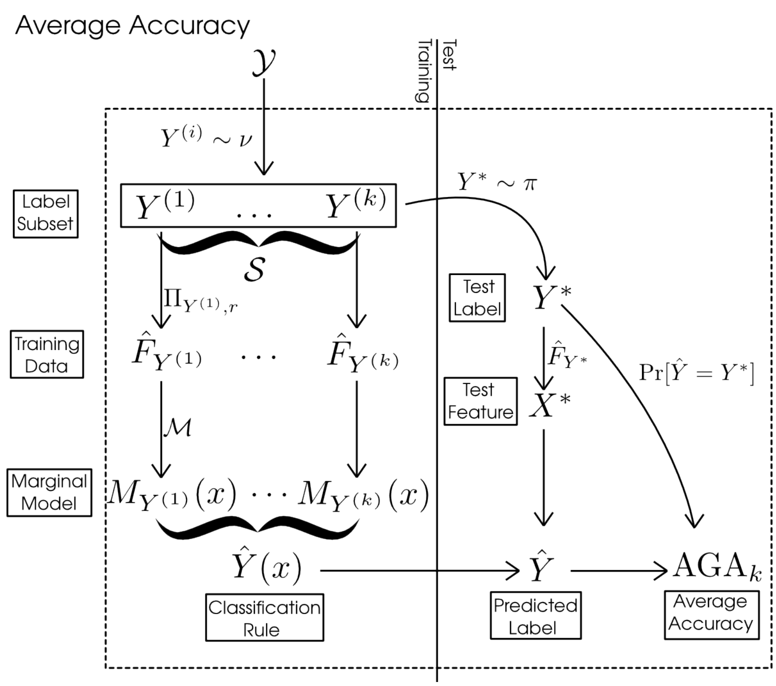


Figure 2: Average generalization accuracy

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:

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 3.

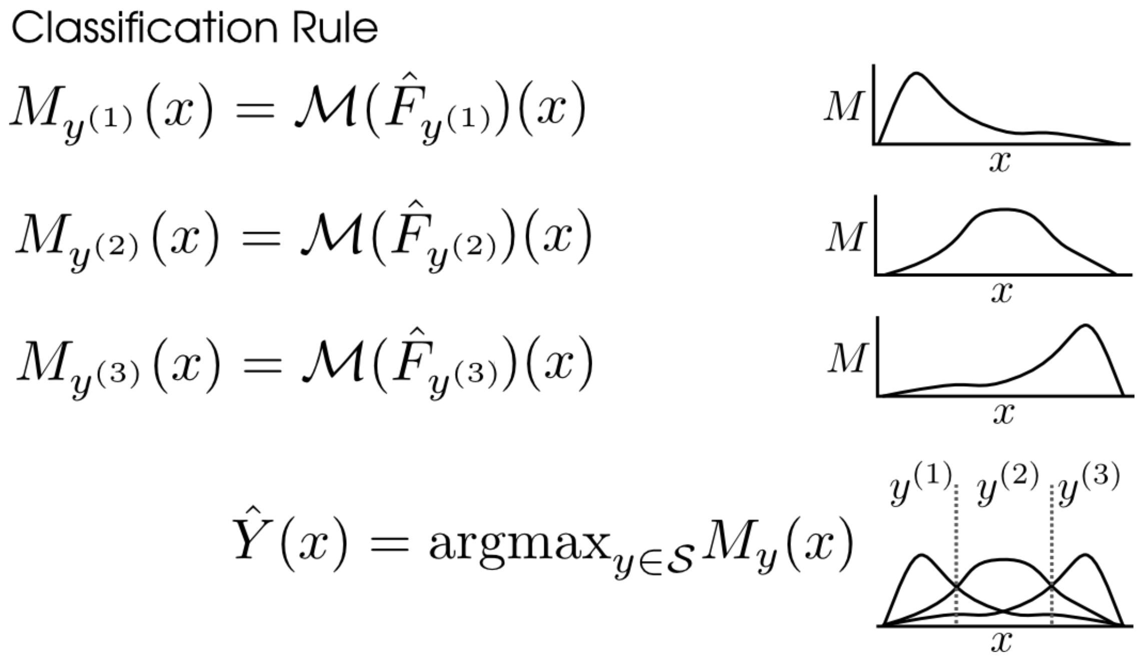


Figure 3: Classification rule [to be updated]

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 $j \neq i$.

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

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

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 $p(x_i|y)$

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

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

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

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

2.3 Toy Example: Bivariate normals

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

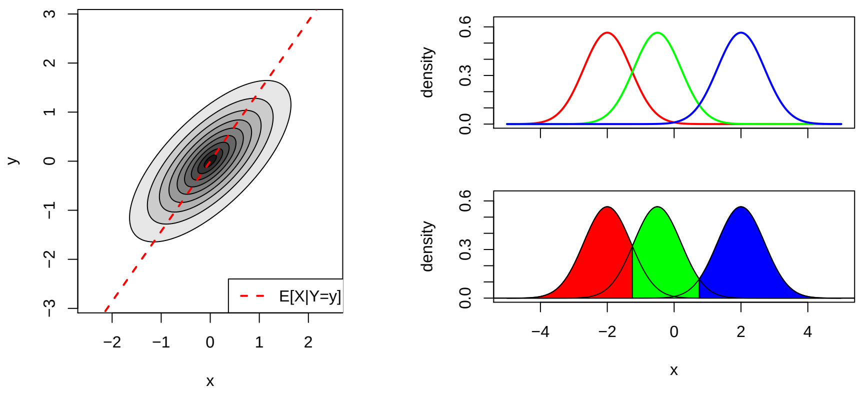
$$(Y, X) \sim N\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right),$$

as illustrated in figure 4(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 4(b, top). The Bayes rule assigns X^* to the class with the highest density $p(x|y_i)$, as illustrated by figure 4(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 4: (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 $\{y^{(1)}, \dots, y^{(k)}\}$ is given by

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

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 $\mathrm{GA}_k(y_1, \dots, y_k)$ for randomly drawn labels $Y_1, \dots, Y_k \stackrel{iid}{\sim} N(0, 1)$; the distributions of GA_k for $k = 2, \dots, 10$ are illustrated in figure 5. 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.

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.

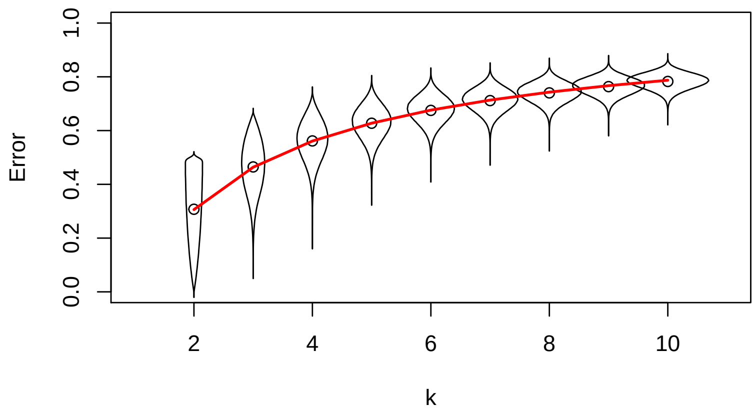


Figure 5: [change figure]The distribution of the classification risk for $k = 2, 3, \ldots, 10$ for the bivariate normal model with $\rho = 0.7$. Circles indicate the average classificatin risk; the red curve is the theoretically computed average risk.

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,

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

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 $[0,1]$:

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

$\square$.

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

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

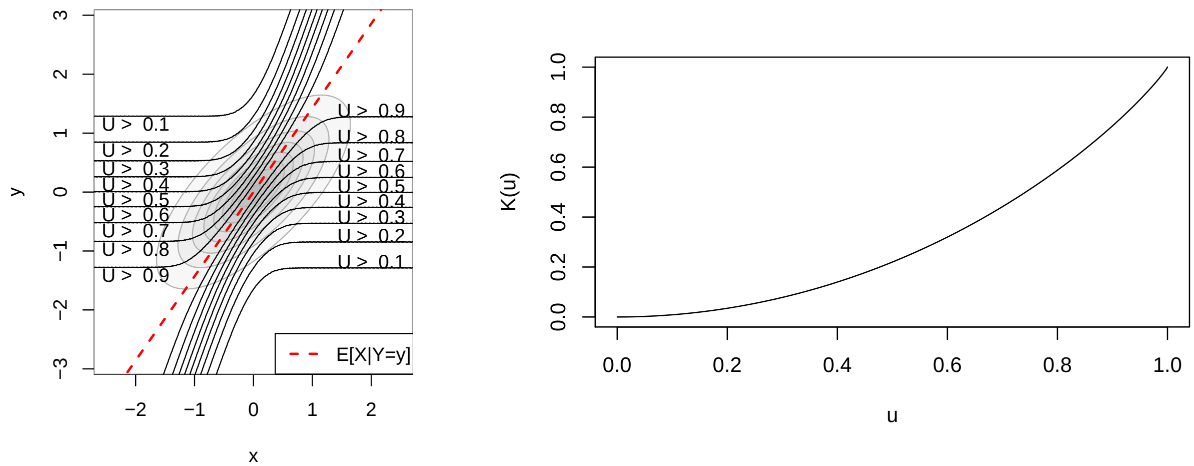
For this model, the favorability function $U_{x^*}(m_y)$ compares the distance between x^* and ρy to the distance between x^* and $\rho Y'$ for a randomly chosen distractor $Y' \sim N(0,1)$:

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

where Φ is the standard normal cumulative distribution function. Figure 6(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 6(b) for the current example with $\rho = 0.7$. The red curve in figure 5 was computed using the formula

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



(a) (b)

Figure 6: (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 7(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 7(b), where the function $\bar{D}(u)$ becomes smaller as ρ increases.

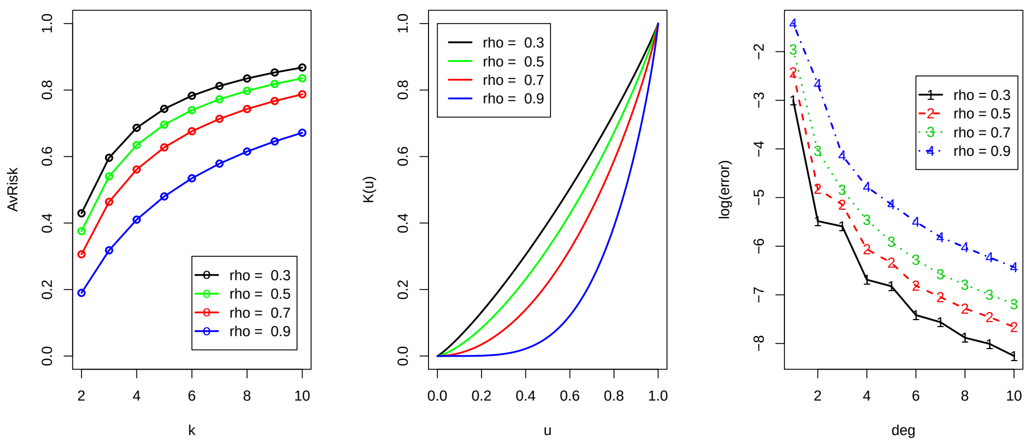
4. Estimation

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, $\text{ATA}_2, ..., \text{ATA}_{k_1}$ (see Eq 4), for its inputs.

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.



(a) (b) (c)

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

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

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

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

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

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

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

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 $(\text{ATA}_2, ..., \text{ATA}_{k_1})$. Specifically, we use a linear spline basis,

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

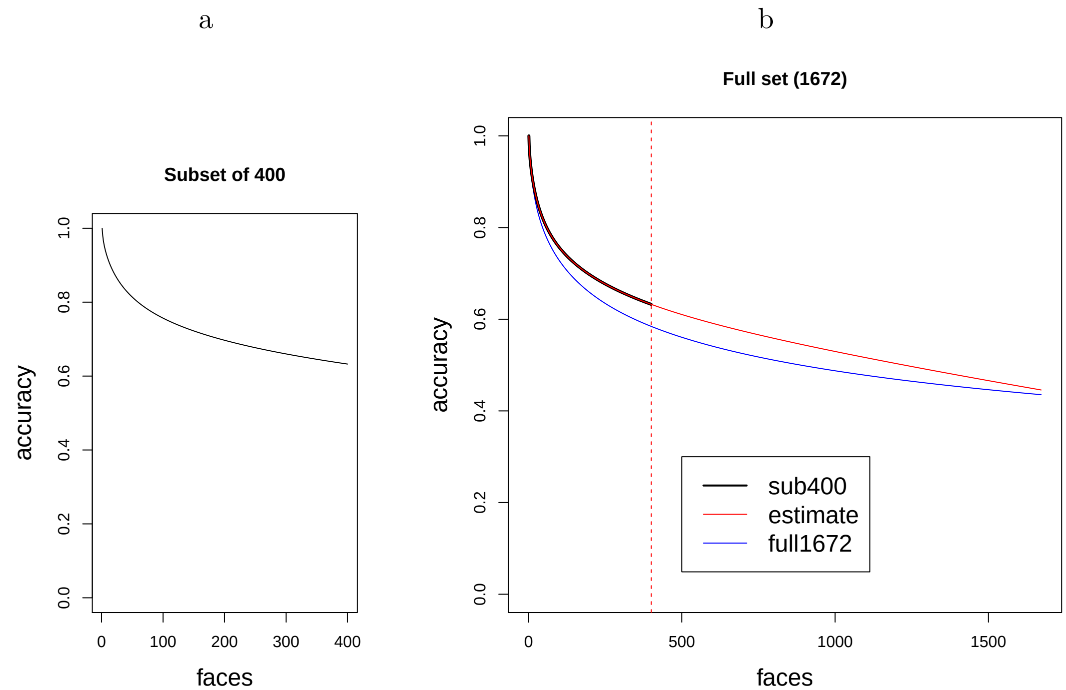


Figure 8: (a) The estimated average accuracy for $k = 2, \ldots, 400$ given a dataset of 400 faces subsampled from Labeled Faces in the Wild. (b) Estimated average accuracy for $k > 400$ on the same dataset, compared to the ground truth (average k-class test accuracy using all 1672 classes).

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

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

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 $A\hat{G}A_{1672}$. 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 [[MISSING]].

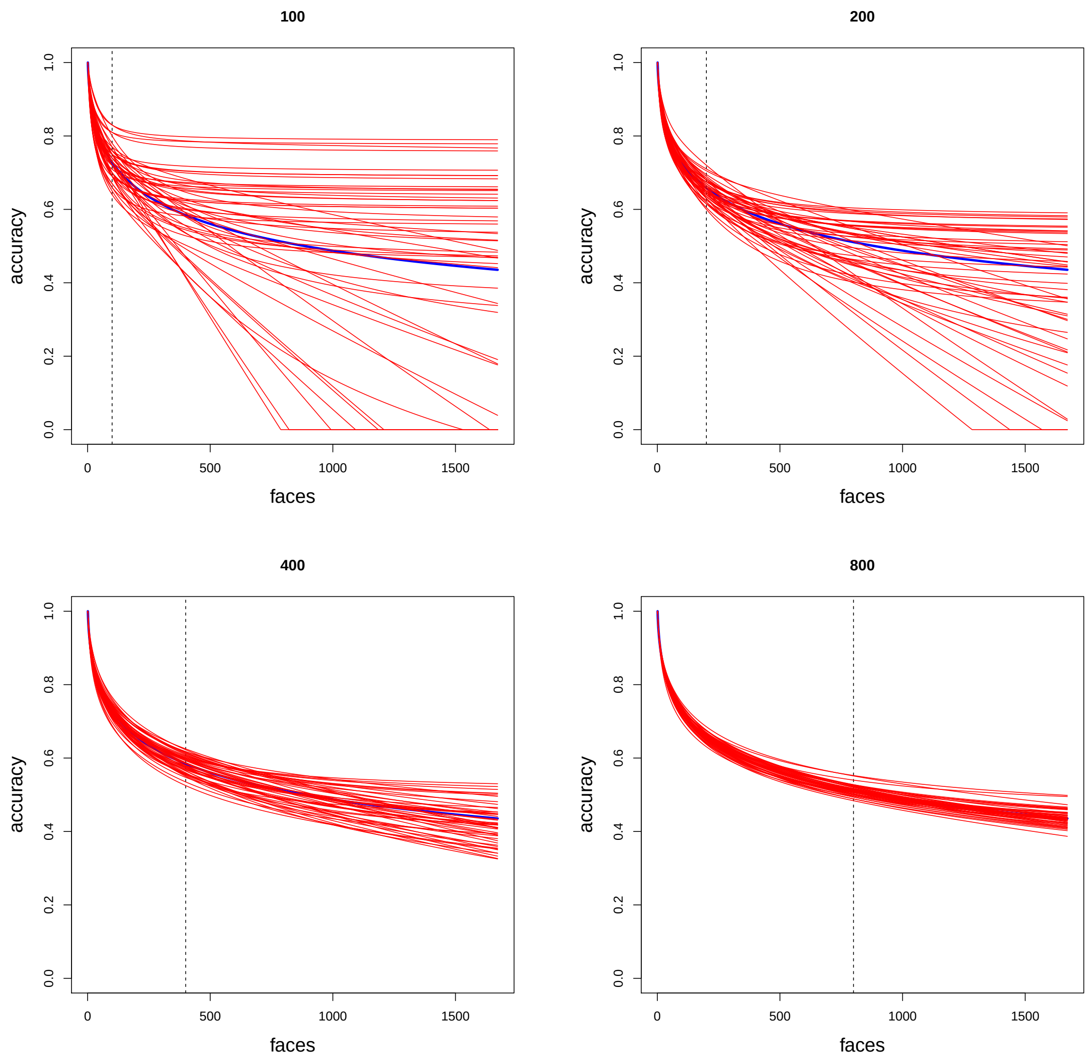


Figure 9: Estimated average accuracy using subsampled datasets of size k, compared to the ground truth (average k-class test accuracy using all 1672 classes).

6. Discussion

Consider approximating $\bar{D}(u)$ by polynomials, or some other function basis, it becomes relevant to consider how well $\bar{D}(u)$ can be approximated by such bases in realistic problems. We see in figure 7(c) that, at least in this toy problem, $\bar{D}(u)$ is well-aproximated by low-order polynomials. However, the approximation becomes less adequate as ρ increases. We can see visually why this is the case: as ρ increases, the curvature of $\bar{D}(u)$ near $u = 1$ increases. Hence, higher-degree polynomials become needed to capture the behavior near $u = 1$. More generally, we observe that in cases where classes are relatively well-separated, it becomes necessary to use increasingly precise approximations in order to extrapolate the average risk.

We have found that in many simulated examples, $D(u)$ appears to be monotone and convex, and we have also found that including the monotonicity and convexity constraint in the estimation procedure improves performance in practical examples.

While further work is still needed to better understand the performance of the proposed performance extrapolation method, both for marginal classifiers (which satisfy the theoretical assumptions) and non-marginal classifiers (which do not), the results obtained in these two examples are encouraging in that sense that useful predictions were obtained both for marginal and non-marginal classifiers.

Acknowledgments

We would like to acknowledge support for XXX

References

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.

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.