Extrapolating expected accuracies for multi-class classification

Charles Zheng, Rakesh Achanta and Yuval Benjamini

February 3, 2017

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.

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 multi-class classification problems: the algorithm observes an input x, and uses the classifier function f 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 $\mathcal{S}_1 \subset \cdots \subset \mathcal{S}_K \subset \mathcal{Y}$, where in the i-th problem, one constructs the classification rule $f^{(i)} : \mathcal{X} \to \mathcal{S}_i$. Supposing that (X, Y) have a joint distribution, define the misclassification error for the i-th problem as

$$\mathrm{Err}^{(i)} = \Pr[f^{(i)}(X) \neq Y | Y \in \mathcal{S}_i].$$

The problem of *performance extrapolation* is the following: using data from only $\mathcal{S}_k$, can one predict the misclassification error on the larger label set $\mathcal{S}_K$, with $K > k$? 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.

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

- Example 1: A researcher develops a classifier for the purpose of labelling images in 10,000 classes. However, for a pilot study, her resources are sufficient to tag only a smaller subset of these classes, perhaps 100. Can she estimate how well the algorithm work on the full set of classes based on an initial "pilot" subsample of class labels?

- 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. [1] 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.

- The stories just described can be viewed as a metaphor for typical paradigm of machine learning research, where academic researchers, working under limited resources, develop novel algorithms and apply

them to relatively small-scale datasets. Those same algorithms may then be adopted by companies and applied to much larger datasets with many more classes. In this scenario, it would be convenient if one could simply assume that performance on the smaller-scale classification problems was highly representative of performance on larger-scale problems.

Previous works have shown that generalizing from a small set of classes to a larger one is not straightforward. In a paper titled "What does classifying more than 10,000 Image Categories Tell Us," Deng and co-authors compared the performance of four different classifiers on three different scales: a small-scale (1,000-class) problem, medium-scale (7,404-class) problem, and large-scale (10,184-class) problem (all from ImageNet.) They found that while the nearest-neighbor classifier outperformed the support vector machine classifier (SVM) in the small and medium scale, the ranking switched in the large scale, where the SVM classifier outperformed nearest-neighbor. As they write in their conclusion, "we cannot always rely on experiments on small datasets to predict performance at large scale." 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 in 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 $\bar{K}(u)$, 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. Since marginal classifiers only comprise a minority of the classifiers used in practice, we applied our methods to a variety of generative and non-generative classifiers in simulations and in one OCR dataset. Our methods have varying success on generative and non-generative classifiers, but seem to work badly for neural networks.

[[Move to Discussion:]] In non-marginal classifiers, the classification rule has a joint dependence on the entire set of classes, and cannot be analyzed by conditioning on individual classes.

Our contribution.

To our knowledge, we are the first to formalize the problem of prediction extrapolation. We develop a general theory for prediction extrapolation under *general class priors* and under bounded cost functions. In addition, we investigate the special case of zero-one loss under uniform priors: we develop a pseudolikelihood-based estimation approach for this special case, and evaluate its performance in real data examples.

2 Framework

2.1 Problem Formulation

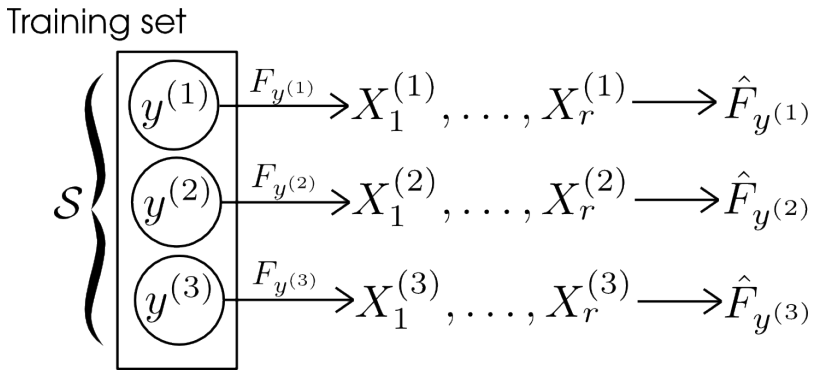

Figure 1: Training set

A *classification task* consists of a subset of labels, $\mathcal{S} \subset \mathcal{Y}$, and a prior distribution π over the label subset. Write $\mathcal{S} = \{y_1, \ldots, y_k\}$, where k is the number of classes. It is convenient to decouple the joint distribution

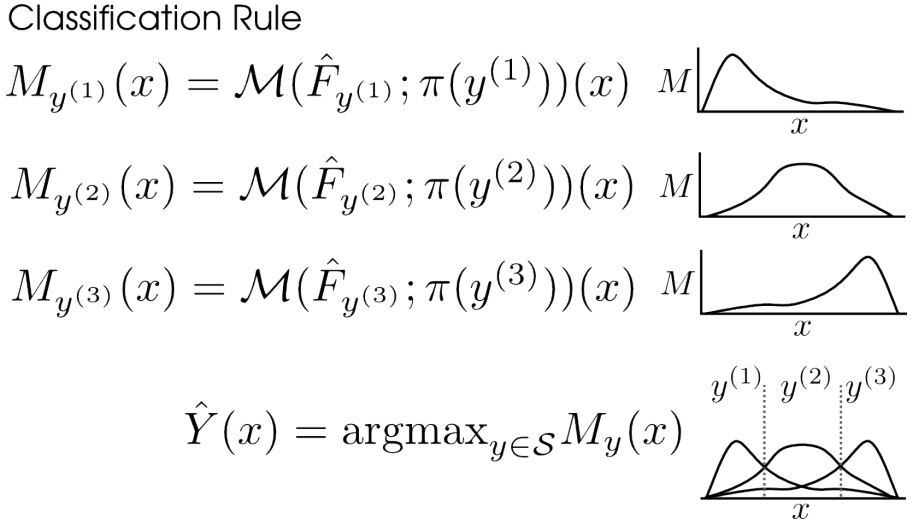

Figure 2: Classification rule

of (X, Y) into a prior distribution over the k labels $\mathcal{S}_k$, and the conditional distribution of elements, or feature vectors describing them, within a label class $X|Y = y \sim F_y$.

We would like to identify the sources of randomness in evaluating a classifier. First, there is the specific choice of k classes for the label set. Second, there is randomness in training the classifier for these classes, which comes from the use of a finite training set. Third, there is the randomness in the observed accuracy when testing the classifier on a test set. In order to separate these three sources, we need to clarify some terms used ambiguously in the classification literature.

We call a *classification rule* a function f which maps feature vectors $x \in \mathcal{X}$ to the set of labels $\mathcal{S}$:

$$f : \mathcal{X} \to \mathcal{S}.$$

For a random class Y drawn according to π and a feature vector drawn under F_Y, the loss of $\ell(f(X), Y)$ is obtained. Therefore, the *risk*, or expected loss, is

$$Risk(f; \pi, \ell) = \int \ell(f(X), Y) dF_Y d\pi.$$

For now, we will assume a 0–1 loss and a uniform prior over the labels in $\mathcal{S}$;

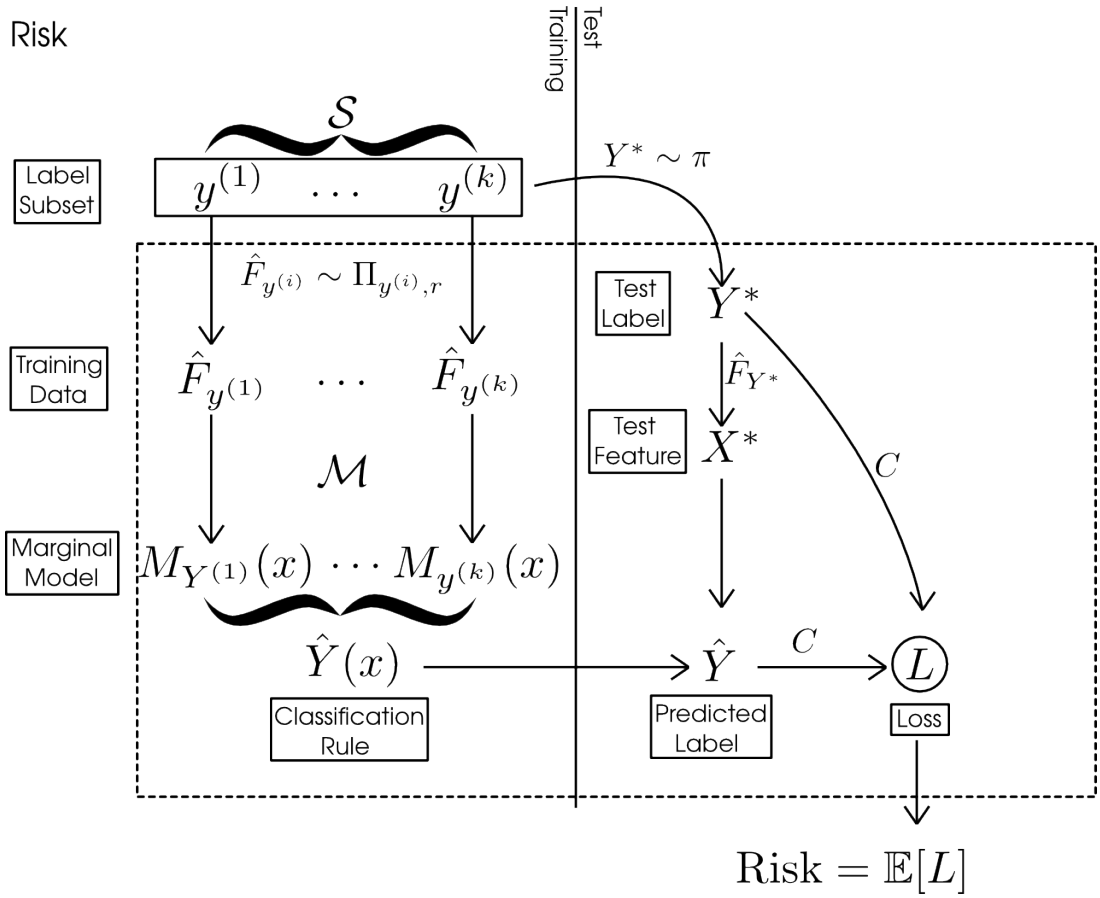

Figure 3: Classification risk

the risk can therefore be re-written as

$$Risk(f; \mathcal{S}, \ell_{01}) = \frac{1}{k} \sum_{y_i \in \mathcal{S}} P(f(X) \neq y_i; X \sim F_{y_i})$$

The classification rule itself can be seen as a random function that is based on the choice of the training set. For convenience, assume that the training set is composed of r_{train} i.i.d examples for each label $y \in \mathcal{S}$ (a total of $k \times r_{train}$). An i.i.d. sample of size r, $X_1, \ldots, X_r \sim F_y$ can also be described as an empirical distribution, using the shorthand $\hat{F}_y$: [[TODO: Should r appear in the notation for $\hat{F}_y$? It is unnecessary to include r in the notation since

we never discuss any situation where r changes.]].

$$\hat{F}_y = \frac{1}{r}\sum_{i=1}^{r} \delta_{x_i^{(y)}}.$$

[[TODO: IS this needed Let $\Pi_{y,r}$ denote the sampling distribution of $\hat{F}_y$. It is used in the following equation defining Risk(F; pi).]]

A *classification model* or *classifier* $\mathcal{F}$ is the algorithm or procedure for producing classification rules given a vector of empirical distributions $(\hat{F}_y)_{y\in\mathcal{S}}$ and given a prior on the labels π. The model maps the empirical distributions and the prior to a classification rule f (Figure 2).

We can therefore describe the r-risk of the model $\mathcal{F}$ as the expected risk of a classification rule $\hat{f}$ trained using a sample of size r from each of labels in $\mathcal{S}_k$. That is,

$$\text{Risk}_r(\mathcal{F};\pi) = \int \text{Risk}(\mathcal{F}(\{\hat{F}_y\}_{y\in\mathcal{S}};\pi)) \prod_{y\in\mathcal{S}} d\Pi_{y,r}(\hat{F}_y).$$

Figure 3 illustrates the variables involved in defining the risk.

The problem of *performance extrapolation* can now be formally defined as follows: Given a known classification task $\mathcal{S}_{k_1} = \{y_1, ..., y_{k_1}\}$ with an observed r_{train} training set and r_{test} test set per class, can we estimate the expected r_{train}-risk of the classifier $\mathcal{F}$ on a second classification task with k_2 labels and an r_{train} training set per class.

2.2 Assumptions

Implicit in our definition of performance extrapolation is that the new set of k_2 is partially or fully unknown at the time of the extrapolation. Therefore, the extrapolation must account also for the randomness in the choice of labels. We will assume that the labels in the two classification tasks are comparable.

Assumption 1A Let $\mathcal{S}_{k_1}, \mathcal{S}_{k_2}$ be the label sets for the first and second classification tasks. Then $\mathcal{S}_{k_1}, \mathcal{S}_{k_2}$ are i.i.d. samples from an infinite population π_Y.

Assumption 1B For $y, y' \sim \pi_Y$,

$$P(y = y') = 0$$

[[TODO: Assumption 1B is implied by the previous text. Is it really necessary as an assumption, or can we deal with it without loss of generality. Assumption 1B is implied by the tie-breaking condition, so we can drop it here.]]

Comments:

1. These assumption are most easily satisfied by taking $\mathcal{Y}$ to be a continuous space and letting π_Y be a density over $\mathcal{Y}$. However, a discrete space with a small enough probability for the classes would work well.

2. Note that here we assumed that the label subsets $\mathcal{S}_{k_1}$ and $\mathcal{S}_{k_2}$ are independent and disjoint. An alternative assumption would be that $\mathcal{S}_{k_1} \subset \mathcal{S}_{k_2}$ with $\mathcal{S}_{k_1}$ being a subsample of $\mathcal{S}_{k_2}$: this assumption can also be addressed, as we will discuss later.

3. In practice, $\mathcal{S}_{k_1}$ is often a convenience sample meant to be similar to $\mathcal{S}_{k_2}$. The theory will be relevant in so far as the assumptions approximate well the true sampling similarity between the $\mathcal{S}_{k_1}$ and $\mathcal{S}_{k_2}$.

4. We can imagine other sampling mechanisms designed to make $\mathcal{S}_{k_1}$ a representative sample from the population, e.g. by stratifying. In this paper we do not discuss these more complex sampling schemes.

[TODO: Is the following paragraph really needed? Somehow we need prior probabilities to be assigned in a sensible way. An easier alternative would just be to have uniform prior probabilities by default, but this would work really poorly in practice.] Now recall that the prior probabilities π_i for each classification task are free for the user to define, unlike the population distribution π_0 of class labels which is assumed to have an objective existence. Since the subsampled or 'small-scale' classification tasks (with label subsets $\mathcal{S}_i$) are presumably intended to approximate the 'full' classification problem (with the label set $\mathcal{Y}$), and since the prior in the full problem is π_0, a sensible choice would be to choose

$$\pi_i(y) = \frac{\pi_0(y)}{\sum_{y' \in \mathcal{S}_i} \pi_0(y')}.$$

as the prior for the ith classification task. As it turns out, such a prior assignment also simplifies the theory, so we will assume that π_i is defined according to the above. [[END of TODO]]

Our analysis will also rely on a property of the classification model. We do not want the classifier to rely to strongly on complicated interaction between the labels in the set. We therefore propose the following property of marginal separability for classification models:

Definition 2.1 *1. The classification rule f is called a* marginal rule *if*

$$f(x) = argmax_{y \in \mathcal{S}} m_y(x),$$

where each function m_y maps $\mathcal{X}$ to $\mathbb{R}$.

2. A classification model that produces marginal classification rules is called a marginal classification model.

In words, a marginal classification rule produces a score for each label, and chooses the label with the highest marginal score. The marginal model converts empirical distributions $\hat{F}_y$ over $\mathcal{X}$ into the marginal scoring function m_y. The *marginal* property allows us to prove strong results about the accuracy of the classifier under i.i.d. sampling assumption. [TODO: Do we need both the marginal model and the marginal classifier? $\mathcal{M}$. We can omit the definition of marginal classifier here, but later we have to have some discussion of marginal vs non-marginal classifiers. So we can move the definition to the discussion, maybe.]]

Comments:

1. The marginal model includes several popular classifiers. A primary example for a marginal model is the estimated Bayes classifier. Let $\hat{f}_y$ be a density estimate obtained from the empirical distribution $\hat{F}_y$. Then, we can use the estimated densities of each class to produce the marginal functions:

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

   if class probabilities are uniform, or

   $$m_y^{EB}(x, p_y) = \log(p_y) + \log(\hat{f}_y(x))$$

   if $\pi = (p_y)_{y \in \mathcal{S}}$. The resulting empirical approximation for the Bayes classifier would be

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

2. Both the Quadratic Discriminant Analysis and the naive Bayes can be seen as specific instances of an estimated Bayes classifier [TODO: is this correct? Gaussian mixture model with flat prior for mu and sigma for each cluster results in QDA]. For QDA, the classification function is given by

$$\mathcal{Q}_{QDA}(\hat{F}, y) = -(y - \mu(\hat{F}))^T \Sigma(\hat{F})^{-1}(y - \mu(\hat{F})) - \log\det(\Sigma(\hat{F})),$$

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

$$\mathcal{Q}_{NB}(\hat{F}, y) = \sum_{i=1}^{n} \log \hat{f}_i(y_i),$$

where $\hat{f}_i$ is a density estimate for the i-th component of $\hat{F}$. [TODO: Examples of classifiers that are not Such as multinomial logistic regression.]

[TODO: where do you want to treat the empirical vs. true class probabilities? I guess you are referring to the distinction between population probabilities and sampling probabilities. If you really want to simplify things, we should just assume that population probabilities, sampling probabilities, and prior probabilities are all uniform, and we can discuss extensions in the same section as for general cost function.]

2.3 Definition of average risk

Since the classification tasks are randomly generated, the r-repeat risk becomes a *random variable* which depends on the random label subset $\mathcal{S}$.

Therefore, define the k-class, r-repeat *average risk* of classifier $\mathcal{F}$ with prior weights π as

$$\text{AvRisk}_{k,r}(\mathcal{F}; \pi) = \mathbf{E}[\text{Risk}_k(\mathcal{F}; \pi)]$$

where the expectation is taken over the distribution of $\mathcal{S} = (Y^{(1)}, \ldots, Y^{(k)})$ when $Y^{(i)} \overset{iid}{\sim} \nu$.

As we can see from Figure 4, the average risk is obtained by averaging over four randomizations:

A1. Drawing the label subset $\mathcal{S}$.

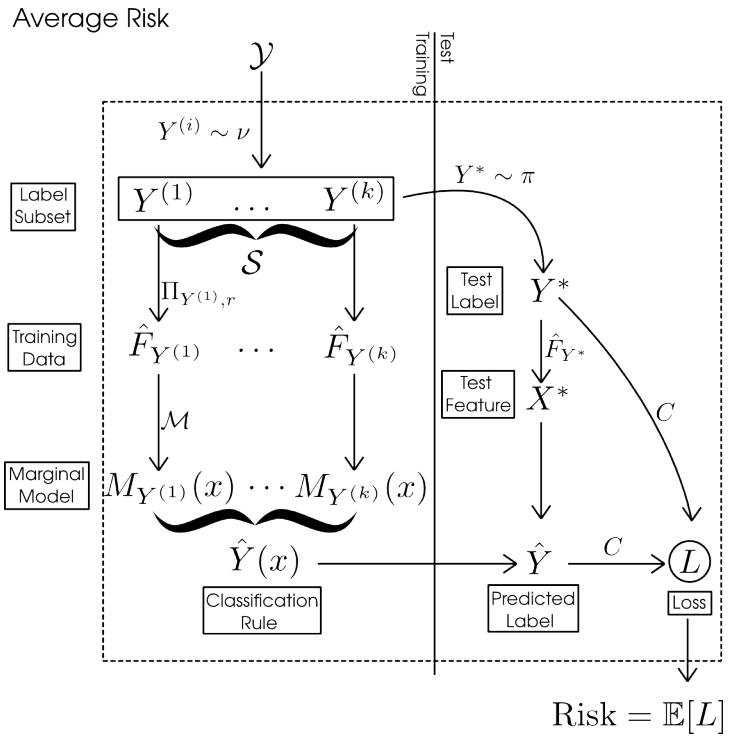

Figure 4: Average risk

A2. Drawing the training dataset.

A3. Drawing Y^* from $\mathcal{S}$ according to π.

A4. Drawing X^* from F_{X^*}.

For the sake of developing a better intuition of the average risk, it is helpful to define a random variable called the *loss*, which is the cost incurred by a single test instance. The loss is determined by quantities from all four randomization steps: the label subset $\mathcal{S} = \{Y^{(1)}, \dots, Y^{(k)}\}$, the training samples $\hat{F}_{Y^{(1)}}, \dots, \hat{F}_{Y^{(k)}}$, and the test point (X^*, Y^*). Formally, we write

$$L = C(\mathcal{F}(\{\hat{F}_y\}_{y \in \mathcal{S}}; \pi)(X^*), Y^*).$$

Now note that the k-class, r-repeat average risk is the expected loss,

$$\text{AvRisk}_{k,r,\nu}(\mathcal{F}) = \mathbf{E}[L] = \mathbf{E}[C(\mathcal{F}(\{\hat{F}_y\}_{y \in \mathcal{S}}; \pi)(X^*), Y^*)]. \tag{1}$$

where the expectation is taken over the joint distribution of all the quantities $\{Y^{(1)}, \ldots, Y^{(k)}, \hat{F}_{Y^{(1)}}, \ldots, \hat{F}_{Y^{(k)}}, (X^*, Y^*)\}$.

We will aim to develop a method for estimating the *average risk*. In the case where the classification tasks are independently generated, the average risk is the best predictor (in mean-squared error) for the (random) risk.

Figure 5: Average risk

3 Performance extrapolation for marginal classification models

Having outlined our assumption for randomized label subsets, the focus of our theory moves towards understanding the k-class average risk: that is, the expected risk of $\mathcal{F}$ when a random subset $\mathcal{S}$ of size k is drawn.

We obtain a method for estimating the risk in the second classification task using data from the first. The insight behind our estimation method is obtained via an analysis of the average risk of the classification task.

3.1 Easy special cases

Let us first mention two easy special cases, which can be handled using existing machine learning methodology.

3.1.1 Equal numbers of classes

In the special case where $k_1 = k_2 = k$: that is, where the label subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ are the same size, it is clear to see that any unbiased estimate of the risk of the classifier $\mathcal{F}$ for the first classification problem is an unbiased estimate of the average k-class risk. The *test risk* gives one such unbiased estimate of the average k-class risk.

Recall that the data consists of class labels $y^{(i)}$ and prior weights $\pi_1(y^{(i)})$ for $i = 1, \ldots, k_1$, as well as training sample $\hat{F}_{y^{(i)}}$ and test sample $(x_1^{(i)}, \ldots, x_{r_{test}}^{(i)})$ for $i = 1, \ldots, k_1$.

For any given test observation $x_j^{(i)}$, we obtain the predicted label $\hat{y}_j^{(i)}$ by finding the class with the highest margin $M_{i,j,\ell}$,

$$\hat{y}_j^{(i)} = y_{\text{argmax}_\ell M_{i,j,\ell}},$$

where

$$M_{i,j,\ell} = \mathcal{M}(\hat{F}_{y^{(\ell)}}; \pi_1(y^{(\ell)}))(x_j^{(i)}).$$

The test risk is the average cost over test observations,

$$\text{Test Risk} = \frac{1}{r_{test}k} \sum_{i=1}^{k} \sum_{j=1}^{r_{test}} C(\hat{y}_j^{(i)}, y^{(i)}). \tag{2}$$

For each test observation, define the ranks of the margins by

$$R_{i,j,\ell} = \sum_{m \neq \ell} I\{M_{i,j,\ell} \geq M_{i,j,m}\}.$$

Therefore, $\hat{y}_j^{(i)}$ is equal to ℓ if and only if $R_{i,j,\ell} = k$. Thus, an equivalent expression for test risk is

$$\text{Test Risk} = \frac{1}{r_{test}k} \sum_{i=1}^{k} \sum_{\ell=1}^{k} \sum_{j=1}^{r_{test}} C_{ij} I\{R_{ij\ell} = k\}. \tag{3}$$

where

$$C_{ij} = C(y^{(j)}, y^{(i)}).$$

Besides the test risk, other methods, such as cross-validation, can also be used to obtain estimates of the average k-class risk.

3.1.2 Fewer classes

Suppose we have data for k_1 classes, and we wish to estimate AvRisk_{k_2} for $k_2 \leq k_1$. Let $\mathcal{S}_1 = \{y_1, \ldots, y_{k_1}\}$. To obtain a classification problem with k_2 classes, we can simply pick a subset S of size k_2 from $\mathcal{S}_1$, and throw away all the training and test data from the other classes $\mathcal{S} \setminus S$. Then, the test risk (3) gives an unbiased estimate of the AvRisk_{k_2}.

Of course, one could obtain a better estimator of the average risk by averaging over all the subsets $S \subset \mathcal{S}_1$ of size k_2. For general classifiers, this may require retraining a classifier over each subset. However, for marginal classifiers, one can compute the average test risk over all $\binom{k_1}{k_2}$ subsets easily.

The reason why the efficient computation is possible is because the test risk for each subproblem can be determined by looking at the margins $M_{i,j,\ell}$, which remain the same as long as both i and ℓ are included in the subsample S.

The computational trick is to look at each combination of test observation $x_j^{(i)}$ and class label $y^{(\ell)}$, and to count count the number of subsets $N_{i,j,\ell}$ where (i) both i and ℓ are included in S, and (ii) $\hat{y}_j^{(i)} = y^{(\ell)}$. Then it should be clear that the average test risk over all subsets is equal to

$$\text{AvTestRisk}_{k_2} = \frac{1}{\binom{k_1}{k_2}} \frac{1}{r_{test} k_2} \sum_{i=1}^{k_1} \sum_{\ell \neq i} \sum_{j=1}^{r_{test}} C_{i\ell} N_{i,j,\ell}. \tag{4}$$

Now it is just a matter of simple combinatorics to compute $N_{i,j,\ell}$. We require both $y^{(i)}$ and $y^{(\ell)}$ to be included in S. This implies that if $M_{i,j,i} > M_{i,j,\ell}$, then $y^{(\ell)}$ will never have the highest margin in any of those subsets, so $N_{i,j,\ell} = 0$.

Otherwise, there are $R_{i,j,\ell} - 1$ elements in $\mathcal{S}_1$ with a lower margin than $y^{(\ell)}$. Since $i \neq \ell$, then there are $k_2 - 2$ elements in $S \setminus \{i, \ell\}$, so therefore $N_{i,j,\ell} = \binom{R_{i,j,\ell}-2}{k_2-2}$. Therefore, we can write

$$N_{i,j,\ell} = I\{R_{i,j,\ell} > R_{i,j,i}\} \binom{R_{i,j,\ell} - 2}{k_2 - 2} \tag{5}$$

Therefore, the challenging case is when $k_2 > k_1$: we want to predict the performance of the classification model in a setting with more labels than we currently see in the training set.

3.2 Analysis of the average risk

As we pointed out in the previous section, the challenging case for the analysis is the "undersampled" regime where we wish to predict the loss on a larger label set. Given data with k_1 classes, we already have means to estimate the average risk for all $k \leq k_1$, so the challenge is to understand how the risk will "extrapolate" to $k > k_1$. Hence, the goal of the current analysis is to isolate the effect of k, the size of the label subset, on the average risk.

Our strategy is to analyze the average risk (1) by means of *conditioning on* the true label and its training sample, $(y^*, \hat{F}_{y^*})$, and the test feature x^* while *averaging* over all the other random variables. Define the *conditional average risk* $\text{CondRisk}_k((y^*, \hat{F}_{y^*}), x^*)$ as

$$\text{CondRisk}_k((y^*, \hat{F}_{y^*}), x^*) = \mathbf{E}[L | Y^* = y^*, X^* = x^*, \hat{F}_{Y^*} = \hat{F}_{y^*}].$$

Figure 5 illustrates the variables which are fixed under conditioning and the variables which are randomized. Compare to figure 4.

Without loss of generality, we can write the label subset $\mathcal{S} = \{Y^*, Y^{(1)}, \ldots, Y^{(k-1)}\}$. Note that due to independence, $Y^{(1)}, \ldots, Y^{(k-1)}$ are still i.i.d. from π_0 even conditioning on $Y^* = y^*$. Therefore, the conditional risk can be obtained via the following alternative order of randomizations:

C0. Fix $y^*, \hat{F}_y^*$, and x^*. Note that $M_{y^*}(x^*) = \mathcal{M}(\hat{F}_{y^*}; \pi(y^*))(x^*)$ is also fixed.

C1. Draw the *incorrect labels* $Y^{(1)}, \ldots, Y^{(k)}$ i.i.d. from ν. (Note that $Y^{(i)} \neq y^*$ with probability 1 due to the continuity assumptions on $\mathcal{Y}$ and ν.)

C2. Draw the training samples for the incorrect labels $\hat{F}_{Y^{(1)}}, \ldots, \hat{F}_{Y^{(k-1)}}$. This determines

$$\hat{Y} = \text{argmax}_{y \in \mathcal{S}} M_y(x^*)$$

and hence

$$L = C(\hat{Y}, y^*).$$

Compared to four randomization steps listed in section 2.3, we have essentially conditioned on steps A3 and A4 and randomized over steps A1 and A2.

Having defined the conditional average risk, we will now further decompose it to expose its dependence on k. We make the following additional technical assumptions:

- *Scaling property of margins*: if $\mathcal{M}(\hat{F}_1, \pi_1)(x) > \mathcal{M}(\hat{F}_2, \pi_2)(x)$ then also $\mathcal{M}(\hat{F}_1, c\pi_1)(x) > \mathcal{M}(\hat{F}_2, c\pi_2)(x)$.
- *Tie-breaking condition*: for all $x \in \mathcal{X}$, $\mathcal{M}(\hat{F}_Y, \pi_1)(x) = \mathcal{M}(\hat{F}_{Y'}, \pi_2)(x)$ with zero probability for $Y \neq Y'$ drawn from ν.

The scaling property of margins is satisfied by most of the marginal classifiers which are used in practice, and as such we do not consider it to be a strong assumption. Meanwhile, the tie-breaking condition is a technical assumption which allows us to neglect the specification of a tie-breaking rule in the case that margins are tied. In practice, one can simply break ties randomly, which is mathematically equivalent to adding a small amount of random noise ϵ to the function $\mathcal{M}$.

Now, in order to analyze the k-class behavior of the conditional average risk, we begin by considering the *two-class* situation.

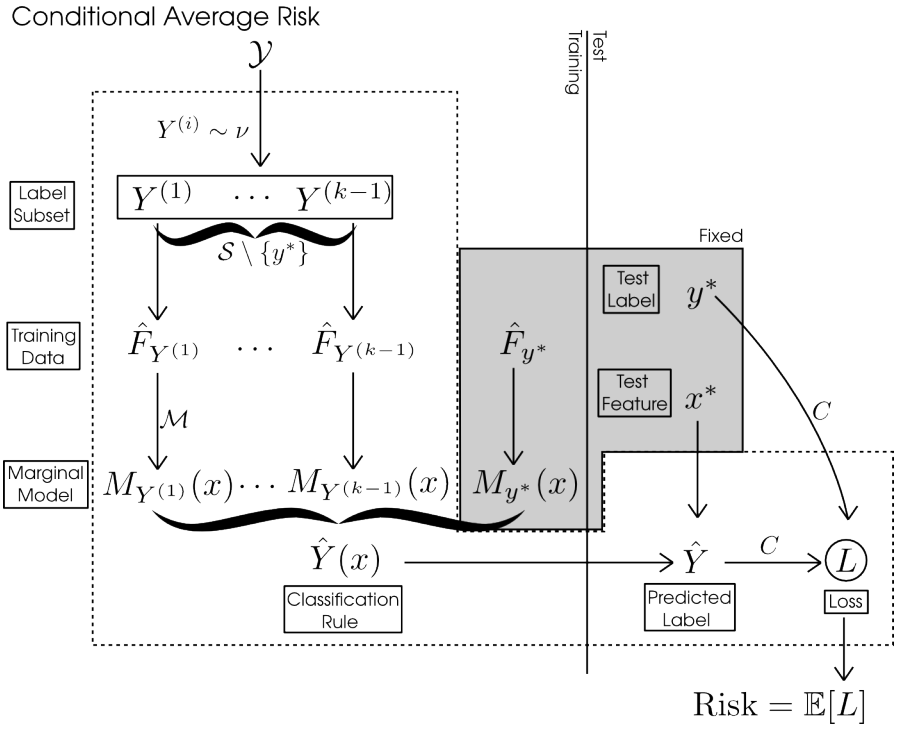

Figure 6: Conditional average risk

In the two-class situation, we have a true label y^* and one incorrect label, Y. Define the *U-function* $U_{x^*}(y^*, \hat{F}_{y^*})$ as the *probability of correct classification* in the two-class case. The classification is correct if the margin $M_{y^*}(x^*)$ is greater than the margin $M_Y(x^*)$, and incorrect otherwise. Since we are fixing x^* and $(y^*, \hat{F}_{y^*})$, the probability of correct classification is obtained by taking an expectation:

$$U_{x^*}(y^*, \hat{F}_{y^*}) = \Pr[M_{y^*}(x^*) > \mathcal{M}(\hat{F}_Y, \pi_0(Y))(x^*)] \tag{6}$$

$$= \int_{\mathcal{Y}} I\{M_{y^*}(x^*) > \mathcal{M}(\hat{F}_y, \pi_0(y))(x)\} d\Pi_{y,r}(\hat{F}_y) d\pi_0(y). \tag{7}$$

See also figure 6 for an graphical illustration of the definition.

An important property of the U-function, and the basis for its name, is that the random variable $U_x(Y, \hat{F}_Y)$ for $Y \sim \nu$ and $\hat{F}_Y \sim \Pi_{Y,r}$ is uniformly distributed for all $x \in \mathcal{X}$. This is proved in Lemma A.1 in the appendix.

Now, we will see how the U-function allows us to understand the k-class case. Suppose we have true label y^* and incorrect labels $Y^{(1)}, \ldots, Y^{(k-1)}$.

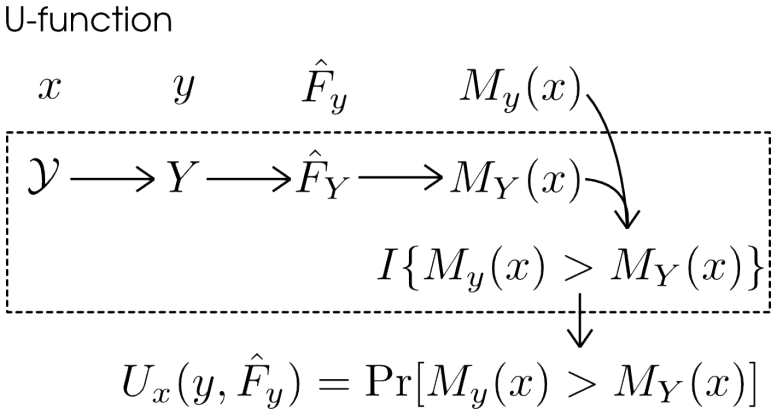

Figure 7: U-functions

Note that the U-function $U_{x^*}(y, \hat{F}_y)$ is monotonic in $M_y(x^*)$. Therefore,

$$\hat{Y} = \text{argmax}_{y \in \mathcal{S}} M_y(x^*) = \text{argmax}_{y \in \mathcal{S}} U_{x^*}(y, \hat{F}_y).$$

Therefore, we have a correct classification if and only if the U-function value for the correct label is greater than the maximum U-function values for the incorrect labels:

$$\Pr[\hat{Y} = y^*] = \Pr[U_{x^*}(y^*, \hat{F}_{y^*}) > \max_{i=1}^{k-1} U_{x^*}(Y^{(i)}, \hat{F}_{Y^{(i)}})] = \Pr[u^* > U_{max}].$$

where $u^* = U_{x^*}(y^*, \hat{F}_{y^*})$ and $U_{max,k-1} = \max_{i=1}^{k-1} U_{x^*}(Y^{(i)}, \hat{F}_{Y^{(i)}})$. But now, observe that we know the distribution of $U_{max,k-1}$! Since $U_{x^*}(Y^{(i)}, \hat{F}_{Y^{(i)}})$ are i.i.d. uniform, we know that

$$U_{max,k-1} \sim \text{Beta}(k-1, 1). \tag{8}$$

We now have the insights needed to analyze the simplest special case: zero-one loss.

Special case: 0-1 loss. For zero-one loss, which is $C(y, y') = I\{y = y'\}$, we have $L = 1$ if and only if $U_{max} > u^*$ and $L = 0$ otherwise. Therefore, the conditional average risk is

$$\text{CondRisk}_k((y^*, \hat{F}_{y^*}), x^*) = \Pr[U_{max} > u^*] = \int_{u^*}^{1} (k-1) u^{k-2} du.$$

Now the average risk can be obtained by integrating over the distribution of $U^* = U_{x^*}(y^*, \hat{F}_{y^*})$. We have

$$\begin{aligned}\text{AvRisk}_k &= \mathbf{E}[\int_{U^*}^1 (k-1)u^{k-2}du] \\ &= \mathbf{E}[\int_0^1 I\{u \geq U^*\}(k-1)u^{k-2}du] \\ &= (k-1)\int_0^1 \Pr[U^* \leq u]u^{k-2}du.\end{aligned}$$

Or equivalently,

$$\text{AvRisk}_{k,r,\nu}((y^*, \hat{F}_{y^*}), x^*) = (k-1)\int \bar{K}(u)u^{k-2}du.$$

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

$$\bar{K}(u) = \Pr[U_{x^*}(y^*, \hat{F}_{y^*}) \leq u].$$

We have expressed the average risk expressed as a weighted integral of a certain function $\bar{K}(u)$ defined on $u \in [0,1]$. We have clearly isolated the part of the average risk which is independent of k–the univariate function $\bar{K}(u)$, and the part which is dependent on k–which is the density of U_{max}.

In section 3.3, we will develop estimators of $\bar{K}(u)$ in order to estimate the k-class average risk. But now let us return to the general case.

General loss functions. The case for general cost functions is somewhat more complicated, since knowledge of U_{max} is not sufficient to determine L. In short, this is because U_{max} by itself is insufficient to determine $\hat{Y}$, and therefore $L = C(\hat{Y}, y^*)$. However, we can resolve this issue by noting that for the purposes of computing the expected loss, it suffices to have the *conditional distribution* of $\hat{Y}$ given U_{max}. Even though U_{max} does not deterministically map onto a unique $\hat{Y}$, it determines a conditional distribution of $\hat{Y}$ which allows us to compute $\mathbf{E}[L|U_{max}, x^*, y^*, \hat{F}_{y^*}]$.

Now, a key fact is that the conditional distribution of $\hat{Y}$ given U_{max} *does not depend* on k. To see this fact, suppose without loss of generality that $\hat{Y} = Y^{(k-1)}$. Then the joint density of $Y^{(1)}, \ldots, Y^{(k-1)}$ given $U_{max} = u$ can be written

$$p(y^{(1)}, \ldots, y^{(k-1)}) \propto \nu(y^{(k-1)}) \frac{d}{dt} \Pr[U_{x^*}(y^{(k-1)}, \hat{F}_{y^{(k-1)}}) \leq t]|_{t=u} \prod_{i=1}^{k-2} \nu(y^{(i)}) \Pr[U_{x^*}(y^{(k-1)}, \hat{F}_{y^{(k-1)}}) < u]$$

up to a normalizing constant. Note that the term $\frac{d}{dt}\Pr[U_{x^*}(y^{(k-1)}, \hat{F}_{y^{(k-1)}}) \leq t]$ is the density of the random variable $U_{x^*}(Y^{(k-1)}, \hat{F}_{Y^{(k-1)}})$. From the density, we can see that $Y^{(1)}, \ldots, Y^{(k-1)}$ are conditionally independent given $U_{max} = u$, hence the marginal density of $\hat{Y} = Y^{(k-1)}$ can be written

$$p(\hat{y}) \propto \nu(\hat{y}) \frac{d}{dt} \Pr[U_{x^*}(y^{(k-1)}, \hat{F}_{y^{(k-1)}}) \leq t]|_{t=u}.$$

The only property of the conditional distribution of $\hat{Y}|U_{max} = u$ that is needed is the expectation of $L = C(\hat{Y}, y^*)$. Therefore, define the *conditional expected loss* $K((y^*, \hat{F}_{y^*}), x^*, u)$ by

$$K((y^*, \hat{F}_{y^*}), x^*, u) = \begin{cases} 0 \text{ if } u < u^* \\ \mathbf{E}[C(\hat{Y}, y^*)|U_{max} = u, x^*, y^*, \hat{F}_{y^*}] \text{ otherwise.} \end{cases} \tag{9}$$

We have the two cases $u < u^*$ and $u > u^*$ since when $U_{max} < u^*$, the correct label is chosen and the loss is zero. Otherwise, an incorrect label is chosen, and the expected loss must be calculated using the conditional distribution of $\hat{Y}$.

Again, since the conditional distribution of $\hat{Y}|U_{max}, x^*, (y^*, \hat{F}_{y^*})$ is independent of k, the conditional cost function is also independent of k.

With the conditional cost function and the distribution of U_{max} both in hand, we can compute the average conditional risk

$$\text{CondRisk}_k((y^*, \hat{F}_{y^*}), x^*) = (k-1) \int K((y^*, \hat{F}_{y^*}), x^*, u) u^{k-2} du.$$

Now the average risk can be obtained by integrating over $(Y^*, \hat{F}_{Y^*})$, and X^*.

$$\text{AvRisk}_{k,r}((y^*, \hat{F}_{y^*}), x^*) = (k-1) \int \bar{K}(u) u^{k-2} du.$$

where

$$\bar{K}(u) = \int K((y^*, \hat{F}_{y^*}), x^*, u) \nu(y^*) dy dF_{y^*}(x^*) d\Pi_{y^*,r}(\hat{F}_{y^*}). \tag{10}$$

This is the key result behind our estimation method, and we restate it in the following theorem.

Theorem 3.1 *Suppose* π_0, $\{F_y\}_{y\in\mathcal{Y}}$ *and marginal classifier* $\mathcal{F}$ *satisfy the marginal scaling condition tie-breaking condition. Then, under the definitions* (6), (9), *and* (10), *we have*

$$AvRisk_{k,r}((y^*, \hat{F}_{y^*}), x^*) = (k-1)\int \bar{K}(u)u^{k-2}du. \tag{11}$$

The proof is given in the appendix.

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 Average Risk_k requires some means of estimating the unknown function $\bar{K}$–which we discuss in the following.

3.3 Estimation in the general case

Now we address the problem of estimating $\text{AvRisk}_{k_2,r_{train}}$ from data. As we have seen from Theorem 3.1, the k-class average risk of a marginal classifier $\mathcal{M}$ is a functional of a object called $\bar{K}(u)$, which depends marginal model $\mathcal{M}$ of the classifier, the joint distribution of labels Y and features X when Y is drawn from the sampling density ν.

Therefore, the strategy we take is to attempt to estimate $\bar{K}$ for then given classification model, and then plug in our estimate of $\bar{K}$ into the integral (11) to obtain an estimate of $\text{AvRisk}_{k_2,r_{train}}$.

Having decided to estimate $\bar{K}$, there is then the question of what kind of model we should assume for $\bar{K}$. While a nonparametric approach may be ideal, for the case of general loss functions we will adopt a parametric model: that is the subject of this section.

We assume the linear model

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

where $h_\ell(u)$ are known basis functions, and β are the model parameters to be estimated. We propose two approaches for *unbiased* estimation of $\text{AvRisk}_{k_2,r_{train}}$. The first approach estimates the coefficients β via the unbiased estimates of k-class average risk obtained from (4). The second approach estimates β by estimating $\bar{K}(u)$ directly. However, the second approach requires the use of the *regression adjustment* method taken from measurement

error models, and therefore can only yield unbiased estimates for *polynomial* basis functions $h_\ell(u)$.

If we plug in the assumed linear model (12) into the identity (11), then we get

$$\begin{aligned} \mathrm{AvRisk}_{k,r_{train}} &= (k-2)\int \bar{K}(u)u^{k-2}du && (13)\\ &= (k-2)\int_0^1 \sum_{\ell=1}^{m} \beta_\ell h_\ell(u)u^{k-2}du && (14)\\ &= \sum_{\ell=1}^{m} \beta_\ell H_{\ell,k} && (15) \end{aligned}$$

where

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

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 $\mathrm{AvTestRisk}_k$ are unbiased estimates of $\mathrm{AvRisk}_{k,r_{train}}$, this implies that the regression estimate

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

is unbiased for β, under any choice of positive weights w_k. The estimate of $\mathrm{AvRisk}_{k_2,r_{train}}$ is similarly obtained from (15), via

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

3.4 Large-Sample Theory

How good are the estimated average risks (17)? Let us investigate the accuracy of the estimates in the limit where $k_1 \to \infty$.

If we fix the number of classes k_2 which defines the estimation target, then we need not use the estimator (17), since once $k_1 > k_2$, we can use the $\mathrm{AvTestRisk}_{k_2}$ as an estimator instead, which can easily be shown to a have

a convergence rate of $O(1/\sqrt{k_1})$ to the true average risk. Therefore, if we want to quantify the performance of the regression-based estimator (17), it does not make sense to look at asymptotic settings where k_2 is fixed. One approach is to specify a setting where k_2 changes as a function of k_1. However, the approach we will take is to look at the minimax error: that is, to look at the maximum discrepancy between the estimate and the true average risk over all k_2 simultaneously. The performance criterion is the minimax error, defined

$$\text{MinimaxError} = \sup_{k_2 > 2} |\widehat{\text{AvRisk}}_{k_2, r_{train}} - \text{AvRisk}_{k_2, r_{train}}|. \tag{18}$$

Note that if the parametric model (12) is correct, then

$$\text{AvRisk}_{k_2, r_{train}} = \sum_{\ell=1}^{m} \beta_\ell H_{\ell, k_2} = \langle \vec{H}_{k_2}, \beta \rangle$$

where $\vec{H}_{k_2} = (H_{\ell, k_2})_{\ell=1}^m$. Then, we get

$$\text{MinimaxError} = \sup_{k_2 > 2} |\langle \vec{H}_{k_2}, \beta - \hat{\beta} \rangle|.$$

If we assume that all the basis functions $h_\ell(u)$ are bounded by a common constant M, then it follows that H_ℓ, k are also bounded by the same constant M, and we have

$$\text{MinimaxError} \leq M||\beta - \hat{\beta}||_1 \leq M\sqrt{m}||\beta - \hat{\beta}||_2$$

Therefore, any convergence rate we can establish for $\hat{\beta}$ is inherited by the minimax error. Meanwhile, we can show that choosing k_0 sufficiently large that $(\vec{H}_2, \ldots, \vec{H}_{k_0})$ is full-rank, and setting weights $w_k = I\{k \leq k_0\}$, then the resulting $\hat{\beta}$ converges to the true β at the usual $O(1/\sqrt{n})$ rate. We state the result in the following theorem.

Theorem 3.2 *Consider a sequence of problems where the model $\mathcal{M}$, r_{train}, r_{test}, joint distribution $\{F_y\}_{y \in \mathcal{Y}}$, and class sampling distribution η are fixed as $k_1 \to \infty$. Further assume that the function $\bar{K}(u)$ defined by $\{F_y\}_{y \in \mathcal{Y}}$, η, and $\mathcal{M}$ satisfies*

$$\bar{K}(u) = \sum_{\ell=1}^{m} \beta_\ell h_\ell(u)$$

for some basis functions $h_\ell(u)$. Let k_0 be an integer sufficiently large so that

$$Rank(\vec{H}_2, \ldots, \vec{H}_{k_0}) = m.$$

Then, defining

$$\hat{\beta} = argmin_\beta \sum_{k=2}^{k_0} \left(AvTestRisk_k - \sum_{\ell=1}^{m} \beta_\ell H_{\ell,k} \right)^2$$

there exists some constant $C < \infty$ such that

$$\lim_{k_1 \to \infty} \sqrt{k_1} ||\hat{\beta} - \beta||_2 = C.$$

Proof. Note that the statistics AvTestRisk_k are U-statistics of the k_1 pairs of test and training samples. Therefore, by Hoeffding 1948, it follows that $(\text{AvTestRisk}_2, \ldots, \text{AvTestRisk}_{k_0})$ is asymptotically normal with covariance satisfying

$$\lim_{k_1 \to \infty} k_1 \text{Cov}(\text{AvTestRisk}_2, \ldots, \text{AvTestRisk}_{k_0}) = \Sigma,$$

for some positive semidefinite matrix Σ. Defining $\boldsymbol{H}$ to be the matrix with rows $\vec{H}_2, \ldots, \vec{H}_{k_0}$, this then implies that

$$\lim_{k_1 \to \infty} k_1 \text{Cov}(\hat{\beta}) = (\boldsymbol{H}^T \boldsymbol{H})^{-1} \boldsymbol{H}^T \Sigma \boldsymbol{H} (\boldsymbol{H}^T \boldsymbol{H})^{-1}.$$

It follows that defining

$$C = \sqrt{\text{tr}(\boldsymbol{H}^T \boldsymbol{H})^{-1} \boldsymbol{H}^T \Sigma \boldsymbol{H} (\boldsymbol{H}^T \boldsymbol{H})^{-1}}$$

we have

$$\lim_{k_1 \to \infty} \sqrt{k_1} ||\hat{\beta} - \beta||_2 = C.$$

4 Results

A Appendix

A.1 Proofs

Lemma A.1 *Defining $U_{y,\hat{F}_y}(x)$ as in* (6).

Classifier	Test err$^{(20)}$	Test err$^{(400)}$	$\hat{p}_{400}^{EXP}$	$\hat{p}_{400}^{POS}$	$\hat{p}_{400}^{(5)}$
Naive Bayes	0.049	0.399	0.108	**0.142**	0.079
Logistic	0.078	0.289	0.166	**0.188**	0.130
SVM	0.140	0.455	0.299	**0.313**	0.227
ϵ-NN	0.049	0.409	0.084	**0.590**	0.102
Deep neural net	0.005	0.014	**0.011**	0.093	0.010

Table 1: Performance extrapolation: predicting the accuracy on 400 classes using data from 20 classes on a Telugu character dataset. $\epsilon = 0.002$ for ϵ-nearest neighbors.

B Background: Multiclass terminology

In this section we review the key terminology for multi-class classification and discuss examples of problems and algorithms which we will use throughout the paper to serve as concrete examples. We assume some degree of familiarity with statistical learning: however, this section can probably be skipped by the expert. Meanwhile, those new to the field might be aided by having a good introduction to the subject at hand, such as (Hastie et al, ESL) or (?? other book.)

While a *binary classification* problem generally refers to a class with two labels, $\mathcal{Y} = \{0, 1\}$, problems with three or more classes are called *multi-class classification* problems. The most famous dataset for illustrating a multi-class classification problem is Fisher's iris data (Fisher 1936), where the classification task is to assign a flower to one of three iris species based on four features: the lengths and widths of the sepals and the lengths and widths of the petals.

In classification problems, it is assumed that each observation belongs exclusively to a single class. In contrast, in *multi-label* classification, each observation can belong to multiple classes at the same time, or none at all. We do not address multi-label classification in this paper: however, we remark that any multi-label classification problem can be recoded as a single-label classification problem [find a reference so we don't have to explain this.]

The performance of a classification rule on a problem is evaluated by specifying a *cost function.* If the true class is y, but the classifier outputs y', the severity of this misclassification is quantified by $C(y', y)$. The most common cost function is *zero-one loss*: the cost is zero for correct classifications,

and the cost is one for all incorrect classifications, i.e. $C(y', y) = \delta_y(y')$.

One setting where alternative cost functions are used is when there exists *hierarchical structure* of the label sets. For example, in image recognition, the label "golden retriever" may be a member of the class "dog," which is in itself another label. If we work under the single-label framework, then a picture of a golden retriever might be considered to have the true class of "golden retriever." While labelling the picture as "dog" would be semantically correct, we might prefer the more specific label. But while on a technical level we may consider "dog" to be the incorrect label for the picture, we would not want to overly penalize the assignment of "dog" to the picture. Therefore, in hierarchichal problems it is often appropriate to use a cost function which is reflective of the *semantic distance* between two labels, rather than the strict zero-one loss.

In our terminology, a *classification model* is an algorithm which learns a *classification rule* from *training data*. Examples of multi-class classification models include k-nearest neighbors, multinomial logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), decision trees, and random forests, as well as the two 'divide and conquer' approaches, one-vs-one (OVO) and one-vs-all (OVA) (Friedman et al, 2008.)

The *generalization risk* is the expected cost over the population of label-feature pairs. Given *test data* sampled from the population, it is possible to obtain an unbiased estimate of the risk fof a classification rule.

References

[1] Kay, K. N., Naselaris, T., Prenger, R. J., & Gallant, J. L. (2008). "Identifying natural images from human brain activity." *Nature*, 452(March), 352-355.

[2] Deng, J., Berg, A. C., Li, K., & Fei-Fei, L. (2010). "What does classifying more than 10,000 image categories tell us?" *Lecture Notes in Computer Science*, 6315 LNCS(PART 5), 71-84.

[3] Garfield, S., Stefan W., & Devlin, S. (2005). "Spoken language classification using hybrid classifier combination." *International Journal of Hybrid Intelligent Systems* 2.1: 13-33.

[4] Anonymous, A. (2016). "Estimating mutual information in high dimensions via classification error." Submitted to *NIPS 2016.*

[5] Tewari, A., & Bartlett, P. L. (2007). "On the Consistency of Multiclass Classification Methods." *Journal of Machine Learning Research*, 8, 1007-1025.

[6] Hastie, T., Tibshirani, R., & Friedman, J., (2008). *The elements of statistical learning.* Vol. 1. Springer, Berlin: Springer series in statistics.

[7] Arnold, Barry C., & Strauss, D. (1991). "Pseudolikelihood estimation: some examples." *Sankhya: The Indian Journal of Statistics, Series B*: 233-243.

[8] Cox, D.R., & Hinkley, D.V. (1974). *Theoretical statistics.* Chapman and Hall. ISBN 0-412-12420-3

[9] Lawson, C. L., & Hanson, R. J. (1974). *Solving least squares problems.* Vol. 161. Englewood Cliffs, NJ: Prentice-hall.

[10] Hong, J., Mohan, K. & Zeng, D. (2014). "CVX. jl: A Convex Modeling Environment in Julia."

[11] Domahidi, A., Chu, E., & Boyd, S. (2013). "ECOS: An SOCP solver for embedded systems." *Control Conference (ECC), 2013 European. IEEE.*

[12] Achanta, R., & Hastie, T. (2015) "Telugu OCR Framework using Deep Learning." arXiv preprint arXiv:1509.05962 .