Extrapolating expected accuracies for multi-class classification

Charles Zheng, Rakesh Achanta and Yuval Benjamini

November 18, 2016

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

Machine learning models are becoming increasingly employed in scientific and industrial applications. A common problem in these settings in *multi-class classification*, where the goal is to label objects (e.g. images, sentences, etc.) from a set of finitely many labels. Example applications:

- In biology, labeling images of cancerous cells by the type of cancer.
- In language detection, labeling a sentence by the language of the sentence.

- In face recognition, labeling a photograph of a person with their name.

In multi-class classification, one observes pairs (x, y) where $y \in \mathcal{Y}$ are class labels, and $x \in \mathcal{X} \subset \mathbb{R}^p$ are feature vectors (e.g. images of cells.) The goal is to construct a classification rule for predicting the label of a new data point; generally, the classification rule $f : \mathcal{X} \to \mathcal{Y}$ is learned from previously observed data points. In many applications of multi-class classification, such as face recognition or image recognition, the space of potential labels is practically infinite. In such a setting, one might 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 $f^{(i)} : \mathcal{X} \to \mathcal{Y}_i$. Supposing that (X, Y) have a joint distribution, define the misclassification error for the i-th problem as

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

Using data from only $\mathcal{Z}_k$, can one predict the misclassification error (or some other performance metric) on the larger label set $\mathcal{Z}_K$, with $K > k$? This is the problem of *performance extrapolation.*

A practical instance of performance extrapolation occurs in neuroimaging studies, where the number of classes k is limited by experimental considerations. Kay et al. [1] obtained fMRI brain scans which record how a single subject's visual cortex responds to natural images. The label set $\mathcal{Y}$ corresponds to the space of all grayscale photographs of natural images, and the set $\mathcal{S}$ is a subset of 1750 photographs used in the experiment. They construct a classifier which achieves over 0.75 accuracy for classifying the 1750 photographs; based on exponential extrapolation, they estimate that it would take on the order of $10^{9.5}$ photographs before the accuracy of the model drops below 0.10! Directly validating this estimate would take immense resources, so it would be useful to develop the theory needed to understand how to compute such extrapolations in a principled way.

However, in the fully general setting, it is impossible on construct nontrivial bounds on the accuracy achieved on the new classes $\mathcal{S}_K \setminus \mathcal{S}_k$ based only on knowledge of $\mathcal{S}_k$: after all, $\mathcal{S}_k$ could consist entirely of well-separated classes while the new classes $\mathcal{S}_K \setminus \mathcal{S}_k$ consist entirely of highly inseparable classes, or vice-versa. Thus, the most important assumption for our theory is that of *exchangeable sampling.* The labels in $\mathcal{S}_i$ are assumed to be an exchangeable sample from $\mathcal{Y}$. The condition of exchangeability ensures that the separability of random subsets of $\mathcal{Y}$ can be inferred by looking at the empirical distributions in $\mathcal{S}_k$, and therefore that some estimate of the achievable accuracy on $\mathcal{S}_K$ can be obtained.

In addition to the assumption of exchangeability, we consider a restricted set of classifiers. We focus on *marginal classifiers*, which are classifiers that work by training a model separately on 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 3, 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 function called the *conditional risk* depends on the true distribution 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 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. In section 5, we empirically study the performance of our classifiers. Since generative 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.

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.

1.1 Background: multi-class classification

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 classifier 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.

The hierarchy of labels is often represented by a tree T, where the vertices V are the set of labels, and the edges indicate superclass-subclass relationships. One of the earlier examples is Chakrabarti (1998), who classified online texts into topic taxonomies. One of the taxonomies, reprinted from their paper, is shown in Figure 1.

Cost functions for hierarchical class function are commonly derived from graph distances induced by the tree T. For example, one might define $C(y', y)$ to be the *graph distance* between y and y'–that is, the minimal number of

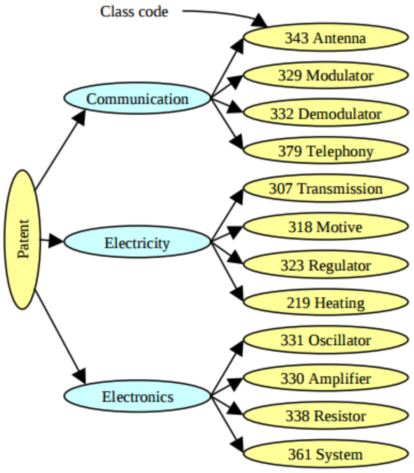

Figure 1: A topic taxonomy reprinted from Chakrabarti (1998)

edges in the graph for any path between y and y'.

[examples of multi-class classifiers? LDA, QDA, OVA, OVO?]

1.2 Potential applications

Applications involving an extremely large number of labels, such as image labelling, fall into the recently coined category of *extreme classification.* A common issue in extreme classification applications is the difficulty of collecting an initial dataset which contains enough data for all of the classes in the label set $\mathcal{Y}$.

To take a hypothetical (but realistic) example, a researcher might be interested in developing a classifier for the purpose of labelling images. She might begin with a list of 10,000 keywords which defines the label set $\mathcal{Y}$. Ideally, she would obtain a dataset by running a Google image search on each of the 10,000 keywords, and taking 20 images from the ith keyword as the training data for the ith class. However, initially, she only has the resources to do this for a much smaller number of keywords–say 100 keywords. Yet her goal is to develop a new algorithm for multi-class classification which works well on the larger set of labels. Can she get an idea of how well here algorithm will work on the full set of classes based on an initial "pilot" subsample of class labels?

The story 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. However, previous works have shown that such a simplistic assumption cannot be justified. 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." On the other hand, if more principled and accurate approaches can be found for predicting large-scale performance from smaller datasets, this would give academic researchers a way to get an idea of the large-scale performance of their methods without having to collect the data themselves, and it would also give companies a way to speed up their iterative development cycles, since performance extrapolation can reveal models with bad scaling properties in the pilot stages of development.

Outside of industry, there are increasingly many applications of multi-class classification in scientific experiments. As we have already seen from the Kay et. al. example, neuroscientists are interested in how well the brain activity in various regions of the brain can discriminate between different classes of stimuli. In the practical matter of experimental design, there is a tradeoff associated with how many classes of stimuli to include in the experiment. If too many classes are used, the p-values calculated from test performance may rise above the significance cutoff due to insufficient sample size for training and testing the classifier, as well as an increase in the difficulty of the classification task. However, when too few classes are used, while the p-values may become much stronger, the generalizability of the experiment is sacrificed because an overly simplistic classification task fails to represent the full complexity of the stimuli being studied. A better understanding of how the difficulty of the classification task scales with the number of classes could be useful for scientists seeking the best tradeoff between power and generalizability in choosing the number of classes in the experimental design.

While our primary goal is to motivate and formulate the question rather than to obtain optimal methods, we are optimistic that good methods for performance extrapolation can be found, and that such methods could aid the development of classifers in academia and industry, as well as help the scientists who use machine learning in their analysis pipeline design their experiments more effectively.

2 Framework

2.1 Problem Formulation

This section lays out the basic framework which is necessary to *formulate* the problem. However, in the rest of the paper, we will also adopt some additional assumptions in order to solve the problem we pose in this section.

Let $\mathcal{Y}$ be a collection of labels and $\mathcal{X}$ be a space of feature vectors. For each label $y \in \mathcal{Y}$, there exists a distribution F_y supported on $\mathcal{X}$. Also suppose that there exists a *cost function* $C(\hat{y}, y)$ which measures the cost of incorrectly labelling an instance as $\hat{y}$ when the correct label is y. Further suppose that $0 \leq C(y, y') \leq \infty$ and $C(y, y) = 0$ for all $y, y' \in \mathcal{Y}$.

A *classification task* consists of a subset of labels, $\mathcal{S} \subset \mathcal{Y}$, and a prior distribution π over the label subset. A *classification rule* for the task consists of a function f which maps feature vectors $x \in \mathcal{X}$ to labels in $\mathcal{S}$:

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

The classification task defines the *risk* of a classification rule. Under the classification task, a label y is drawn from the distribution π. Then, we draw $x \sim F_y$. The label assigned by the classification rule is $\hat{y} = f(x)$. The *loss* incurred is $C(\hat{y}, y)$. The *risk* of the classification rule is the expected loss under the class distribution π:

$$\text{Risk}(f) = \mathbf{E}_\pi[C(\hat{y}, y)] = \int_{\mathcal{S}} d\pi(y) \int_{\mathcal{X}} C(f(x), y) dF_y(x).$$

A *classification model* $\mathcal{F}$ is an algorithm or procedure for producing classification rules given an empirical distributions $\hat{F}_y$ for each $y \in \mathcal{S}$, and a vector of prior probabilities π. The model maps a distribution G and a vector π to a classification rule f.

We consider *datasets* of size $|\mathcal{S}|r$ for a given classification task consisting r i.i.d. observations $x_i^{(y)} \sim F_y$ for each $y \in \mathcal{S}$. Then define

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

The sampling distribution of $\hat{F}_y$ is referred to as the size-r sampling distribution $\Pi_{y,r}$.

The n-sample *risk* of the classification model $\mathcal{F}$ is the expected risk of a classification rule $\hat{f} = \mathcal{F}(\hat{G})$ for a random dataset of size n for the classification task, $\hat{G} \sim \Pi_n$. That is,

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

The problem of *performance extrapolation* is as follows. Suppose we have two classification tasks: the ith classification task is specified by label subset $\mathcal{S}_i$, prior distribution π_i. We observe data from the first classification task consisting of a dataset of size n_1. The goal is to estimate the n_2-sample risk of a $\mathcal{F}$ on the second classification task, $\text{Risk}_{n_2}(\mathcal{F};\pi_2)$.

2.2 Additional assumptions

In order to obtain a tractable solution to the problem of performance extrapolation, we make a number of special assumptions on the nature of the classification tasks, and the classifiers themselves, which make the problem much easier.

Firstly, we assume that the label space $\mathcal{Y}$ is a continuum: in fact, that $\mathcal{Y}$ is a subset of d-dimensional Euclidean space. Note this is not such a strong assumption as it might seem, since cases where there are k discrete labels can be equivalently formulated as continous models where the the continuum can be partitioned into k equivalence classes, and in which the cost between two label y, y' is a function only of their equivalence classes.

We work with bounded cost functions. Without loss of generality, assume that

$$\sup_{y,y'\in\mathcal{Y}} C(y,y') \leq 1.$$

With regards to the classification tasks, we assume that there exists some prior density ν_0 over $\mathcal{Y}$, and that the label subsets $\mathcal{S}_i = \{y^{(1)}, \ldots, y^{(k_i)}\}$ are obtained by iid samples with replacement from the density ν_0. (An alternative assumption would be that $\mathcal{S}_1 \subset \mathcal{S}_2$ with $\mathcal{S}_1$ being a subsample of $\mathcal{S}_2$: this assumption can also be addressed, as we wil discuss later.)

Next, suppose there exists some other density ν_1 over $\mathcal{Y}$, and that the prior probabilities for each classification task are given by

$$\pi_i(y) = \frac{\nu_1(y)}{\sum_{y' \in \mathcal{S}_i} \nu_1(y')}.$$

Define π_0 as the distribution over $\mathcal{Y}$ with density proportional to $\nu_0 \nu_1$. More precisely, suppose that the density of π is written

$$(\mathbf{??})\pi_0(y) = \frac{\nu_0(y)\nu_1(y)}{\zeta} \tag{1}$$

where ζ is a normalizing constant. Then the marginal distribution of any element of $\mathcal{S}$ is given by π_0.

Further, let us assume that we have more repeats per class in the first classification task than in the second, $r_1 > r_2$. We discuss the possibility of relaxing this condition in the Discussion.

Since the classification tasks are randomly generated, 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.

We make some rather strong assumptions with regards to the classifiers. The classifier $\mathcal{F}$ produces classification rules f which depend on *marginal scoring rules*, m_y for $y \in \mathcal{S}$. Each marginal scoring rule m_i is a mapping

$$m_y : \mathcal{X} \to \mathbb{R}.$$

The classification rule chooses the class with the highest marginal score,

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

The marginal scoring rules m_i, in turn, are generated by a marginal model $\mathcal{M}$. The marginal model converts empirical distributions $\hat{F}$ over $\mathcal{X}$, and an (empirical) prior class probability, into a marginal scoring function $m : \mathcal{X} \times \mathbb{R} \to \mathbb{R}$. For example, one could take

$$m(x, p) = \log(p) + \log(\hat{f}(x)).$$

where $\hat{f}$ is a density estimate obtained from $\hat{F}$. We call such a classification model $\mathcal{F}$ a *marginal classifier*, and such marginal classifiers are completely specified by the marginal model $\mathcal{M}$.

Quadratic discriminant analysis and Naive Bayes are two examples of marginal classification models. The *marginal* property allows us to prove strong results about the accuracy of the classifier under the exchangeable sampling assumption, as we see in Section [].

2.3 Local polynomial regression

Explain background.

Introduce the notation $\{(w_i, x_i, y_i)\}_{i=1}^n$: ordered triples of weight, predictor and response.

2.4 Measurement error models

Explain background.

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.

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. Since various methods, such as cross-validation can be used to obtain close-to-unbiased estimates of the risk in a given classification problem, the problem is essentially solved for this special case.

Meanwhile, in the case where $k_2 < k_1$, the problem can be solved by repeatedly subsampling label sets of size k_2 from $\mathcal{S}_1$ and averaging unbiased estimates of the risk of each subsampled classification task. Aside from computational issues with respect to computing or approximating the average of $\binom{k_1}{k_2}$ empirical accuracies, the problem is again more or less solved by using existing methods.

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

The average risk is obtained by averaging over four randomizations:

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

2. Drawing the training dataset.

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

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

In other words, one can define a random variable L

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

which clearly depends on all of the random quantities $\mathcal{S}, \hat{F}_y, \pi, Y^*, X^*$, and where

$$\text{Average Risk}_k(\mathcal{F}) = \mathbf{E}[L]$$

where the expectation is taken over all four randomization steps.

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.

We find that we can do this by *conditioning on* the pair (x^*, y^*) while *averaging* over the first two steps. Define the *conditional risk* $R_k(y^*, x^*)$ as

$$R_k(y^*, x^*) = \mathbf{E}[L|Y^* = y, X^* = x].$$

Let G denote the joint distribution over (X, Y) obtained by drawing $Y \sim \pi_0$ and $X \sim F_Y$. Since Y^* has the marginal distribution π_0, it follows that

$$\text{Average Risk}_k(\mathcal{F}) = \int\int R_k(y^*, x^*) dG(x^*, y^*). \tag{2}$$

What we have done is to rewrite the average risk as the expectation of R_k, which depends on k, according to a measure G which does *not* depend on k.

However, we will further decompose R_k into k-dependent and k-independent components.

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 $x \in \mathcal{X}$, $\mathcal{M}(\hat{F}_1, \pi_1)(x) = \mathcal{M}(\hat{F}_2, \pi_2)(x)$ with zero probability.

Define the U-functions

$$\begin{aligned} U_x(y) &= \Pr[\mathcal{F}(\hat{F}_y, \pi_0(y))(x) > \mathcal{F}(\hat{F}_Y, \pi_0(Y))(x)] & (3) \\ &= \int_{\mathcal{Y}} I\{\mathcal{F}(\hat{F}_y, \pi_0(y))(x) > \mathcal{F}(\hat{F}_{y'}, \pi_0(y'))(x)\} d\Pi_{y,r}(\hat{F}_y) d\Pi_{y',r}(\hat{F}_{y'}) d\pi_0(y'). & (4) \end{aligned}$$

Under the scaling property of margins, we have

$$U_x(y) = \Pr[\mathcal{F}(\hat{F}_y, \pi(y))(x) > \mathcal{F}(\hat{F}_Y, \pi(Y))(x)]$$

for the *random* π corresponding to $\mathcal{S}$. Hence, $U_x(y)$ gives the probability that an observation x would be assigned to class y, supposing that the only two choices for labels were y and Y', with Y' drawn uniformly from π_0.

Note that the random variable $U_x(Y)$ for $Y \sim \pi_0$ is uniformly distributed for all $x \in \mathcal{X}$ (hence the name "U-function").

Define the *conditional cost function* $K(y^*, x^*, u)$ by

$$K(y^*, x^*, u) = \mathbb{E}[C(Y, y^*) I\{U_{x^*}(Y) > U_{x^*}(y^*)\} | U_{x^*}(y) = u]. \tag{5}$$

The conditional cost function gives the expected cost conditional on x^*, y^*, and the U_{x^*}-value of the incorrect label with the largest margin.

Obtaining the conditional risk $R_k(y^*, x^*)$ from the conditional cost function requires the following observation. Let the $(k-1)$ incorrect labels in $\mathcal{S}$ be denoted by $y^{(1)}, \ldots, y^{(k-1)}$, and define $U_i = U_{X^*}(y^{(i)})$. Let U_{max} denote the U_{x^*}-value of the incorrect label: we have

$$U_{max} = \max_{i=1}^{k-1} U_i.$$

Meanwhile, by definition, we have

$$R_k(y^*, x^*) = \mathbf{E}[K(y^*, x^*, U_{max})].$$

But we know the density of U_{max}! Recall that U_i are iid uniform, and therefore U_{max} has density $p(u) = ku^{k-1}$. We therefore have

$$R_k(y^*, x^*) = k \int K(y^*, x^*, u) u^{k-1} du.$$

Returning to equation (2), we obtain

$$\text{Average Risk}_k(\mathcal{F}) = k \int u^{k-1} du \int K(y^*, x^*, u) dG(x^*, y^*) = k \int u^{k-1} \bar{K}(u) du,$$

where

$$\bar{K}(u) = \int K(y^*, x^*, u) dG(x^*, y^*). \tag{6}$$

The average risk is 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 weighting density ku^{k-1} (which is the Beta(k, 1) density.)

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

Theorem 3.1 *Suppose π, $\{F_y\}_{y \in \mathcal{Y}}$ and marginal classifier $\mathcal{F}$ satisfy the marginal scaling condition tie-breaking condition. Then, under the definitions* (3)*,* (5)*, and* (6)*, we have*

$$AvRisk_k(\mathcal{F}) = k \int u^{k-1} \bar{K}(u) du. \tag{7}$$

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

Now we address the problem of estimating Average Risk$_{k_2}$ from data.

First, let us assume a d-th order polynomial model

$$\bar{K}(u) = \sum_{\ell=0}^{d} \beta_\ell u^\ell.$$

Recall that the data consists of $k_1 < k_2$ classes, with r_1 repeats per class. The set of class labels is $\mathcal{S}_1 = \{y^{(1)}, \ldots, y^{(k_1)}\}$. For each class $i = 1, \ldots, k_1$, we have repeats $x_j^{(i)}$ for $j = 1, \ldots, r_1$.

Define $r_{test} = r_1 - r_2$. Let us take the first r_{test} repeats per class, and form the *test set* $\{x_j^{(i)}\}_{j=1,i=1}^{r_{test},k_1}$. From the remaining r_2 repeats, we form the empirical distributions

$$\hat{F}_{y^{(i)}} = \frac{1}{r_2} \sum_{j=r_{test}+1}^{r_1} \delta_{x_j^{(i)}}.$$

The marginal model $\mathcal{M}$ yields margins for each point in the test set for each label in $\mathcal{S}_1$. Define the margins

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

The predicted label for each test point is

$$\hat{y}_{i,j} = y^{(\text{argmax}_{\ell \in \{1,\ldots,k\}} M_{i,j}^\ell)}.$$

Therefore, an unbiased estimate of the risk (which is also an unbiased estimate of the k_1-class average risk) is

$$\text{Test Risk} = \frac{1}{r_{test}} \sum_{i=1}^{k} \sum_{j=1}^{r_{test}} \eta_1(y^{(i)}) C(\hat{y}_{i,j}, y^{(i)}).$$

Now we turn to the question of estimating the function $\bar{K}(u)$. Suppose that hypothetically, we could have observed the quantities $U_{i,j,\ell}$, defined

$$U_{i,j,\ell} = U_{x_j^{(i)}}(y^{(\ell)}).$$

Also define

$$C_{i,\ell} = C(y^{(\ell)}, y^{(i)})$$

and

$$w_i = \eta_1(y^{(i)}).$$

Then $\bar{K}(u)$ could be estimated via a d-th order polynomial regression

$$\hat{\beta} = \text{argmin}_\beta \sum_{i=1,j=1,\ell=1}^{r_{test},k_1,k_1} w_i \left(C_i^\ell - \sum_{h=0}^{d} \beta_h U_{i,j,\ell}^d \right)^2$$

for the dataset $\{(w_i, U_{i,j,\ell}, C_{i,\ell})\}_{i=1,j=1,\ell=1}^{r_{test},k_1,k_1}$. However, this is not possible in practice because $U_{i,j,\ell}$ are not directly observed.

Instead, we can obtain unbiased estimates via

$$\hat{U}_{i,j,\ell} = \frac{1}{(k-1)\zeta} \sum_{m \neq \ell} \eta_1(y^{(m)}) I\{M_{i,j}^\ell > M_{i,j}^m\}$$

However, if we were to simply treat $\hat{U}_{i,j,\ell}$ as a proxy for the unobserved $U_{i,j,\ell}$, and apply polynomial regression to the dataset

$$\{(w_i, \hat{U}_{i,j,\ell}, C_{i,\ell})\}_{i=1,j=1,\ell=1}^{r_{test},k_1,k_1},$$

the estimated $\widehat{\bar{K}(u)}$ would be biased, since we have *errors-in-covariates.* It is necessary to make use of the *covariate adjustment* technique. Covariate adjustment is justified since the error in the covariates is conditionally independent of the response given the true covariates:

$$\hat{U}_{i,j,\ell} \perp C_i^\ell | U_{i,j,\ell}.$$

In the naive polynomial regression, the predictors are the powers of the unbiased estimates, $\hat{U}_{i,j,\ell}^h$ for $h = 0, \ldots, d$. The issue is that while $\hat{U}_{i,j,\ell}$ is unbiased for $U_{i,j,\ell}$, the higher powers of $\hat{U}_{i,j,\ell}$ are *not* unbiased estimators of the higher powers of $U_{i,j,\ell}$. Covariate adjustment in this case amounts to replacing the naive estimates $\hat{U}_{i,j,\ell}^h$ with unbiased estimators.

There exist U-statistic estimators of the higher powers of $\hat{U}_{i,j,\ell}^h$. For instance, for $h = 2$, the estimator is

$$\hat{U}_{i,j,\ell}^{(2)} = \frac{1}{\zeta^2 k(k-1)} \sum_{m_1 \neq m_2 \neq j} \eta_1(y^{(m_1)}) I\{M_{i,j}^\ell > M_{i,j}^{m_2}\} \eta_1(y^{(m_2)}) I\{M_{i,j}^\ell > M_{i,j}^{m_2}\}.$$

In general, the U-statistic is

$$\hat{U}_{i,j,\ell}^{(h)} = \frac{(k-h)!}{\zeta^h k!} \sum_{m_1 \neq \cdots \neq m_h \neq j} \prod_{z=1}^h \eta_1(y^{(m_z)}) I\{M_{i,j}^\ell > M_{i,j}^{m_z}\}.$$

In summation, the algorithm is as follows:

3.4 Monotonicity assumption

Assuming that $\bar{K}(u)$ in monotone in $u \in [0,1]$. Justification and implications.

4 Special case: uniform prior and zero-one loss

For the special case of zero-one loss and uniform prior, the theory becomes simplified and additional methods of estimation are possible.

Define

$$\gamma_m = \int_0^1 \bar{K}(u) \frac{k_1!}{(m-1)!(k_1-m)!} u^{m-1}(1-u)^{k_1-m} du$$

for $m = 1, \ldots, k_1$.

Define the *ranks*

$$R_{i,j,\ell} = (k-1)\hat{U}_{i,j,\ell}.$$

Due to the uniform prior, $R_{ij}^\ell \in \{0, \ldots, k_1 - 1\}$.

Define

$$C_{ij}^{(h)} = \sum_{\ell=1}^k I\{R_{i,j,\ell} = h-1\} C_{i,\ell},$$

i.e., the cost incurred for the class with the hth smallest rank for the observation $x_i^{(\ell)}$.

Note that the test risk for k_1 classes can be written as

$$\text{Test Risk}_k = \frac{1}{r_{test} k_1} \sum_{i=1,j=1}^{k_1, r_{test}} C_{ij}^{(k_1)}.$$

Since C_{ij}^{ℓ} is now a binary random variable, we have

$$C_{ij}^{(h)} \sim \text{Bernoulli}(\gamma_h).$$

If $C_{ij}^{(h)}$ were independent, one could estimate γ_h by maximizing the log-likelihood

$$\mathcal{L}(\vec{\gamma}) = \sum_{i=1,j=1,h=1}^{k_1, r_{test}, k_1} C_{ij}^{(h)} \log \gamma_h + (1 - C_{ij})^{(h)} \log(1 - \gamma_h) \tag{8}$$

However, since $C_{ij}^{(h)}$ are not independent, the equation (8) is not a likelihood, but a *pseudolikelihood*. Nevertheless, one can attempt to estimate γ_h using the method.

The basic idea of using pseudolikelihood leads to many different practical approaches for estimating the average k-class risk. We tried the following approaches:

1. Estimate unconstrained γ_h, then find a function $\bar{K}(u)$ which satisfies the moment constraints implied by the estimates $\hat{\gamma}_h$. Estimate AvgRisk_k by plugging in the estimated $\hat{K}(u)$ into (7).

2. Let $\hat{K}(u)$ be constrained by the test risk (which is unbiased for the k_1-class average risk,)

   $$\text{Test Risk}_l = \int_0^1 \hat{K}(u) k_1 u^{k_1 - 1} du.$$

   Under this moment constraint, estimate $\hat{K}(u)$ using pseudolikelihood.

3. Either of the above two approaches, plus a monotonicity constraint on $\hat{K}(u)$.