Extrapolating expected accuracies for multi-class classification

Charles Zheng, Rakesh Achanta and Yuval Benjamini

November 21, 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.

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? This is the problem of *performance extrapolation.*

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. Neuroscientists are 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. 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 classes; 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.

More generally, however, a theory of performance extrapolation could be useful for aiding in the design of neuroscience experiments: specially with regards to the choice of how many stimuli to use in the experiment, where the scientist faces the following tradeoff. 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.

1.1 Problem statement

To state the problem more formally, recall that 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 $f^{(i)} : \mathcal{X} \to \mathcal{S}_i$. 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 prediction extrapolation is the following: using data from only $\mathcal{S}_k$, can one predict the misclassification error (or some other performance metric) on the larger label set $\mathcal{S}_K$, with $K > k$?

In the fully general setting, it is impossible on construct non-trivial 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 *i.i.d. sampling.* The labels in $\mathcal{S}_i$ are assumed to be an i.i.d. sample from $\mathcal{Y}$. The condition of i.i.d. sampling 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 i.i.d. sampling, 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 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.

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.

1.2 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 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.

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 $0 \leq C(y, y') < \infty$ and $C(y, y) = 0$ for all $y, y' \in \mathcal{Y}$. (Recall that in practice, the cost function is specified by the user to reflect the needs of the application.)

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. 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:

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

For label $y \in \mathcal{Y}$, define a sample of size r as a sequence of i.i.d. observations $X_1, \ldots, X_r \sim F_y$. A sample can either be represented as a vector of points, or as an *empirical distribution* $\hat{F}_y$

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

Let $\Pi_{y,r}$ denote the sampling distribution of $\hat{F}_y$.

In the literature, the term *classifier* is used ambiguously to refer to either what we call the *classification model* or the *classification rule.* Here, we take a *classification model* $\mathcal{F}$ to mean 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 (Figure 2.) Within this paper, we use the word *classifier* as a shorthand for *classification model* when there is no danger of ambiguity.

We suppose that the *training set* $\{\hat{F}_y\}_{y \in \mathcal{S}}$ consists of samples of size r_{train} for each $y \in \mathcal{S}$ (Figure 1.) Additionally, we assume that one has access to a *test set* of size r_{test}. Notationally, we always represent the training set

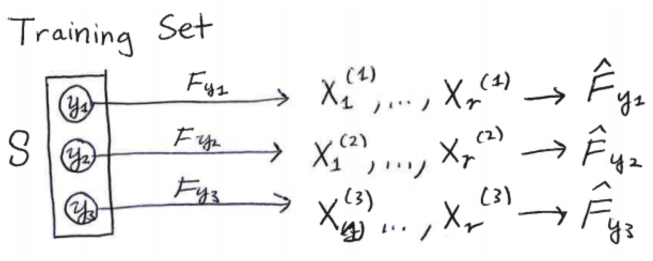
Training Set
S
y_1
y_2
y_3
F_{y_1}
F_{y_2}
F_{y_3}
$X_1^{(1)}, \ldots, X_r^{(1)} \rightarrow \hat{F}_{y_1}$
$X_1^{(2)}, \ldots, X_r^{(2)} \rightarrow \hat{F}_{y_2}$
$X_1^{(3)}, \ldots, X_r^{(3)} \rightarrow \hat{F}_{y_3}$

Figure 1: Training set

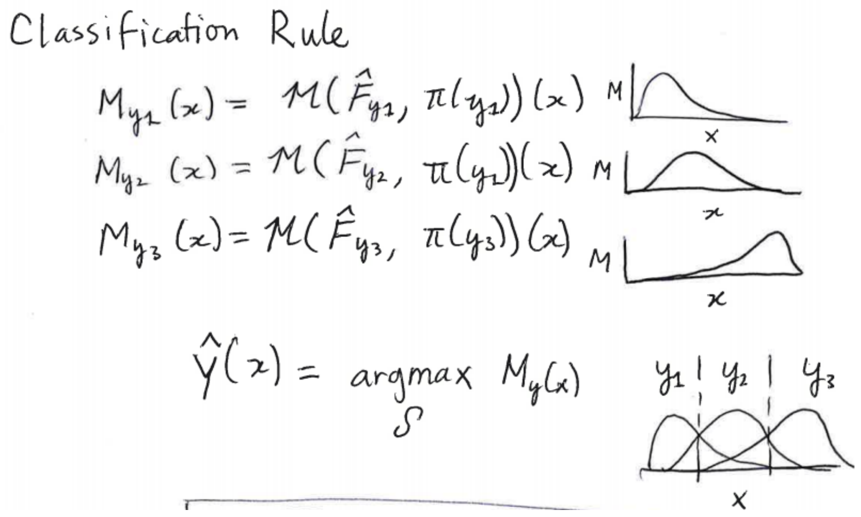
Classification Rule
$M_{y_1}(x) = \mathcal{M}(\hat{F}_{y_1}, \pi(y_1))(x)$
$M_{y_2}(x) = \mathcal{M}(\hat{F}_{y_2}, \pi(y_2))(x)$
$M_{y_3}(x) = \mathcal{M}(\hat{F}_{y_3}, \pi(y_3))(x)$
M
x
M
x
M
x
$\hat{y}(x) = \underset{\mathcal{S}}{\text{argmax}}\ M_y(x)$
y_1
y_2
y_3
x

Figure 2: Classification rule

samples for each class as empirical distributions $\hat{F}_y$, while the test set samples are written as vectors $(X_1^{(y)}, \ldots, X_{r_{test}}^{(y)})$.

The r-repeat *risk* of the classification model $\mathcal{F}$ is the expected risk of a classification rule $\hat{f} = \mathcal{F}(\hat{F}_{y_1}, \ldots, \hat{F}_{y_k})$ for the classification task, $\hat{F}_y \sim \Pi_{y,r}$. 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* 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

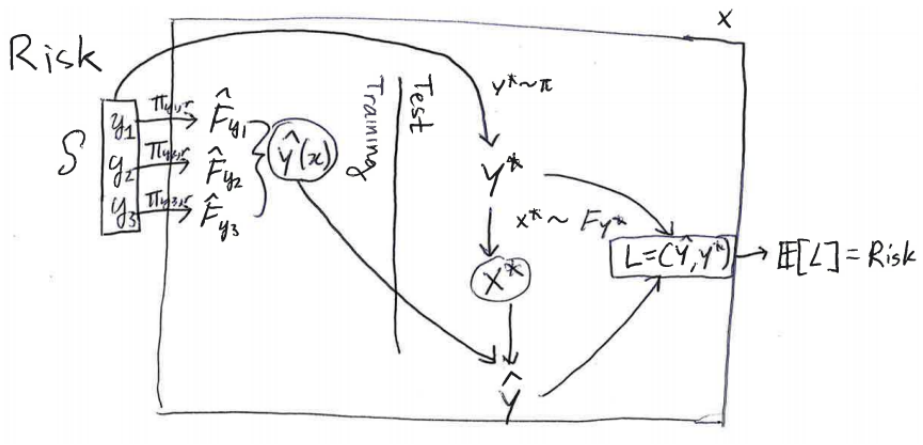

Figure 3: Classification risk

task consisting of a dataset consisting of a training samples of size r_{train} per class, plus an independent test samples of size r_{test} per class. The goal is to estimate the r_{train}-sample risk of a $\mathcal{F}$ on the second classification task, $\text{Risk}_{r_{train}}(\mathcal{F}; \pi_2)$.

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, since such an assumption simplifies the theory, and because unbounded cost functions can be analyzed by taking a limit of bounded approximations. Furthermore, without loss of generality, we can assume that

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

With regards to the classification tasks, (i) we assume that there exists some prior density π_0 over $\mathcal{Y}$, and that (ii) the label subsets $\mathcal{S}_i = \{y^{(1,i)}, \dots, y^{(k_i,i)}\}$ are obtained by iid samples with replacement from the density ν_i. We pause to remark on these assumptions.

i. The assumption of a prior density, when combined with the assumption that $\mathcal{Y}$ is a continuum, implies that the probability of any particular $y \in \mathcal{Y}$ is zero. Therefore, this assumption would seem to violate most applications of classification, where the labels come from a discrete distribution. However, any discrete classification problem can be recoded as an equivalent continuous classification problem. Note that any discrete space $\mathcal{Y}$ can be expanded to a continuous space $\tilde{\mathcal{Y}}$ by means of augmenting the label y with an uninformative auxillary variable $u \in [0,1]$. By defining the cost function $\tilde{C}((y,u),(y',u')) = C(y,y')$ and prior distribution $\tilde{\pi}_0 = \pi \times U$ where U is the uniform distribution, we obtain an equivalent classification problem which satisfies our assumptions.

ii.(a.) We allow the sampling distribution of classes in the label subset to differ from the prior probabilities, because π_0 reflects the population distribution of Y while ν_i reflects the mechanism used to pick classes for the label subset. In experimental settings, ν_i is under the control of the experimenter, so it need not be chosen to be identical to π_0.

ii.(b.) Note that here we assumed that the label subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ are independent: as a consequence of (i), this means that they are disjoint with probability one. 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 will discuss later.

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.

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 i.i.d. sampling assumption, as we see in Section [].

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

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

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 $\mathcal{S}$.

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 $\mathcal{S} = \{Y^{(1)}, \ldots, Y^{(k)}\}$, the training samples $\hat{F}_{Y^{(1)}}, \ldots, \hat{F}_{Y^{(k)}}$, 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.

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.

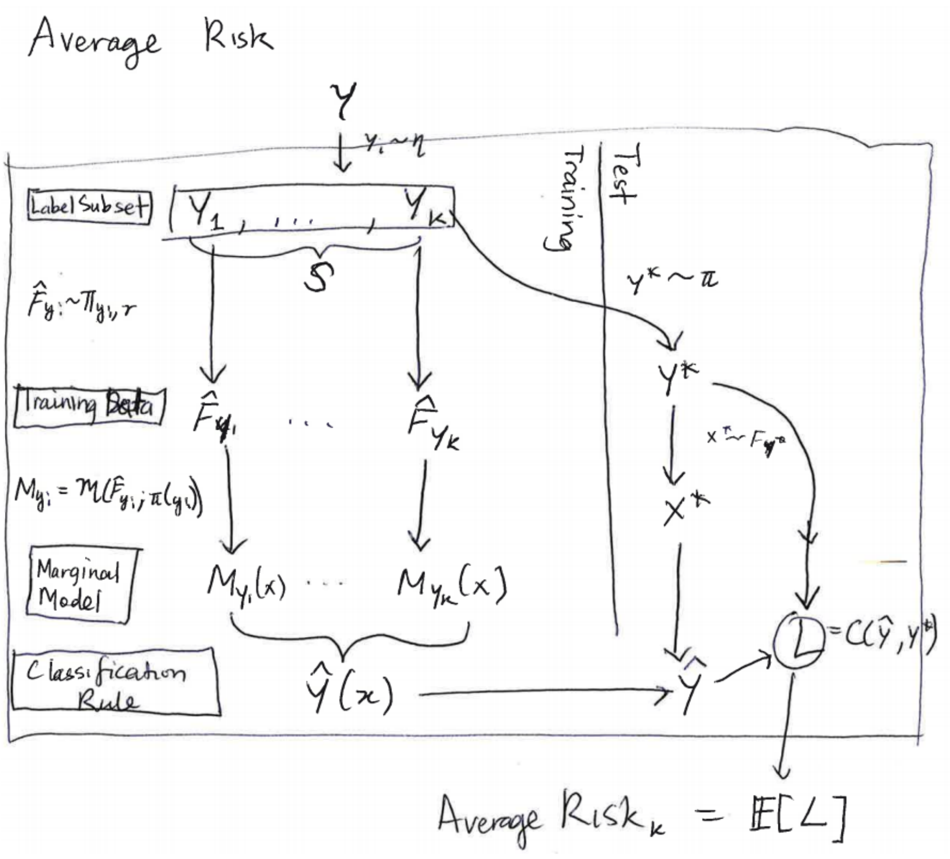

Figure 4: Average risk

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

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.

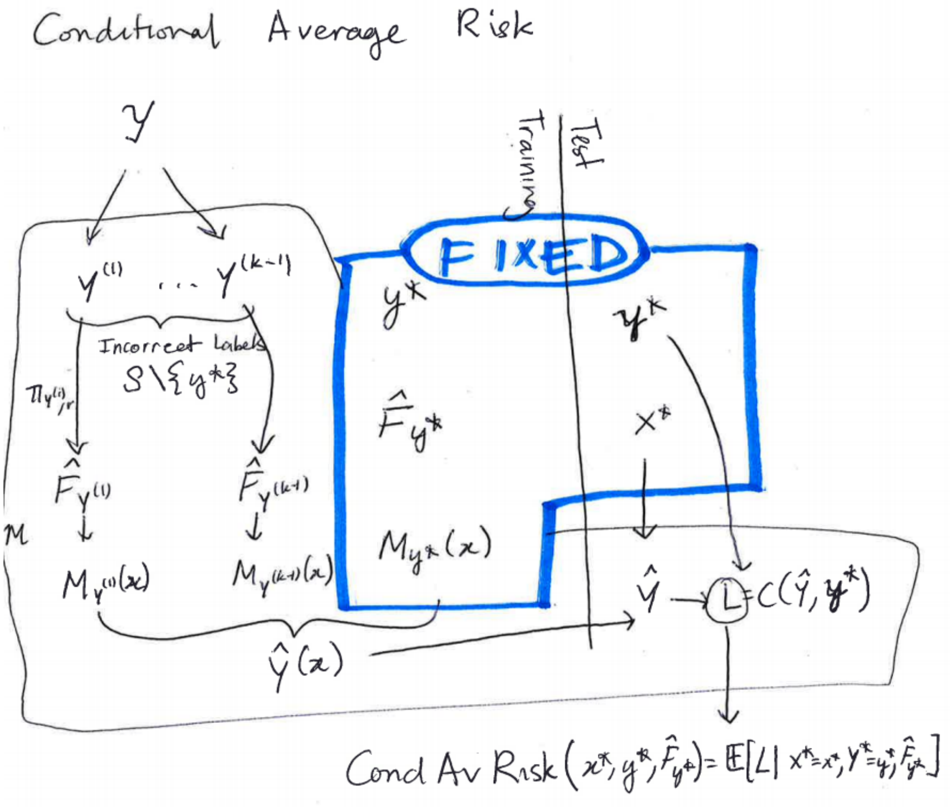

Figure 5: Conditional average risk

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.

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:

$$\begin{aligned} U_{x^*}(y^*, \hat{F}_{y^*}) &= \Pr[M_{y^*}(x^*) > \mathcal{M}(\hat{F}_Y, \pi_0(Y))(x^*)] && (2) \\ &= \int_{\mathcal{Y}} I\{M_{y^*}(x^*) > \mathcal{M}(\hat{F}_y, \pi_0(y))(x)\} d\Pi_{y,r}(\hat{F}_y) d\pi_0(y). && (3) \end{aligned}$$

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 $Y^{(1)}, \ldots, Y^{(k-1)}$. 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

$$U_{max,k-1} \sim \text{Beta}(k-1, 1). \tag{4}$$

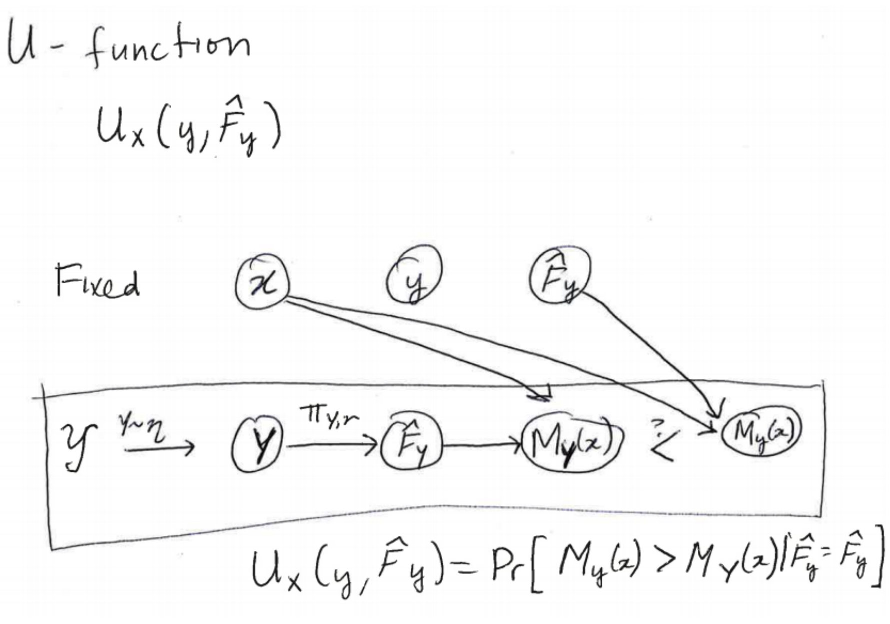

Figure 6: U-functions

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

$$K((y^*, \hat{F}_{y^*})x^*, u) = \begin{cases} 0 \text{ if } u < u^* \\ \mathbf{E}[C(\hat{Y}, y^*)|U_{max} = u, x^*, y^*, \hat{F}_{y^*}] \text{ otherwise.} \end{cases} \tag{5}$$

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^*.

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

where

$$\bar{K}(u) = \int K((y^*, \hat{F}_{y^*}), x^*, u) \nu(y^*) dy dF_{y^*}(x^*) d\Pi_{y^*,r}(\hat{F}_{y^*}). \tag{6}$$

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* (2)*,* (5)*, and* (6)*, we have*

$$AvRisk_{k,r,\nu}((y^*, \hat{F}_{y^*}), x^*) = (k-1) \int \bar{K}(u) u^{k-2} du. \tag{7}$$

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 in the general case

[Note: In the final version we might just assume $\nu_1 = \nu_2$ throughout the whole paper to simplify things. But I have written the general case for now.]

Now we address the problem of estimating $\text{AvRisk}_{k_2, r_{train}, \nu_2}$ 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 (7) to obtain an estimate of $\text{AvRisk}_{k_2, r_{train}, \nu_2}$. 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. On the other hand, for the special case of zero-one loss (Section 4), we take a nonparametric approach.

Even restricting ourselves to parametric models, there are wide variety of parametric families one might consider for $\bar{K}(u)$. As it turns out, the d-th order polynomial model is unique for enabling unbiased estimation of the average risk. Therefore, let us assume

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

Recall that the data consists of $k_1 < k_2$ classes, $\mathcal{S}_1 = \{y^{(1)}, \dots, y^{(k_1)}\}$. For each $y^{(i)}$ we have training sample $\hat{F}_{y^{(i)}}$, and r_{test} test repeats per class, $(x_1^{(i)}, \dots, x_{r_{test}}^{(i)})$.

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

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

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

$$\text{Test Risk} = \frac{1}{r_{test}} \sum_{i=1}^{k} \sum_{j=1}^{r_{test}} \frac{\nu_2(y^{(i)})}{\nu_1(y^{(i)})} C(\hat{y}_{i,j}, y^{(i)}).$$

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

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

Also define

$$C_{i,\ell} = C(y^{(\ell)}, y^{(i)}) I\{U_{i,j,\ell} > U_{i,j,i}\}$$

and

$$w_i = \frac{\nu_2(y^{(i)})}{\nu_1(y^{(i)})}.$$

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

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

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 the importance-sampling-reweighted mean

$$\hat{U}_{i,j,\ell} = \frac{1}{(k-1)\zeta} \sum_{m \neq \ell} \frac{\nu_2(y^{(m)})}{\nu_1(y^{(m)})} I\{M_{i,j}^\ell > M_{i,j}^m\}$$

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, $\hat{U}^h_{i,j,\ell}$ 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 $\hat{U}^h_{i,j,\ell}$ with unbiased estimators.

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

$$\hat{U}^{(2)}_{i,j,\ell} = \frac{1}{\zeta^2 k(k-1)} \sum_{m_1 \neq m_2 \neq j} \frac{\nu_2(y^{(m_1)})}{\nu_1(y^{(m_2)})} I\{M^\ell_{i,j} > M^{m_2}_{i,j}\} \frac{\nu_2(y^{(m_2)})}{\nu_1(y^{(m_2)})} I\{M^\ell_{i,j} > M^{m_2}_{i,j}\}.$$

In general, the U-statistic is

$$\hat{U}^{(h)}_{i,j,\ell} = \frac{(k-h)!}{\zeta^h k!} \sum_{m_1 \neq \cdots \neq m_h \neq j} \prod_{z=1}^{h} \frac{\nu_2(y^{(m_z)})}{\nu_1(y^{(m_z)})} I\{M^\ell_{i,j} > M^{m_z}_{i,j}\}.$$

In summation, the algorithm is as follows:

3.4 Convergence analysis

Taking a fairly conservative analysis, we will use the universal variance bound

$$\text{Var}[\bar{C}_m] \leq \frac{1}{4r}$$

which holds given the condition $\sup_{\mathcal{Y}^2} C(y, y') \leq 1$.

We have the explicit formulas

$$W_\ell = \frac{\ell!(K-\ell)!}{K!}$$

and

$$Z_{m,\ell} = \frac{(m+\ell-1)!(k-1)!}{(m-1)!(k+\ell-1)!}.$$

The variance of the unbiased estimate of average risk is bounded by

$$\text{Var}[\widehat{AvRisk}_k(\mathcal{F})] \leq \frac{1}{4r}\vec{W}^T(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\vec{W}.$$

The bound depends only on the quantities r, d, k, K, and can be easily computed numerically. However, it is easy to see that in the $r \to \infty$ limit, consistent estimation results. It remains to understand how the asymptotic performance of our estimation procedure depends on the parameters d, k, K.

Define

$$\kappa(k, K, d) = \vec{W}^T(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\vec{W}.$$

3.5 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 $\hat{K}(u)$.

A Appendix

A.1 Measurement error models

We make use of measurement error regression models in section 3.3: in particular, the regression adjustment method. In this section we focus on the specific measurement error models and methods that are used in this work: for an overview of the subject, the reader can consult (Caroll, 2005).

In the traditional setting of linear regression, one observes data $(\boldsymbol{x}_i, y_i)_{i=1}^n$ where the data is generated from the linear model

$$Y_i = \boldsymbol{x}_i^T \beta + \epsilon_i$$

for some random noise variables ϵ_i that satisfy $\mathbf{E}[\epsilon_i] = 0$. However, in measurement error models, we assume that $\boldsymbol{X}_i$ is not directly observed. (We write $\boldsymbol{X}_i$ instead of $\boldsymbol{x}_i$, since in some measurement error models the true covariates are also assumed to be random.) Instead, one only has access to a corrupted version of the covariates, $\boldsymbol{Z}_i$. We say that $\boldsymbol{Z}_i$ are *noisy measurements* of $\boldsymbol{X}_i$. Depending on the noise model for $\boldsymbol{Z}_i | \boldsymbol{X}_i$, the OLS coefficients for the regression of Y on $\boldsymbol{Z}$

$$\hat{\beta}_{naive} = \text{argmin}_\beta \sum_{i=1}^n (y_i - \boldsymbol{z}_i^T \beta)^2 \tag{9}$$

may be a biased estimated of β.

Suppose we can true the true covariates $\boldsymbol{X}_i$ as random variables. We say that the measurement error is *nondifferential* if $\boldsymbol{Z}_i$ is conditionally independent of Y_i given $\boldsymbol{X}_i$. In other words, the distribution of Y only depends on $(\boldsymbol{X}, \boldsymbol{Z})$ through $\boldsymbol{X}$. For nondifferential measurement error, it is possible to obtain an unbiased estimate of β by using *regression calibration.*

Regression calibration requires the ability to estimate or exactly compute the conditional expectation $\mathbf{E}[\boldsymbol{X}_i | \boldsymbol{Z}_i]$. For simplicity, suppose that there exists a known function g such that $g(\boldsymbol{z}) = \mathbf{E}[\boldsymbol{X} | \boldsymbol{Z} = \boldsymbol{z}]$. Then, define $\hat{\boldsymbol{x}}_i = g(\boldsymbol{z}_i)$ for $i = 1, \ldots, n$. The calibrated estimate of β is

$$\hat{\beta}_{rc} = \text{argmin}_\beta \sum_{i=1}^n (y_i - \hat{\boldsymbol{x}}_i^T \beta)^2 \tag{10}$$

We can see that the estimate is unbiased by the following identity. Observe that

$$\begin{aligned}\mathbf{E}[Y|\hat{\boldsymbol{X}}] &= \mathbf{E}[Y|\boldsymbol{Z}]\\ &= \mathbf{E}[\boldsymbol{X}^T\beta + \epsilon|\boldsymbol{Z}]\\ &= \mathbf{E}[\boldsymbol{X}|\boldsymbol{Z}]^T\beta\\ &= \hat{\boldsymbol{X}}^T\beta.\end{aligned}$$

This identity shows that β is the population regression coefficient of Y on $\hat{\boldsymbol{X}}$. Therefore, applying ordinary least squares to the data $(\hat{\boldsymbol{x}}_i, y_i)$ will produce an unbiased estimate of the population regression coefficient–that is, β.

In the *Berkson* measurement error model, we have

$$\boldsymbol{x}_i = \boldsymbol{z}_i + \eta_i$$

where η_i are independent error variates such that $\eta_i \perp \epsilon_i|\boldsymbol{z}_i$. Thus, in the Berkson model, the measurement error is nondifferentiable, and regression calibration can be applied to estimate β. Furthermore, supposing that $\mathbf{E}[\eta_i] = 0$, then we can simply take $\hat{\boldsymbol{x}} = \boldsymbol{z}_i$–in other words, the naive OLS estimate is equivalent to the regression calibration estimate, and therefore no correction is needed. However, suppose we have a *nonlinear additive model* of the form

$$Y_i = \sum_{j=1}^{d} \beta_j h_j(X_i) + \epsilon_i.$$

Then, even if $\mathbf{E}[X_i|Z_i] = Z_i$, it is not true in general that $\mathbf{E}[h_j(X_i)|Z_i] = Z_i$. Therefore, one needs to construct unbiased (or approximately unbiased) estimates $H_{i,j}$ such that

$$\mathbf{E}[H_{i,j}|Z_i] = \mathbf{E}[h_j(X_i)|Z_i]$$

and estimate β via

$$\hat{\beta}_{rc} = \text{argmin}_\beta \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} \beta_j H_{i,j} \right)^2.$$

A.2 Regression with importance sampling

This section describes a fairly specialized problem in linear regression which we encounter in section 3.3.

Suppose $\{F_\theta\}$ is a family of bivariate distributions for random variates $(\boldsymbol{X}, Y)$, with $\boldsymbol{X} \in \mathbb{R}^p$ and $Y \in \mathbb{R}$, indexed by parameter $\theta \in \Theta$. Let G, H be distributions on Θ. Define F^* as the bivariate distribution obtained by mixing over $\theta \sim G$:

$$F^*(A) = \int_\Theta F_\theta(A) dG(\theta)$$

for any measurable set $A \in \mathbb{R}^p \times \mathbb{R}$. Now suppose that Y has a conditional expectation which is linear in $\boldsymbol{X}$ under F^*:

$$\mathbf{E}_{F^*}[Y|\boldsymbol{X}] = \boldsymbol{X}^T \beta$$

for some $\beta \in \mathbb{R}^p$. Equivalently,

$$\int_{A\times\mathbb{R}} yI(\boldsymbol{x} \in A) dF^*(\boldsymbol{x}, y) = \beta^T \left(\int_{A\times\mathbb{R}} \boldsymbol{x} I(\boldsymbol{x} \in A) dF^*(\boldsymbol{x}, y) \right)$$

for all measurable $A \subset \mathbb{R}^p$.

However, the conditional expectation $\mathbf{E}_{F_\theta}[Y|\boldsymbol{X}]$ need not be linear in $\boldsymbol{X}$ for $\theta \in \Theta$.

Now, suppose we observe data triples (θ_i, x_i, y_i) which are generated as follows:

- Draw $\theta_i \sim H$.
- Draw $(X_i, Y_i) \sim F_\theta$.

Let η be a Radon-Nikodym derivative of H with respect to G, so that $H = \eta G$. Then an unbiased estimate of β can be obtained via an importance sampling estimate:

$$\hat{\beta} = \text{argmin}_\beta \sum_{i=1}^n \frac{1}{\eta(\theta_i)} (y_i - \boldsymbol{x}_i^T \beta).$$

A.3 Proofs

Lemma A.1 *Defining* $U_{y,\hat{F}_y}(x)$ *as in* (2).

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 .