STANFORD UNIVERSITY

DOCTORAL THESIS

Information, Prediction, and Supervised Learning

Author:
Charles ZHENG

Supervisor:
Dr. Trevor HASTIE and Dr. Jonathan TAYLOR

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Department of Statistics

March 7, 2017

Declaration of Authorship

I, Charles ZHENG, declare that this thesis titled, "Information, Prediction, and Supervised Learning" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Stanford University

Abstract

Faculty Name
Department of Statistics

Doctor of Philosophy

Information, Prediction, and Supervised Learning

by Charles ZHENG

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor...

Contents

Chapter 1

Introduction

1.1 Introduction

The study of complex systems.

1.2 Supervised learning

The generalization error of the learner as a statistic.

1.2.1 General characaterization of supervised learning

1.3 Mutual information

1.3.1 Definition and history

1.3.2 Usage in neuroscience

1.4 Generalizations of information

1.4.1 Information axioms

1.4.2 Information coefficients based on supervised learning

Chapter 2

Randomized classification

2.1 Motivation

2.1.1 Facial recognition example

2.2 Setup

2.2.1 Sampling scheme

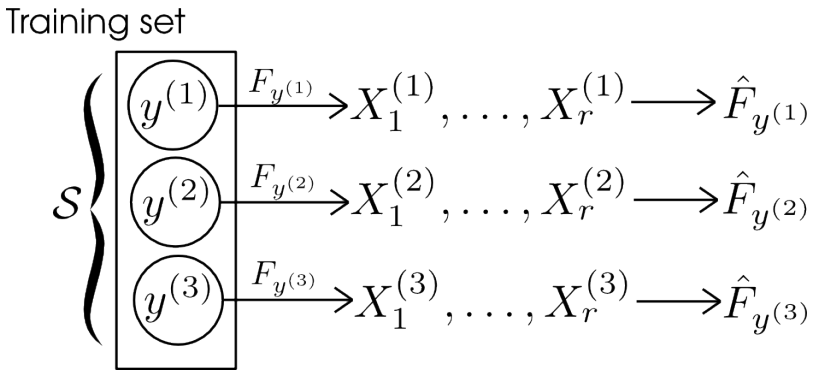

FIGURE 2.1: Training set

A *classification task* consists of a subset of labels, $\mathcal{S} \subset \mathcal{Y}$. Write $\mathcal{S} = \{y_1, \ldots, y_k\}$, where k is the number of classes. It is convenient to decouple the joint distribution 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 the uniform distribution[1] on $\mathcal{S}$ and a feature vector drawn under F_Y, the loss of $\ell(f(X), Y)$ is obtained. The *risk*, or expected

[1] See the discussion for extensions to non-uniform priors.

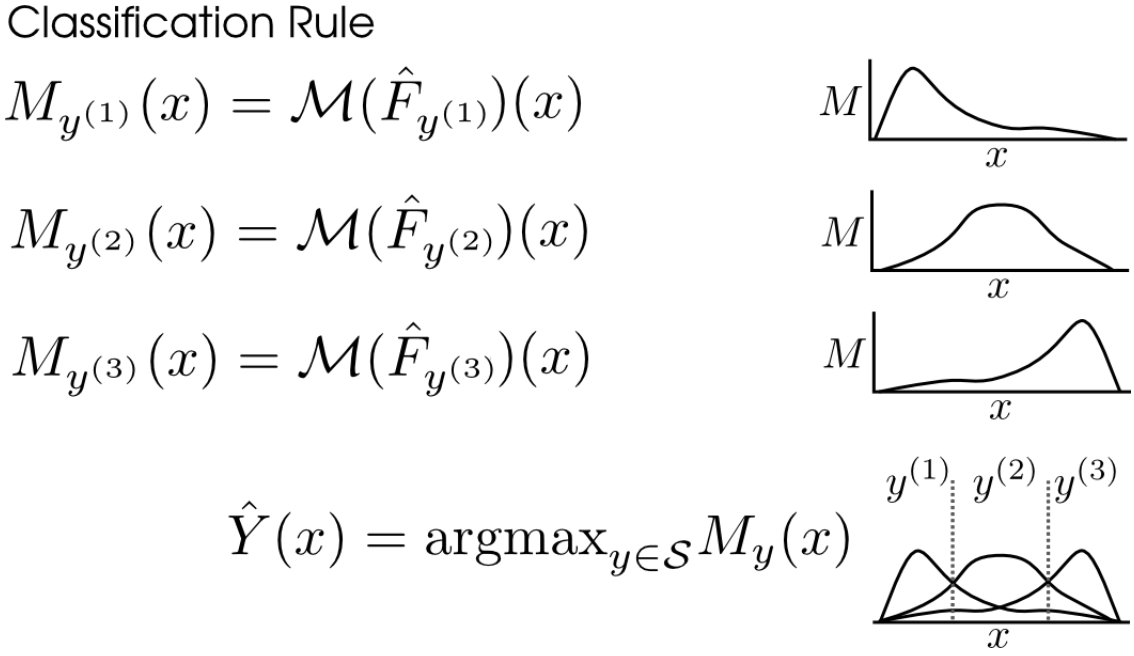

FIGURE 2.2: Classification rule

loss, of the classification rule is

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

For now, we will assume a 0–1 loss and a uniform prior over the labels in $\mathcal{S}$. Therefore, the risk can be rewritten as

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

The classification rule itself can be seen as a random function that depends on the sampling of the training set. For convenience, assume that the training set is composed of r i.i.d examples for each label $y \in \mathcal{S}$ (a total of $k \times r$). 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 $\hat{F}_y$.

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

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

We can therefore describe the r-repeat 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,

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

Figure **??** illustrates the variables involved in defining the risk.

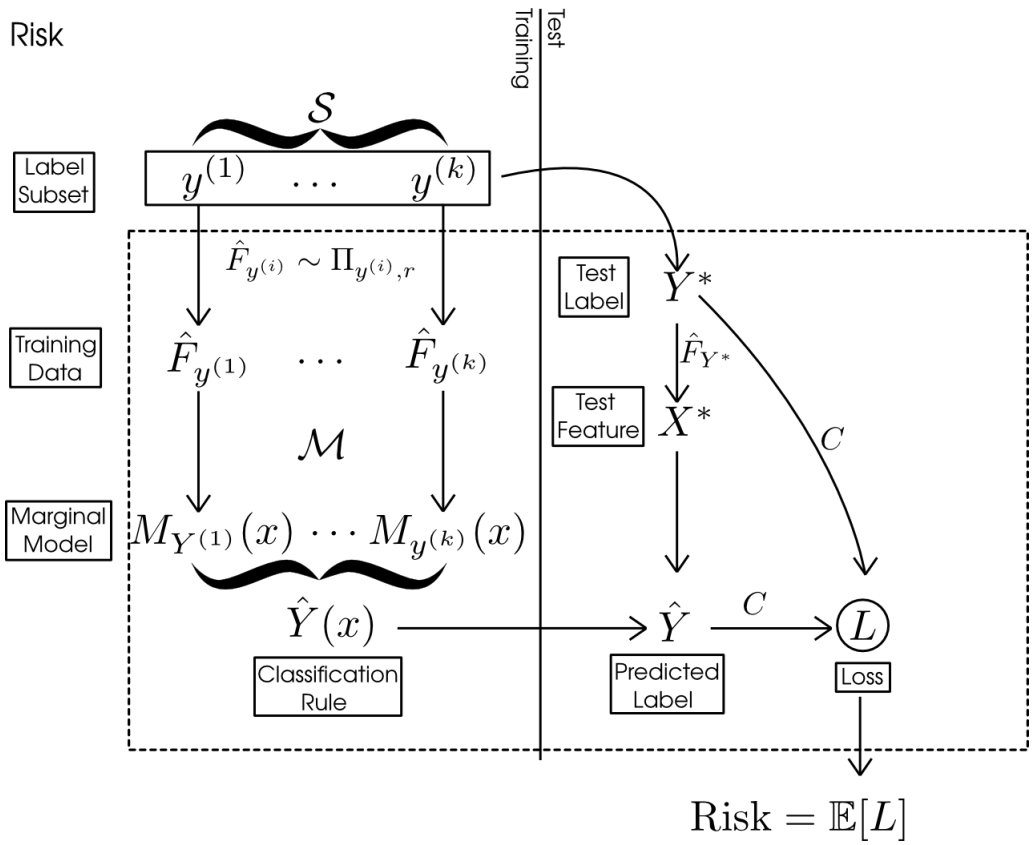

FIGURE 2.3: Classification risk

2.2.2 Average accuracy

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 $\mathcal{F}$ as

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

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

As we can see from Figure **??**, 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^* uniformly at random from $\mathcal{S}$.

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

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

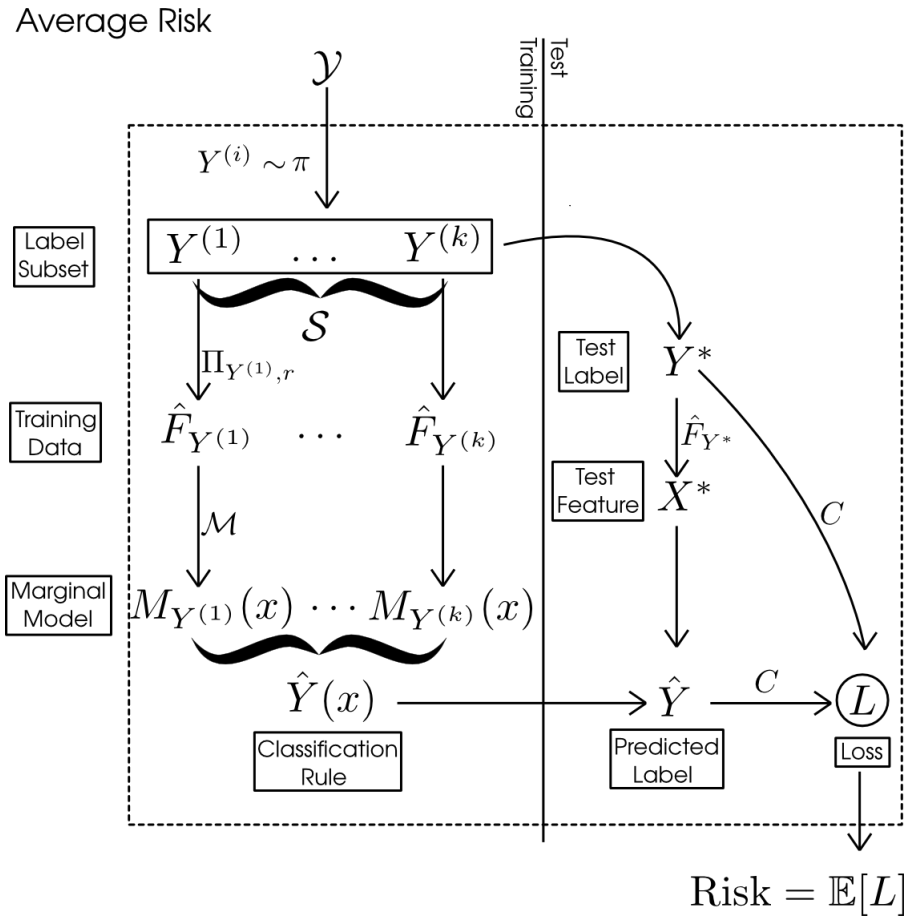

FIGURE 2.4: Average risk

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

$$\text{AvRisk}_{k,r,\nu}(\mathcal{F}) = \mathbf{E}[L] = \mathbf{E}[C(\mathcal{F}(\{\hat{F}_y\}_{y\in\mathcal{S}})(X^*), Y^*)]. \tag{2.1}$$

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

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.

2.3 Estimation of average accuracy

2.3.1 Subsampling method

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 $y^{(i)}$ for $i = 1, \dots, k_1$, as well as training sample $\hat{F}_{y^{(i)}}$ and test sample $(x_1^{(i)}, \dots, x_{r_{test}}^{(i)})$ for $i = 1, \dots, k_1$.

For any given test observation $x_j^{(i)}$, we obtain the predicted label $\hat{y}_j^{(i)}$ by computing the margin for each class,

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

for $\ell = 1, \ldots, k$, and by finding the class with the highest margin $M_{i,j,\ell}$,

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

The test risk is the average cost over test observations,

$$\text{Test Risk} = \frac{1}{r_{test}k} \sum_{i=1}^{k} \sum_{j=1}^{r_{test}} C(\hat{y}_j^{(i)}, y^{(i)}). \tag{2.2}$$

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

$$\text{Test Risk} = \frac{1}{r_{test}k} \sum_{i=1}^{k} \sum_{\ell=1}^{k} \sum_{j=1}^{r_{test}} C_{ij} I\{R_{ij\ell} = k\}. \tag{2.3}$$

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.

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 (**??**) 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 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

$$\text{AvTestRisk}_{k_2} = \frac{1}{\binom{k_1}{k_2}} \frac{1}{r_{test}k_2} \sum_{i=1}^{k_1} \sum_{\ell \neq i} \sum_{j=1}^{r_{test}} C_{i\ell} N_{i,j,\ell}. \tag{2.4}$$

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

$$N_{i,j,\ell} = I\{R_{i,j,\ell} > R_{i,j,i}\}\binom{R_{i,j,\ell} - 2}{k_2 - 2} \tag{2.5}$$

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.

2.3.2 Extrapolation

2.4 Average Bayes accuracy

The generalization accuracy of any classification rule is upper-bounded by the accuracy of the optimal classification rule, or *Bayes rule.* That is, one can define the *Bayes accuracy* as

$$\text{BA} = \sup_f \text{GA}(f).$$

And due to Bayes' theorem, the optimal classification rule f^* which achieves the Bayes accuracy can be given explicitly: it is the maximum a posteriori (MAP) rule

$$f^*(y) = \text{argmax}_{i=1}^k \, p(y|x^{(i)}).$$

Of course, it is not possible to construct this rule in practice since the joint distribution is unknown. Instead, a reasonable approach is to try a variety of classifiers, producing rules $f_1, \ldots, f_m$, and taking the best generalization accuracy as an estimate of the Bayes accuracy.

2.4.1 Definitions

Suppose X and Y are continuous random variables (or vectors) which have a joint distribution with density $p(x, y)$. Let $p(x) = \int p(x, y)dy$ and $p(y) = \int p(x, y)dx$ denote the respective marginal distributions, and $p(y|x) = p(x, y)/p(x)$ denote the conditional distribution.

ABA_k, or k-class Average Bayes accuracy is defined as follows. Let $X_1, ..., X_K$ be iid from $p(x)$, and draw Z uniformly from $1, .., k$. Draw $Y \sim p(y|X_Z)$. Then, the average Bayes accuracy is defined as

$$\text{ABA}_k[p(x, y)] = \sup_f \Pr[f(X_1, ..., X_k, Y) = Z]$$

where the supremum is taken over all functions f. A function f which achieves the supremum is

$$f_{Bayes}(x_1, ..., x_k, y) = \text{argmax}_{z\in\{1,...,k\}} p(y|x_z),$$

where an arbitrary rule can be employed to break ties. Such a function f_{Bayes} is called a *Bayes classification rule*. It follows that ABA_k is given explicitly by

$$\text{ABA}_k = \frac{1}{k} \int \left[\prod_{i=1}^k p(x_i)dx_i\right] \int dy \max_i p(y|x_i),$$

as stated in the following theorem.

Theorem 2.4.1 *For a joint distribution $p(x, y)$, define*

$$ABA_k[p(x,y)] = \sup_f \Pr[f(x_1, ..., x_k, y) = Z]$$

where $X_1, ..., X_K$ are iid from $p(x)$, Z is uniform from $1, .., k$, and $Y \sim p(y|X_Z)$, and the supremum is taken over all functions $f : \mathcal{X}^k \times \mathcal{Y} \to \{1, ..., k\}$. Then,

$$ABA_k = \frac{1}{k} \int \left[\prod_{i=1}^{k} p(x_i) dx_i \right] \int dy \max_i p(y|x_i).$$

Proof. First, we claim that the supremum is attained by choosing

$$f(x_1, ..., x_k, y) = \text{argmax}_{z \in \{1,...,k\}} p(y|x_z).$$

To show this claim, write

$$\sup_f \Pr[f(X_1, ..., X_k, Y) = Z] = \sup_f \frac{1}{k} \int p_X(x_1) \dots p_X(x_k) p(y|x_{f(x_1,...,x_k,y)}) dx_1 \dots dx_k dy$$

We see that maximizing $\Pr[f(X_1, ..., X_k, Y) = Z]$ over functions f additively decomposes into infinitely many subproblems, where in each subproblem we are given $\{x_1, ..., x_k, y\} \in \mathcal{X}^k \times \mathcal{Y}$, and our goal is to choose $f(x_1, ..., x_k, y)$ from the set $\{1, ..., k\}$ in order to maximize the quantity $p(y|x_{f(x_1,...,x_k,y)})$. In each subproblem, the maximum is attained by setting $f(x_1, ..., x_k, y) = \text{argmax}_z p(y|x_z)$–and the resulting function f attains the supremum to the functional optimization problem. This proves the claim.

We therefore have

$$p(y|x_{f(x_1,...,x_k,y)}) = \max_{i=1}^{k} p(y|x_i).$$

Therefore, we can write

$$\begin{aligned} \text{ABA}_k[p(x,y)] &= \sup_f \Pr[f(X_1, ..., X_k, Y) = Z] \\ &= \frac{1}{k} \int p_X(x_1) \dots p_X(x_k) p(y|x_{f(x_1,...,x_k,y)}) dx_1 \dots dx_k dy. \\ &= \frac{1}{k} \int p_X(x_1) \dots p_X(x_k) \max_{i=1}^{k} p(y|x_i) dx_1 \dots dx_k dy. \end{aligned}$$

2.5 Variability of Bayes Accuracy

We have

$$\text{ABA}_k = \mathbf{E}[\text{BA}(X_1, ..., X_k)]$$

where the expectation is over the independent sampling of $X_1, ..., X_k$ from $p(x)$.

Therefore, $\text{BA}_k = \text{BA}(X_1, ..., X_k)$ is already an unbiased estimator of ABA_k. However, to get confidence intervals for ABA_k, we also need to know the variability.

We have the following upper bound on the variability.

Theorem 2.5.1 *Given joint density $p(x, y)$, for $X_1, ..., X_k \overset{iid}{\sim} p(x)$, we have*

$$Var[BA(X_1, ..., X_k)] \leq \frac{1}{4k}.$$

Proof. According to the Efron-Stein lemma,

$$\text{Var}[\text{BA}(X_1, ..., X_k)] \leq \sum_{i=1}^{k} \mathbf{E}[\text{Var}[\text{BA}|X_1, ..., X_{i-1}, X_{i+1}, ..., X_k]].$$

which is the same as

$$\text{Var}[\text{BA}(X_1, ..., X_k)] \leq k\mathbf{E}[\text{Var}[\text{BA}|X_1, ..., X_{k-1}]].$$

The term $\text{Var}[\text{BA}|X_1, ..., X_{k-1}]$ is the variance of $\text{BA}(X_1, ..., X_k)$ conditional on fixing the first $k-1$ curves $p(y|x_1), ..., p(y|x_{k-1})$ and allowing the final curve $p(y|X_k)$ to vary randomly.

Note the following trivial results

$$-p(y|x_k) + \max_{i=1}^{k} p(y|x_i) \leq \max_{i=1}^{k-1} p(y|x_i) \leq \max_{i=1}^{k} p(y|x_i).$$

This implies

$$\text{BA}(X_1, ..., X_k) - \frac{1}{k} \leq \frac{k-1}{k}\text{BA}(X_1, ..., X_{k-1}) \leq \text{BA}(X_1, ..., X_k).$$

i.e. conditional on $(X_1, ..., X_{k-1})$, BA_k is supported on an interval of size $1/k$. Therefore,

$$\text{Var}[\text{BA}|X_1, ..., X_{k-1}] \leq \frac{1}{4k^2}$$

since $\frac{1}{4c^2}$ is the maximal variance for any r.v. with support of length c. □

2.5.1 Inference of average Bayes accuracy

2.5.2 Classification without model selection

Recall the notation used in section 2.1: the k stimuli exemplars are denoted $\{x^{(1)}, \ldots, x^{(k)}\}$ and the r responses for the ith class are given by $y^{(i),1}, \ldots, y^{(i),r}$.

Recall that *data-splitting*, one creates a *training set* consisting of r_1 repeats per class,

$$\{(x^{(1)}, y^{(1),1}), \ldots, (x^1, y^{(1),r_1}), \ldots, (x^{(k)}, y^{(k),1}), \ldots, (x^{(m)}, y^{(m),r_1})\}$$

and a *test set* consisting of the remaining $r_2 = r - r_1$ repeats.

$$\{(x^{(1)}, y^{(1),r_1+1}), \ldots, (x^{(1)}, y^{(1),r}), \ldots, (x^{(k)}, y^{(k),r_1+1}), \ldots, (x^{(k)}, y^{(k),r_1})\}.$$

One inputs the training data into the classifier to obtain the classification rule f,

$$f = \mathcal{F}(\{(x^{(1)}, y^{(1),1}), \ldots, (x^{(1)}, y^{(1),r_1}), \ldots, (x^{(k)}, y^{(k),1}, \ldots, (x^{(k)}, y^{(k),r_1})\}).$$

The test statistic of interest is the test error, defined as

$$\widehat{\text{GA}} = \frac{1}{kr_2} \sum_{i=1}^{k} \sum_{j=r_1+1}^{r} \text{I}(f(y^{(i),j}) \neq i).$$

Since $kr_2\widehat{\text{GA}}$ is a sum of independent binary random variables, from Hoeffding's inequality, we have

$$\Pr[\widehat{\text{GA}} > \text{GA} + \frac{t}{kr_2}] \leq 2e^{-2kr_2t^2}.$$

Therefore,

$$\underline{\text{GA}}_\alpha = \widehat{\text{GA}} - \sqrt{\frac{-\log(\alpha/2)}{2kr_2}}$$

is a $(1-\alpha)$ lower confidence bound for $\text{GA}(f)$. But, since

$$\text{GA}(f) \leq \text{BA}(x^{(1)}, \ldots, x^{(k)}),$$

it follows that $\underline{GA}_\alpha$ is also a $(1-\alpha)$ lower confidence bound for $\text{BA}(x^{(1)}, \ldots, x^{(k)})$.

Next, consider the variance bound for BA. From Chebyshev's inequality,

$$\Pr[|\text{BA}(X^{(1)}, \ldots, X^{(k)}) - \text{ABA}_k| > \frac{1}{\sqrt{4\alpha k}}] \leq \alpha.$$

Combining these facts, we get the following result.

Theorem 2.5.2 *The following is a $(1-\alpha)$ lower confidence bound for ABA_k:*

$$\underline{ABA}_k = \widehat{GA} - \sqrt{\frac{-\log(\alpha/4)}{2kr_2}} - \frac{1}{\sqrt{2\alpha k}}.$$

That is, for all joint densities $p(x, y)$,

$$\Pr[\underline{ABA}_K > ABA_k] \leq \alpha.$$

Proof. Suppose that both $\text{BA}(X^{(1)}, \ldots, X^{(k)}) \leq \text{ABA}_k + \frac{1}{\sqrt{2\alpha k}}$ and $\underline{\text{GA}}_{\alpha/2} \leq \text{GA}$. Then it follows that

$$\underline{\text{GA}}_{\alpha/2} \leq \text{BA}(X^{(1)}, \ldots, X^{(k)}) \leq \text{ABA}_k + \frac{1}{\sqrt{2\alpha k}}$$

and hence

$$\underline{\text{ABA}}_k = \underline{\text{GA}}_{\alpha/2} - \frac{1}{\sqrt{2\alpha k}} \leq \text{ABA}_k.$$

Therefore, in order for a type I error to occur, either $\text{BA}(X^{(1)}, \ldots, X^{(k)}) > \text{ABA}_k + \frac{1}{\sqrt{2\alpha k}}$ or $\underline{\text{GA}}_{\alpha/2} > \text{GA}$. But each of these two events has probability of at most $\alpha/2$, hence the union of the probabilities is at most α. □

2.5.3 Classification with model selection

In practice, it is common to evaluate multiple classifiers on the test set, ultimately *selecting* the classifier with the best test performance. Due to selection, the test accuracy $\widehat{\text{GA}}$ of the selected classifier becomes biased upwards with respect to the true generalization accuracy. Nevertheless, we can correct for the selection effect using the Bonferroni correction.

Suppose the investigator begins with classifiers $\mathcal{F}_1, \ldots, \mathcal{F}_\ell$, and obtains corresponding classification rules $f_1, \ldots, f_\ell$ via

$$f_i = \mathcal{F}_i(\{(x^{(1)}, y^{(1),1}), \ldots, (x^{(1)}, y^{(1),r_1}), \ldots, (x^{(k)}, y^{(k),1}, \ldots, (x^{(k)}, y^{(k),r_1})\}).$$

for $i = 1, \ldots, \ell$. Next, they evaluate the test accuracies $\widehat{\mathrm{GA}}(f_i)$ according to (**??**). Since $\mathrm{BA}(x^{(1)}, \ldots, x^{(k)}) \geq \max_i \mathrm{GA}(f_i)$, we have the following lemma.

Lemma 2.5.1 *The following is a* $(1-\alpha)$ *lower confidence bound for* $BA(x^{(1)}, \ldots, x^{(k)})$*:*

$$\underline{BA}_\alpha(x^{(1)}, \ldots, x^{(k)}) = \max_{i=1}^{\ell} \underline{GA}_{\alpha/\ell}(f_i) = \max_{i=1}^{\ell} \widehat{GA}(f_i) - \sqrt{\frac{-\log(\alpha/(2\ell))}{2kr_2}}.$$

Proof. In order for type I error to occur, $\underline{\mathrm{GA}}_{\alpha/\ell}(f_i) \geq \mathrm{BA}(x^{(1)}, \ldots, x^{(k)}) \geq \mathrm{GA}(f_i)$ for some $i = 1, \ldots, \ell$. For each i, the event occurs with probability at most α/ℓ. Therefore, by the union bound, the probability of type I error is at most α. □

It remains to apply the variance bound for Bayes accuracy to obtain a lower confidence bound for ABA_k:

$$\underline{\mathrm{ABA}}_k = \underline{\mathrm{BA}}_{\alpha/2} - \frac{1}{\sqrt{2\alpha k}}$$

2.6 Identification task

The identification task originated as a method for evaluating the quality of encoding models in neuroscience (Kay 2008).

2.6.1 Experimental design

We consider experiments in which a single subject is presented with a sequence of T stimuli: each stimulus is presented during a 'task window' of a fixed duration. The stimuli are represented by real-valued feature vectors $\vec{X}$; let p be the dimensionality of the feature space. The brain activity of the subject is recorded, yielding a q-dimensional vector $\vec{Y}$: in practice, $\vec{Y}$ could consist of discretized time series data or mean firing rates for spike-sorted neurons, or BOLD response for voxels, depending on the recording modality. Let $\vec{X}^{(t)}$ denote the feature vector of the stimulus, and let $\vec{Y}^{(t)}$ denote the vector of intensities (e.g. BOLD response, mean spike) for the tth task window in the sequence.

2.6.2 Data splitting

The T stimulus-response pairs $(\vec{X}, \vec{Y})$ are randomly partitioned into a *training set* of size N and a *test set* of size $M = T - N$. Form the $N \times p$ data matrix $\boldsymbol{X}^{tr}$ by stacking the features of the N training set stimuli as row vectors, and stack the corresponding responses as row vectors to form the $N \times q$ matrix $\boldsymbol{Y}^{tr}$. Similarly, define $\boldsymbol{X}^{te}$ as the $N \times p$ matrix of test stimuli and $\boldsymbol{Y}^{te}$ as the $N \times q$ matrix of corresponding test responses.

2.6.3 Probabilistic encoding model

The data is used to estimate a stimulus-based encoding model Kay et al. 2008Naselaris et al. 2011Mitchell et al. 2008. The conditional mean response $\mathbf{E}[\boldsymbol{Y}|\boldsymbol{X}]$ is modelled as a linear transformation of the stimulus features,

$$\vec{Y} = \boldsymbol{B}^T \vec{X} + \boldsymbol{\epsilon}$$

where $\boldsymbol{B}$ is a $p \times q$ coefficient matrix and $\boldsymbol{\epsilon}$ is a noise variable with an assumed multivariate normal distribution, $\boldsymbol{\epsilon} \sim N(0, \Sigma)$. Hence, the conditional density of $\vec{Y}|\vec{X}$ is given by the multivariate normal density

$$p(\vec{y}|\vec{x}) = -\frac{1}{(2\pi|\Sigma|)^{-q/2}} \exp\left[-\frac{1}{2}(\vec{y} - \boldsymbol{B}^T\vec{x})^T\Sigma^{-1}(\vec{y} - \boldsymbol{B}^T\vec{x})\right].$$

The coefficient B can be estimated from the training set data $(\boldsymbol{X}^{tr}, \boldsymbol{Y}^{tr})$ using a variety of methods for regularized regression, for instance, the elastic net Zou and Hastie 2005, where each column of $\boldsymbol{B} = (\beta_1, \ldots, \beta_q)$ is estimated via

$$\hat{\beta}_i = \text{argmin}_\beta ||\boldsymbol{Y}_i^{tr} - \boldsymbol{X}^{tr}\beta||^2 + \lambda_1||\beta||_1 + \lambda_2||\beta||_2^2,$$

where λ_1 and λ_2 are regularization parameters which can be chosen via cross-validation Hastie, Tibshirani, and Friedman 2009 separately for each column i.

After forming the estimated coefficient matrix $\hat{\boldsymbol{B}} = (\hat{\beta}_1, \ldots, \hat{\beta}_q)$, we estimate the noise covariance Σ via a shrunken covariance estimateLedoit and Wolf 2004Daniels and Kass 2001 from the residuals,

$$\hat{\Sigma} = \frac{1}{N}((1-\lambda)S + \lambda\text{Diag}(S))$$

where

$$S = (\boldsymbol{Y}^{tr} - \boldsymbol{X}^{tr}\boldsymbol{B})^T(\boldsymbol{Y}^{tr} - \boldsymbol{X}^{tr}\boldsymbol{B}).$$

2.6.4 Converting the encoding model to a decoding model

Bayes' rule can be used to convert a probabilistic encoding model into a decoding model Naselaris et al. 2011. The Bayesian decoding model gives the posterior probability of the stimulus given the response,

$$p(\vec{x}|\vec{y}) = p(\vec{y}|\vec{x})\frac{p(\vec{x})}{p(\vec{y})}.$$

In an *identification task*Kay et al. 2008, a response $\boldsymbol{y}$ is generated by presenting the subject to a stimulus which is randomly chosen from a subset of k stimuli, $S = (\vec{x}^{(1)}, \ldots, \vec{x}^{(k)})$. The decoder is used to select the stimulus in S which is most likely to have generated the response $\boldsymbol{y}$: the performance of the the decoder is measured by the probability of correct identification. In the identification task, the prior probability $p(\vec{x})$ is uniform over the candidate set S. Therefore, the estimated log posterior probability of each candidate stimulus $\vec{x}^{(i)}$ is given by

$$\log\hat{p}(\vec{x}|\vec{y}) = \log\hat{p}(\vec{y}|\vec{x}) + \text{const.} = -\frac{1}{2}(\vec{y} - \hat{\boldsymbol{B}}^T\vec{x})^T\hat{\Sigma}^{-1}(\vec{y} - \hat{\boldsymbol{B}}^T\vec{x}) + \text{const.}$$

where we have elided the inconsequential constant terms. Therefore, the chosen stimulus $\hat{\vec{x}}$ is the stimulus which minimizes the empirical Mahalanobis distance

$$d_{\hat{\Sigma}}(\vec{y}, \hat{\boldsymbol{B}}^T\vec{x}) = (\vec{y} - \hat{\boldsymbol{B}}^T\vec{x})^T\hat{\Sigma}^{-1}(\vec{y} - \hat{\boldsymbol{B}}^T\vec{x})$$

among the stimuli in S, and supposing that the correcty stimulus has index i, the probability of correct identification is

$$\Pr[\text{correct}] = \Pr[d_{\hat{\Sigma}}(\vec{y}, \hat{\boldsymbol{B}}^T \vec{x}^{(i)}) \leq \min_{j \neq i} d_{\hat{\Sigma}}(\vec{y}, \hat{\boldsymbol{B}}^T \vec{x}^{(j)})].$$

2.6.5 Computation of identification accuracy curve

The probability of correct identification varies depending on the choice of stimulus set S. Therefore, to obtain a well-defined measure of decoder precision, we define the k-class *identification risk* as the expected acccuracy when the set S is constructed by drawing $x^{(1)}, \ldots, x^{(k)}$ independently from the prior distribution $p(\vec{x})$.

An unbiased estimate of the k-class identification risk for any $k \leq M$ can be obtained, where M is the number of test observations. The idea is to evaluate the empirical accuracy (the proportion of correct identifications) over all combinations of $\binom{M}{k}$ stimulus subsets S times all k choices for the correct stimulus within S. Yet, this empirical accuracy can be computed without explicitly looping over all $\binom{kM}{k}$ combinations via a computational trick.

Suppose without loss of generality that the indices of the test observations are $i = 1, \ldots, M$. Define

$$M_{i,j} = \log \hat{p}(\vec{x}^{(j)} | \vec{y}^{(i)})$$

Furthermore, define

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

The computational trick is to look at each combination of test response $\vec{y}^{(i)}$ and stimulus $\vec{x}^{(\ell)}$, and to count count the number of subsets $N_{i,\ell}$ where (i) both i and ℓ are included in S, and (ii) $\hat{x}^{(i)} = \vec{x}^{(\ell)}$. One can then verify that the empirical accuracy over all subsets is equal to

$$\text{EmpAcc}_k = 1 - \frac{1}{\binom{M}{k}} \frac{1}{k} \sum_{i=1}^{k} \sum_{\ell \neq i} C_{i\ell} N_{i,\ell}. \tag{2.6}$$

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

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

$$N_{i,\ell} = I\{R_{i,\ell} > R_{i,i}\} \binom{R_{i,\ell} - 2}{k - 2} \tag{2.7}$$

The *identification accuracy curve* is defined as the function which maps $k \in 2, 3, \ldots$ to the k-class identification risk. Therefore, an estimate of a portion of the curve can be obtained by estimating the k-class identification risk for $k = 2, \ldots, M$.

Chapter 3

Extrapolating average accuracy

3.1 Motivation

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

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

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 $\bar{D}(u)$, one can compute the average risk for any number of classes. In section 5, we empirically study the performance curves of classifiers on sequences of classification tasks. Since marginal classifiers only comprise a minority of the classifiers used in practice, we applied our methods to a variety of marginal and non-marginal classifiers in simulations and in one OCR dataset. Our methods have varying success on marginal and non-marginal 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. [[TODO: mention estimation results, theory]]

3.1.1 Facial recognition example

3.2 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 1 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.

Comments:

1. These assumption are most easily satisfied by taking $\mathcal{Y}$ to be a continuous space and letting π 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 insofar 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.

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

Definition 3.2.1 *1. The classification rule f is called a* marginal rule *if*

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

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

2. Define a marginal model $\mathcal{M}$ as a mapping from empirical distributions to margin functions,

$$\mathcal{M}(\hat{F}_y) = m_y(x).$$

3. A classifier that produces marginal classification rules

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

by use of a marginal model, i.e. such that $m_y = \mathcal{M}(\hat{F}_y)$ for some marginal model $\mathcal{M}$, is called a marginal classifier.

In words, a marginal classification rule produces a *margin*, or score, for each label, and chooses the label with the highest margin. The marginal model converts empirical distributions $\hat{F}_y$ over $\mathcal{X}$ into the margin function m_y. The *marginal* property allows us to prove strong results about the accuracy of the classifier under i.i.d. sampling assumptions.

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 margin functions:

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

   The resulting empirical approximation for the Bayes classifier (further assuming a uniform prior π) 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 classifiers can be seen as specific instances of an estimated Bayes classifier [1]. For QDA, the margin function is given by

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

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

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

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

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

3.3 Analysis of average risk

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

Theorem 3.3.1 *Suppose π, $\{F_y\}_{y \in \mathcal{Y}}$ and marginal classifier $\mathcal{F}$ satisfy the tie-breaking condition. Then, under the definitions* (**??**), (**??**), *and* (**??**), *we have*

$$\textit{AvRisk}_{k,r} = (k-1) \int \bar{D}(u) u^{k-2} du. \tag{3.1}$$

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

The tie-breaking condition referred in the theorem is defined as follows.

- *Tie-breaking condition*: for all $x \in \mathcal{X}$, $\mathcal{M}(\hat{F}_Y)(x) = \mathcal{M}(\hat{F}_{Y'})(x)$ with zero probability for Y, Y' independently drawn from π.

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 $\mathcal{M}$.

Our strategy is to analyze the average risk (**??**) 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 **??** illustrates the variables which are fixed under conditioning and the variables which are randomized. Compare to figure **??**.

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 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 $M_{y^*}(x^*) = \mathcal{M}(\hat{F}_{y^*})(x^*)$ is also fixed.

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

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 **??**, we have essentially conditioned on steps A3 and A4 and randomized over steps A1 and A2.

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:

$$U_{x^*}(y^*, \hat{F}_{y^*}) = \Pr[M_{y^*}(x^*) > \mathcal{M}(\hat{F}_Y)(x^*)] \tag{3.2}$$

$$= \int_{\mathcal{Y}} I\{M_{y^*}(x^*) > \mathcal{M}(\hat{F}_y)(x)\} d\Pi_{y,r}(\hat{F}_y) d\pi(y). \tag{3.3}$$

See also figure **??** 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 $Y \sim \pi$ and $\hat{F}_Y \sim \Pi_{Y,r}$ is uniformly distributed for all $x \in \mathcal{X}$. This is proved in Lemma **??** 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

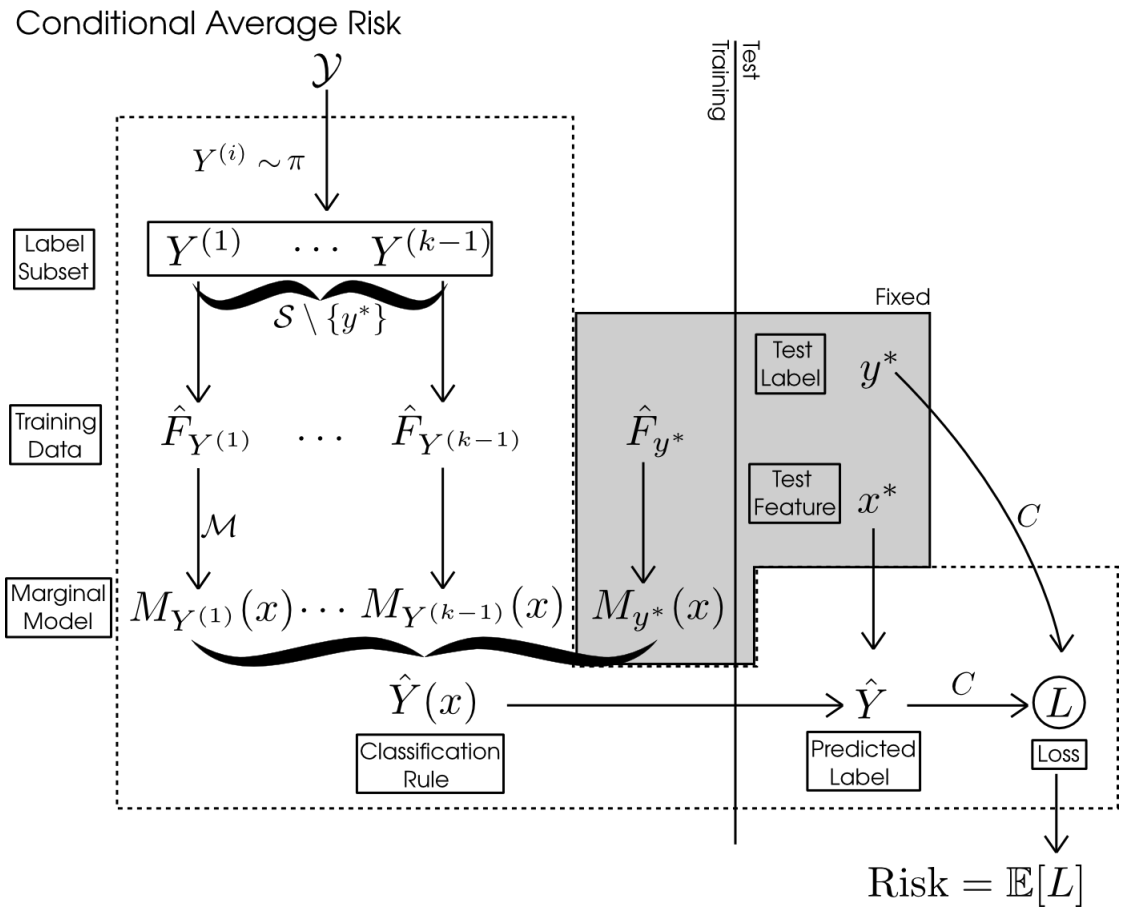

FIGURE 3.1: Conditional average risk

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

$$U_{max,k-1} \sim \text{Beta}(k-1, 1). \tag{3.4}$$

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

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

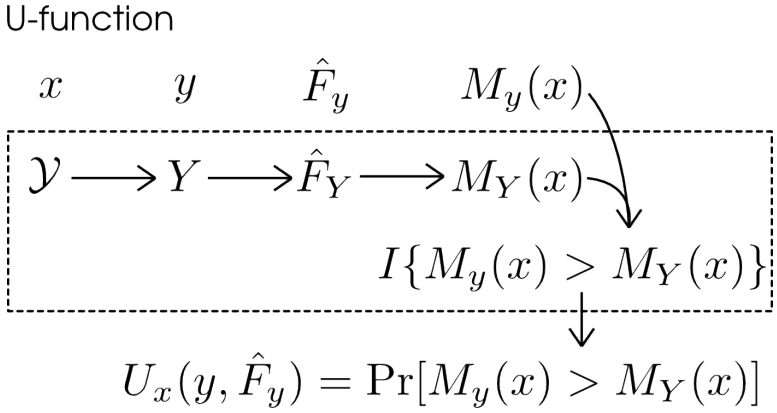

FIGURE 3.2: U-functions

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,

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

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

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

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

In section **??**, we will develop estimators of $\bar{D}(u)$ 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

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

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

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

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* $D((y^*, \hat{F}_{y^*}), x^*, u)$ by

$$D((y^*, \hat{F}_{y^*}), x^*, u) = \begin{cases} 0 \text{ if } u < u^* \\ \mathbf{E}[C(\hat{Y}, y^*)|U_{max} = u, x^*, y^*, \hat{F}_{y^*}] \text{ otherwise.} \end{cases} \tag{3.5}$$

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

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

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

$$\text{AvRisk}_{k,r} = (k-1) \int \bar{D}(u) u^{k-2} du.$$

where

$$\bar{D}(u) = \int D((y^*, \hat{F}_{y^*}), x^*, u) \pi(y^*) dy dF_{y^*}(x^*) d\Pi_{y^*,r}(\hat{F}_{y^*}). \tag{3.6}$$

This is the key result behind our estimation method, which was stated in theorem **??**. 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 $\bar{D}$–which we discuss in the following.

3.4 Estimation

Now we address the problem of estimating $\text{AvRisk}_{k_2,r_{train}}$ from data. As we have seen from Theorem **??**, the k-class average risk of a marginal classifier $\mathcal{M}$ is a functional of a object called $\bar{D}(u)$, 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 $\bar{D}$ for then given classification model, and then plug in our estimate of $\bar{D}$ into the integral (**??**) to obtain an estimate of $\text{AvRisk}_{k_2,r_{train}}$.

Having decided to estimate $\bar{D}$, there is then the question of what kind of model we should assume for $\bar{D}$. 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.

Let us assume the linear model

$$\bar{D}(u) = \sum_{\ell=1}^{m} \beta_\ell h_\ell(u), \tag{3.7}$$

where $h_\ell(u)$ are known basis functions, and β are the model parameters to be estimated. We can obtain *unbiased* estimation of $\text{AvRisk}_{k_2,r_{train}}$ via the unbiased estimates of k-class average risk obtained from (**??**).

If we plug in the assumed linear model (**??**) into the identity (**??**), then we get

$$\begin{aligned}
\text{AvRisk}_{k,r_{train}} &= (k-2)\int \bar{D}(u)u^{k-2}du && (3.8)\\
&= (k-2)\int_0^1 \sum_{\ell=1}^{m} \beta_\ell h_\ell(u)u^{k-2}du && (3.9)\\
&= \sum_{\ell=1}^{m} \beta_\ell H_{\ell,k} && (3.10)
\end{aligned}$$

where

$$H_{\ell,k} = (k-2)\int_0^1 h_\ell(u)u^{k-2}du. \tag{3.11}$$

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

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

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

$$\widehat{\text{AvRisk}}_{k_2,r_{train}} = \sum_{\ell=1}^{m} \hat{\beta}_\ell H_{\ell,k_2}. \tag{3.12}$$

3.4.1 Large-Sample Theory

How good are the estimated average risks (**??**)? Let us investigate the accuracy of the estimates in the limit where $k_1 \to \infty$, first in the case where the model (**??**) is correctly specified, and then considering possible model misspecification.

If we fix the number of classes k_2 which defines the estimation target, then we need not use the estimator (**??**), since once $k_1 > k_2$, we can use the 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 (**??**), 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

$$\text{MinimaxError} = \sup_{k_2>2} |\widehat{\text{AvRisk}}_{k_2,r_{train}} - \text{AvRisk}_{k_2,r_{train}}|. \tag{3.13}$$

Well-specified case.

Let us first assume that the parametric model (**??**) 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 3.4.1 *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 $\bar{D}(u)$ defined by $\{F_y\}_{y\in\mathcal{Y}}$, η, and $\mathcal{M}$ satisfies*

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

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

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

Then, defining

$$\hat{\beta} = \textit{argmin}_\beta \sum_{k=2}^{k_0} \left(\textit{AvTestRisk}_k - \sum_{\ell=1}^{m} \beta_\ell H_{\ell,k} \right)^2$$

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

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

□.

Misspecified case.

Now consider the more realistic setting where the model (**??**) is misspecified. We quantify the degree of misspecification by the ℓ_∞ error on [0,1]. Define

$$\delta = \inf_{\beta} \|\bar{D}(u) - \sum_{\ell=1}^{m} \beta_\ell h_\ell(u)\|_\infty,$$

and let $\tilde{\beta}$ be the coefficients β which attain the infimum, with $\tilde{D}(u) = \sum_{\ell=1}^m \tilde{\beta}_\ell h_\ell(u)$. To deal with this case, refer to the theory in section **??**. For each $u = [0, 1]$, find a matrix $A(u)$ such that (i) the first column equals

$$A_1(u) = (h_1(u), \ldots, h_m(u))$$

and that (ii) the rest of the columns are orthogonal to the first, and (iii) $A(u)$ is full-rank. Then define $Z(u) = XA(u)$, and consider the column vector

$$Z_{1|-1}(u) = (I - P_{Z_{-1}})Z_1(u).$$

It can be shown that $Z_{1|-1}(u)$ is well-defined, regardless of how $A(u)$ is chosen. Then, by the theory in section **??**, the extra bias due to approximation error for predicting $\hat{D}(u)$ is given by

$$\text{Bias}^2(u) = \frac{||Z_{1|-1}(u)||_1^2}{||Z_{1|-1}(u)||_2^4}.$$

Define the maximum bias as

$$\text{Bias}^2_{max} = \sup_{u\in[0,1]} \text{Bias}^2(u).$$

From the analysis of the well-specified case, we know that the variance component of the prediction risk decreases at order $O(1/k)$. Therefore, the misspecified minimax error is of order

$$\text{MinimaxError} = O(1/\sqrt{k}) + \text{Bias}^2_{max}.$$

3.5 Examples

Chapter 4

Inference of mutual information

4.1 Motivation

4.1.1 Gene expression dataset example

4.2 Identification loss

4.3 Average Bayes accuracy and Mutual information

4.4 Lower confidence bound

4.5 Example

Chapter 5

High-dimensional inference of mutual information

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

`\hypersetup{urlcolor=red}`, or

`\hypersetup{citecolor=green}`, or

`\hypersetup{allcolor=blue}`.

If you want to completely hide the links, you can use:

`\hypersetup{allcolors=.}`, or even better:

`\hypersetup{hidelinks}`.

If you want to have obvious links in the PDF but not the printed text, use:

`\hypersetup{colorlinks=false}`.

Bibliography

Daniels, Michael J. and Robert E. Kass (2001). "Shrinkage Estimators for Covariance Matrices". In: *Biometrics* 57.4, pp. 1173–1184. ISSN: 0006341X. DOI: 10.1111/j.0006-341X.2001.01173.x. URL: http://doi.wiley.com/10.1111/j.0006-341X.2001.01173.x.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning*. 2nd ed. Vol. 1. Springer, pp. 337–387. ISBN: 9780387848570. DOI: 10.1007/b94608. arXiv: 1010.3003. URL: http://www.springerlink.com/index/10.1007/b94608.

Kay, Kendrick N et al. (2008). "Identifying natural images from human brain activity." In: *Nature* 452.March, pp. 352–355. ISSN: 0028-0836. DOI: 10.1038/nature06713.

Ledoit, Olivier and Michael Wolf (2004). "Honey, I Shrunk the Sample Covariance Matrix". In: *The Journal of Portfolio Management* 30.4, pp. 110–119. ISSN: 0095-4918. DOI: 10.3905/jpm.2004.110.

Mitchell, Tom M. et al. (2008). "Predicting Human Brain Activity Associated with the Meanings of Nouns". In: *Science* 320.5880.

Naselaris, Thomas et al. (2011). "Encoding and decoding in fMRI". In: *NeuroImage* 56.2, pp. 400–410. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2010.07.073. URL: http://dx.doi.org/10.1016/j.neuroimage.2010.07.073.

Zou, Hui and Trevor Hastie (2005). "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2, pp. 301–320. ISSN: 1369-7412. DOI: 10.1111/j.1467-9868.2005.00503.x. URL: http://doi.wiley.com/10.1111/j.1467-9868.2005.00503.x.