

SUPERVISED EVALUATION OF REPRESENTATIONS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF STATISTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Charles Zheng

May 2017

Abstract

Intuitively speaking, a good data representation reveals meaningful differences between inputs, while being relatively invariant to differences between inputs due to irrelevant factors or noise. In this work, we consider criteria for formally defining the quality of a representation, which all make use of the availability of a *response variable* Y to distinguish between meaningful and meaningless variation. Hence, these are criteria for *supervised* evaluation of representations. We consider three particular criteria: the mutual information between the representation and the response, the average accuracy of a randomized classification task, and the identification accuracy. We discuss methods for estimating all three quantities, and also show how these three quantities are interrelated. Besides the application of evaluating representations, our work also has relevance for estimation of mutual information in high-dimensional data, for obtaining performance guarantees for recognition systems, and for making statements about the generalizability of certain kinds of classification-based experiments which are found in neuroscience.

Acknowledgements

First of all, I would like to thank my co-advisors, Trevor Hastie and Jonathan Taylor. Trevor once told me that “I am a bit of an unusual student,” but by the same token, Trevor is also an exceptional advisor. He really cares about making an impact in science and industry, and also understands the difficulty of applied work in statistics. Many times when I was frustrated about the difficulty of translating theoretical ideas into practically useful tools, Trevor would inspire me to never give up on trying to bridge that gap between theory and practice. I also learned a great deal from him about how to best present and communicate my ideas, which has helped immensely in my collaborations. I am no less fortunate to have had access to Jonathan, whose wide-spanning technical expertise and passion for both mathematical problem-solving and computation made him a valuable resource to consult whenever I ran into any problem which seemed beyond my grasp. I can always count on Jonathan, with his sharp eye for detail and his intensely curious mind, to provide numerous corrections, suggestions, or connections anytime that I show him my work in progress, and this applies especially to the thesis you now hold in your hands (or—more likely—which is now displayed on your monitor.)

In addition to Trevor and Jonathan, I had an additional, informal advisor in my collaborator Yuval Benjamini. Our extensive joint work, which forms the basis for this thesis, was inspired by the class he taught on statistical analysis of neuroscience data. I must also thank Yuval for his extraordinary support and assistance during my job search.

I would also like to thank my committee members. It has always been a pleasure to work with Russell Poldrack, and I am especially grateful to him for helping me

find a job! Bradley Efron has been an amazing inspiration and support to me as a student, and I am truly honored to have had the chance to discuss my research with one of the living legends of statistics. I am also deeply indebted to Tsachy Weissman for being willing to lend his expertise in information theory, a field ended up quite central to my research, despite my lack of knowledge in the subject. Of course, I would like to thank all of my committee members for their valuable comments and questions they provided during my oral defense, and I would like to thank my reading committee for agreeing to read my thesis and for the feedback they have provided.

I have also relied extensively on many other faculty and staff members at Stanford. It has always been a pleasure to work with Balasubramanian Narasimhan, or “Naras,” whose expertise at statistical computing makes him a truly unique asset to the department. My time with Naras, both as his teaching assistant and as his student, brought me to a higher level of expertise and appreciation for the beauty of programming. I am also grateful to Art Owen, David Siegmund, Wing Wong, Robert Tibshirani, Persi Diaconis, Richard Olshen, and Brian Wandell for many useful discussions. I am also greatly indebted to the department staff, not only for their work in keeping the department running, but also for the help they’ve personally given to me over the years for matters both great and small: Cindy Kirby, Caroline Gates, Susie Ementon, Ellen van Stone, Helen Tombropoulos, Heather Murthy, and Emily Lauderdale.

I would also like to thank the collaborators I have had over the years: Ariel Rokem, Franco Pestilli, Sanmi Koyejo, Jingshu Wang, Arzav Jain, Reza Zadeh, Bai Jiang, Nick Boyd, Rakesh Achanta, Qingyuan Zhao, and Asaf Weinstein. No less important are the friends at Stanford whom I have to blame for making the five years of my Ph. D. seem to pass in the blink of an eye: Rakesh Achanta, Chaojun Wang, Xin Zheng, Christof Seiler, Asaf Weinstein, Veniamin Morgenstern, Pratheepa Jeganathan, Lucy Xia, Kris Sankaran, Subharata Sen, Joey Arthur, Junyang Qian, Nan Bi, Jun Yan, Yuming Kuang, Ya Le, Zeyu Zheng, Ziyu Wang, Jiantao Jiao, Lucas Janson, Shuo Xie, Joshua Loftus, Murat Erdogdu, Qianyuan Zhao, Paulo Orenstein, Keli Liu, Stephen Bates, Hatef Monajemi, Edgar Dobriban, Cyrus DiCiccio, Oleg Lazarev, Zhou Fan, Peter Diao, Fanya Becks, Kelvin Gu, Yan Chen, Xiaoyan Han,

Haben Michael, Jiada Liu, Frederik Mikelsen, Roman Croessmann, and many others.

Finally, I am grateful to my family: my father, Buhong Zheng; my mother, Xue-meì Du; and my younger brother, Bruce Zheng, for supporting me during my Ph. D. studies, and even volunteering as a test audience for my job talk!

Note on attribution

The content in chapters 1,2, 4, and 5 is based on joint work with Yuval Benjamini. Chapter 3 is based on joint work with Yuval Benjamini and Rakesh Achanta. All theoretical results are due to the author.

Contents

Abstract	iv
Acknowledgements	v
1 Introduction	1
1.1 Finding the correct representation	1
1.1.1 Example: Receptive-field models for vision	1
1.1.2 Example: Face-recognition algorithms	4
1.1.3 What makes a good representation?	7
1.1.4 Supervised evaluation of representations	9
1.1.5 Related Work	10
1.2 Overview	13
1.2.1 Theme and variations	13
1.2.2 Organization	17
1.3 Information and Discrimination	18
1.3.1 Introduction	18
1.3.2 Supervised learning	20
1.3.3 Identification accuracy	30
1.3.4 Information Theory	36
1.3.5 Comparisons	44
2 Randomized classification	46
2.1 Recognition tasks	47
2.2 Randomized classification	48

2.2.1	Motivation	48
2.2.2	Setup	49
2.2.3	Assumptions	50
2.3	Estimation of average accuracy	56
2.3.1	Subsampling method	58
2.3.2	Extrapolation	58
2.3.3	Variance bounds	59
2.4	Reproducibility and Average Bayes accuracy	59
2.4.1	Motivation	59
2.4.2	Setup	60
2.4.3	Identities	61
2.4.4	Variability of Bayes Accuracy	62
2.4.5	Inference of average Bayes accuracy	63
2.4.6	Implications for reproducibility	64
2.4.7	Application to identification task	65
3	Extrapolating average accuracy	68
3.1	Introduction	68
3.2	Analysis of average risk	70
3.3	Estimation	75
3.4	Examples	77
3.4.1	Facial recognition example	77
3.4.2	Telugu OCR example	79
4	Inference of mutual information	82
4.1	Motivation	82
4.2	Average Bayes accuracy and Mutual information	84
4.2.1	Problem formulation and result	84
4.2.2	Reduction	85
4.2.3	Proof of theorem	91
4.3	Estimation	93
4.4	Simulation	94

5	High-dimensional inference of mutual information	96
5.1	Motivation	96
5.2	Theory	98
5.2.1	Assumptions	98
5.2.2	Limiting universality	99
5.3	Examples	104
5.3.1	Simulation	105
5.3.2	Real data example	107
6	Discussion	111
A	Appendix for Chapter 4	114
A.1	Proofs	114
	Bibliography	127

List of Tables

3.1	Performance extrapolation: predicting the accuracy on 400 classes using data from 20 classes on a Telugu character dataset. $\epsilon = 0.002$ for ϵ -nearest neighbors.	81
-----	---	----

List of Figures

1.1	Visual pathway in humans. Image credit to Ratznum under CC 2.5 license.	2
1.2	Examples of Gabor filters of varying size and orientation. From Haghighat, Zonouz, and Abdel-Mottaleb 2015.	3
1.3	Face recognition problem	5
1.4	Triplet loss function for training face representations. From Amos, Ludwiczuk, and Satyanarayanan 2016	5
1.5	Comparing the discrimination tasks in multi-class classification and information theory.	21
1.6	Different variants of regression models	31
1.7	Mean-squared error (a) versus identification accuracy (b) for evaluating a multivariate predictive model.	33
1.8	Identification accuracies in toy example	36
2.1	Training set	51
2.2	Classification rule	54
2.3	Generalization accuracy	55
2.4	Average generalization accuracy	56
3.1	Conditional accuracy	72
3.2	U-functions	73

3.3	(a) The estimated average accuracy for $k = 2, \dots, 400$ given a dataset of 400 faces subsampled from Labeled Faces in the Wild. (b) Estimated average accuracy for $k > 400$ on the same dataset, compared to the ground truth (average k -class test accuracy using all 1672 classes). . .	78
3.4	Estimated average accuracy using subsampled datasets of size k , compared to the ground truth (average k -class test accuracy using all 1672 classes).	80
4.1	Estimation of mutual information in simulated example.	95
5.1	Left: The function $\pi_k(\mu)$ for $k = \{2, 10\}$	98
5.2	Simulation for inferring mutual information in a gaussian random classification model	106
5.3	Estimated mutual information for different subsets of V1 voxels . . .	109
5.4	Dependence of estimated mutual information on k	110

Chapter 1

Introduction

1.1 Finding the correct representation

A fundamental question in the cognitive sciences is how humans and other organisms perceive complex stimuli, such as faces, objects, and sounds. A highly related question in artificial intelligence is how to engineer systems that can learn how to identify objects, faces, and parse the meaning of language. Through the last couple of decades, breakthroughs in both the understanding of cognition, and developments in artificial intelligence, both suggest that *nonlinear representations* are key for making sense of complex stimuli, regardless of whether the perceiver is a biological or algorithmic.

1.1.1 Example: Receptive-field models for vision

Let us begin with the biological case. By looking at the neural pathways involved in mammalian vision, neuroscientists know that vision begins in the retina, where light-sensitive cells (rods and cones) detect incoming photons. The signals from the rods and cones are aggregated by retinal cells, and then transmitted sequentially through a series of structures within the brain. The dominant pathway goes from the retina to the optic chiasm, then to the lateral geniculate nucleus, and finally to the visual cortex (Figure 1.1). The visual cortex, in turn, is divided into subregions V1 through V6. It is an active area of research to study the specialized roles of each subregion

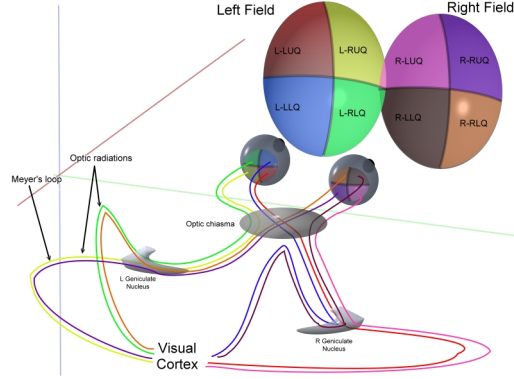


Figure 1.1: Visual pathway in humans. Image credit to Ratznum under CC 2.5 license.

with regards to visual processing.

Functional MRI (fMRI) studies of vision provide one means of testing theories about the workings of the visual cortex. In an exemplary study, Kay et al. 2008 model the response of the BOLD fMRI signal (a proxy measure of neural activity) to greyscale natural images presented to a human subject. The data takes the form of pairs (\vec{z}_i, \vec{y}_i) , where \vec{z}_i is the pixel intensities of the presented image, and \vec{y}_i is a three-dimensional map of BOLD signal, represented as a numerical vector with one real-valued intensity per voxel.

Kay et al. test two different models for the *receptive field* (RF) of V1 voxels. A receptive field model, in this case, specifies a specific set of transformations for explaining how visual information is *represented* in the V1 area of the brain. Under one RF model, the activity of V1 voxels can be explained by *retinotopic* receptive fields, in which the raw image \vec{z}_i is represented by a library of local luminance and contrast maps. Under the second RF model, the activity of V1 voxels is explained by *Gabor filter* receptive fields, consisting of sinusoidal filters which are sensitive to position, frequency, and orientation (Figure 1.2).

Each receptive field model corresponds to a family of *representations*, which is a collection $\vec{g} = (g_1, \dots, g_m)$ of linear or nonlinear transformations of the visual stimulus \vec{z} . Let z_j denote the intensity of the j th pixel in the visual stimulus, and let $\ell_j = (r_j, c_j)$ indicate the row and column coordinates of the j th pixel. Under the retinotopic

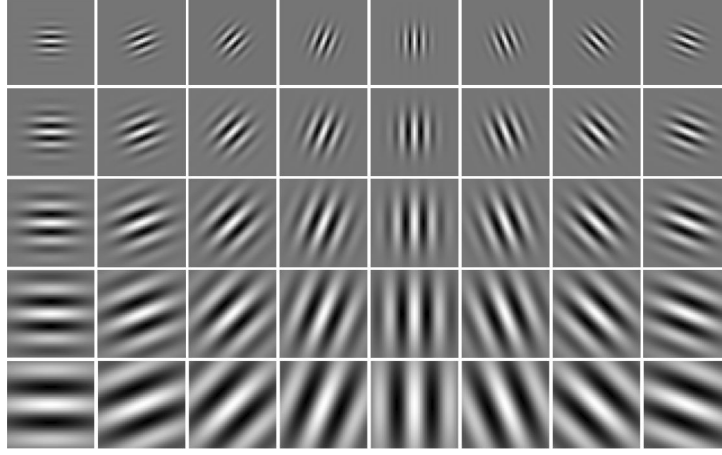


Figure 1.2: Examples of Gabor filters of varying size and orientation. From Haghighat, Zonouz, and Abdel-Mottaleb 2015.

model, the transformations consist of locally-weighted mean-luminance and contrast operations,

$$L(\vec{z}) = \frac{\sum_j w_j z_j}{\sum_j w_j}$$

$$C(\vec{z}) = \sqrt{\frac{\sum_j w_j (z_j - L(\vec{z}))^2}{\sum_j w_j}}$$

where w_j are weights from a symmetric bivariate Gaussian distribution (but whose center μ and spread σ^2 are free parameters),

$$w_j = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2} \|\ell_j - \mu\|^2}.$$

Under the Gabor filter model, the transformations consist of local wavelet transforms of the form

$$g(\vec{z}) = \left\| \sum_j e^{-i\langle \theta, \ell_j \rangle} w_j z_j \right\|^2$$

where $\|\cdot\|^2$ is the squared modulus of a complex number, θ is a free parameter which describes the frequency and orientation of the wavelet, and w_j is defined the same way as in the retinotopic RF model.

The retinotopic RF model is known in the literature to be a good model of receptive fields in early visual areas (such as the retina—hence the nomenclature.) However, Kay et al. are interested in testing whether the Gabor filter model, which is a popular model for neurons in V1, is better supported by the data.

In order to compare the two different RF models, each of the candidate RF models is used to fit an *encoding model*—a forward model for predicting the voxel activations in V1, \vec{y}^{V1} , from the representations defined by the RF model, $\vec{g}(\vec{z})$. Kay et al. consider sparse linear encoding models of the form

$$\vec{y}^{V1} = \mathbf{B}^T \vec{g}(\vec{z}) + \vec{b} + \vec{\epsilon}$$

where \mathbf{B} , a sparse coefficient matrix and \vec{b} , a offset vector, are parameters to be estimated from the data, and $\vec{\epsilon}$ is a noise variable. The quality of each encoding model is assessed using *data-splitting* and the *identification risk* of the model—these methods will be explained in the following background sections. Kay et al. found that the encoding model based on Gabor filter receptive fields clearly outperformed the encoding model based on the retinotopic RF field—supporting the hypothesis that V1 *represents* visual information primarily in the form of Gabor filters.

1.1.2 Example: Face-recognition algorithms

Facial recognition is an important technology with applications in security and in social media, such as automatic tagging of photographs on Facebook. The basic problem is illustrated in Figure 1.3: given a collection of tagged and cropped photographs $\{(\vec{z}_j^{(i)}, y^{(i)})\}$, where $y^{(i)}$ is the label, and $\vec{z}_j^{(i)}$ is a vector containing the numeric features of the photograph (e.g. pixels), assign labels y to untagged photographs \vec{z}_* . Here, the notation $\vec{z}_j^{(i)}$ indicates the j th labelled photograph in the database belonging to the i th individual.

Decades of research into facial recognition has confirmed that careful *feature-engineering* or *representation-learning* is the key to achieving human-level performance on the face recognition task. The feature-engineering approach involves crafting algorithms to locate landmarks in the image (the corners of the eyes, nose, mouth,

















Label	Training			Test
$y^{(1)}=\text{Amelia}$	$\vec{z}_1^{(1)} = $ 	$\vec{z}_2^{(1)} = $ 	$\vec{z}_3^{(1)} = $ 	$\vec{z}_*^{(1)} = $ 
$y^{(2)}=\text{Jean-Pierre}$	$\vec{z}_1^{(2)} = $ 	$\vec{z}_2^{(2)} = $ 	$\vec{z}_3^{(2)} = $ 	$\vec{z}_*^{(2)} = $ 
$y^{(3)}=\text{Liza}$	$\vec{z}_1^{(3)} = $ 	$\vec{z}_2^{(3)} = $ 	$\vec{z}_3^{(3)} = $ 	$\vec{z}_4^{(3)} = $ 
$y^{(4)}=\text{Patricia}$	$\vec{z}_1^{(4)} = $ 	$\vec{z}_2^{(4)} = $ 	$\vec{z}_3^{(4)} = $ 	$\vec{z}_4^{(4)} = $ 

Figure 1.3: Face recognition problem

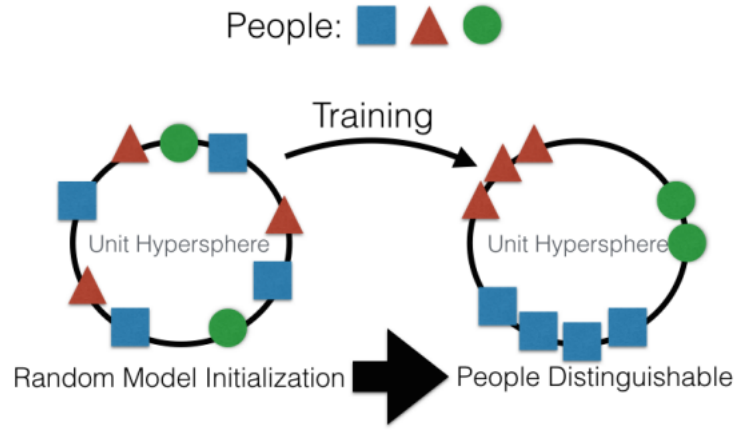


Figure 1.4: Triplet loss function for training face representations. From Amos, Ludwiczuk, and Satyanarayanan 2016

etc.) and to use distances between landmarks as features. The most sophisticated approaches extract features by means of first fitting a three-dimensional model of the face to the photograph.

More recently, fully automated feature-learning, or *representation-learning*, using deep convolutional networks (CNN) has yielded record performance. Google’s FaceNet (Schroff, Kalenichenko, and Philbin 2015), using learned features from a deep CNN, achieved an accuracy of 0.9964 ± 0.0009 on the Labeled Faces in the Wild (LFW) benchmark dataset, outperforming Facebook’s DeepFace (which uses both a deep CNN, and 3D modeling, with an accuracy of 0.9735 ± 0.0025 , Taigman et al. 2014) and a human benchmark (accuracy 0.9753, Kumar et al. 2009).

The method that FaceNet uses to learn a representation $\vec{g}(\vec{z})$ (a collection of

nonlinear mappings of the input image) is highly interesting. The representation \vec{g} is parameterized by a deep CNN architecture: in other words, the basis functions g_i are the end result of composing several layers of nonlinear transformations as specified by the hierarchical architecture of the CNN. However, for our purposes, the modeling and algorithmic details of the CNN are not important, and we refer the interested reader to (LeCun and Ranzato 2013) for a reference on principles of convolutional neural networks. At a higher level of abstraction, we can say that the representation $\vec{g}_\theta(\vec{z})$ lies in a class of nonlinear functions, parameterized by some (possibly large) vector of parameters, θ . The triplet loss function used by FaceNet defines the objective function used to estimate θ ; in other words, the triplet loss function measures the quality of the representation $\vec{g}_\theta(\vec{z})$.

The intuition behind the triplet loss function is that a good representation $\vec{g}(\vec{z})$ should cause faces of the same person to cluster, as illustrated in Figure 1.4. Therefore, the triplet loss function encourages inputs \vec{z}, \vec{z}' that belong to the same class (that is, faces which belong to the same person) to have a representations $\vec{g}(\vec{z}), \vec{g}(\vec{z}')$ that are close to each other in terms of Euclidean distance, while inputs \vec{z}, \vec{z}^* which belong to different classes are encouraged to have representations $\vec{g}(\vec{z}), \vec{g}(\vec{z}^*)$ which are far apart in terms of Euclidean distance. Note also that for the triplet loss, we require the representations \vec{g} to be normalized to have unit norm, so that the maximum distance between two representations is 2.

Recall that the training data consists of images $\{\vec{z}_j^{(i)}\}$ where i indexes the person (or class) and j indexes the repeats from the same class. Define a *triplet* as a triple consisting of an *anchor*, a *positive example* from the same class as the anchor, and a *negative example* from a different class from the anchor,

$$\left(\underbrace{\vec{z}_j^{(i)}}_{\text{anchor}}, \underbrace{\vec{z}_k^{(i)}}_{\text{positive example}}, \underbrace{\vec{z}_\ell^{(m)}}_{\text{negative example}} \right)$$

where $j \neq k$ and $m \neq i$. For instance, in a training set with N classes and M training examples per class, we can form $N(N-1)M^2(M-1)$ triplets. The triplet loss is then

defined as

$$\text{TripletLoss}_\theta = \sum_{j \neq k} \sum_{m \neq i} \sum_{\ell} [\|\vec{g}_\theta(\vec{z}_j^{(i)}) - \vec{g}_\theta(\vec{z}_k^{(i)})\|^2 + \alpha - \|\vec{g}_\theta(\vec{z}_j^{(i)}) - \vec{g}_\theta(\vec{z}_\ell^{(m)})\|^2]_+$$

where α is a tuning parameter (defining the desired separation between inter-cluster distance and between-cluster distance). In the case of FaceNet, stochastic gradient descent with backpropagation is used to update the CNN parameters θ over mini-batches of triplets.

1.1.3 What makes a good representation?

One of the big questions in representation learning is how to define or evaluate the quality of a representation (Bengio, Courville, and Vincent 2013). When, as in face recognition, the end goal of the representation learning is to obtain more accurate predictions or classifications within a machine learning pipeline, an obvious criterion for the quality of the representation is the prediction or classification accuracy that can be attained after using that particular representation as the feature set for a classification or regression model.

However, this result-oriented approach to evaluating representations has two drawbacks. Firstly, it may be difficult to work with a performance metric (such classification or regression accuracy) as a quality metric, since obtaining the performance metric requires training a model and then testing it on data, which can be computationally costly and may not yield a differentiable objective function. Secondly, one of the appealing qualities of a ‘good’ representation is that it should enable good performance in a *variety* of different tasks. Limiting the definition of ‘good’ to performance on a single task seemingly ignores the requirement that a representation should be general across tasks.

Thinking about generative models suggests different avenues for evaluating representations. One such generative model is that the observations \vec{z} (e.g. images of faces) originate from some latent objects \vec{t} (e.g. a person’s head). We can think of the observations \vec{z} as being generated by some mechanism which depends on the

attributes of the latent objects, \vec{t} , as well as some *nuisance parameters* or *degrees of freedom* $\vec{\xi}$ (such as the pose, or lighting of the face) which modulate how the features of \vec{t} are expressed (or perhaps masked) in the observed data \vec{z} . Presumably, for the task at hand, e.g. identifying the person, only the latent objects \vec{t} are important, and not the nuisance parameters. However, it is worth noting that for a different task, such as ‘pose identification’ (rather than face identification), it may be the case that the roles of the nuisance parameters $\vec{\xi}$ and the latent objects \vec{t} are switched—as the saying¹ goes, “one man’s noise is another man’s signal”.

Bengio, Courville, and Vincent 2013 suggest that an ideal representation, rather than discriminating between ‘signal’ and ‘noise’, would *disentangle* the effects of *all factors* while discarding a minimum amount of information. In other words, an ideal representation would map \vec{z} onto some estimate of $(\vec{t}, \vec{\xi})$ which separates the effect of the latent objects from nuisance parameters, and also allows for *reconstruction* of observation from the representation. We will come across similar ideas when we discuss *auto-encoders* in section 1.1.5.

However, in this work, we take a more simplistic approach, where we *enforce* a distinction between one set of factors, \vec{t} , as the ‘signal’, and the other factors, $\vec{\xi}$, as the ‘noise’, and where we consider an ideal representation to be one that keeps the signal while discarding the noise. The ‘signal-only’ approach to representations is sufficient for most current applications, including the two examples of ‘representation evaluation’ that we just presented—the facial recognition problem, and the evaluation of receptive field models in fMRI data.

In the case of facial recognition, the ‘signal’ is the features of the face that persist across different perspectives and illumination, while the ‘noise’ is the effect of pose, illumination, transient features such as hairstyle and makeup, and occlusive accessories such as sunglasses. The effect of extracting the signal while reducing the noise is to shrink inputs that share the same latent variables—faces from the same person—towards each other, as illustrated in Figure 1.4.

Meanwhile, in the case of the functional MRI study, it is the V1 neurons themselves which define what is the ‘signal’ and what is the ‘noise’ in the input. The V1 neurons

¹The quotation is commonly attributed to Edward Ng, 1990.

only respond to certain features in the data, and ignore others. Therefore, the goal of the receptive field model is to extract the information in the data that is relevant to V1 (such as, perhaps, local angular frequencies in the image) and discard other information (e.g. intensities of individual pixels).

1.1.4 Supervised evaluation of representations

In both examples, we not only have inputs \vec{z} but also some form of *side information* that helps us distinguish between signal and noise. In the case of facial recognition, the side information is the labels y which label the photographs. In the case of the functional MRI study, the side information is the V1 intensities \vec{y}^{V1} which allow us to infer which features of the image are salient to V1 neurons.

Thus, the unifying theme of this thesis is how to evaluate (possibly nonlinear) representations $\vec{g}(\vec{z})$ of inputs \vec{z} when we are given pairs (\vec{z}_i, y_i) of input vectors as well as some form of ‘side-information’ y_i , which we will call *the response variable*, that gives us some basis for distinguishing signal from noise. In analogy with the terminology of *supervised learning*, wherein a response y is available to provide feedback to the learning algorithm, we use the terminology *supervised evaluation of representations* to refer to methods for evaluating representations with the aid of a response y . In this work, we consider three particular methods for supervised evaluation of representations.

1. The *mutual information* $I(\vec{g}(\vec{Z}); Y)$ between the representation and the response variable.
2. In the case of discrete response variables Y : the k -class average classification accuracy.
3. In the case of continuous response variables Y : the *identification accuracy*.

The mutual information is a classical measure of dependence that was first developed by Claude Shannon as one of the key concepts in information theory. The k -class average classification accuracy is a concept that has not been (to our knowledge)

previously introduced in the literature, but it is highly related to the *identification accuracy*, which was introduced by the same functional MRI study of natural images (Kay et al. 2008) that we discussed in section 1.1.1. To our knowledge, we are the first to investigate the properties of the identification task from a theoretical perspective.

Comparing these methods, the advantage of both the k -class average classification accuracy and the identification accuracy is that they are relatively easy to compute, even in high-dimensional data, because they are both based on error metrics for supervised learning tasks. Meanwhile, the mutual information is extremely difficult to estimate in high-dimensional data. However, the advantage of the mutual information is that it does not depend on arbitrary tuning parameters, while both the k -class average classification accuracy and identification risk depend on the choice of a tuning parameter k .

However, one of the main theoretical contributions of this work is to show how all of these three methods: mutual information, k -class average classification accuracy, and identification accuracy, are highly related. In particular, we establish methods for lower-bounding the mutual information from either the k -class average classification accuracy or identification accuracy.

1.1.5 Related Work

As we hoped to convey in the introduction, the problem of finding and evaluating representations is a hot topic in multiple disciplines, from neuroscience to machine learning. Consequently, the space of possible approaches to the problem is vast. We limit our study to a few highly interconnected and (in our opinion) interesting approaches to the problem of evaluating representations, in the special case when a *response* variable Y is available and where one wants to take advantage of the side-information provided by this response.

However, many other ideas exist for evaluating representations. One extremely notable family of approaches, which lies totally outside the scope of this thesis, is *unsupervised* methods for evaluating representations— methods which do not require

access to an external response variable Y . Obviously, this is highly interesting, because in many applications one does not have easy access to such a response variable. One family of methods—including restricted Boltzmann machines and gaussian restricted Boltzmann machines—fits a parametric distribution to the inputs \vec{z} (Hinton and Salakhutdinov 2006). The representations are obtained as summary statistics of the latent variables in the model, and the quality of the representation is assessed via the *likelihood* of the parametric model. *Auto-encoders* form another family of methods (Baldi 2012). Representations, or *encoders* \vec{g} are paired with *decoders* \vec{g}^{-1} that infer the original input from the representation. The quality of the representation \vec{g} is based on the reconstruction error obtained by comparing the original input to the inverse of the representation,

$$\|\vec{z} - \vec{g}^{-1}(\vec{g}(\vec{z}))\|^2.$$

In the case that \vec{g} is of smaller dimensionality than \vec{z} , this forces the representation to extract highly explanatory ‘latent factors’ that explain most of the variation in \vec{z} . (If this sounds familiar, it may be because Principal Component Analysis can be interpreted as an auto-encoder model: the principal components minimize the reconstruction error over all linear encoding/decoding rules.) However, one can also consider *over-complete* representations of higher dimensionality than \vec{z} . In order to prevent the identity map (which would trivially have zero reconstruction error) from being the optimal representation, a variety of different approaches can be taken to modify the objective function. One is to require the auto-encoder (the composition of the encoder and decoder) to recover the original input \vec{z} from a *noisy* input $\tilde{z} = \vec{z} + \vec{\epsilon}$. Another approach is to *regularize* the encoder, for instance, requiring sparsity in the output of the encoder.

With regards to *supervised* evaluation of representations, one can find extremely similar ideas in the methodology of *representation-similarity analysis*, which was introduced by Kriegeskorte, Mur, and Bandettini 2008 to the neuroscience community, and which has already grown incredibly popular within the field given the short span of time since its introduction. However, the methodology is based on much more

classical work in statistics and psychometrics on *distance-based inference*. The idea is that if one has multiple *views* of the same object, say, the pixel values \vec{z}_i of an image, a semantic labeling y_i ('house' or 'chair'), as well as a subject's response \vec{x}_i to the image, as measured by fMRI, then all of these different views can be *compared* by means of their *inter-object distance matrices*. That is, if we have distinct objects indexed by $i = 1, \dots, n$, then one can form an $n \times n$ distance matrix for each view: for instance, $D_{\vec{z}}$, the matrix of all pairwise Euclidean distances between pixel vectors; D_y , a binary matrix indicating pairs of identical labels with 0 and non-identical labels with 1; and $D_{\vec{x}}$, a matrix of pairwise Euclidean distances between fMRI images. One can then compare these resulting distance matrices (e.g. in terms of correlation) to determine which *views* are similar to each other, and which are dissimilar. For instance, one may find that distances within 'brain-space', $D_{\vec{x}}$, are much more similar to semantic distances D_y than raw pixel distances $D_{\vec{z}}$.

One could easily adapt the ideas in representational-similarity analysis towards the supervised evaluation of representations. A representation \vec{g} is good if the resulting distance matrix $D_{\vec{g}}$ of pairwise distances between representations is similar to the distance matrix D_y between responses. In fact, one could interpret the *triplet-loss* objective function as enforcing a kind of *representational similarity* between face representations $\vec{g}(\vec{z})$ and labels y . Two faces with the same label have a distance of 0 within D_y , and therefore, they should have a small distance within $D_{\vec{g}}$. Two faces with different labels have distance 1 within D_y ; therefore, they should have at least α distance within $D_{\vec{g}}$.

However, the connection between representational-similarity analysis and supervised evaluation of representations remains unexplored in this work. We leave it to future research.

1.2 Overview

1.2.1 Theme and variations

We have seen that the main *theme* of the thesis is the supervised evaluation of representations. However, a number of *subthemes* arise from similar problems in related disciplines, and additional applications of our methods.

Subtheme: Recognition systems. We have seen that *recognition systems*, such as facial recognition systems, which are tasked with identifying objects from data, depend on finding a good representation of the data. Recognition systems and representations are also highly linked because one way to define ‘what makes a good representation?’ is that a good representation should enable accurate recognition. However, one issue that a formal definition of how to evaluate the quality of a recognition system has been missing in the literature. Our proposal for modelling recognition problems, and for evaluating recognition systems, is through the formalism of *randomized classification*, which defines parameters for multi-class classification problems (think of the problem of classifying a face to K possible people) where the classes have been drawn randomly.

Subtheme: Information geometry. An intuitive notion of quality for representations is that the distance between representations should reflect *meaningful differences* (or ‘signal’) between the underlying objects \vec{t} rather than the effect of the degrees of freedom $\vec{\xi}$ in the representation. However, the proper measure of distance in the representation space is arguably the *statistical distance* rather than geometric (e.g. Euclidean) distance. That is, if we consider the nuisance parameters $\vec{\xi}$ as random variables, then the distance between a representations $\vec{g}(\vec{z})$ and $\vec{g}(\vec{z}')$ should reflect the power with which we can conduct a *statistical hypothesis test* for determining whether the representations originate from the same latent objects,

$$H_0 : \vec{t} = \vec{t}'.$$

This leads us to consider the ideas in *information geometry*, which is the study of

spaces of *distributions* $\{f_\theta\}_{\theta \in \Theta}$ in which distance is measured by some type of statistical distance or divergence, e.g. Kullback-Liebler divergence (Amari and Nagaoka 2007). To fit our problem into the framework of information geometry, we would consider the latent objects \vec{t} as playing the role of the parameter θ , and the induced distribution of $\vec{g}(\vec{z})$ as the distribution f_θ . It is important to note however, that this emphasis on parameter spaces is complemented by the concept of *duality* between the space of distributions and the space of observations. The concept can be formalized in exponential families, where a sample from f_θ can be represented in the distributional space as the MLE estimate $f_{\hat{\theta}}$, and where the process of estimation is seen to correspond to projection operators.

Within this framework, one can consider the *metric entropy* of a space Θ , which is a measure of the *volume* of the space according to statistical distance. A ball $B_{\theta,r}$ centered at parameter θ and with radius r is defined as the set of parameters θ' such that the statistical distance between f_θ and $f_{\theta'}$ is less than r :

$$d(f_\theta, f_{\theta'}) < r.$$

The δ -*metric entropy* of the space Θ is defined as the minimum number of balls of radius δ needed to cover Θ (Adler and Taylor 2009). While we will not employ the formal tools of information geometry in this work, we take inspiration from some of their intuitions. Instead, we use *information theory*, a closely related field, to provide much of the formalism for our theory.

Subtheme: Information theory. Extremely similar notions of *volume* appear in information theory, which is the study of how to design systems for transmitting messages between a sender and a receiver over a possibly noisy channel. We will review more of the background of information theory later in this chapter. For now, we note that the analogy between information theory and information geometry is that now the *encoded message* plays the role of the parameter θ , and we are concerned with the space of the distributions f_θ of *received messages*. The *capacity* of a channel is a measure of the *volume* of the space. The channel capacity is defined in terms of *mutual information*, which plays the analogue of the logarithm of the *metric entropy*.

This can be seen clearly if we consider the Euclidean case for metric entropy: the log-metric entropy is closely related to the difference of the log-volume of the space and the log-volume of the ball $B_{\theta,\delta}$. Meanwhile, mutual information $I(T; R)$ is defined as the difference between the entropy of R (the recieved message) and the conditional entropy of R given T (the transmitted message):

$$I(T; R) \stackrel{def}{=} \underbrace{H(R)}_{\text{entropy}} - \underbrace{H(R|T)}_{\text{conditional entropy}} .$$

Here the entropy $H(R)$ is analagous to the log-volume of the entire space, while the conditional entropy $H(R|T)$ measures the log-volume of the ball which is centered at the parameter T . While mutual information is not defined explicitly in terms of packing or covering numbers, as we see in the Euclidean example for metric entropy, both packing and covering numbers are approximately equivalent to volume ratios. Another difference between the mutual information and the metric entropy is that the mutual information is concerned with volume in the *observation* space (the space of recieved messages R) rather than the *parameter space*. However, due to the concept of duality, we can see that one arrives at similar definitions of volume whether we choose to use the parameter space, or its dual, the observation space.

Subtheme: Estimation of mutual information. Besides serving (in our case) as a measure of statistical volume, the mutual information enjoys numerous other desirable properties such as symmetry, invariance under bijections, and independent additivity, as we will review later in the chapter. Due to these properties, the mutual information is an ideal measure of dependence for many problems; therefore, in a variety of applications, including many in neuroscience, it is desirable to estimate the mutual information of some empirically observed joint distribution. However, this is a highly nontrivial functional estimation problem in high dimensions. By connecting mutual information to more easily estimated quantities such as average classification accuracy, our work provides novel estimators of mutual information, which we show to have better scaling properties in many high-dimensional problems than previous approaches for estimating mutual information.

Subtheme: Connections between information theory and supervised learning. Information theory, statistics, and machine learning have many interconnections, as testified by the many applications of information-theoretic inequalities in statistical and machine learning research. By studying both information-theoretic and classification-based methods for evaluating representations, we uncover additional links between information theory and classification. Fano’s inequality, which bounds the mutual information in terms of Bayes accuracy of classification (BA),

$$I(X; Y) \geq \log(k) - H(\text{BA}) - (1 - \text{BA}) \log(k - 1),$$

is one of the earliest results bridging the two worlds of information theory and supervised learning. However, its application is limited to *discrete* and *uniformly* distributed X . Our work in Chapter 4 provides an extension of Fano’s inequality to the case of continuous (X, Y) , through means of the Bayes accuracy of *identification*.

Subtheme: Geometric inference from random samples. Regardless of which definition of ‘volume’ one employs, a natural question is how to estimate this ‘volume’ from empirical data. That is, we wish to infer a geometric characteristic of the space—the volume—from a random sample of observations drawn from the space. Meanwhile, a complementary question that was already extensively studied in information theory is the question of how to *construct* a collection of points in the random space that *optimizes* another geometric characteristic—the overlap between points. It was established by Shannon that the *randomization* method provides such a construction—a randomly drawn collection of points has asymptotically optimal properties in terms of maximum overlap (as measured by decoding error.) In information theory, these random constructions pioneered by Shannon continued to be studied in the form of *random code models*.

Returning to the problem of inferring volume from samples, two questions arise—one being how to construct an estimator, and secondly, what is the variance of the estimator. We define volume in terms of mutual information and develop estimators based on *random classification tasks*, which specify the sampling mechanism. Furthermore, we obtain preliminary results on the variability of such estimators in section

2.4.4. We compare our results to existing results in information theory regarding random code models.

Subtheme: Generalizability of experiments. Two of our motivations for studying random classification tasks is (i) to evaluate representations, and (ii) as a model for recognition problems. Yet a third application is for understanding the generalizability of experiments that can be modelled as random classification tasks. For example, many task-fMRI experiments can be modelled as random classification tasks, because the stimuli sets used in the experiment are composed of arbitrary (‘random’) exemplars, and therefore the stimuli set used by one lab may differ from the stimuli set used by another lab, even when they are presumably studying the same task. Intuitively, using larger and more diverse stimuli sets should lead to better generalizability of results to the entire population of stimuli. Our work on random classification—in particular, our variance bounds on the classification accuracy in randomized classification tasks—provides a theoretical basis for understanding how the results of a random classification task allow inference of population parameters, such as the mutual information between the stimulus and the response.

1.2.2 Organization

The rest of the thesis is organized as follows. The remaining sections in this chapter deal with background material on supervised learning and information theory, as well as the application of both to neuroscience, which forms a major motivation for the current work. Chapter 2 introduces the concept of randomized classification, and also establishes some variability bounds which will be used later in the development of inference procedures. Chapter 3 studies the dependence of classification accuracy on the label set size in randomized classification, and a practical method for predicting the accuracy-versus-label set size curve from real data. Chapter 4 and 5 deal with the applications of randomized classification to the estimation of mutual information in continuous data: chapter 4 derives a lower confidence bound for mutual information under very weak assumptions, while chapter 5 works within an asymptotic high-dimensional framework which leads to a more powerful but less robust estimator

estimate of mutual information. We conclude the thesis with a discussion in Chapter 6.

1.3 Information and Discrimination

We now begin our review of background material in supervised learning and information theory. Therefore, a reader who is familiar to both fields could skip most of the following—with the exception of the explanation of *identification accuracy* in section 1.3.3, which is a relatively novel concept in the statistical literature. Also, we hope that even the experienced reader will find some food for thought in our comparison of information theory and supervised learning, and our humble speculations about how current developments may increase the degree of interaction between the two areas.

1.3.1 Introduction

In studying the problem of evaluating representations, we make use of two closely related frameworks: firstly, the multi-class classification framework from the statistics and machine learning literature, and secondly, the concepts of information theory. From a broader perspective, this is hardly unusual, since concepts such as entropy, divergence, and mutual information are commonly applied in theoretical statistics and machine learning. Furthermore, since information theory, theoretical statistics, and machine learning are based on the same foundation of measure-theoretic probability theory, one could go as far as to say that all three disciplines are subfields of applied probability. However, while the three sub-fields may appear very similar from a mathematical perspective, some differences arise if we examine the kinds of intuitions and assumptions that are characteristic of the literature in each area.

A common problem to all three subfields is the inference of some unobserved quantity on the basis of observed quantities. In classical statistics, the problem is to infer an unknown parameter; in supervised learning, the problem is to predict an unobserved label or response Y ; in information theory, the problem is to decode a noisy message. Next, the metric for quantifying achievable performance differs between

the three disciplines. In classical statistics, one is concerned with the variance of the estimated parameter, or equivalently, the Fisher information. In machine learning, one seeks to minimize (in expectation) a *loss* function which measures the discrepancy between the prediction and the truth. In information theory, one can measure the quality of the noisy channel (and therefore, the resulting achievable accuracy) through the *mutual information* $I(X;Y)$ between the sender's encoded message X and the receiver's received message Y . If we specialize within machine learning to the study of classification, then we are concerned with accurate *discrimination* of the input X according to labels Y . Similarly, if we specialize to the problem of hypothesis testing within statistics, the problem is again to *discriminate* between two (or more) different hypotheses regarding the data-generating mechanism.

The concepts of *information* and *discrimination* are quite distinct from an intuitive standpoint; however, they are linked at a fundamental level. A statistical hypothesis test is *informative* because it provides evidence that the data behaves according to a certain hypothesis rather than another. In information theory, even if the receiver cannot conclusively determine the sender's message from the observed signal, the signal still contains *information* if it contains some evidence that favors one set of possible messages over another. The formalism of measure-theoretic probability theory provides yet another example of the conceptual link between information and discrimination².

Either natural or artificially intelligence recognition systems must rely on input data that is *informative* of the optimal response if they are to achieve reasonable discriminative accuracy. In natural environments, mammals rely on a combination of visual, auditory, and tactile cues to recognize potential threats in the environment. Mammalian brains integrate all of this sensory information in order to make more

²Supposing Ω is a probability space defined with respect to a σ -algebra \mathcal{F} , we can represent our state of knowledge with a filtration (or sub- σ -algebra) $\mathcal{F}' \subseteq \mathcal{F}$. Complete knowledge (zero uncertainty) is represented by the full σ -algebra: that is, $\mathcal{F}' = \mathcal{F}$. Partial knowledge is represented by a coarser filtration, $\mathcal{F} \subset \mathcal{F}'$. The filtration, of course, indicates that our knowledge is sufficient to *discriminate* the outcome space Ω into a number of finitely or infinitely many categories. The more information we have, (or, the closer we come to complete knowledge of the outcome), the more finely we can discriminate the realized outcomes given by $\omega \in \Omega$.

rapid and reliable decisions. Generally, increased diversity and quality of the available sources of information will lead to more accurate recognition (say, of possible environmental threats.)

This link between the information content of the input and the achievable discrimination accuracy was first quantified by Claude Shannon via the concept of *mutual information*. The mutual information $I(X; Y)$ quantifies the information content that an input X holds about a target of interest, Y . For instance, in the case of facial identification, the discrimination target Y is a label corresponding to the identity of the person, and X is an image of the individual's face. An image corrupted by noise holds less information, and correspondingly leads to lower classification accuracies.

The discrimination problem that Shannon studied—the *noisy-channel decoding problem*, is extremely similar to the multi-class classification problem, but also features some important differences. A side-by side comparison between the schematics of multi-class classification and the noisy channel problem is displayed in Figure 1.5. We will elaborate much further on the comparison illustrated in the figure, but for now, one can note that both the multi-class classification problem and the noisy-channel decoding problem involves the inference of a latent variable Y from an observation X , where X is linked to Y through a conditional distribution F_Y .

We will now briefly review the relevant background for supervised learning and information theory, to give the context for each side of figure 1.5. Afterwards, we will compare and contrast the supervised learning and information theory, and note what kind of cross-talk exists between the two related fields, and what new developments could still arise by way of a dialogue between supervised learning and information theory. One such new development is the *randomized classification* model, since it is a very close analogue of the *random code* model studied in information theory.

1.3.2 Supervised learning

Supervised learning refers to the training of predictive models; both *classification* and *regression* are special cases of supervised learning. But let us begin with the most general recipe for a prediction task, which involves:

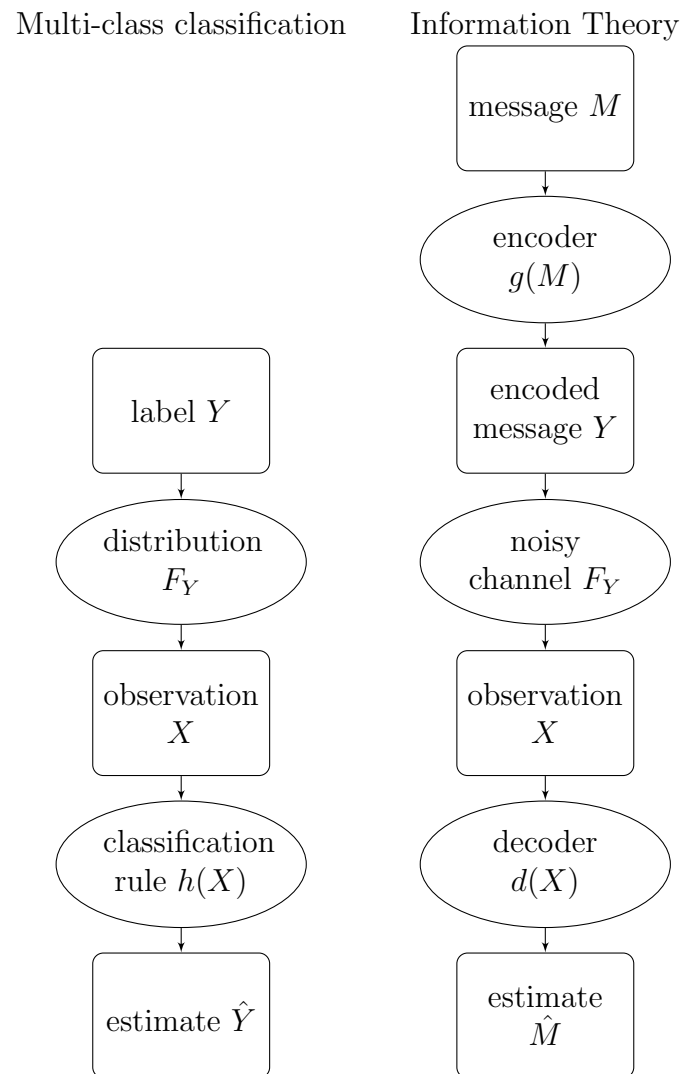


Figure 1.5: Comparing the discrimination tasks in multi-class classification and information theory.

- A predictor space \mathcal{X} defining the possible values the predictor X can take; though typically, $\mathcal{X} = \mathbb{R}^p$.
- A response space \mathcal{Y} defining the possible values the response Y can take;
- An *unknown* population joint distribution G for the pair (\vec{X}, Y) ;
- A *cost* function defining the penalty for incorrect predictions, $C : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$. If Y is the response, and $\hat{Y} = h(\vec{X})$ is the prediction, then the loss for making the prediction \hat{Y} when the truth is Y is given by $C(Y; \hat{Y})$.

The prediction task can be viewed as a game between two players: nature and the statistician. Nature draws an observation (\vec{X}, Y) from the joint distribution G . The statistician only gets to see \vec{X} , and has to make a prediction \hat{Y} for the response Y . Later on, the true value Y is revealed, and the statistician pays the cost $C(Y; \hat{Y})$ depending on how well their prediction matched the actual value of Y .

The various types of prediction tasks include classification, regression, and multivariate variants: such as multi-label classification and multiple-response regression. These special cases are just specializations of the general prediction task to a particular type of response space.

- In *classification*, the response space is finite and discrete. In *binary classification*, the response space \mathcal{Y} consists of two elements, say, $\mathcal{Y} = \{0, 1\}$. Multi-class classification usually refers to the case \mathcal{Y} has more than two elements. The most common cost function for classification is zero-one loss,

$$C(y; \hat{y}) = I(y \neq \hat{y}).$$

- In *regression*, the response space is \mathbb{R} . The most common cost function is squared loss:

$$C(y; \hat{y}) = (y - \hat{y})^2.$$

- In *multi-label classification*, the response space is a product of several finite sets, say $\mathcal{Y} = \mathcal{Y}_1 \times \mathcal{Y}_2 \times \cdots \times \mathcal{Y}_\ell$. That is to say, that the response \vec{Y} consists of a

categorical vector, $\vec{Y} = (Y_1, \dots, Y_\ell)$. More complex types of cost functions can be considered, such as *Jaccard distance*,

$$C(\vec{y}; \hat{\vec{y}}) = \frac{\sum_{i=1}^{\ell} y_i \wedge \hat{y}_i}{\sum_{i=1}^{\ell} y_i \vee \hat{y}_i}.$$

- In *multiple-response regression*, the response space is \mathbb{R}^p . A natural cost function is squared Euclidean distance,

$$C(\vec{y}; \hat{\vec{y}}) = \|\vec{y} - \hat{\vec{y}}\|^2.$$

A *prediction rule* is a function $h : \mathcal{X} \rightarrow \mathcal{Y}$ for predicting Y as a function of \vec{X} . Prediction rules can be found through a variety of means. In some domains, experts manually construct the prediction rules using their domain knowledge. However, the field of *supervised learning* aims to algorithmically construct, or ‘learn’ a good prediction rule from data. In supervised learning, we assume that we have access to a *training set* consisting of n_1 observations $\{(\vec{X}_i, Y_i)\}_{i=1}^{n_1}$, plus a *test set* consisting of n_2 observations $\{(\vec{X}_i, Y_i)\}_{i=n_1+1}^{n_1+n_2}$; usually, we assume that the pairs in both the training and test set have been sampled i.i.d. from the distribution G . As we will elaborate further, the training set is used to construct h , while the test set is used to evaluate the performance of h .

A *learning algorithm* Λ is a procedure for constructing the prediction rule h given training data $\{(\vec{X}_i, Y_i)\}_{i=1}^{n_1}$ as an input. Formally, we write

$$h = \Lambda(\{(\vec{X}_i, Y_i)\}_{i=1}^{n_1}),$$

indicating that h is the output of the function Λ evaluated on the input $\{(\vec{X}_i, Y_i)\}_{i=1}^{n_1}$. But recall that $h : \mathcal{X} \rightarrow \mathcal{Y}$, the classification rule, is also a function! How learning algorithms are implemented in practice can vary considerably; we illustrate just a few of the most common types of learning algorithms:

- *Parametric generative models*. These types of learning algorithms Λ first fit a statistical model to the observed data, then use that model to predict on new

observations. Define a parametric family F_θ of joint distributions (X, Y) . For instance, in linear regression, a commonly studied family is the multivariate normal linear model, where

$$(\vec{X}, Y) \sim N((1, 0, \dots, 0, \beta_0), \begin{pmatrix} \Sigma_X & \Sigma_X \beta \\ \beta^T \Sigma_X & \beta^T \Sigma_X \beta + \Sigma_\epsilon \end{pmatrix}),$$

or equivalently,

$$\vec{X} \sim N((1, 0, \dots, 0), \Sigma_X)$$

$$Y|\vec{X} \sim N(\vec{X}^T \beta, \Sigma_\epsilon).$$

The learning algorithm Λ proceeds by first fitting the parametric model to estimate the parameter $\hat{\theta}$. A variety of methods may be chosen to estimate θ , such as maximum likelihood, penalized maximum likelihood, or Bayesian estimation. Given the fitted statistical model, we can obtain the conditional distribution of Y given \vec{X} . The prediction rule $h(\vec{x})$ is then constructed using this conditional distribution; for instance, taking $h(\vec{x})$ to be the conditional mean of Y given $\vec{X} = \vec{x}$.

- *Discriminative models.* These types of learning algorithms directly attempt to find a good prediction rule, using empirical performance on the training data as a criterion. One typically limits the search over possible prediction rules to a function class \mathcal{H} . We wish to search for an element $h \in \mathcal{H}$ which minimizes the empirical risk on the training set,

$$h = \operatorname{argmin}_{h \in \mathcal{H}} \frac{1}{n_1} \sum_{i=1}^{n_1} \tilde{C}(Y_i; h(\vec{X}_i))$$

Here, \tilde{C} could be taken to be equal to the original cost function C , or could be taken to be a different function, such as a smoothed approximation of C . The advantage of using a smoothed approximation \tilde{C} is that the empirical risk can be made differentiable (whereas the original cost C might be nondifferentiable) and hence the optimization made much more tractable from a numerical standpoint.

This is often the case in binary classification, where C is zero-one loss, but \tilde{C} is the logistic loss

$$\tilde{C}(y; p) = y \log p + (1 - y) \log(1 - p).$$

Further complicating the picture is the fact that often the learning algorithm requires specification of various *hyperparameters*. For instance, lasso regression (Hastie, Tibshirani, and Friedman 2009) is a penalized generative model which finds β minimizing the objective function

$$\beta = \operatorname{argmin}_{\beta} \frac{1}{2} \sum_{i=1}^{n_1} (y_i - \vec{x}_i^T \beta)^2 + \lambda \|\beta\|_1.$$

and then constructs the prediction rule

$$h(\vec{x}) = \vec{x}^T \beta.$$

Here, the L_1 -penalty constant λ needs to be specified by the user. In practice, one can either use prior knowledge or theoretically-justified rules to select λ ; or, more commonly, one uses various procedures to automatically tune λ based on the training data. The most common procedure for automatically selecting λ is cross-validation, with either the “min” or “one standard deviation” rule. We do not go into details here, and refer the interested reader to Hastie, Tibshirani, and Friedman 2009, section 7.10.

Performance evaluation

In practice, we would often like to know how well the prediction rule h will perform on new data. This can be done rigorously if we can assume that the new data pairs (X, Y) will be drawn i.i.d. from some population distribution G , and that the observations in the test set are also drawn i.i.d. from G . The criterion we use to judge the performance of the prediction rule h is the *prediction risk* (also called

generalization error)

$$\text{Risk}(h) = \mathbb{E}_G[C(Y; h(X))].$$

Under the assumption that the test set is drawn i.i.d. from G , then it follows that the test risk (aka *test error*) is an unbiased estimator of the risk.

$$\text{TestRisk}(h) = \frac{1}{n_2} \sum_{i=n_1+1}^{n_1+n_2} C(y_i; h(x_i)).$$

$$\mathbf{E}[\text{TestRisk}(h)] = \text{Risk}(h).$$

Under mild assumptions, one can use the Student-t quantiles to construct a confidence interval for the risk,

$$\text{TestRisk}(h) \pm t_{1-\alpha/2, df=n_2-1} \hat{\text{sd}}(\{C(y_i; h(x_i))\}_{i=n_1+1}^{n_1+n_2})$$

where $t_{1-\alpha/2, df=n_2-1}$ is the $1 - \frac{\alpha}{2}$ quantile of the t-distribution with $n_2 - 1$ degrees of freedom, and $\hat{\text{sd}}$ is the sample standard deviation.

A common pitfall is to attempt to use the *training data*, rather than independent test data, to estimate the risk. The empirical risk on the training data tends to be an underestimate of the true population risk, due to the phenomenon of *overfitting*. That is, the prediction rule h may be capturing the effect of noise in the training data as well as signal.

It is usually the job of the data analyst to make sure that the data has been partitioned into independent training and test data sets before carrying out any analysis. It is an important decision as to how much data to allocate to each of the training and test sets. A larger training set generally results in better prediction rules, but a larger test set allows for more precise estimates of prediction risk.

In any case, once it has been decided to allocate n_1 observations to the training set, and n_2 observations to the test set, one carries out *data-splitting* in order to randomly assign the observations to the training and test sets. The randomization ensures that the i.i.d. sampling assumption is met for both the training and test

set. Concretely speaking, given observations $(\vec{x}_i, y_i)_{i=1}^n$, one draws a random permutation $\sigma : n \rightarrow n$, then takes $\{(\vec{x}_{\sigma_i}, y_{\sigma_i})_{i=1}^{n_1}\}$ as the training set, and the remaining observations $\{(\vec{x}_{\sigma_i}, y_{\sigma_i})_{i=n_1+1}^n\}$ as the test set.

Often it is the case that the number of observations n is so small that one cannot afford to create a large test set. To avoid the tradeoff between having insufficient training data and insufficient test data, one can use the k -fold *cross-validation* procedure. In cross-validation, one uses the entire data set to construct the prediction rule h . Now, in order to estimate the prediction risk, one splits the data into k (approximately) equally-sized partitions. Then, for fold $i = 1, \dots, k$, we take the i th partition as the test set, and merge the remaining $k - 1$ partitions into the training set. The training set is used to construct a new prediction rule, $h^{(i)}$. Then, the test set is used to estimate the risk of $h^{(i)}$, yielding the empirical risk $\text{TestRisk}^{(i)}$. After this has been done for all k folds, we have the cross-validation risk estimates $\text{TestRisk}^{(1)}, \dots, \text{TestRisk}^{(k)}$. The risk of h itself is estimated as

$$\text{CVRisk} = \frac{1}{k} \sum_{i=1}^k \text{TestRisk}^{(i)}.$$

The intuition behind cross-validation is that each cross-validated risk estimate $\text{TestRisk}^{(i)}$ should be an overestimate of the population risk of h , because $h^{(i)}$, being constructed from fewer training data, tends to have a larger population risk than h . Therefore, CVRisk should be an overestimate of the risk of h .

Classification

In classification, the response space \mathcal{Y} is discrete. The prediction rule is called a *classification rule*, and the learning algorithm is called a *classifier*. The elements $y \in \mathcal{Y}$ of the response space are called *labels*. Let $k = |\mathcal{Y}|$ be the number of labels. When a feature vector \vec{x} has the true label i , we can also say that \vec{x} belongs to the i th class.

The most common cost function considered in classification problems is zero-one

loss,

$$C(y; \hat{y}) = I(y \neq \hat{y}).$$

We assume the zero-one loss for the rest of the discussion. This implies that the risk of a classification rule is the probability of misclassification,

$$\text{Risk}(h) = \mathbf{E}[C(Y; h(X))] = \Pr[Y \neq h(X)].$$

A theoretically important (but non-implementable) classification rule is the *Bayes rule*, which achieves optimal prediction risk. However, since the Bayes rule requires knowledge of the population joint distribution, it cannot be constructed in practice. Supposing that (\vec{X}, Y) are drawn from a joint distribution G , then define F_y as the conditional distribution of \vec{X} given $Y = y$. Supposing that F_y has a density f_y , and that the labels Y have a uniform distribution, then the Bayes rule assigns feature vectors \vec{x} to the label with the highest density.

$$h_{\text{Bayes}}(\vec{x}) = \operatorname{argmax}_{y \in \mathcal{Y}} f_y(\vec{x}).$$

Since the response space is discrete, the classification rule h partitions the input space \mathcal{X} into k partitions. The boundaries between adjacent partitions are called *decision boundaries*. A large number of popular classifiers produce *linear decision boundaries*: that is, each decision boundary lies on a hyperplane.

A large number of classifiers create classification rules that are based on *margin functions* (or *discriminant functions*.) A margin function is produced for each label in \mathcal{Y} . The margin function for label y , $m_y : \mathcal{X} \rightarrow \mathbb{R}$ quantifies how likely a feature vector \vec{x} has label y . We say that $m_y(\vec{x})$ is the margin (or *discriminant score*) of \vec{x} for the y th label. The classification rule h , therefore, assigns points to the label having the highest margin for \vec{x} ,

$$h(\vec{x}) = \operatorname{argmax}_{y \in \mathcal{Y}} m_y(\vec{x}).$$

Classifiers with *linear discriminant functions*; that is, which produce margin functions of the form

$$m_y(\vec{x}) = w^T \vec{x}$$

result in *linear decision boundaries*. These include:

- *Linear support vector machines* (Hastie, Tibshirani, and Friedman 2009).
- *Multinomial logistic regression*.
- *Fisher’s linear discriminant analysis*.

Another large class of classifiers—*generative* classifiers—are based on estimating the conditional distribution of \vec{x} within each class. These classifiers use the discriminant function

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

where \hat{f}_y is the estimated density of the distribution F_y . The estimated densities \hat{f}_y also comprise a *generative model* in the sense that they allow the possibility of simulating new data from the class—hence the nomenclature. Different distributional assumptions lead to different classifiers within the generative category. Some examples are:

- *Naive Bayes*. One assumes that F_y is a product distribution on the components of \vec{x} .
- *Fisher’s linear discriminant analysis*. One assumes that $\{F_y\}_{y \in \mathcal{Y}}$ are multivariate normal with common covariance.
- *Quadratic discriminant analysis*. One assumes that $\{F_y\}_{y \in \mathcal{Y}}$ are multivariate normal.

Some other commonly used classifiers include:

- *k-Nearest neighbors*. Uses margin functions $m_y(\vec{x})$ which count how many of the k nearest neighbors of \vec{x} in the training set have the label y .
- *Decision trees*. Recursively partitions the input space \mathcal{X} into smaller and smaller regions, then assigns points \vec{x} to the majority class within the region.

- *Multilayer neural networks.* Learns nonlinear representations of the input space, $g_j(\vec{x})$, then constructs margin functions which are linear combinations of the representations g_j .

Under zero-one loss, it is easy to conduct inference for the prediction risk of h . Under the i.i.d. sampling assumption, the loss of a test observation $L(y_i; h(x_i))$ has a Bernoulli distribution with probability equal to the population risk. Therefore, we have

$$n_2 \text{TestRisk}(h) \sim \text{Bernoulli}(n_2, \text{Risk}(h)).$$

Regression

Regression models predict a continuous response Y as a function of a vector-valued predictor \vec{X} . The model assumed is written as

$$Y = f(\vec{X}) + \epsilon$$

where ϵ is additive noise with zero mean, $\mathbf{E}[\epsilon] = 0$. The function f is called the *regression function* since

$$f(\vec{x}) = \mathbf{E}[Y | \vec{X} = \vec{x}].$$

The goal in regression is to recover the unknown function f .

In *linear regression*, we assume f is linear. However, if we do not assume a particular form for f , we can use *nonparametric regression*.

When \vec{X} is high dimensional, classical regression techniques perform poorly. Yet, if the true function f only depends on a small number of components in \vec{X} , we can still do well if we use *sparse* regression methods. Table 1.6 shows some examples of different regression procedures proposed in the literature which are suited for different scenarios.

1.3.3 Identification accuracy

The identification accuracy originated as a method for evaluating the quality of encoding models in neuroscience (Kay et al. 2008, **Mitchell2008**). However, it can

	<i>Classical</i>	<i>Sparse</i>
<i>Linear</i>	Ordinary Least-Squares (Legendre 1805)	Elastic net (Zou 2008)
<i>Nonpar.</i>	LOWESS (Cleveland 1979)	Random forests (Breiman 2001)

Figure 1.6: Different variants of regression models

generally be applied to evaluate any regression model with a multivariate response \vec{Y} . Furthermore, we argue that it can be an ideal method for evaluating the quality of multivariate representations $\vec{g}(\vec{X})$ given a continuous multivariate response \vec{Y} .

Suppose that we have pairs of vector-valued observations $(\vec{X}_i, \vec{Y}_i)_{i=1}^T$, where the features \vec{X} are p -dimensional, and the response vectors \vec{Y} are q -dimensional. For instance, in the Kay et al. study of natural images, the features \vec{X} are a basis of 10,921 Gabor filters coefficients for the presented images, and the response vectors \vec{Y} are a 3000-dimensional vector of activation coefficients from the visual cortex.

Step 1. Fitting the forward model

We would like to fit a linear³ multivariate regression model of the form

$$\mathbf{E}[\vec{Y} | \vec{X} = \vec{x}] = \vec{x}^T B$$

where B is a $p \times q$ coefficient matrix. Data-splitting is used to partition the data into a training and test set, so that the training set can be used to obtain an estimate of the coefficient matrix, \hat{B} , while the test set can be used to evaluate the quality of the regression model.

To be specific, 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 \mathbf{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 \mathbf{Y}^{tr} . Similarly, define \mathbf{X}^{te} as the $N \times p$ matrix of test stimuli and \mathbf{Y}^{te} as the $N \times q$ matrix

³The discussion applies equally well to nonlinear regression models, but for expositional purposes we focus on the special case of linear models.

of corresponding test responses. Without loss of generality, let us suppose that the indices $i = 1, \dots, M$ correspond to the test set, so that the test observations are $(\vec{x}_i, \vec{y}_i)_{i=1}^M$.

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

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

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 $\hat{\mathbf{B}} = (\hat{\beta}_1, \dots, \hat{\beta}_q)$, we estimate the noise covariance Σ via a shrunk covariance estimate (Ledoit and Wolf 2004, Daniels and Kass 2001) from the residuals,

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

where

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

Step 2. Evaluating the forward model

A usual means of evaluating the linear model specified by the estimate \hat{B} is to evaluate the mean-squared error on the test set,

$$\text{TMSE} = \frac{1}{M} \|\mathbf{Y}^{te} - \mathbf{X}^{te} B\|_F^2,$$

where a lower TMSE indicates a better model. Intuitively, the mean-squared error is the average squared distance between an observation \vec{Y} and its model prediction, \hat{Y} as illustrated in Figure 1.7(a).

However, an alternative criterion for evaluating the multivariate linear model was proposed by Kay et al. 2008. In an *identification task*, as illustrated in Figure 1.7(b),

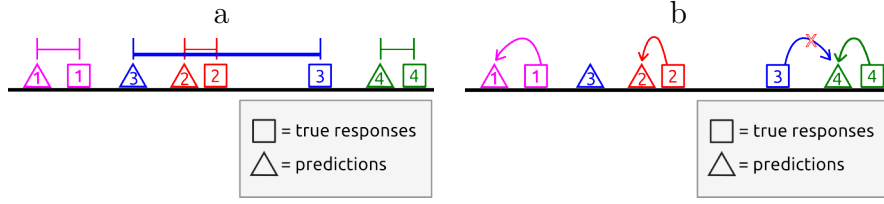


Figure 1.7: Mean-squared error (a) versus identification accuracy (b) for evaluating a multivariate predictive model.

the model is first used to make predictions

$$\hat{y}_i = \hat{B}^T \vec{x}_i$$

for all features $\{\vec{x}_i\}_{i=1}^M$ in the test set. Next, each of the observations \vec{y}_i in the test set is *assigned* to the closest prediction $\{\hat{y}_j\}_{j=1}^M$ within the test set. Here, ‘closest’ is defined in terms of the the empirical Mahalanobis distance.

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

Finally, the M -point test identification accuracy is defined as the fraction of observations which are assigned to the correct prediction,

$$\text{TIA}_M = \frac{1}{M} \sum_{i=1}^M I(d_{\hat{\Sigma}}(\vec{y}_i, \hat{y}_i) \leq \min_{j \neq i} d_{\hat{\Sigma}}(\vec{y}_i, \hat{y}_j)).$$

Advantages of identification accuracy over MSE

A major reason why identification accuracy was adopted for fMRI studies is because it provides an intuitive demonstration of how much information about the stimulus \vec{X} is contained in the response \vec{Y} , in a way that mean-squared error cannot. We formalize this idea by explicitly showing how the empirical identification accuracy can be used to obtain a lower bound of mutual information (Chapter 4). However, some of the intuition can be demonstrated via the following toy example.

Suppose that the response \vec{Y} and predictor \vec{X} are both three-dimensional. The predictor \vec{X} is generated from a standard multivariate normal distribution, and then

\vec{Y} is generated according to the linear model

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

where ϵ is multivariate normal

$$\epsilon \sim N(0, \sigma^2 I).$$

Now consider two different scenarios. In scenario 1, we have the model

$$\vec{Y}^{(1)} = B^{(1)T} \vec{X} + \epsilon$$

with

$$B^{(1)} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

That is, the responses \vec{Y} are only related to the *first* component of \vec{X} . There is no information in \vec{Y} about the other two components of \vec{X} , since \vec{Y} is independent of (X_2, X_3) .

In scenario 2, we have the model

$$\vec{Y}^{(2)} = B^{(2)T} \vec{X} + \epsilon$$

with

$$B^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Unlike in scenario 1, now each component Y_i contains information about a different component X_i of \vec{X} . Therefore, \vec{Y} contains information about all components of \vec{X} ,

Assume that a large amount of training data is available, so that there is practically no estimation error for \hat{B} . Suppose the test set consists of eight points, $(\vec{x}_i, \vec{y}_i)_{i=1}^8$, where

$$\begin{aligned}
\vec{x}_1 &= (-1, -1, -1) & \vec{x}_5 &= (+1, -1, -1) \\
\vec{x}_2 &= (-1, -1, +1) & \vec{x}_6 &= (+1, -1, +1) \\
\vec{x}_3 &= (-1, +1, -1) & \vec{x}_7 &= (+1, +1, -1) \\
\vec{x}_4 &= (-1, +1, +1) & \vec{x}_8 &= (+1, +1, +1)
\end{aligned}$$

Let us compare the two scenarios in terms of what happens when we evaluate the model using mean-squared error. In both cases, the mean-squared error will be approximately $3\sigma^2$. Therefore, MSE does not distinguish between a low-information situation (Scenario 1) and a high-information situation (Scenario 2). Furthermore, since the expected squared norm of Y is the same in both cases, the multivariate R^2 similarly fails to distinguish the two scenarios.

Now consider the identification accuracy. In scenario 1, the fact that \vec{Y} only contains information about X_1 means that it can only separate the two sets $\{\vec{x}_1, \dots, \vec{x}_4\}$ and $\{\vec{x}_5, \dots, \vec{x}_8\}$ from each other, but it cannot discriminate between stimuli \vec{x}, \vec{x}' which have the same first component. It follows that regardless of how small σ^2 is—even in the noiseless case, the identification accuracy can be at most $\frac{1}{4}$. On the other hand, in scenario 2, it is clear that as σ^2 goes to zero, that the accuracy increases to 1. In figure 1.8 we have computed the identification accuracy for the two scenarios as a function of σ^2 . As we can see, for a large range of σ^2 values, the identification accuracy is greater in scenario 2 (reflecting greater information) than scenario 1.

Cross-validated identification accuracy

Notice that the procedure described earlier for computing the empirical identification is non-deterministic, because of the randomness in data-splitting. Therefore, it is extremely useful in practice to repeat the computation of identification accuracy for multiple data splits of the same size, and then average the result.

Concretely, define the leave- k -out cross-validated estimate of identification accuracy as follows:

1. Let L be a large number of Monte Carlo trials. For $i = 1, \dots, L$, carry out data-splitting to create a training set of size $T - k$, and a test set of size K . Let

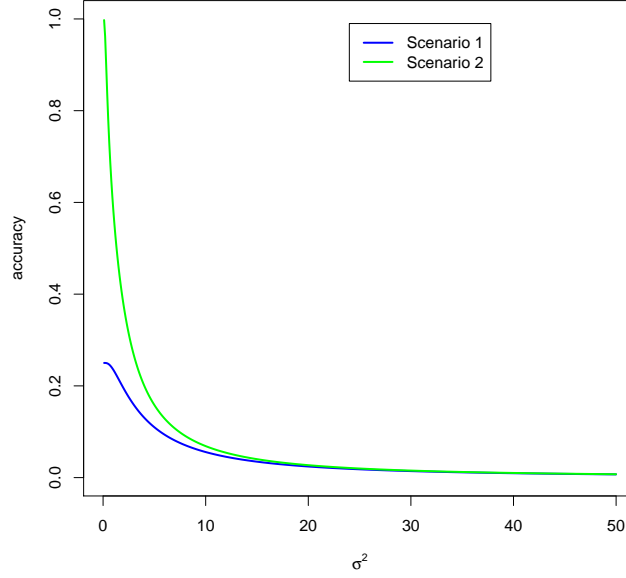


Figure 1.8: Identification accuracies in toy example

$\text{TA}^{(i)}$ be the identification accuracy on the test set.

2. Define the cross-validated identification accuracy as the average over the L quantities computed,

$$\text{TA}_{k,CV} = \frac{1}{L} \sum_{i=1}^L \text{TA}^{(i)}. \quad (1.1)$$

We describe some applications of cross-validated identification accuracy to the inference of mutual information in Chapters 4 and 5.

1.3.4 Information Theory

Information theory is motivated by the question of how to design a message-transmission system, which includes two users—a sender and a receiver, a *channel* that the sender can use in order to communicate to the receiver, and a protocol that specifies:

- a. how the sender can *encode* the message in order to transmit it over the channel.

Morse code is one example of an encoding scheme: a means of translating

plaintext into signals than can be transmitted over a wire (dots and dashes); and

- b. how the receiver can *decode* the signals received from the channel output in order to (probabilistically) recover the original message.

Beginning with Shannon (1948), one constrains the properties of the channel, and studies properties of encoding/decoding protocols to be used with the channel. Two types of channels are studied: *noiseless* channels, which transmit symbols from a fixed alphabet (e.g. “dots” and “dashes”) from the sender to receiver, and *noisy* channels, which transmit symbols from a discrete symbol space \mathcal{Y} to a possibly different symbol space \mathcal{X} in a stochastic fashion. That is, for each input symbol $y \in \mathcal{Y}$, the transmitted symbol⁴ output X is drawn from a distribution F_y that depends on y . It is the study of noisy channels that is of primary interest to us.

We allow the sender to transmit a sequence of L input symbols over the channel, $\vec{Y} = (Y_1, Y_2, \dots, Y_L)$. The receiver will observe the output $\vec{X} = (X_1, X_2, \dots, X_L)$, where each X_i is drawn from F_{Y_i} independently of the previous X_1, \dots, X_{i-1} .

An example of a noisy channel is the *bit-flip* channel. Let $\mathcal{Y} = \mathcal{X} = \{0, 1\}$, so that both the input and output are binary strings. The bit flip channel is given by

$$F_0 = \text{Bernoulli}(\epsilon)$$

$$F_1 = \text{Bernoulli}(1 - \epsilon)$$

so that $X = Y$ with probability $1 - \epsilon$, and $X = 1 - Y$ otherwise.

Now, let us assume that the sender wants to transmit message M , out of a finite set of possible messages $\mathcal{M} = \{1, \dots, m\}$. The message must be encoded into a signal $\vec{Y} \in \mathcal{Y}^L$, which is sent through a stochastic channel F . Thus, the encoding scheme is given by a *codebook* or *encoding function* $g : \{1, \dots, m\} \rightarrow \mathcal{Y}^L$ which specifies how each message i is mapped to an input sequence, $g(i) \in \mathcal{Y}^L$. Conversely, the decoding scheme is given by a decoding function $d(\vec{X})$ which infers the message $\{1, \dots, m\}$

⁴Note that here we have flipped the usual convention in information theory, in which the letter X commonly denotes the input and Y denotes the output. However, we flip the notation in order to match the convention in multi-class classification.

from the recieved signal \vec{X} . Theoretically speaking⁵, a reasonable decoding scheme is the *maximum likelihood decoder*,

$$d(\vec{x}) = \max_{i \in \{1, \dots, m\}} \Pr[\vec{X} = \vec{x} | \vec{Y} = g(i)] = \max_{i \in \{1, \dots, m\}} \prod_{j=1}^L F_{(g(i))_j}(X_j).$$

The design of encoding/decoding schemes with minimal error (or other desirable properties) over a fixed channel is a highly nontrivial problem, which remains a core problem in the information theory literature. However, Shannon's original proof of the noisy channel capacity theorem demonstrates a surprising fact, which is that for large message spaces \mathcal{M} , close-to-optimal information transmission can be achieved by using a *randomized* codebook. In order to discuss the noisy channel capacity theorem and the construction of the randomized codebook, we first need to define the concept of *mutual information*.

Mutual information

If \mathbf{X} and \mathbf{Y} have joint density $p(\mathbf{x}, \mathbf{y})$ with respect to the product measure $\mu_x \times \mu_y$, then the mutual information is defined as

$$I(\mathbf{X}; \mathbf{Y}) = \int p(\mathbf{x}, \mathbf{y}) \log \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} d\mu_x(\mathbf{x}) d\mu_y(\mathbf{y}).$$

where $p(\mathbf{x})$ and $p(\mathbf{y})$ are the marginal densities with respect⁶ to μ_x and μ_y . When the reference measure $\mu_x \times \mu_y$ is unambiguous, note that $I(\mathbf{X}; \mathbf{Y})$ is simply a functional of the joint density $p(\mathbf{x}, \mathbf{y})$. Therefore, we can also use the *functional* notation

$$I[p(\mathbf{x}, \mathbf{y})] = \int p(\mathbf{x}, \mathbf{y}) \log \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} d\mu_x(\mathbf{x}) d\mu_y(\mathbf{y}).$$

The mutual information is a measure of dependence between random vectors \mathbf{X} and \mathbf{Y} , and satisfies a number of important properties.

⁵Practically speaking, the maximum likelihood (ML) decoder may be intractable to implement, and computational considerations mean that development of practical decoders remains a challenging problem.

⁶Note that the mutual information is invariant with respect to change-of-measure.

1. The channel input \mathbf{X} and output \mathbf{Y} can be random vectors of arbitrary dimension, and the mutual information remains a scalar functional of the joint distribution P of (\mathbf{X}, \mathbf{Y}) .
2. When \mathbf{X} and \mathbf{Y} are independent, $I(\mathbf{X}; \mathbf{Y}) = 0$; otherwise, $I(\mathbf{X}; \mathbf{Y}) > 0$.
3. The data-processing inequality: for any vector-valued function \vec{f} of the output space,

$$I(\mathbf{X}; \vec{f}(\mathbf{Y})) \leq I(\mathbf{X}; \mathbf{Y}).$$

4. Symmetry: $I(\mathbf{X}; \mathbf{Y}) = I(\mathbf{Y}; \mathbf{X})$.
5. Independent additivity: if $(\mathbf{X}_1, \mathbf{Y}_1)$ is independent of $(\mathbf{X}_2, \mathbf{Y}_2)$, then

$$I((\mathbf{X}_1, \mathbf{X}_2); (\mathbf{Y}_1, \mathbf{Y}_2)) = I(\mathbf{X}_1; \mathbf{Y}_1) + I(\mathbf{X}_2; \mathbf{Y}_2).$$

Three additional consequences result from the data-processing inequality:

- *Stochastic data-processing inequality.* If \vec{f} is a stochastic function independent of both \mathbf{X} and \mathbf{Y} , then

$$I(\mathbf{X}; \vec{f}(\mathbf{Y})) \leq I(\mathbf{X}; \mathbf{Y}).$$

This can be shown as follows: any stochastic function $\vec{f}(\mathbf{Y})$ can be expressed as a deterministic function $\vec{g}(\mathbf{Y}, W)$, where W is a random variable independent of \mathbf{X} and \mathbf{Y} . By independent additivity,

$$I(\mathbf{X}; \mathbf{Y}) = I(\mathbf{X}; (\mathbf{Y}, W)).$$

Then, by the data-processing inequality,

$$I(\mathbf{X}; \mathbf{Y}) = I(\mathbf{X}; (\mathbf{Y}, W)) \geq I(\mathbf{X}; \vec{g}(\mathbf{Y}, W)) = I(\mathbf{X}; \vec{f}(\mathbf{Y})).$$

- *Invariance under bijections.* If \vec{f} has an inverse \vec{f}^{-1} , then

$$I(\mathbf{X}; \vec{f}(\mathbf{Y})) \leq I(\mathbf{X}; \mathbf{Y}) = I(\mathbf{X}; \vec{f}^{-1}(\vec{f}(\mathbf{Y}))) \leq I(\mathbf{X}; \vec{f}(\mathbf{Y})),$$

therefore, $I(\mathbf{X}; \vec{f}(\mathbf{Y})) = I(\mathbf{X}; \mathbf{Y})$.

- *Monotonicity with respect to inclusion of outputs.* Suppose we have an output ensemble $(\mathbf{Y}_1, \mathbf{Y}_2)$. Then the individual component \mathbf{Y}_1 can be obtained as a projection of the ensemble. By the data-processing inequality, we therefore have

$$I(\mathbf{X}; \mathbf{Y}_1) \leq I(\mathbf{X}; (\mathbf{Y}_1, \mathbf{Y}_2)).$$

Intuitively, if we observe both \mathbf{Y}_1 and \mathbf{Y}_2 , this can only *increase* the information we have about \mathbf{X} compared to the case where we only observe \mathbf{Y}_1 by itself.

And it is the property of *invariance under bijections*, inclusive of non-linear bijections, which qualifies mutual information as a *non-linear measure of dependence*. Linear correlations are invariant under scaling and translation, but not invariant to *nonlinear* bijections.

Besides the formal definition, there are a number of well-known alternative characterizations of mutual information in terms of other information-theoretic quantities: the *entropy* H :

$$H_\mu(\mathbf{X}) = - \int p(\mathbf{X}) \log p(\mathbf{X}) d\mu(\mathbf{X}),$$

and the *conditional entropy*:

$$H_\mu(\mathbf{X}|\mathbf{Y}) = - \int p(\mathbf{Y}) d\mu_y(\mathbf{Y}) \int p(\mathbf{X}|\mathbf{Y}) \log p(\mathbf{X}|\mathbf{Y}) d\mu_x(\mathbf{X}).$$

Some care needs to be taken with entropy and conditional entropy since they are not invariant with respect to change-of-measure: hence the use of the subscript in the notation H_μ . In particular, there is a difference between *discrete entropy* (when μ is the counting measure) and *differential entropy* (when μ is p -dimensional Lebesgue measure.) Intuitively, entropy measures an observer's uncertainty of the random variable \mathbf{X} , supposing the observer has no prior information other than the distribution

of \mathbf{X} . Conditional entropy measures the *expected uncertainty* of \mathbf{X} supposing the observer observes \mathbf{Y} .

The following identities characterize mutual information in terms of entropy:

$$I(\mathbf{X}; \mathbf{Y}) = H_{\mu_x \times \mu_y}((\mathbf{X}, \mathbf{Y})) - H_{\mu_x}(\mathbf{X}) - H_{\mu_y}(\mathbf{Y}).$$

$$I(\mathbf{X}; \mathbf{Y}) = H_{\mu_y}(\mathbf{Y}) - H_{\mu_y}(\mathbf{Y}|\mathbf{X}). \quad (1.2)$$

Loosely speaking, $H_{\mu_y}(\mathbf{Y})$ is the uncertainty of \mathbf{Y} before having observed \mathbf{X} , and $H_{\mu_y}(\mathbf{Y}|\mathbf{X})$ is the uncertainty of \mathbf{Y} after having observed \mathbf{X} , hence $H_{\mu_y}(\mathbf{Y}) - H_{\mu_y}(\mathbf{Y}|\mathbf{X})$ is how much the observation of \mathbf{X} has *reduced* the uncertainty of \mathbf{Y} . Stated in words,

$$I(\mathbf{X}; \mathbf{Y}) = \text{average reduction of uncertainty about } \mathbf{Y} \text{ upon observing } \mathbf{X}.$$

The identity (1.2) is also noteworthy as being practically important for estimation of mutual information. Since the entropies in question only depend on the marginal and conditional distributions of \mathbf{Y} , the problem of estimating $I(\mathbf{X}; \mathbf{Y})$ can be reduced from a $\dim(\mathbf{X}) + \dim(\mathbf{Y})$ -dimensional nonparametric estimation problem to a $\dim(\mathbf{Y})$ -dimensional problem: hence this identity is a basis of several methods of estimation used in neuroscience, such as Gastpar (2014).

Channel capacity and randomized codebooks

As a general measure of dependence, mutual information has enjoyed numerous and diverse applications outside of information theory. However, its original role in Shannon's paper was to define the quantity known as *channel capacity* of a noisy channel.

Let us first note that the channel capacity of a noiseless channel with S symbols is simply $\log S$. The justification is that if we allow L symbols to be sent, then S^L possible messages can be encoded. Therefore, the channel capacity of a noiseless channel can be understood as the logarithm of the number of possible messages to be transmitted divided by the length of the sequence, with is $\log S$.

However, how can the idea of channel capacity be generalized to the noisy case? At first glance, it would seem like no comparison is possible, because no matter how

many symbols L the sender is allowed to transmit, the probability of decoding error may remain nonzero. Consider the bit-flip channel, where $X = Y$ with probability $1 - \epsilon$ and $X = 1 - Y$ otherwise. Given two different messages, $M \in \{1, 2\}$, a reasonable encoding scheme is for the sender to transmit a string of L repeated zeros for $M = 1$, and an string of L repeated ones for $M = 2$.

$$Y_1 = Y_2 = \dots = Y_L = M - 1.$$

The receiver should guess $M = 1$ if she receives more zeros than ones, and guess $M = 2$ otherwise. However, for any L , the decoding error will always be nonzero. Therefore there seems to be no analogy to the noiseless channel, where zero decoding error can be achieved.

Shannon's idea was to invent an asymptotic definition of channel capacity. Consider a sequence of problems where the number of messages M is increasing to infinity. In the m th coding problem, where $M = m$, let (g_m, d_m) be an encoder/decoder pair (or *protocol*), where g_m produces strings of length L_m . Let e_m be the maximum error probability over all messages $1, \dots, m$ when using the protocol (g_m, d_m) . Now, let us require that we choose (g_m, d_m) so that the error probability vanishes in the limit:

$$\lim_{m \rightarrow \infty} e_m \rightarrow 0.$$

We can define the channel capacity to be the best possible limiting ratio

$$C = \lim_{m \rightarrow \infty} \frac{\log m}{L_m}$$

over all sequences of protocols that have vanishing error probability. Note that this definition yields $C = \log S$ for the noiseless channel, but can also be extended to the noisy channel case. Remarkably, Shannon finds an explicit formula for the noisy channel capacity, which is proved in his noisy channel capacity theorem. We will now discuss how to calculate the capacity of a noisy channel.

First, let us define the set of joint distributions which can be realized in the noisy channel. Let p_y be a probability distribution over input symbols \mathcal{Y} . If we transmit

input Y randomly according to $Y \sim p_y$, the induced joint distribution $p(Y, X)$ is given by

$$p(y, x) = p_y(y)F_y(\{x\}).$$

The set \mathcal{P} is simply the collection of all such distributions: that is,

$$\mathcal{P} = \{p(y, x) \text{ such that } p(x|y) = F_y(\{x\}) \text{ for all } (x, y) \in \mathcal{X} \times \mathcal{Y}\}.$$

Suppose we have a noisy channel with transmission probabilities given by $\{F_y\}_{y \in \mathcal{Y}}$. Shannon came with the following result:

$$C = \max_{p \in \mathcal{P}} I[p(y, x)].$$

The noisy channel capacity is given by the maximal mutual information $I(Y; X)$ over all joint distributions of (Y, X) that can be realized in the channel.

To show that $C = \max_p I[p(y, x)]$ is the noisy channel capacity, then, (i) we need to show that there exists a sequence of codes with length $L = \frac{\log M}{C}$ which achieves vanishing⁷ decoding error as $M \rightarrow \infty$, and (ii) we need to show that any code with a shorter length has non-vanishing decoding error. We omit the proof of (i) and (ii), which can be found in any textbook on information theory, such as Cover and Thomas 2006. However, for our purposes, it is very much worth discussing the construction that shows direction (i) of the proof—the achievability of channel capacity.

For a given channel $\{F_y\}$, let $p^* \in \mathcal{P}$ be the distribution which maximizes $I[p(y, x)]$. Let p_y^* be the marginal distribution of Y , and let $L = \lceil \frac{\log M}{C} \rceil$. Now we can define the random code. Let $g(i) = (Y_1^{(i)}, \dots, Y_L^{(i)})$ where $Y_j^{(i)}$ are iid draws from p_y^* for $i = 1, \dots, M$ and $j = 1, \dots, L$. Shannon proved that average decoding error, taken over the distribution of random codebooks, goes to zero as $M \rightarrow \infty$. This implies the existence of a deterministic sequence of codebooks with the same property, hence establishing (i).

⁷Shannon's noisy channel capacity theorem shows a much stronger property—that the *maximum* decoding error over all messages has to vanish. However, for our purposes, we will limit our discussion to a weaker form of the noisy channel capacity theorem which is only concerned with average decoding error over all messages.

1.3.5 Comparisons

We see that both the multi-class classification problem and the noisy channel model present examples of discrimination problems where one must recover some latent variable Y from observations X , where X is related to Y through the family of conditional distributions F_Y . One difference is that while in multi-class classification, F_Y is unknown and has to be inferred from data, in the noisy channel model, the stochastic properties of the channel F_Y are usually assumed to be known. A second difference is that in the noisy channel model, there is a choice in how to specify the encoding function $g(M)$, which affects subsequent performance. Finally, in the broader research context, machine learning research has traditionally focused on multi-class problems with relatively few classes, while information theory tends to consider problems in asymptotic regimes where the number of possible messages m is taken to infinity. These differences are perhaps sufficient to explain why little overlap exists in the respective literatures between multi-class classification and the noisy channel model.

However, an interesting development in the machine learning community has been the application of multi-class classification to problems with increasingly large and complex label sets. Consider the following timeline of representative papers in the multi-class classification literature:

- Fisher’s Iris data set, Fisher 1936, $K = 3$ classes
- Letter recognition, Frey and Slate 1991, $K = 26$ classes
- Michalski’s soybean dataset, Micaltsd 1980, $K = 15$ classes
- The NIST handwritten digits data set, Grother 1995, $K = 10$ classes
- Phoneme recognition on the TIMIT dataset, Clarkson and Moreno 1999, $K = 39$ classes
- Object categorization using Corel images, Duygulu et al. 2002 $K = 371$ classes
- Object categorization for ImageNet dataset, Deng et al. 2010, $K = 10,184$ classes

- The 2nd Kaggle large-scale hierarchical text classification challenge (LSHTC), Partalas et al. 2015, $K = 325,056$

As we can see, in recent times we begin to see classification problems with extremely large label sets. In such large-scale classification problems, or ‘extreme’ classification problems, results for $K \rightarrow \infty$ numbers of classes, like those found in information theory, begin to look more applicable.

This work focuses on a particular intersection between multi-class classification and information theory, which is the study of *random classification tasks*. In numerous domains of applied mathematics, it has been found that systems with large numbers of components can be modelled using randomized versions of those same systems, which are more tractable to mathematical analysis: for example, studying the properties of networks by studying random graphs in graph theory, or studying the performance of combinatorial optimization algorithms for random problem instances. Similarly, it makes sense to posit randomized models of multi-class discrimination problems. Since information theorists were the first to study discrimination problems with large number of classes, we find in the information theory literature a long tradition of the study of *random code* models. This thesis is dedicated to the study of the analogue of random code models in the multi-class classification setting: models of *randomized classification*, which we motivate and analyze in the next chapter.

Chapter 2

Randomized classification

As we foreshadowed in the introduction, randomized classification is also one of the three methods we consider for evaluating representations. Yet, two other applications of randomized classification are (i) for providing a formalism for evaluating *recognition systems*, and (ii) for studying generalizability of certain classification-based experiments. The application of recognition systems provides the most intuitive way of understanding the randomized classification task; therefore, in this chapter, we begin with a discussion in section 2.1 of recognition tasks, and within this context, motivate the definition of a randomized classification task in section 2.2. We propose to use the randomized classification task to model the problem of recognition, and to evaluate performance via the *average accuracy*. Next, to put our ideas into practice, we need ways to estimate the average accuracy from data, which we address in 2.3.

Meanwhile, the problem of generalizing classification experiments provides a natural motivation for studying the variance of classification accuracy within a randomized classification task, which we cover in section 2.4. Meanwhile, another one of the methods we consider—the identification task, is closely connected with the randomized classification task. We discuss how our results in 2.4 can also be applied to the identification accuracy.

2.1 Recognition tasks

Human brains have a remarkable ability to recognize objects, faces, spoken syllables and words, and written symbols or words, and this recognition ability is essential for everyday life. While researchers in artificial intelligence have attempted to meet human benchmarks for these classical recognition tasks for the last few decades, only very recent advances in machine learning, such as deep neural networks, have allowed algorithmic recognition algorithms to approach or exceed human performance (LeCun, Bengio, and Hinton 2015).

Within the statistics and machine learning literature, the usual formalism for studying a recognition task is to pose it as a *multi-class classification* problem. One delineates a finite set of distinct entities which are to be recognized and distinguished, which is the *label set* \mathcal{Y} . The input data is assumed to take the form of a finite-dimensional real *feature vector* $X \in \mathbb{R}^p$. Each input instance is associated with exactly one true label $Y \in \mathcal{Y}$. The solution to the classification problem takes the form of an algorithmically implemented *classification rule* h that maps vectors X to predicted labels $\hat{Y} \in \mathcal{Y}$. The classification rule can be constructed in a data-dependent way: that is, one collects a number of labelled *training observations* (X_1, Y_1) which is used to inform the construction of the classification rule h . The quality of the classification rule h is measured by *generalization accuracy*

$$\text{GA}(h) = \Pr[h(X) = Y],$$

where the probability is defined with reference to the unknown population joint distribution of (X, Y) .

However, a limitation of the usual multi-class classification framework for studying recognition problems is the assumption that the label set \mathcal{Y} is finite and known in advance. When considering human recognition capabilities, it is clear that this is not the case. Our ability to recognize faces is not limited to some pre-defined, fixed set of faces; similarly with our ability to recognize objects in the environment. Humans learn to recognize novel faces and objects on a daily basis. And, if artificial intelligence is to fully match the human capability for recognition, it must also possess the ability

to add new categories of entities to its label set over time; however, at present, there currently exists a void in the machine learning literature on the subject of the online learning of new classes in the data.

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

2.2 Randomized classification

2.2.1 Motivation

The formalism of classification is inadequate for studying many practical questions related to the generalizability of the facial recognition system. Using test data, we estimate the generalization accuracy of a recognition system. However, these estimated accuracies apply only to the particular collection of individuals $\{y^{(1)}, \dots, y^{(k)}\}$. If we were to add a new individual $y^{(k+1)}$ to the dataset, for instance, when photographs are uploaded on Facebook containing a new user, this defines a totally new classification problem because the expanded set of labels $\{y^{(1)}, \dots, y^{(k+1)}\}$ defines a different response space than the old set of labels $\{y^{(1)}, \dots, y^{(k)}\}$. Yet, these two classification problems are clearly linked. To take another example, a client might want to run a facial recognition system developed by lab A on their own database of individuals. In this case, there might be no overlap between the people in lab A's database and the people in the client's database. And yet, the client might still expect the performance figures reported by lab A to be informative of how well the recognition system will do on their own database!

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

2.2.2 Setup

By ‘recognition system,’ we really mean a *learning algorithm* Λ which can take training data as input, and produce a classification rule for recognizing faces. While a classification rule is bound to a specific label set, a learning algorithm can be applied to datasets with arbitrary label sets, and be continually updated as new labels are added to the label set. To ‘update’ a facial recognition system with new data means to apply the learning algorithm to the updated database.

Now we can formalize what it means to generalize performance from one problem to another. A *classification problem* P is specified by a label set \mathcal{Y} , a predictor space \mathcal{X} , a joint distribution G on $\mathcal{X} \times \mathcal{Y}$, and a *sampling scheme* S for obtaining training data (for example, to obtain n observations from G i.i.d.). The sampling scheme is needed because we cannot say much about how the learning algorithm will perform unless we know how much training data it is going to have. The generalization accuracy (GA) of the algorithm Λ on the classification problem $P = (\mathcal{Y}, \mathcal{X}, G, S)$ is defined as the expected risk of the resulting classification rule h ,

$$\text{GA}_P(\Lambda) = \mathbf{E}[\text{GA}(h)]$$

where h is produced by applying Λ to the training data, sampled by S . The expectation is taken over the randomness in the generation of the training data.

At this point we have defined a general transfer learning problem: given two different classification problems $P_1 = (\mathcal{Y}_1, \mathcal{X}_1, G_1, S_1)$ and $P_2 = (\mathcal{Y}_2, \mathcal{X}_2, G_2, S_2)$, what can we say about the relationship between $\text{GA}_{P_1}(\Lambda)$ and $\text{GA}_{P_2}(\Lambda)$? Not much, unless we make many more assumptions about how P_1 and P_2 are linked.

The basic approach we will take is to assume that both P_1 and P_2 have been generated randomly via a common mechanism. In the original motivating context of facial recognition, this is to say that two different label sets \mathcal{Y}_1 and \mathcal{Y}_2 are linked because they both belong to a common population of labels \mathcal{Y} , i.e., the population of all possible humans, and to further assume that both have been *sampled*, in the same manner, from \mathcal{Y} .

The study of how to make inferences about the risk in P_2 given information about the performance achieved in P_1 , granted a set of assumptions on how P_1 and P_2 have been randomly generated (and are thereby linked through shared randomization mechanisms) forms the basis of the subject of *randomized classification*.

As we noted in the introduction, the problem of randomized classification has a close ancestor in the study of *random code models* in information theory. There, the problem is to understand the *decoding performance* (the analogue to risk) of an encoding/decoding system P which has a randomized code space \mathcal{Y} . Where random code models have a random codebook which is a sample over a distribution of all possible codes, randomized classification problems have a random label set that is a sample of a larger label space. However, the results we obtain for randomized classification are more general in nature than the existing results available for random code models, because work on random codes is generally limited to asymptotic settings, whereas we obtain finite- k results, and because random code models assume a specific product-distribution structure on (X, Y) which is not appropriate for classification problems.

2.2.3 Assumptions

The randomized classification model we study has the following features. We assume that there exists an infinite (or even continuous) label space \mathcal{Y} and a response space

$\mathcal{X} \in \mathbb{R}^p$. For each label $y \in \mathcal{Y}$, there exists a distribution of features F_y . That is to say, that for a feature-label pair (X, Y) , that the conditional distribution of X given $Y = y$ is given by F_y . Furthermore, we assume that there exists a prior distribution π on the label space \mathcal{Y} .

A random classification task P can be generated as follows. The label set $\mathcal{S} = \{Y^{(1)}, \dots, Y^{(k)}\}$ is generated by drawing labels $Y^{(1)}, \dots, Y^{(k)}$ i.i.d. from π . The joint distribution G of pairs (X, Y) is uniquely specified by the two conditions that (i) the marginal distribution of Y is uniform over \mathcal{S} , and (ii) the conditional distribution of X given $Y = Y^{(i)}$ is $F_{Y^{(i)}}$. We sample both a training set and a test set. The training set is obtained by sampling r_1 observations $X_{j,train}^{(i)}$ i.i.d. from $F_{Y^{(i)}}$ for $j = 1, \dots, r_1$. The test set is likewise obtained by sampling r_2 observations $X_j^{(i)}$ i.i.d. from $F_{Y^{(i)}}$ for $j = 1, \dots, r_2$. For notational convenience, we represent the training set as the set of empirical distributions $\{\hat{F}_{Y^{(i)}}\}_{i=1}^k$ where

$$\hat{F}_{Y^{(i)}} = \frac{1}{r_1} \sum_{j=1}^{r_1} \delta_{X_{j,train}^{(i)}}.$$

Figure 2.1 illustrates the sampling scheme for generating the training set.

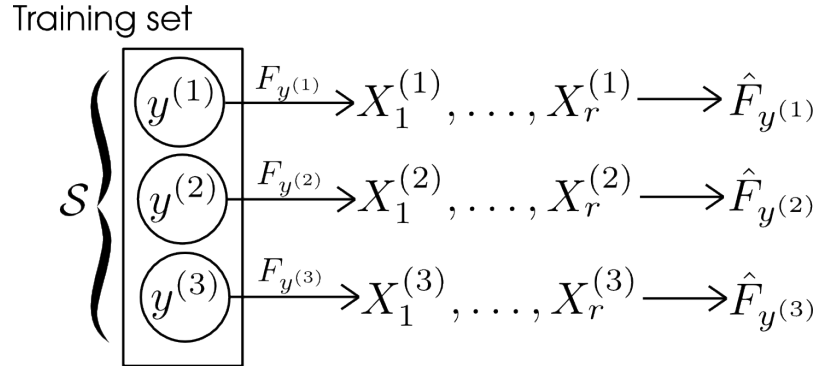


Figure 2.1: Training set

Our analysis will also rely on a property of the classifier. 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 2.2.1 1. The classification rule h is called a *marginal rule* if

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

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

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

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

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

2. Both the Quadratic Discriminant Analysis and the naive Bayes classifiers can be seen as specific instances of an estimated Bayes classifier¹. 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.

The operation of a marginal classifier is illustrated in figure 2.2. Since a marginal classifier is specified entirely by its marginal model \mathcal{M} , we will take the notational convention of referring to a marginal classifier as \mathcal{M} .

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 estimated risk when testing the classifier on a test set.

If we fix a particular realization of the random label set $\mathcal{S} = \{y^{(1)}, \dots, y^{(k)}\}$ as well as the training set $\{\hat{F}_{y^{(i)}}\}_{i=1}^k$, then the classifier $h(x)$ is fixed, and only the third source of randomness (in test risk) applies. However, the true generalization accuracy

¹QDA 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 $p(x_i|y)$

Classification Rule

$$M_{y^{(1)}}(x) = \mathcal{M}(\hat{F}_{y^{(1)}})(x)$$

$$M_{y^{(2)}}(x) = \mathcal{M}(\hat{F}_{y^{(2)}})(x)$$

$$M_{y^{(3)}}(x) = \mathcal{M}(\hat{F}_{y^{(3)}})(x)$$

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

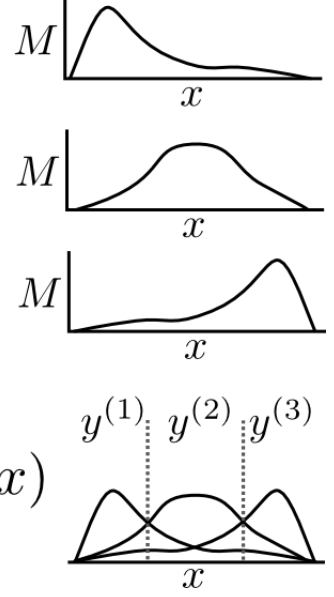


Figure 2.2: Classification rule

of the classifier is deterministic:

$$\begin{aligned} \text{GA}(h) &= \Pr[Y = h(X) | Y \sim \text{Unif}(\mathcal{S}), \mathcal{S}, \{\hat{F}_{y^{(i)}}\}_{i=1}^k] \\ &= \frac{1}{k} \sum_{i=1}^k \Pr[m_{y^{(i)}}(x) = \max_j m_{y^{(j)}}(x) | X \sim F_{y^{(i)}}, \mathcal{S}, \{\hat{F}_{y^{(i)}}\}_{i=1}^k] \\ &= \frac{1}{k} \sum_{i=1}^k \int_{\mathcal{X}} I(m_{y^{(i)}}(x) = \max_j m_{y^{(j)}}(x)) dF_{y^{(i)}}(x), \end{aligned}$$

recalling that $m_{y^{(i)}}(x) \stackrel{\text{def}}{=} \mathcal{M}(\hat{F}_{y^{(i)}})(x)$ is the margin function obtained from the training data of the i th class.

If we *fix* a particular realization of the random label set $\mathcal{S} = \{y^{(1)}, \dots, y^{(k)}\}$, then we can define the (generalization) accuracy specific to that label set. However, the training data $\{\hat{F}_{y^{(i)}}\}_{i=1}^k$ will be random. Let us denote the *distribution* of the empirical distribution \hat{F}_y constructed from sample size r as $\Pi_{y,r}$. The accuracy of the classifier

\mathcal{M} on label set \mathcal{S} is given by

$$\begin{aligned} \text{GA}_{\mathcal{S}}(\mathcal{M}) &= \Pr[Y = h(X) | Y \sim \text{Unif}(\mathcal{S}), \hat{F}_{y^{(i)}} \sim \Pi_{y^{(i)}, r_1}] \\ &= \frac{1}{k} \sum_{i=1}^k \int I(\mathcal{M}(\hat{F}_{y^{(i)}})(x) = \max_j \mathcal{M}(\hat{F}_{y^{(j)}})(x)) dF_{y^{(i)}}(x) \prod_{\ell=1}^k d\Pi_{y^{(\ell)}, r_1}(\hat{F}_{y^{(\ell)}}). \end{aligned}$$

The calculation of the accuracy (for fixed label set \mathcal{S}) is illustrated in figure 2.3.

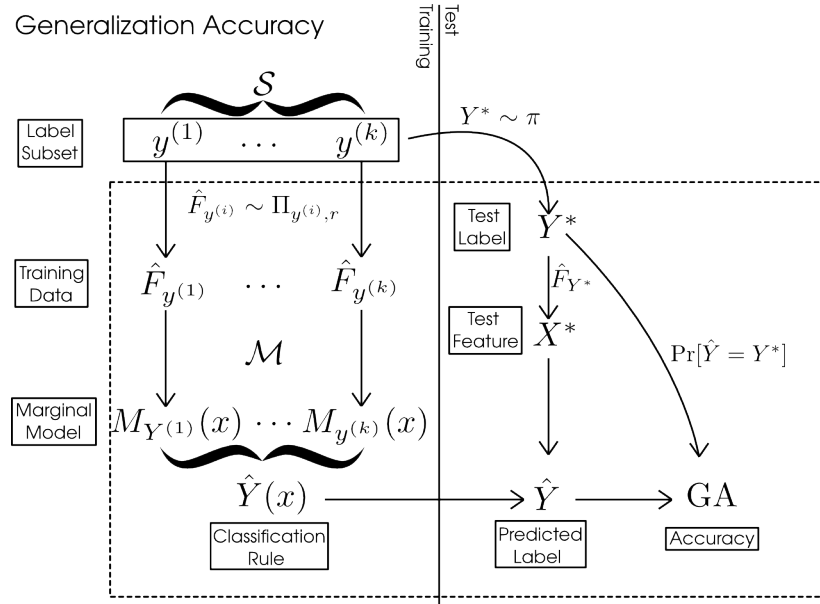


Figure 2.3: Generalization accuracy

Finally, suppose we do not fix any of the random quantities in the classification task P , and merely specify k , the number of classes, and r_1 , the number of repeats in the training set. Then the k -class, r -repeat *average generalization accuracy* of a marginal classifier \mathcal{M} is defined as

$$\begin{aligned} \text{AGA}_{k, r_1}(\mathcal{M}) &= \mathbf{E}[\text{GA}_{\mathcal{S}}(\mathcal{M}) | Y^{(1)}, \dots, Y^{(k)} \sim \pi] \\ &= \frac{1}{k} \sum_{i=1}^k \int I(\mathcal{M}(\hat{F}_{y^{(i)}})(x) = \max_j \mathcal{M}(\hat{F}_{y^{(j)}})(x)) dF_{y^{(i)}}(x) \prod_{\ell=1}^k d\Pi_{y^{(\ell)}, r_1}(\hat{F}_{y^{(\ell)}}) d\pi(y^{(\ell)}). \\ &= \int I(\mathcal{M}(\hat{F}_{y^{(1)}})(x) = \max_j \mathcal{M}(\hat{F}_{y^{(j)}})(x)) dF_{y^{(1)}}(x) \prod_{\ell=1}^k d\Pi_{y^{(\ell)}, r_1}(\hat{F}_{y^{(\ell)}}) d\pi(y^{(\ell)}). \end{aligned}$$

where the last line follows from noting that all k summands in the previous line are identical. The definition of average generalization accuracy is illustrated in Figure 2.4.

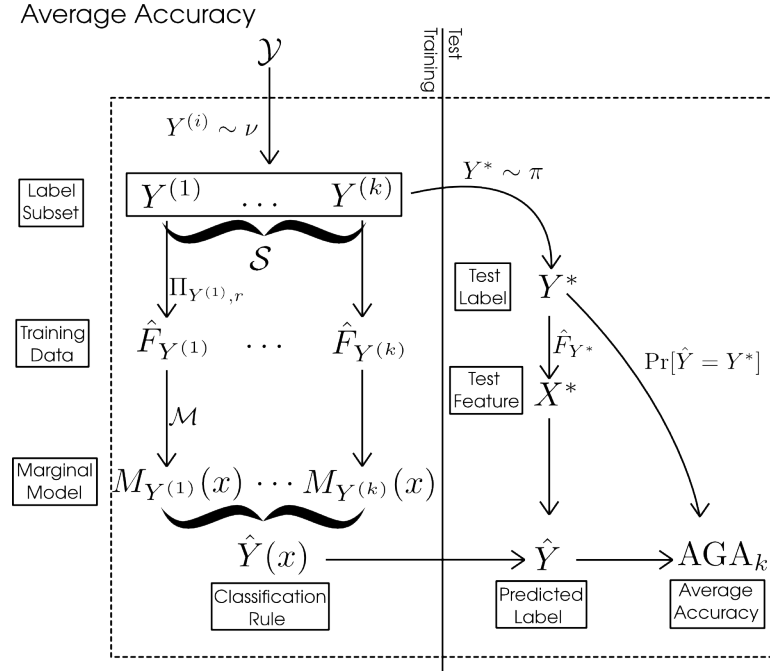


Figure 2.4: Average generalization accuracy

Having defined the average (generalization) accuracy for the randomized classification task, we begin to develop the theory of how to *estimate* the average accuracy in the next section.

2.3 Estimation of average accuracy

Suppose we have training and test data for a classification task P_1 with k_1 classes, r_1 -repeat training data and r_2 -repeat test data. That is, we have label set $\mathcal{S}_1 = \{y^{(i)}\}_{i=1}^{k_1}$, as well as training sample $\hat{F}_{y^{(i)}}$ and test sample $(x_1^{(i)}, \dots, x_{r_2}^{(i)})$ for $i = 1, \dots, k_1$. How can we estimate the k, r -average accuracy of a marginal classifier \mathcal{M} for arbitrary k and r ?

Let us start with the case $k = k_1$ and $r = r_1$. Then the answer is simple: construct

the classification rule h using marginal model \mathcal{M} from the training data. Then the test accuracy of h is an unbiased estimator of $\text{AGA}_{k,r}$.

This follows from definition. Observe that AGA_{k_1,r_1} is the expected prediction risk for the classification rule h for a randomized classification problem P with k_1 classes and r_1 -repeat training data. Of course, the classification task P_1 that we have been given is a random draw from the desired distribution of random classification problems. Therefore, the prediction risk of h constructed from P_1 is unbiased for AGA_{k_1,r_1} , and since test accuracy is unbiased for generalization accuracy, it follows that the test accuracy of h is an unbiased estimator of $\text{AGA}_{k,r}$, as we claimed.

In following sections, we consider more complicated cases where $k_1 \neq k$. However, before proceeding, let us first review the procedure for computing the test accuracy.

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 $\ell = 1, \dots, k_1$, and by finding the class with the highest margin $M_{i,j,\ell}$,

$$\hat{y}_j^{(i)} = y^{(\arg\max_{\ell} M_{i,j,\ell})}.$$

The test accuracy is the fraction of correct classification over test observations,

$$\text{TA} = \frac{1}{r_2 k_1} \sum_{i=1}^{k_1} \sum_{j=1}^{r_2} I(\hat{y}_j^{(i)} = y^{(i)}). \quad (2.1)$$

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 $y^{(i)}$ if and only if $R_{i,j,i} = k_1$. Thus, an equivalent expression

for test accuracy is

$$\text{TA} = \frac{1}{r_2 k_1} \sum_{i=1}^{k_1} \sum_{j=1}^{r_2} I\{R_{iji} = k_1\}. \quad (2.2)$$

2.3.1 Subsampling method

Next, let us consider the case where $k < k_1$ and $r = r_1$. Define a classification problem P_2 with label set \mathcal{S}_2 obtained by sampling k labels uniformly without replacement from \mathcal{S}_1 . Let the training and test data for P_2 be obtained by taking the training data and test data from P_1 belonging to labels in \mathcal{S}_1 . It follows that P_2 is a randomized classification task with k labels, r_1 -repeat training data and r_2 -repeat test data. Therefore, by the previous argument, the test accuracy for a classification rule h constructed using the training data in P_2 provides an unbiased estimate of AGA_{k,r_1} .

However, we can get a much better unbiased estimate of AGA_{k,r_1} by averaging over the randomization of \mathcal{S}_2 . Naïvely, this requires us to train and evaluate $\binom{k_1}{k}$ classification rules. However, due to the special structure of marginal classifiers, we do the computation in the same order of computation as evaluating a single classification rule (assuming that the computational bottleneck is in training the classifier.)

This is because the rank R_{iji} of the correct label, i , for the $x_j^{(i)}$ allows us to determine how many subsets \mathcal{S}_2 will result in a correct classification. For example $x_j^{(i)}$, there are $R_{iji} - 1$ labels with a lower margin than the correct label i . Therefore, as long as one of the classes in \mathcal{S}_2 is i , and the other $k - 1$ labels are from the set of $R_{iji} - 1$ labels with lower margin than i , the classification of $x_j^{(i)}$ will be correct. This implies that there are $\binom{R_{iji}-1}{k-1}$ such subsets \mathcal{S}_2 where $x_j^{(i)}$ is classified correctly, and therefore

$$\text{AverageTA}_{k,r_1} = \frac{1}{\binom{k_1}{k}} \frac{1}{r_2 k} \sum_{i=1}^{k_1} \sum_{j=1}^{r_2} \binom{R_{iji}-1}{k-1}. \quad (2.3)$$

2.3.2 Extrapolation

A much more challenging case is when $k_2 > k_1$: that is, we want to predict the performance of the classification model in a setting with more labels than we currently

see in the training set. This is the subject of Chapter 3.

2.3.3 Variance bounds

By now we have developed unbiased estimators for average generalization accuracy in the special case $k \leq k_1$, and the following chapter will present methods for the more difficult case $k > k_1$. However, to get useful inference statements, we also have to understand the variance of these estimators. For the large part, this is still work-in-progress. However, some first steps towards addressing this problem are described in the next section.

2.4 Reproducibility and Average Bayes accuracy

2.4.1 Motivation

In task fMRI experiments, one typically obtains data of the form (\vec{x}_i, y_i) , where \vec{x}_i are activity patterns obtained from the fMRI scan for a region of interest, and y_i are categories for tasks. The labels y_i are limited to a discrete label set $\mathcal{S} = \{y^{(1)}, \dots, y^{(k)}\}$. Data-splitting is used to obtain a test accuracy TA for predicting y from \vec{x} . The test accuracy is then interpreted as evidence for the specialization of that region of interest for the task at hand: a high accuracy suggests that the region is specialized for the task, while a low accuracy suggests that the region is not specialized for the task.

However, a limitation of this approach is the poor reproducibility of the test accuracy TA. Besides the dependence of TA on the particular subject participating in the experiment, the randomness in sampling training and test data, and the classifier used, there is an even more fundamental issue in that the classification task is often inconsistent from lab to lab. That is because the task exemplars—that is, the set of labels $\{y^{(i)}\}_{i=1}^k$, may be arbitrarily specified, and therefore even ignoring the effect of subject, sampling of the data, and classifier, the generalization accuracy depends on an arbitrary choice of exemplars.

On the other hand, fixing in advance the set of exemplars $\{y^{(i)}\}_{i=1}^k$ is also not a satisfactory solution, since the objective of the experiment is to understand the general relation between the task and the region of interest, and not the relationship between a particular set of task exemplars $\{y^{(i)}\}_{i=1}^k$ and the region of interest.

Randomized classification provides a solution for addressing both the variability in estimation target and generalization to the population, as long as one can justify the assumption that the labels $\{y^{(i)}\}_{i=1}^k$ are a random sample from some population of task exemplars π . While the generalization accuracy GA for any particular, fixed set of exemplars is *not* a population parameter, the average generalization accuracy AGA *is* defined with reference to a population, albeit also dependent on a specific classifier and sampling scheme. Meanwhile, the limitations on reproducibility due to differing choices of labels sets can be understood based on the variability properties of GA.

However, one can argue that randomized classification does not go far enough to ensure generalizability of results, because AGA still depends on the sampling scheme (the amount of training data) and the choice of classifier, which are both arbitrary experimental choices. Therefore, our proposal is to treat the average *Bayes* accuracy as the estimation target of interest. The average Bayes accuracy is defined independently of the classifier and sampling scheme, and we will develop tools for inferring probabilistic lower bounds (lower confidence bounds) of the average Bayes accuracy in this section.

2.4.2 Setup

Define the generalization accuracy of a classification rule f as the complement of its risk (under zero-one loss),

$$\text{GA}(f) = \Pr[Y = f(X)].$$

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

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

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, \dots, f_m , and taking the best generalization accuracy as an estimate of the Bayes accuracy.

Now consider a randomized classification task where the labels $\{Y^{(1)}, \dots, Y^{(k)}\}$ are drawn iid from a population π , and where the observations X for label Y are drawn from the conditional distribution F_Y . In this case, the Bayes accuracy is a random variable depending on the label set, since

$$\text{BA}(Y^{(1)}, \dots, Y^{(k)}) = \frac{1}{k} \sum_{i=1}^k \Pr[\operatorname{argmax}_{i=1}^k p(x|y^{(i)}) = i | X \sim F_{Y^{(i)}}].$$

The k -class Average Bayes accuracy is defined as the average Bayes accuracy,

$$\text{ABA}_k = \mathbf{E}[\text{BA}(Y^{(1)}, \dots, Y^{(k)})]$$

where the expectation is taken over the joint distribution of $\{Y^{(1)}, \dots, Y^{(k)}\}$.

2.4.3 Identities

The following theorem gives a convenient formula for computing ABA_k .

Theorem 2.4.1 *Suppose (X, Y) have a joint distribution with a density $p(x, y)$, with a marginal density $\pi(y)$. Then for a randomized classification task with $\{Y^{(1)}, \dots, Y^{(k)}\}$*

are drawn iid from π , the k -class average Bayes accuracy can be computed as

$$ABA_k[p(x, y)] = \frac{1}{k} \int \left[\prod_{i=1}^k \pi(y_i) dy_i \right] \int dx \max_i p(x|y_i).$$

Proof. Write

$$\begin{aligned} ABA_k[p(x, y)] &= \mathbf{E}[BA(Y^{(1)}, \dots, Y^{(k)})] \\ &= \frac{1}{k} \int \pi(y_1) \dots \pi(y_k) \sum_{i=1}^k I\{\operatorname{argmax}_{i=1}^k p(x|y_i) = i\} p(x|y_i) dx dy_1 \dots dy_k \\ &= \frac{1}{k} \int \pi(y_1) \dots \pi(y_k) p(x|y_{\operatorname{argmax}_{i=1}^k p(x|y_i)}) dy_1 \dots dy_k dx. \\ &= \frac{1}{k} \int \pi(y_1) \dots \pi(y_k) \max_{i=1}^k p(x|y_i) dy_1 \dots dy_k dx. \end{aligned}$$

2.4.4 Variability of Bayes Accuracy

By definition, $BA_k = BA(X_1, \dots, X_k)$ is already an unbiased estimator of ABA_k . However, to get confidence intervals for ABA_k , we also need to know the variability of BA_k .

We have the following upper bound on the variability.

Theorem 2.4.2 *For a randomized classification task with $\{Y^{(1)}, \dots, Y^{(k)}\}$ are drawn iid from π , the variability of the Bayes accuracy can be bounded as*

$$\operatorname{Var}[BA(Y^{(1)}, \dots, Y^{(k)})] \leq \frac{1}{4k}.$$

Proof. According to the Efron-Stein lemma,

$$\operatorname{Var}[BA(Y^{(1)}, \dots, Y^{(k)})] \leq \sum_{i=1}^k \mathbf{E}[\operatorname{Var}[BA|Y^{(1)}, \dots, Y^{(i-1)}, Y^{(i+1)}, \dots, Y^{(k)}]].$$

which is the same as

$$\operatorname{Var}[BA(Y^{(1)}, \dots, Y^{(k)})] \leq k \mathbf{E}[\operatorname{Var}[BA|Y^{(1)}, \dots, Y^{(k-1)}]].$$

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

Note the following trivial results

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

This implies

$$\text{BA}(Y^{(1)}, \dots, Y^{(k)}) - \frac{1}{k} \leq \frac{k-1}{k} \text{BA}(Y^{(1)}, \dots, Y^{(k-1)}) \leq \text{BA}(Y^{(1)}, \dots, Y^{(k)}).$$

i.e. conditional on $(Y^{(1)}, \dots, Y^{(k-1)})$, BA_k is supported on an interval of size $1/k$. Therefore,

$$\text{Var}[\text{BA}|Y^{(1)}, \dots, Y^{(k-1)}] \leq \frac{1}{4k^2}$$

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

2.4.5 Inference of average Bayes accuracy

Recall the procedure used to estimate generalization error: by applying *data-splitting*, one creates a *training set* consisting of r_1 repeats per class, and a *test set* consisting of the remaining $r_2 = r - r_1$ repeats. One inputs the training data into the classifier to obtain the classification rule f , and computes the test accuracy,

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

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

$$\text{GA}(f) \leq \text{BA}(y^{(1)}, \dots, y^{(k)}),$$

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

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

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

Combining these facts, we get the following result.

Theorem 2.4.3 *The following is a $(1 - \alpha)$ lower confidence bound for ABA_k :*

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

That is,

$$\Pr[\underline{\text{ABA}}_k > \text{ABA}_k] \leq \alpha.$$

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

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

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 $\text{BA}(Y^{(1)}, \dots, Y^{(k)}) > \text{ABA}_k + \frac{1}{\sqrt{2\alpha k}}$ 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 . \square

2.4.6 Implications for reproducibility

Returning to the original problem of experimental reproducibility or generalizability to a larger population under the assumption that the task exemplars have been drawn

from a population. Then it follows from our analysis that both reproducibility and generalizability are assured if experimental parameters enable inference of a lower confidence bound $\underline{\text{ABA}}_k$ which is close to the true average Bayes accuracy. This would require two conditions:

1. The training data is sufficiently large, and the classifier is chosen so that the generalization accuracy is close to the Bayes accuracy.
2. The number of classes k is sufficiently large so that the Bayes accuracy $\text{BA}(Y^{(1)}, \dots, Y^{(k)})$ is close to the average Bayes accuracy.

Under those two conditions, the lower confidence bounds $\underline{\text{ABA}}_k$ have a distribution which is concentrated close to the true population parameter ABA_k , which ensures both reproducibility (in the estimates produced by two realizations of the same randomized classification task) and generalizability to the population parameter ABA .

Our analysis of the variance of BA_k gives an easy criterion for ensuring that k is large enough, since k needs to be inversely proportional to the desired variance. However, we have not developed methods for checking condition 1. In practice, when a large number of different classifiers achieve similar accuracies, and when performance is not particularly affected by training on a fractional subsample of the training data, this can be taken as evidence that the generalization accuracy is close to Bayes accuracy. However, it remains to theoretically characterize the assumptions that make this possible (e.g. smoothness of the model, low signal-to-noise ratio) and to develop formal tests for convergence to Bayes accuracy.

2.4.7 Application to identification task

In the same way that average Bayes accuracy provides an upper bound for the average generalization error in the randomized classification task, the average Bayes accuracy can also bound the expected cross-validated identification accuracy (1.1).

This is clear if we define a *generalized identification task*, which is constructed as follows:

Generalized identification task

1. Draw $\vec{X}_1, \dots, \vec{X}_k$ from the marginal distribution of \vec{X} .
2. Draw index J uniformly from $\{1, \dots, k\}$.
3. Generate \vec{Y} from the conditional distribution $p(\vec{y}|\vec{X} = \vec{x}_j)$.
4. Let g be a function which takes a response vector \vec{Y} and k feature vectors $\vec{X}_1, \dots, \vec{X}_k$ and outputs an index in $\{1, \dots, k\}$

$$g : \mathcal{Y} \times \mathcal{X}^k \rightarrow \{1, \dots, k\}.$$

Define the *identification generalization accuracy* of g as

$$\Pr[g(\vec{Y}; \vec{X}_1, \dots, \vec{X}_k) = J]$$

under the above setup.

We will now claim the following. *Claim 1:* the identification task introduced in section 1.3.3 is a special case of the generalized identification task. *Claim 2:* one can define a randomized classification task such that the k -class average Bayes accuracy is an upper bound of the identification generalization accuracy.

Claim 1 can be argued as follows. In the identification task introduced in section 1.3.3, when we fix a training set, the expected test identification accuracy is equivalent to the identification generalization accuracy of a rule g , which can be written as

$$g(\vec{Y}; \vec{X}_1, \dots, \vec{X}_k) = \operatorname{argmin}_j (\vec{Y} - f(\vec{X}_j))^T \hat{\Sigma}^{-1} (\vec{Y} - f(\vec{X}_j)).$$

where $f(\vec{x})$ is the estimate of the regression function $\mathbf{E}[\vec{Y}|\vec{X} = \vec{x}]$, and $\hat{\Sigma}$ is the estimated covariance matrix of the noise.

Claim 2 follows because the Bayes identification rule,

$$g(\vec{Y}; \vec{X}_1, \dots, \vec{X}_k) = \operatorname{argmax}_j p(\vec{Y}|\vec{X} = \vec{x})$$

maximizes the identification generalization accuracy over all functions g . But now consider a randomized classification task where \vec{X} are the labels, and \vec{Y} are the feature

vectors (not that this is the reverse of the convention used in the rest of the chapter.) Then we see that the Bayes identification rule g is also the Bayes classification rule for the label set $\{\vec{X}_1, \dots, \vec{X}_k\}$.

Since the average Bayes accuracy is an upper-bound for the expected identification accuracy over any training-test split, it follows that ABA_k is also an upper bound for the expected CV-identification accuracy: that is,

$$\text{ABA}_k \geq \mathbf{E}[\text{TA}_{k,CV}] \quad (2.4)$$

for $\text{TA}_{k,CV}$ as defined in (1.1).

We use this fact in Chapters 4 and 5 to link identification accuracy to mutual information via the average Bayes accuracy.

Chapter 3

Extrapolating average accuracy

3.1 Introduction

In this chapter, we continue the discussion of Chapter 2, but focus specifically on the question of how to estimate the k -class average accuracy, $\text{AGA}_{k,r}$, based on data from a problem P_1 with $k_1 < k$ classes, and $r = r_1$ -repeat training data. This problem of *prediction extrapolation* is especially interesting for a number of applied problems.

- 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 will 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. 2008 obtained fMRI brain scans which record how a single subject's visual cortex responds to natural images. They wanted to know how well the brain signals could discriminate between different images. For a set of 1750 photographs, they constructed a classifier which achieved over 0.75 accuracy of classification. Based on exponential extrapolation, they estimate that it would take on the order of $10^{9.5}$ classes before the accuracy of the model drops below

0.10! A theory of performance extrapolation could be useful for the purpose of making such extrapolations in a more principled way.

- 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.

However, in order to formally develop a method for extrapolating performance, it is necessary to specify the model for how the small-scale problem is related to the larger-scale problem. The framework of randomized classification introduced in Chapter 2 is one possible choice for such a model: the small-scale problem and large-scale problem are linked by having classes drawn from the same population.

The rest of the chapter is organized as follows. We begin by continuing to analyze the average accuracy $\text{AGA}_{k,r}$, which results in an explicit formula for the average accuracy. The formula reveals that all of the information needed to compute the

average accuracy is contained in a one-dimensional function $\bar{D}(u)$, and therefore that estimation of the average accuracy can be accomplished via estimation of the unknown function $\bar{D}(u)$. This allows the development of a class of unbiased estimators of $\bar{D}(u)$, presented in section 3.3 given the assumption of a known parametric form for $\bar{D}(u)$. We analyze the performance of the estimator in both the well-specified and misspecified case. We demonstrate our method to a face-recognition example in section 3.4. Additionally, in Chapter 5 we compare the estimator to an alternative, information-theory based estimator in simulated and real-data examples.

3.2 Analysis of average risk

The result of our analysis is to expose the average accuracy $AGA_{k,r}$ 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 3.2.1 *Suppose π , $\{F_y\}_{y \in \mathcal{Y}}$ and marginal classifier \mathcal{F} satisfy the tie-breaking condition. Then, under the definitions (3.2) and (3.5), we have*

$$AGA_{k,r} = 1 - (k-1) \int \bar{D}(u) u^{k-2} du. \quad (3.1)$$

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} .

As we can see from Figure 2.4, the average accuracy 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^*} .

Our strategy is to analyze the average accuracy 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 accuracy* $\text{CondAcc}_k((y^*, \hat{F}_{y^*}), x^*)$ as

$$\text{CondAcc}_k((y^*, \hat{F}_{y^*}), x^*) = \Pr[\arg\max_{y \in \mathcal{S}} \mathcal{M}(\{\hat{F}_y\})(X^*) = Y^* | Y^* = y^*, X^* = x^*, \hat{F}_{Y^*} = \hat{F}_{y^*}].$$

Figure 3.1 illustrates the variables which are fixed under conditioning and the variables which are randomized. Compare to figure 2.4.

Without loss of generality, we can write the label subset $\mathcal{S} = \{Y^*, Y^{(1)}, \dots, Y^{(k-1)}\}$. Note that due to independence, $Y^{(1)}, \dots, Y^{(k-1)}$ 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* $Y^{(1)}, \dots, Y^{(k-1)}$ 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 $\hat{F}_{Y^{(1)}}, \dots, \hat{F}_{Y^{(k-1)}}$. This determines

$$\hat{Y} = \arg\max_{y \in \mathcal{S}} M_y(x^*)$$

and hence, whether or not the classification is correct for (x^*, y^*)

Compared to four randomization steps for the average risk, 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 accuracy, we begin by considering the *two-class* situation.

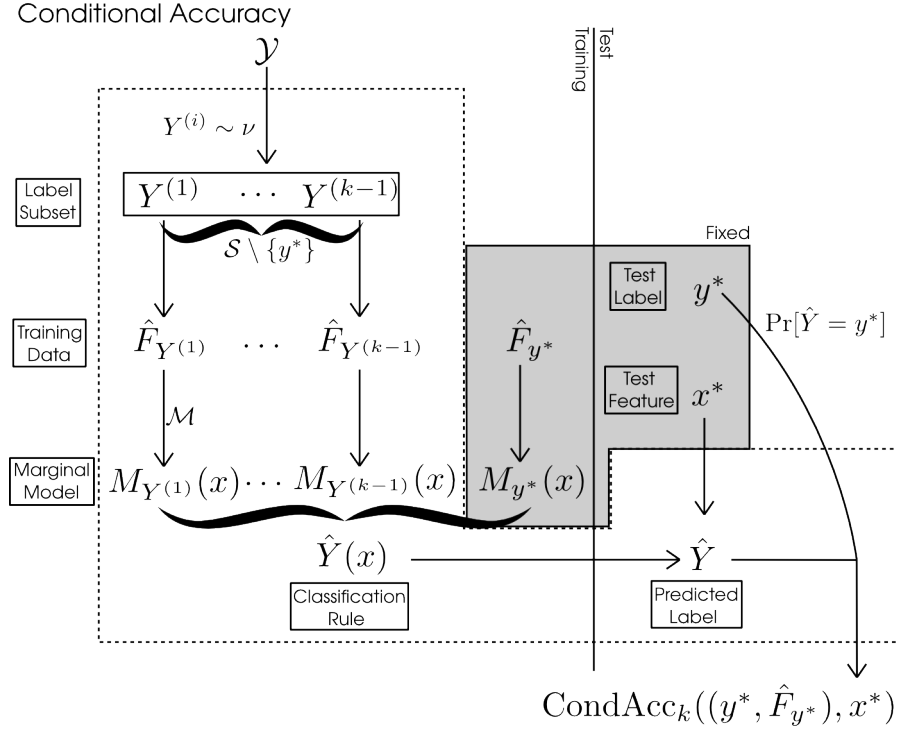


Figure 3.1: Conditional accuracy

In the two-class situation, we have a true label y^* , a training sample \hat{F}_{y^*} , and one incorrect label, Y . Define the *U-function* $U_{x^*}(y^*, \hat{F}_{y^*})$ as the conditional accuracy (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:

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

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

See also figure 3.2 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 the following Lemma.

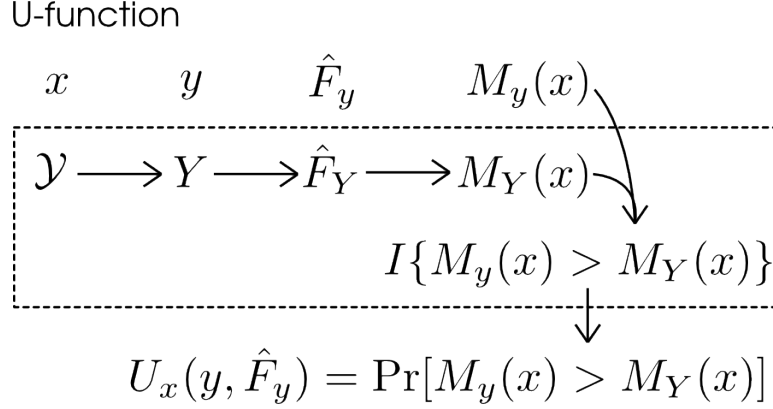


Figure 3.2: U-functions

Lemma 3.2.1 Suppose π , $\{F_y\}_{y \in \mathcal{Y}}$ and marginal classifier \mathcal{F} satisfy the tie-breaking condition. Take $x \in \mathcal{X}$. Defining $U_{y, \hat{F}_y}(x)$ as in (3.2), and defining the random variable U by

$$U = U_{Y, \hat{F}_Y}(x)$$

for $Y \sim \pi$, $\hat{F}_Y \sim \Pi_{Y,r}$, the distribution of U is uniform on $[0, 1]$, i.e.

$$\Pr[U \leq u] = \max\{u, 1\}.$$

Proof of Lemma.

Define the variable $Z = \mathcal{M}(\hat{F}_Y)(x)$ for $Y \sim \pi$. By the tie-breaking condition, Z has a continuous density on $[0, 1]$. Consider the survivor function of Z , $g(z) = \Pr[Z \geq z]$. From the definition (3.2), we see that

$$U = g(\mathcal{M}(\hat{F}_Y)(x)) = g(Z).$$

Now note that the survivor function of any continuous random variable, when applied to itself, is uniformly distributed. \square

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 $Y^{(1)}, \dots, Y^{(k-1)}$. Note that the

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

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

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

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

Therefore, in the general case, the conditional accuracy is

$$\text{CondAcc}_k((y^*, \hat{F}_{y^*}), x^*) = \Pr[U_{max} \leq u^*] = \int_0^{u^*} (k-1)u^{k-2}du.$$

Now the average accuracy can be obtained by integrating over the distribution of $U^* = U_{x^*}(y^*, \hat{F}_{y^*})$, which we state in the following proof of theorem 3.2.1.

Proof of Theorem 3.2.1. We have

$$\begin{aligned} \text{AGA}_{k,r} &= \mathbf{E}\left[\int_0^{U^*} (k-1)u^{k-2}du\right] \\ &= \mathbf{E}\left[\int_0^1 I\{u \leq U^*\} (k-1)u^{k-2}du\right] \\ &= (k-1) \int_0^1 \Pr[U^* \geq u] u^{k-2}du. \end{aligned}$$

Or equivalently,

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

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

$$\bar{D}(u) = \Pr[U_{x^*}(y^*, \hat{F}_{y^*}) \leq u]. \quad (3.5)$$

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 3.3, we will develop estimators of $\bar{D}(u)$ in order to estimate the k -class average risk.

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.

3.3 Estimation

Now we address the problem of estimating AGA_{k_2, r_1} from data. As we have seen from Theorem 3.2.1, the k -class average accuracy 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 (3.1) to obtain an estimate of $\text{AGA}_{k_2, r_{train}}$.

Having decided to estimate \bar{D} , there is then the question of what kind of model we should assume for \bar{D} . In this work, we assume that some parametric model¹ is available for \bar{D} .

¹While a nonparametric approach may be more ideal, we leave this to future work.

Let us assume the linear model

$$\bar{D}(u) = \sum_{\ell=1}^m \beta_{\ell} h_{\ell}(u), \quad (3.6)$$

where $h_{\ell}(u)$ are known basis functions, and β are the model parameters to be estimated. We can obtain *unbiased* estimation of $\text{AGA}_{k_2, r_{\text{train}}}$ via the unbiased estimates of k -class average risk obtained from (2.3).

If we plug in the assumed linear model (3.6) into the identity (3.1), then we get

$$1 - \text{AGA}_{k, r_{\text{train}}} = (k-1) \int \bar{D}(u) u^{k-2} du \quad (3.7)$$

$$= (k-1) \int_0^1 \sum_{\ell=1}^m \beta_{\ell} h_{\ell}(u) u^{k-2} du \quad (3.8)$$

$$= \sum_{\ell=1}^m \beta_{\ell} H_{\ell, k} \quad (3.9)$$

where

$$H_{\ell, k} = (k-1) \int_0^1 h_{\ell}(u) u^{k-2} du. \quad (3.10)$$

The constants $H_{\ell, k}$ are moments of the basis function h_{ℓ} : hence we call this method the *moment method*. Note that $H_{\ell, k}$ can be precomputed numerically for any $k \geq 2$.

Now, since the test accuracies TA_k are unbiased estimates of $\text{AGA}_{k, r_{\text{train}}}$, this implies that the regression estimate

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

is unbiased for β , under any choice of positive weights w_k . The estimate of AGA_{k_2, r_1} is similarly obtained from (3.9), via

$$\widehat{\text{AGA}_{k_2, r_1}} = 1 - \sum_{\ell=1}^m \hat{\beta}_{\ell} H_{\ell, k_2}. \quad (3.11)$$

3.4 Examples

3.4.1 Facial recognition example

From the “Labeled Faces in the Wild” dataset (Huang et al. 2007), we selected 1672 individuals with at least 2 face photos. We form a dataset consisting of photo-label pairs $(\vec{z}_j^{(i)}, y^{(i)})$ for $i = 1, \dots, 1672$ and $j = 1, 2$ by randomly selecting 2 face photos for each of the 1672 individuals.

We implement a face recognition system based on one nearest-neighbor and Open-Face (Amos, Ludwiczuk, and Satyanarayanan 2016) for feature extraction. For each photo \vec{z} , a 128-dimensional feature vector \vec{x} is obtained as follows.

1. The computer vision library DLib is used to detect landmarks in \vec{z} , and to apply a nonlinear transformation to align \vec{z} to a template.
2. The aligned photograph is downsampled to a 96×96 input, which is fed into a pre-trained deep convolutional neural network to obtain the 128-dimensional feature vector \vec{x} .

Therefore, we obtain feature-label pairs $(\vec{z}_j^{(i)}, y^{(i)})$ for $i = 1, \dots, 1672$ and $j = 1, 2$.

The recognition system then works as follows. Suppose we want to perform facial recognition on a subset of the individuals, $I \subset \{1, \dots, 1672\}$. Then, for all $i \in I$, we load one feature vector-label pair, into the system, $(\vec{x}_1^{(i)}, y^{(i)})$. In order to identify a new photo \vec{z}^* , we obtain the feature vector \vec{x}^* , and guess the label \hat{y} based on example with the minimal distance to \vec{x}^* ,

$$\hat{y} = y^{(i^*)}$$

where

$$i^* = \operatorname{argmin}_I d(\vec{x}, \vec{x}_1^{(i)}).$$

The test accuracy is assessed on the unused repeat for all individuals in I . Note that the assumptions of our estimation method are met in this example because one-nearest neighbor is a marginal classifier. One can define the margin functions as

$$(\mathcal{M}(\vec{x}))(\vec{x}^*) = -\|\vec{x} - \vec{x}^*\|^2.$$

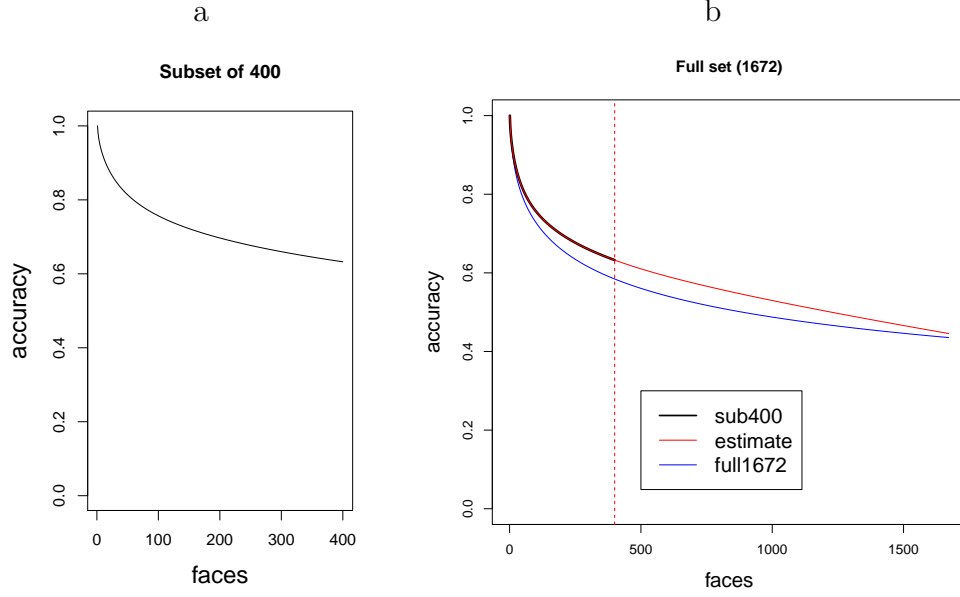


Figure 3.3: (a) The estimated average accuracy for $k = 2, \dots, 400$ given a dataset of 400 faces subsampled from Labeled Faces in the Wild. (b) Estimated average accuracy for $k > 400$ on the same dataset, compared to the ground truth (average k -class test accuracy using all 1672 classes).

However, k -nearest neighbor for $k > 1$ is not marginal.

While we can apply our extrapolation method to estimate the average accuracy for any number of faces k , it will not be possible to validate our estimates if we use the full dataset. Therefore, we take the average accuracies computed using the subsampling method (2.3) on the full dataset as a *ground truth* to compare to the average accuracy estimated from a *subsampled* dataset. Therefore, we simulate the problem of performance extrapolation from a database of K faces by subsampling a dataset of size K from the LFW dataset.

To do the performance extrapolation, we use a linear spline basis,

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

for $\ell = 1, \dots, m$. Here we take $m = 10000$. We model the function $\bar{D}(u)$ as a

non-negative linear combination of basis functions,

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

with $\beta_{\ell} \geq 0$. The resulting estimated generalization accuracies, computed using (3.1), are plotted in Figure (3.3) (b). As we already mentioned, to assess the quality of the estimated average accuracy, we compare them to the ‘ground truth’ accuracy curve obtained by using all 1672 examples to compute the the average test risk.

To get an idea of the accuracy and variance of the accuracy curve estimates for varying sample sizes K , we repeat this procedure multiple times for $K \in \{100, 200, 400, 800\}$. The results, again compared to the ground truth computed from the full data set, are illustrated in figure 3.4.

3.4.2 Telugu OCR example

While the previous example was a perfect fit to the assumptions of the framework, we also want to see how well the method works when some of the assumptions may be violated. Towards this end we apply performance extrapolation in an optical character recognition example (Achanta and Hastie 2015) for predicting the accuracy on 400 classes of Telugu characters from a subsample of size $K = 20$. We consider the use of five different classifiers: Naive Bayes, logistic regression, SVM, ϵ -nearest neighbors², and a deep convolutional neural network³. Of the five classifiers, only Naive Bayes satisfies the marginal classifier assumption. Again, we compare the result of our model to the ground truth obtained by using the full dataset.

The results are displayed in Table 3.1. To our surprise, the worst absolute difference between ground truth and estimated average accuracy was in the case of Naive bayes ($|\delta| = 0.257$) which satisfies the marginal property. All other classifiers had absolute errors less than 0.2. It is also interesting to note that, even though Naive Bayes and ϵ -nearest neighbors have the same test accuracy on the subset, that the

² k -nearest neighbors with $k = \epsilon n$ for fixed $\epsilon > 0$

³The network architecture is as follows: 48x48-4C3-MP2-6C3-8C3-MP2-32C3-50C3-MP2-200C3-SM.

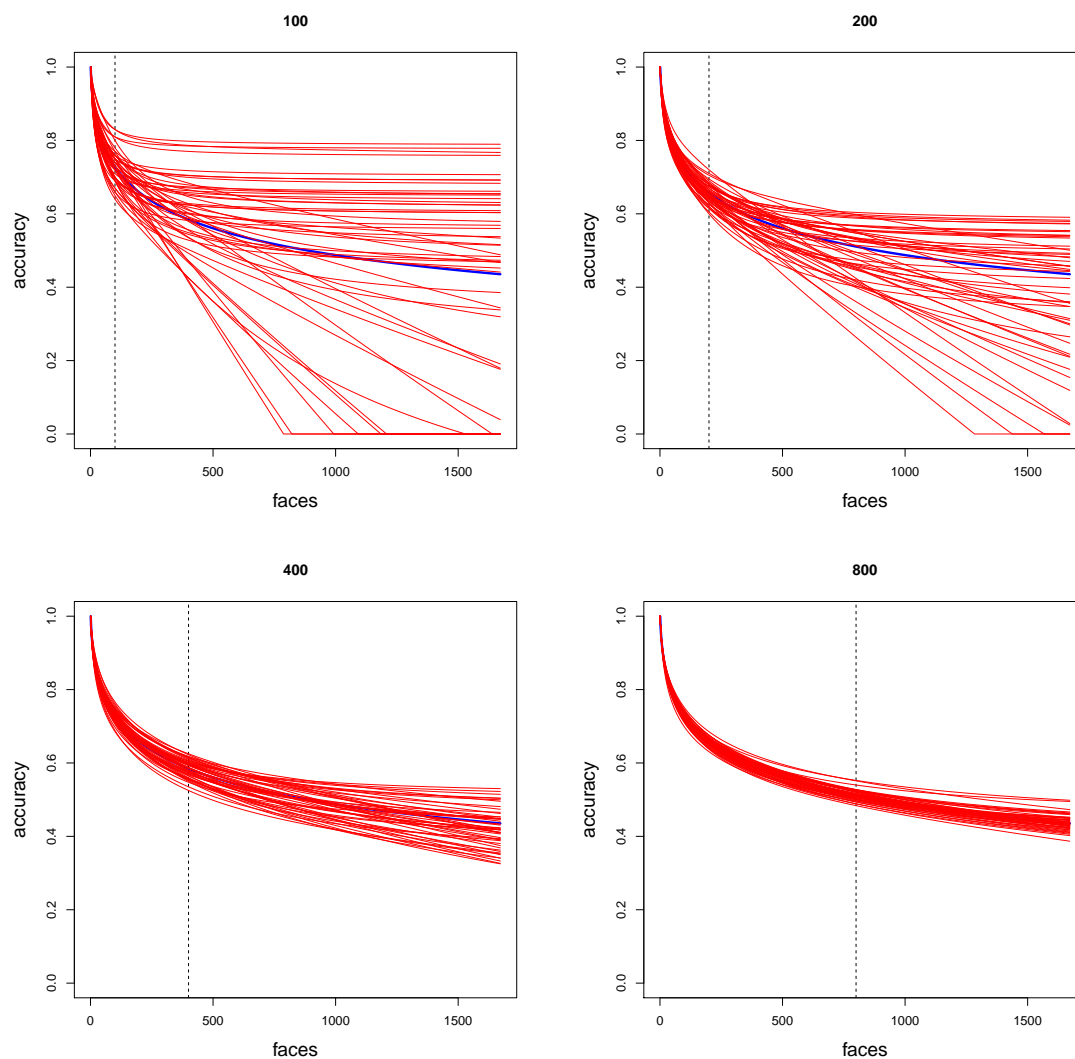


Figure 3.4: Estimated average accuracy using subsampled datasets of size k , compared to the ground truth (average k -class test accuracy using all 1672 classes).

Classifier	Test acc ⁽²⁰⁾	Test acc ⁽⁴⁰⁰⁾	$\hat{A}G A_{400}$
Naive Bayes	0.951	0.601	0.858
Logistic	0.922	0.711	0.812
SVM	0.860	0.545	0.687
ϵ -NN	0.951	0.591	0.410
Deep neural net	0.995	0.986	0.907

Table 3.1: Performance extrapolation: predicting the accuracy on 400 classes using data from 20 classes on a Telugu character dataset. $\epsilon = 0.002$ for ϵ -nearest neighbors.

predicted accuracy on 400 differs greatly between them (0.858 vs 0.400). Furthermore, the difference is in the right direction: Naive Bayes is predicted to work better on 400 classes than ϵ -nearest neighbors, which appears to be the case based on the 400-class test accuracy.

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

Chapter 4

Inference of mutual information

4.1 Motivation

As we discussed in the introduction, the mutual information $I(X; Y)$ provides one method for the supervised evaluation of representations. Recall that Shannon’s mutual information $I(X; Y)$ is fundamentally a measure of dependence between random variables X and Y , defined as

$$I(X; Y) = \int p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy.$$

Various properties of $I(\vec{g}(\vec{Z}); Y)$, such as sensitivity to nonlinear relationships, symmetry, and invariance to bijective transformations, make it ideal for quantifying the information between a representation of an input vector $\vec{g}(\vec{Z})$ and a (possibly vector-valued) response Y .

However, current methods estimating mutual information for high-dimensional data either require large and over-parameterized generative models, or work only for discrete responses. For continuous (X, Y) , one can tractably estimate mutual information by assuming a multivariate Gaussian model: however, this approach essentially assumes a linear relationship between the input and output, and hence fails to quantify nonlinear dependencies. Meanwhile, in the case of a discrete response Y , one can obtain a lower bound on the mutual information by using the confusion

matrix of a classifier. This is the most popular approach for estimating mutual information in neuroimaging studies, but suffers from known shortcomings (Gastpar, Gill, and Theunissen 2009, Quiñero and Panzeri 2009). The idea of linking classification performance to mutual information dates back to the beginnings of information theory: Shannon’s original motivation was to characterize the minimum achievable error probability of a noisy communication channel. More explicitly, Fano’s inequality (Fano 1961) provides a lower bound on mutual information in relation to the optimal prediction error, or Bayes error. Fano’s inequality can be further refined to obtain a tighter lower bound on mutual information (Tebbe and Dwyer 1968.) However, a shortcoming to all of these classification-based approaches is that if Y is not already discrete, the discretization of the response Y necessarily loses information.

In this chapter, we develop an analogue of Fano’s inequality for the *identification task*, rather than the classification task, which can therefore be applied to the case of a continuous response Y . In this way, we derive a new machine-learning based estimator of mutual information $I(X; Y)$ which can be applied without the need to discretize a continuous response.

As we saw in Chapter 2, the identification task is highly related to the randomized classification framework. Therefore, we analyse the identification risk by means of the k -class average Bayes accuracy, which provides an upper bound to identification accuracy. Our main theoretical contributions are (i) the derivation of a tight lower bound on mutual information as a function of k -class average Bayes accuracy, and (ii) the derivation of an asymptotic relationship between the K -class average Bayes accuracy and the mutual information. Our method therefore estimates the K -class Bayes identification accuracy to be translated into estimate of the mutual information.

Our method complements the collection of existing approaches for estimating mutual information in the sense that it allows the leveraging of different kinds of prior assumptions than ones previously considered in the literature. Our target application is when the relationship between the random variables (X, Y) is well-described by a parametric regression model $\mathbf{E}[Y|X = x] = f_\theta(x)$. An important special case is when a *sparse* relationship exists between the predictor and response. As we will demonstrate, exploiting sparsity assumptions can vastly improve the efficiency of

estimation in high-dimensional settings.

The rest of the chapter is organized as follows. Section 4.2 establishes the theoretical results linking average Bayes accuracy and mutual information, and section 4.3 describes our proposed identification-based estimator of mutual information based on the theory. We present one simulation example in section 4.4, but will we also see some real data applications in the next chapter.

4.2 Average Bayes accuracy and Mutual information

4.2.1 Problem formulation and result

Let \mathcal{P} denote the collection of all joint densities $p(x, y)$ on finite-dimensional Euclidean space. For $\iota \in [0, \infty)$ define $C_k(\iota)$ to be the largest k -class average Bayes error attained by any distribution $p(x, y)$ with mutual information not exceeding ι :

$$C_k(\iota) = \sup_{p \in \mathcal{P}: I[p(x, y)] \leq \iota} \text{ABA}_k[p(x, y)].$$

A priori, $C_k(\iota)$ exists since ABA_k is bounded between 0 and 1. Furthermore, C_k is nondecreasing since the domain of the supremum is monotonically increasing with ι .

It follows that for any density $p(x, y)$, we have

$$\text{ABA}_k[p(x, y)] \leq C_k(I[p(x, y)]).$$

Hence C_k provides an upper bound for average Bayes error in terms of mutual information.

Conversely we have

$$I[p(x, y)] \geq C_k^{-1}(\text{ABA}_k[p(x, y)])$$

so that C_k^{-1} provides a lower bound for mutual information in terms of average Bayes

error.

On the other hand, there is no nontrivial *lower* bound for average Bayes error in terms of mutual information, nor upper bound for mutual information in terms of average Bayes error, since

$$\inf_{p \in \mathcal{P}: I[p(x, y)] \leq \iota} \text{ABA}_k[p(x, y)] = \frac{1}{k}.$$

regardless of ι .

The goal of this work is to attempt to compute or approximate the functions C_k and C_k^{-1} .

In the following sections we determine the value of $C_k(\iota)$, leading to the following result.

Theorem 4.2.1 *For any $\iota > 0$, there exists $c_\iota \geq 0$ such that defining*

$$Q_c(t) = \frac{\exp[ct^{k-1}]}{\int_0^1 \exp[ct^{k-1}]},$$

we have

$$\int_0^1 Q_{c_\iota}(t) \log Q_{c_\iota}(t) dt = \iota.$$

Then,

$$C_k(\iota) = \int_0^1 Q_{c_\iota}(t) t^{k-1} dt.$$

We obtain this result by first reducing the problem to the case of densities with uniform marginals, then doing the optimization over the reduced space.

4.2.2 Reduction

Let $p(x, y)$ be a density supported on $\mathcal{X} \times \mathcal{Y}$, where \mathcal{X} is a subset of \mathbb{R}^{d_1} with a non-empty interior, and \mathcal{Y} is a subset of \mathbb{R}^{d_2} , also with non-empty interior, and such that $p(x)$ is uniform on \mathcal{X} and $p(y)$ is uniform on \mathcal{Y} .

Now let \mathcal{P}^{unif} denote the set of such distributions: in other words, \mathcal{P}^{unif} is the space of joint densities in Euclidean space with uniform marginals over the marginal

supports. In this section, we prove that

$$C_k(\iota) = \sup_{p \in \mathcal{P}: I[p(x, y)] \leq \iota} \text{ABA}_k[p(x, y)] = \sup_{p \in \mathcal{P}^{unif}: I[p(x, y)] \leq \iota} \text{ABA}_k[p(x, y)],$$

thus reducing the problem of optimizing over the space of all densities to the problem of optimizing over densities with uniform marginals.

Also define $\mathcal{P}^{bounded}$ to be the space of all densities $p(x, y)$ with finite-volume support. Since uniform distributions can only be defined over sets of finite volume, we have

$$\mathcal{P}^{unif} \subset \mathcal{P}^{bounded} \subset \mathcal{P}.$$

Therefore, it is necessary to first show that

$$\sup_{p \in \mathcal{P}: I[p(x, y)] \leq \iota} \text{ABA}_k[p(x, y)] = \sup_{p \in \mathcal{P}^{bounded}: I[p(x, y)] \leq \iota} \text{ABA}_k[p(x, y)].$$

This is accomplished via the following lemma.

Lemma 4.2.1 (*Truncation*). *Let $p(x, y)$ be a density on $\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$. For all $\epsilon > 0$, there exists a subset $\mathcal{X} \subset \mathbb{R}^{d_x}$ with finite volume with respect to d_x -dimensional Lebesgue measure, and a subset $\mathcal{Y} \subset \mathbb{R}^{d_y}$ with finite volume with respect to d_y -dimensional Lebesgue measure, such that defining*

$$\tilde{p}(x, y) = \frac{I\{(x, y) \in \mathcal{X} \times \mathcal{Y}\}}{\int_{\mathcal{X} \times \mathcal{Y}} p(x, y) dx dy} p(x, y),$$

we have

$$|I[p] - I[\tilde{p}]| < \epsilon$$

and

$$|\text{ABA}_k[p] - \text{ABA}_k[\tilde{p}]| < \epsilon.$$

Proof. Recall the definition of the Shannon entropy H :

$$H[p(x)] = - \int p(x) \log p(x) dx.$$

It is a well-known in information theory that

$$I[p(x, y)] = H[p(x)] + H[p(y)] - H[p(x, y)].$$

There exists a sequence $(\mathcal{X}_i, \mathcal{Y}_i)_{i=1}^\infty$ where $(\mathcal{X}_i)_{i=1}^\infty$ is an increasing sequence of finite-volume subsets of \mathbb{R}^{d_x} and $(\mathcal{Y}_i)_{i=1}^\infty$ is an increasing sequence of finite-volume subsets of \mathbb{R}^{d_y} , and $\lim_{i \rightarrow \infty} \mathcal{X}_i = \mathbb{R}^{d_x}$, $\lim_{i \rightarrow \infty} \mathcal{Y}_i = \mathbb{R}^{d_y}$. Define

$$\tilde{p}_i(x, y) = \frac{I\{(x, y) \in \mathcal{X}_i \times \mathcal{Y}_i\}}{\int_{\mathcal{X}_i \times \mathcal{Y}_i} p(x, y) dx dy} p(x, y)$$

Note that \tilde{p}_i gives the conditional distribution of (X, Y) conditional on $(X, Y) \in \mathcal{X}_i \times \mathcal{Y}_i$. Furthermore, it is convenient to define $\tilde{p}_\infty = p$. We can find some i_1 , such that for all $i \geq i_1$, we have

$$\begin{aligned} \left| \int_{x \notin \mathcal{X}_i} p(x) \log p(x) dx \right| &< \frac{\epsilon}{6} \\ \left| \int_{y \notin \mathcal{Y}_i} p(y) \log p(y) dy \right| &< \frac{\epsilon}{6} \\ \left| \int_{(x, y) \notin \mathcal{X}_i \times \mathcal{Y}_i} p(x, y) \log p(x, y) dx dy \right| &< \frac{\epsilon}{6} \end{aligned}$$

and also such that

$$-\log \left[\int_{x, y \in \mathcal{X}_i \times \mathcal{Y}_i} p(x, y) dx dy \right] < \frac{\epsilon}{2}$$

Then, it follows that

$$|I[p] - I[\tilde{p}_i]| < \epsilon$$

for all $i \geq i_1$.

Now we turn to the analysis of average Bayes error. Let f_i denote the Bayes k -class classifier for $\tilde{p}_i(x, y)$ and f_∞ the Bayes k -class classifier for $p(x, y)$: recall that by definition,

$$\text{ABA}_k[\tilde{p}_i] = \Pr_{\tilde{p}_i}[f_i(Y^{(1)}, \dots, Y^{(k)}, X) = Y^{(1)}]$$

where the probability is taken with respect to the joint distribution of $Y^{(1)}, \dots, Y^{(k)}$ drawn from π , and X drawn from the conditional distribution $F_{Y^{(1)}}$. Define

$$\epsilon_i = \Pr_p[(Y^{(1)}, \dots, Y^{(k)}, X) \notin \mathcal{Y}_i^k \times \mathcal{X}_i];$$

by continuity of probability we have $\lim_i \epsilon_i \rightarrow 0$. We claim that

$$|\text{ABA}_k[\tilde{p}_i] - \text{ABA}_k[p]| \leq \epsilon_i.$$

Given the claim, the proof is completed by finding $i > i_1$ such that $\epsilon_i < \epsilon$, and defining $\mathcal{X} = \mathcal{X}_i$, $\mathcal{Y} = \mathcal{Y}_i$.

Consider using f_i to obtain a classification rule for $p(x, y)$: define

$$\tilde{f}_i = \begin{cases} f_i(y^{(1)}, \dots, y^{(k)}, x) & \text{when } (y^{(1)}, \dots, y^{(k)}, x) \in \mathcal{X}_i^k \times \mathcal{Y}_i \\ 0 & \text{otherwise.} \end{cases}$$

We have

$$\begin{aligned} \text{ABA}_k[p] &= \sup_f \Pr_p[f(Y^{(1)}, \dots, Y^{(k)}, X) = Y^{(1)}] \\ &\geq \\ &= (1 - \epsilon_i) \Pr_p[f_i(Y^{(1)}, \dots, Y^{(k)}, X) = Y^{(1)} | (Y^{(1)}, \dots, Y^{(k)}, X) \in \mathcal{X}_i^k \times \mathcal{Y}_i] \\ &\quad + \epsilon_i \Pr_p[f_i(Y^{(1)}, \dots, Y^{(k)}, X) = Y^{(1)} | (Y^{(1)}, \dots, Y^{(k)}, X) \notin \mathcal{X}_i^k \times \mathcal{Y}_i] \\ &= (1 - \epsilon_i) \Pr_{\tilde{p}}[f_i(Y^{(1)}, \dots, Y^{(k)}, X) = Y^{(1)}] + \epsilon_i 0 \\ &= (1 - \epsilon_i) \text{ABA}_k[\tilde{p}_i] \geq \text{ABA}_k[\tilde{p}_i] - \epsilon_i. \end{aligned}$$

In other words, when \tilde{p}_i is close to p , the Bayes classification rule for \tilde{p}_i obtains close to the Bayes rate when the data is generated under p .

Now consider the reverse scenario of using f_p to perform classification under \tilde{p}_i . This is equivalent to generating data under $p(x, y)$, performing classification using f , then only evaluating classification accuracy conditional on $(Y^{(1)}, \dots, Y^{(k)}, X) \in$

$\mathcal{X}_i^k \times \mathcal{Y}_i$. Therefore,

$$\begin{aligned}
\text{ABA}_k[\tilde{p}_i] &= \sup_f \Pr_{\tilde{p}_i}[f(Y^{(1)}, \dots, Y^{(k)}, X) = Y^{(1)}] \\
&\geq \Pr_{\tilde{p}_i}[f_i(Y^{(1)}, \dots, Y^{(k)}, X) = Y^{(1)}] \\
&= \Pr_p[f_i(Y^{(1)}, \dots, Y^{(k)}, X) = Y^{(1)} | (Y^{(1)}, \dots, Y^{(k)}, X) \in \mathcal{X}_i^k \times \mathcal{Y}_i] \\
&= \frac{1}{1 - \epsilon_i} \Pr_p[I\{(X^{(1)}, \dots, X^{(k)}, Y) \in \mathcal{X}_i^k \times \mathcal{Y}_i\} \text{ and } f_i(Y^{(1)}, \dots, Y^{(k)}, X) = Y^{(1)}] \\
&\geq \frac{1}{1 - \epsilon_i} \left(1 - \Pr_p[I\{(Y^{(1)}, \dots, Y^{(k)}, X) \notin \mathcal{X}_i^k \times \mathcal{Y}_i\}] - \Pr_p[f_p(Y^{(1)}, \dots, Y^{(k)}, X) \neq Y^{(1)}] \right) \\
&= \frac{\text{ABA}_k[p] - \epsilon_i}{1 - \epsilon_i} \geq \text{ABA}_k[p] - \epsilon_i.
\end{aligned}$$

In other words, when \tilde{p}_i is close to p , the Bayes classification rule for p obtains close to the Bayes rate when the data is generated under \tilde{p}_i .

Combining the two directions gives $|\text{ABA}_k[\tilde{p}_i] - \text{ABA}_k[p]| \leq \epsilon_i$, as claimed. \square

One can go from bounded-volume sets to uniform distributions by adding auxillary variables. To illustrate the intuition, consider a density $p(x)$ on a set of bounded volume, \mathcal{X} . Introduce a variable W such that conditional on $X = x$, we have w uniform on $[0, p(x)]$. It follows that the joint density $p(x, w) = 1$ and is supported on a set $\mathcal{X}' = \mathcal{X} \times [0, \infty]$. Furthermore, \mathcal{X}' is of bounded volume (in fact, of volume 1) since

$$\int_{\mathcal{X}'} dx = \int_{\mathcal{X}'} p(x, w) dx = 1.$$

Therefore, to accomplish the reduction from \mathcal{P} to \mathcal{P}^{unif} , we start with a density $p(x, y) \in \mathcal{P}$, and using Lemma 4.2.1, find a suitable finite-volume truncation $\tilde{p}(x, y)$. Finally, we introduce auxillary variables w and z so that the expanded joint distribution $p(x, w, y, z)$ has uniform marginals $p(x, w)$ and $p(y, z)$. However, we still need to check that the introduction of auxillary variables preserves the mutual information and average Bayes error; this is the content of the next lemma.

Lemma 4.2.2 *Suppose X, Y, W, Z are continuous random variables, and that $W \perp$*

$Y|Z$, $Z \perp X|Y$, and $W \perp Z|(X, Y)$. Then,

$$I[p(x, y)] = I[p((x, w), (y, z))]$$

Proof. Due to conditional independence relationships, we have

$$p((x, w), (y, z)) = p(x, y)p(w|x)p(z|y).$$

It follows that

$$\begin{aligned} I[p((x, w), (y, z))] &= \int dx dw dy dz \, p(x, y)p(w|x)p(z|w) \log \frac{p((x, w), (y, z))}{p(x, w)p(y, z)} \\ &= \int dx dw dy dz \, p(x, y)p(w|x)p(z|w) \log \frac{p(x, y)p(w|x)p(z|y)}{p(x)p(y)p(w|x)p(z|y)} \\ &= \int dx dw dy dz \, p(x, y)p(w|x)p(z|w) \log \frac{p(x, y)}{p(x)p(y)} \\ &= \int dx dy \, p(x, y) \log \frac{p(x, y)}{p(x)p(y)} = I[p(x, y)]. \end{aligned}$$

Also,

$$\begin{aligned} \text{ABA}_k[p((x, w), (y, z))] &= \int \left[\prod_{i=1}^k p(x_i, w_i) dx_i dw_i \right] \int dy dz \, \max_i p(y, z|x_i, w_i). \\ &= \int \left[\prod_{i=1}^k p(x_i, w_i) dx_i dw_i \right] \int dy \, \max_i p(y|x_i) \int dz \, p(z|y). \\ &= \int \left[\prod_{i=1}^k p(x_i) dx_i \right] \left[\prod_{i=1}^k \int dw_i p(w_i|x_i) \right] \int dy \, \max_i p(y|x_i) \\ &= \text{ABA}_k[p(x, y)]. \end{aligned}$$

□

Combining these lemmas gives the needed reduction, given by the following theorem.

Theorem 4.2.2 (*Reduction.*)

$$\sup_{p \in \mathcal{P}: I[p(x, y)] \leq \iota} \text{ABA}_k[p(x, y)] = \sup_{p \in \mathcal{P}^{unif}: I[p(x, y)] \leq \iota} \text{ABA}_k[p(x, y)].$$

The proof is trivial given the previous two lemmas.

4.2.3 Proof of theorem

Proof of theorem 4.2.1

Using Theorem 4.2.2, we have

$$C_k(\iota) = \sup_{p \in \mathcal{P}^{unif}: I[p(x, y)] \leq \iota} \text{ABA}_k[p(x, y)].$$

Define $\sigma(\iota) = \int_0^1 Q_{c_\iota}(t) t^{k-1} dt$: our goal is to establish that $C_k(\iota) = \sigma(\iota)$. Note that $\sigma(\iota)$ is the same function which appears in Lemma A.1.5 and the same bound as established in Lemma A.1.4.

Define the density $p_\iota(x, y)$ where

$$p_\iota(x, y) = \begin{cases} g_\iota(y - x) & \text{for } x \geq y \\ g_\iota(1 + y - x) & \text{for } x < y \end{cases}$$

where

$$g_\iota(x) = \frac{d}{dx} G_\iota(x)$$

and G_ι is the inverse of Q_c .

One can verify that $I[p_\iota] = \iota$, and

$$\text{ABA}_k[p] = \int_0^1 Q_{c_\iota}(t) t^{k-1} dt.$$

This establishes that

$$C_k(\iota) \geq \int_0^1 Q_{c_\iota}(t) t^{k-1} dt.$$

It remains to show that for all $p \in \mathcal{P}^{unif}$ with $I[p] \leq \iota$, that $\text{ABA}_k[p] \leq \text{ABA}_k[p_\iota]$.

Take $p \in \mathcal{P}^{unif}$ such that $I[p] \leq \iota$. Letting $Y^{(1)}, \dots, Y^{(k)} \sim \text{Unif}[0, 1]$, and $X \sim \text{Unif}[0, 1]$ define $Z_i(x) = p(x|Y_i)$. We have $\mathbf{E}(Z_1(x))$ and,

$$I[p(x, y)] = \mathbf{E}(Z_1(X) \log Z_1(X))$$

while

$$\text{ABA}_k[p(x, y)] = k^{-1} \mathbf{E}(\max_i Z_i(X)).$$

Letting G_x be the distribution of $Z(x)$, we have

$$E[G_x] = 1$$

$$I[p(x, y)] = \mathbf{E}(I[G_X])$$

$$\text{ABA}_k[p(x, y)] = \mathbf{E}(\psi_k[G_X])$$

where the expectation is taken over $X \sim \text{Unif}[0, 1]$ and where $E[G]$, $I[G]$, and $\psi_k[G]$ are defined as in Lemma A.1.4.

Define the random variable $J = I[G_Y]$. We have

$$\begin{aligned} \text{ABA}_k[p(x, y)] &= \mathbf{E}(\psi_k[G_X]) \\ &= \int_0^1 \psi_k[G_X] dy \\ &\leq \int_0^1 \left(\sup_{G: I[G] \leq I[G_x]} \psi_k[G] \right) dy \\ &= \int_0^1 \sigma(I[G_x]) dy = \mathbf{E}[\sigma(J)]. \end{aligned}$$

Now, since f is concave by Lemma A.1.5, we can apply Jensen's inequality to conclude that

$$\text{ABA}_k[p(x, y)] = \mathbf{E}[\sigma(J)] \leq \sigma(\mathbf{E}[J]) = \sigma(\iota),$$

which completes the proof. \square

4.3 Estimation

Now let us return to the problem of estimating the mutual information $I(X; Y)$ based on observations $(x_i, y_i)_{i=1}^n$ drawn iid from the joint distribution.

We would like to use the cross-validated identification accuracy (1.1) as a basis for estimating the mutual information. How this works is as follows.

Define the k -class average identification accuracy as the expected value of the cross-validated identification accuracy (1.1),

$$\text{AIA}_k \stackrel{\text{def}}{=} \mathbf{E}[\text{TA}_{k,CV}].$$

As we argued in section 2.4.7, the k -class average identification accuracy is a lower bound for the k -class average Bayes accuracy,

$$\text{AIA}_k \leq \text{ABA}_k.$$

Finally, applying the theorem 4.2.1, we have

$$I(X; Y) \geq C_k^{-1}(\text{ABA}_k) \geq C_k^{-1}(\text{AIA}_k).$$

Replacing AIA_k with its estimate, $\text{TA}_{k,CV}$, yields the estimator

$$\hat{I}_{\text{Ident}}(X; Y) \stackrel{\text{def}}{=} C_k^{-1}(\text{TA}_{k,CV})$$

Since we use a lower bounding theorem to obtain the estimate, the estimated mutual information tends to be an underestimate of the true result. Therefore, we get a better estimate if we choose a regression model which maximizes the identification accuracy. For example, when it is known that a sparse linear relationship is a good fit for the regression function $\mathbf{E}[Y|X = x]$, then it would be advised to use sparse linear regression models, such as elastic net (Zou and Hastie 2005), to obtain the identification accuracy.

4.4 Simulation

In order to demonstrate the advantages of our estimator, we ran a simple simulation study under relatively ideal conditions. Our simulation model was a sparse multivariate linear model of the form

$$(Y_1, Y_2) = (X_1, X_2)^T B + \epsilon$$

where B is a randomly generated coefficient matrix. Then we added extra noise dimensions X_3, X_4, \dots, X_p for p ranging up to 600. For data with sample size $n = 1000$, we compared the estimates of mutual information $I(X; Y)$ obtained using the nearest-neighbor estimator (Mnatsakanov et al. 2008, implemented in **FNN**) with our method, using both ordinary least-squares and LASSO regression (using cross-validation to choose λ , as implemented in Friedman, Hastie, and Tibshirani 2010).

The results of the simulation are displayed in Figure 4.1. We see that for small p , the nearest-neighbor method and our method, for both OLS and LASSO perform similarly. As p increases to 10, the nearest neighbor estimator rapidly deteriorates, while OLS and LASSO-based estimators remain at similar levels. For p larger than 100, the LASSO-based estimator begins to outperform the OLS-based estimator, as it better exploits sparsity. We see that the LASSO-based estimator is highly robust to addition of noise predictors up to $p = 600$.

We will see additional examples in the next chapter, including a simulation example involving a dense model.

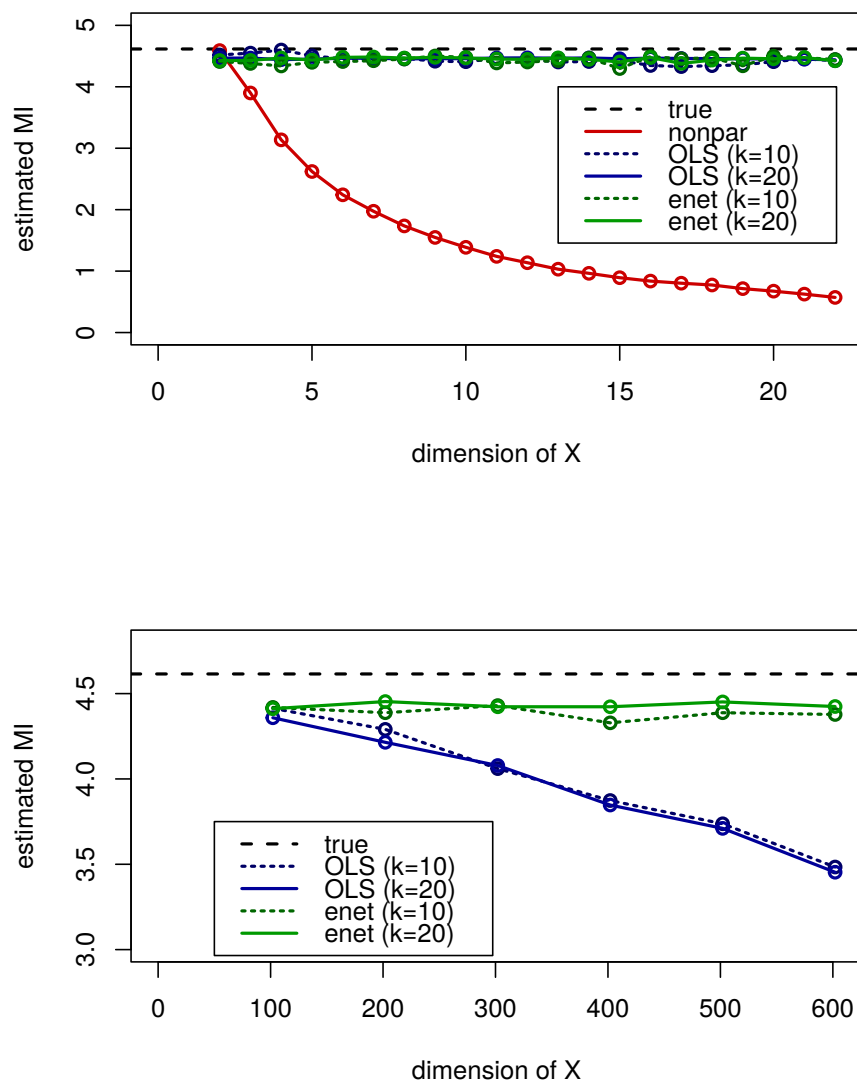


Figure 4.1: Estimation of mutual information in simulated example.

Chapter 5

High-dimensional inference of mutual information

5.1 Motivation

In the previous chapter, we saw that the identification accuracy of predicting Y given X can be used to yield a lower bound on $I(X; Y)$. This relied on the inequality relating k -class average Bayes accuracy and mutual information. However, because in some cases the mutual information may be much higher than the lower bound, the estimator \hat{I}_{Ident} may remain an underestimate of mutual information, even under favorable sample size conditions. In other words, the estimator \hat{I}_{Ident} is not consistent. On the other hand, the theory behind \hat{I}_{Ident} relies on extremely weak assumptions. This makes the estimator quite generally applicable, but on the other hand, much better estimators may be available in certain applications where much stronger assumptions can be made.

In this chapter we develop yet another estimator of mutual information based on identification accuracy, but this time making relatively strong assumptions relating to the dimensionality of the data and the dependence structure of the components. As we will spell out in more detail in section 5.2.1, we assume a setting where both (X, Y) are high-dimensional, and where the true mutual information $I(X; Y)$ is relatively low. Furthermore, we require the components of X to have “low dependence,” in a

certain sense.

We expect that these assumptions are well-matched to applications such as fMRI data, where the effective dimensionality of the data is large, but where the noise level limits the mutual information between X and Y .

The estimator of mutual information we develop in this chapter is based on exploiting a universality property that arises in high-dimensions. This universality phenomenon allows us to establish a relationship between the mutual information $I(X; Y)$ and the k -class average Bayes accuracy, ABA_k . In short, we will identify a function π_k (which depends on k),

$$\text{ABA}_k \approx \pi_k(\sqrt{2I(X; Y)}) \quad (5.1)$$

and that this approximation becomes accurate under a limit where $I(X; Y)$ is small relative to the dimensionality of X , and under the condition that the components of X are approximately independent. The function π_k is given by

$$\pi_k(c) = \int_{\mathbb{R}} \phi(z - c) \Phi(z)^{k-1} dz.$$

This formula is not new to the information theory literature: it appears as the error rate of an orthogonal constellation (Cioffi 2014). What is surprising is that the same formula can be used to approximate the error rate in much more general class of classification problems¹—this is precisely the universality result which provides the basis for our proposed estimator.

Figure 5.1 displays the plot of π_k for several values of k . For all values of k , $\pi_k(\mu)$ is monotonically increasing in μ , and tends to 1 as $\mu \rightarrow \infty$, which is what we expect since if $I(X; Y)$ is large, then we should be able to achieve perfect accuracy. Another intuitive fact is that $\pi_k(0) = \frac{1}{k}$, since after all, an uninformative response cannot lead to above-chance classification accuracy.

¹An intuitive explanation for this fact is that points from any high-dimensional distribution lie in an orthogonal configuration with high probability.

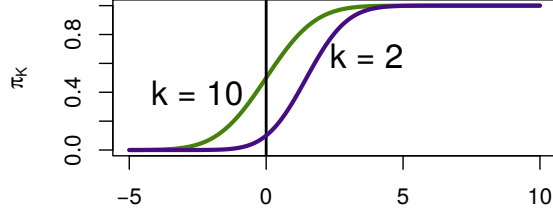


Figure 5.1: Left: The function $\pi_k(\mu)$ for $k = \{2, 10\}$.

The estimator we propose is

$$\hat{I}_{HD} = \frac{1}{2}(\pi_k^{-1}(\text{ABA}_k))^2, \quad (5.2)$$

obtained by inverting the relation (5.1), then substituting an estimate of the identification accuracy TA_k for the ABA_k .

5.2 Theory

5.2.1 Assumptions

The theory applies to a high-dimensional limit where $I(X; Y)$ tends to a constant.

- A1. $\lim_{d \rightarrow \infty} I(X^{[d]}; Y^{[d]}) = \iota < \infty$.
- A2. There exists a sequence of scaling constants $a_{ij}^{[d]}$ and $b_{ij}^{[d]}$ such that the random vector $(a_{ij} \ell_{ij}^{[d]} + b_{ij}^{[d]})_{i,j=1,\dots,k}$ converges in distribution to a multivariate normal distribution, where $\ell_{ij} = \log p(y^{(i)} | x^{(i)})$ for independent $y^{(i)} \sim p(y | x^{(i)})$.
- A3. Define

$$u^{[d]}(x, y) = \log p^{[d]}(x, y) - \log p^{[d]}(x) - \log p^{[d]}(y).$$

There exists a sequence of scaling constants $a^{[d]}$, $b^{[d]}$ such that

$$a^{[d]} u^{[d]}(X^{(1)}, Y^{(2)}) + b^{[d]}$$

converges in distribution to a univariate normal distribution.

A4. For all $i \neq k$,

$$\lim_{d \rightarrow \infty} \text{Cov}[u^{[d]}(X^{(i)}, Y^{(j)}), u^{[d]}(X^{(k)}, Y^{(j)})] = 0.$$

Assumptions A1-A4 are satisfied in a variety of natural models. One example is a multivariate Gaussian sequence model where $X \sim N(0, \Sigma_d)$ and $Y = X + E$ with $E \sim N(0, \Sigma_e)$, where Σ_d and Σ_e are $d \times d$ covariance matrices, and where X and E are independent. Then, if $d\Sigma_d$ and Σ_e have limiting spectra H and G respectively, the joint densities $p(x, y)$ for $d = 1, \dots$, satisfy assumptions A1 - A4. Another example is the multivariate logistic model, which we describe in section 3. We further discuss the rationale behind A1-A4 in the supplement, along with the detailed proof.

5.2.2 Limiting universality

We obtain the universality result in two steps. First, we link the average Bayes error to the moments of some statistics Z_i . Secondly, we use Taylor approximation in order to express $I(X; Y)$ in terms of the moments of Z_i . Connecting these two pieces yields the formula (5.1).

Let us start by rewriting the average Bayes error:

$$\text{ABA}_k = \Pr[p(X|Y^{(1)}) > \max_{j=2}^k p(X|Y^{(j)})].$$

where the expectation is taken over the joint distribution $Y^{(1)}, \dots, Y^{(k)} \sim \pi$ and $X \sim F_{Y^{(1)}}$. Defining the statistic $Z_i = \log p(X|Y^{(i)}) - \log p(X|Y^{(1)})$, we obtain $\text{ABA}_k = \Pr[\max_{j=2}^k Z_j < 0]$. The key assumption we need is that Z_2, \dots, Z_k are asymptotically multivariate normal. If so, the following lemma allows us to obtain a formula for the average Bayes accuracy.

Lemma 1. *Suppose $(\zeta_1, \zeta_2, \dots, \zeta_\ell)$ are jointly multivariate normal, with $\mathbf{E}[\zeta_i] = -\alpha$, $\text{Var}(\zeta_i) = \beta \geq 0$, $\text{Cov}(\zeta_i, \zeta_j) = \gamma$ for $i = 1, \dots, \ell$ and $j = 1, \dots, \ell$ with $i \neq j$.*

Then, letting

$$\mu = \frac{\mathbf{E}[-\zeta_1]}{\sqrt{\frac{1}{2} \text{Var}(\zeta_1 - \zeta_2)}} = \frac{\alpha}{\sqrt{\beta - \gamma}},$$

$$\nu^2 = \frac{\text{Cov}(\zeta_1, \zeta_2)}{\frac{1}{2} \text{Var}(\zeta_1 - \zeta_2)} = \frac{\gamma}{\beta - \gamma},$$

we have

$$\begin{aligned} \Pr[0 > \max_{i=1}^k \zeta_i] &= \Pr[W > M_{k-1}] \\ &= \int \frac{1}{\sqrt{2\pi\nu^2}} e^{-\frac{(w-\mu)^2}{2\nu^2}} \Phi(w)^{k-1} dw, \end{aligned}$$

where $W \sim N(\mu, \nu^2)$ and M_{k-1} is the maximum of $k-1$ independent standard normal variates, which are independent of W .

Remark. To see why the assumption that Z_2, \dots, Z_k are multivariate normal might be justified, suppose that X and Y have the same dimensionality d , and that joint density factorizes as

$$p(x, y) = \prod_{i=1}^d p_i(x_i, y_i)$$

where $x_i^{(j)}, y_i$ are the i th scalar components of the vectors x and y . Then,

$$Z_i = \sum_{m=1}^d \log p_m(x_m | y_m^{(i)}) - \log p_m(x_m | y_m^{(1)}).$$

The d terms $\log p_m(x_m | y_m^{(i)}) - \log p_m(x_m | y_m^{(1)})$ are independent across the indices m , but dependent between the $i = 2, \dots, k$. Therefore, the multivariate central limit theorem can be applied to conclude that the vector (Z_2, \dots, Z_k) can be scaled to converge to a multivariate normal distribution. While the componentwise independence condition is not a realistic assumption, the key property of multivariate normality of (Z_2, \dots, Z_k) holds under more general conditions, and appears reasonable in practice.

Proof. We can construct independent normal variates G_0, G_1, \dots, G_k such that

$$G_0 \sim N(0, \gamma)$$

$$G_i \sim N(0, \beta - \gamma) \text{ for } i = 1, \dots, \ell$$

such that

$$\zeta_i = -\alpha + G_0 + G_i \text{ for } i = 1, \dots, \ell$$

Hence

$$\begin{aligned} \Pr[0 > \max_{i=1}^k \zeta_i] &= \Pr[\min_{i=1}^k G_0 + G_i - \alpha > 0] \\ &= \Pr[\min_{i=1}^k G_i > \alpha - G_0] \\ &= \Pr[\min_{i=1}^k \frac{G_i}{\sqrt{\beta - \gamma}} > \frac{\alpha - G_0}{\sqrt{\beta - \gamma}}]. \end{aligned}$$

Since $\frac{G_i}{\sqrt{\beta - \gamma}}$ are iid standard normal variates, and since $-\frac{\alpha - G_0}{\sqrt{\beta - \gamma}} \sim N(\mu, \nu^2)$ for μ and ν^2 given in the statement of the Lemma, the proof is completed via a straightforward computation. \square

It remains to link the moments of Z_i to $I(X; Y)$. This is accomplished by approximating the logarithmic term by the Taylor expansion

$$\log \frac{p(x, y)}{p(x)p(y)} \approx \frac{p(x, y) - p(x)p(y)}{p(x)p(y)} - \left(\frac{p(x, y) - p(x)p(y)}{p(x)p(y)} \right)^2 + \dots$$

A number of assumptions are needed to ensure that needed approximations are sufficiently accurate; and additionally, in order to apply the central limit theorem, we need to consider a *limiting sequence* of problems with increasing dimensionality. We now state a theorem.

Theorem 1. *Let $p^{[d]}(x, y)$ be a sequence of joint densities for $d = 1, 2, \dots$. Further assume conditions A1 - A4. Then for ABA_k as defined above, we have*

$$\lim_{d \rightarrow \infty} ABA_k = \pi_k(\sqrt{2\iota})$$

where

$$\pi_k(c) = \int_{\mathbb{R}} \phi(z - c) \Phi(z)^{k-1} dz$$

where ϕ and Φ are the standard normal density function and cumulative distribution

function, respectively.

Proof.

For $i = 2, \dots, k$, define

$$Z_i = \log p(X|Y^{(i)}) - \log p(X|Y^{(1)}).$$

Then, we claim that $\vec{Z} = (Z_2, \dots, Z_k)$ converges in distribution to

$$\vec{Z} \sim N \left(-2\iota, \begin{bmatrix} 4\iota & 2\iota & \cdots & 2\iota \\ 2\iota & 4\iota & \cdots & 2\iota \\ \vdots & \vdots & \ddots & \vdots \\ 2\iota & 2\iota & \cdots & 4\iota \end{bmatrix} \right).$$

Combining the claim with the lemma (stated below this proof) yields the desired result.

To prove the claim, it suffices to derive the limiting moments

$$\mathbf{E}[Z_i] \rightarrow -2\iota,$$

$$\text{Var}[Z_i] \rightarrow 4\iota,$$

$$\text{Cov}[Z_i, Z_j] \rightarrow 2\iota,$$

for $i \neq j$, since then assumption A2 implies the existence of a multivariate normal limiting distribution with the given moments.

Before deriving the limiting moments, note the following identities. Let $Y' = Y^{(2)}$.

$$\mathbf{E}[e^{u(X, Y')}] = \int p(x)p(y)e^{u(x, y)} dx dy = \int p(x, y) dx dy = 1.$$

Therefore, from assumption A3 and the formula for gaussian exponential moments, we have

$$\lim_{d \rightarrow \infty} \mathbf{E}[u(X, Y')] - \frac{1}{2} \text{Var}[u(X, Y')] = 0.$$

Let $\sigma^2 = \lim_{d \rightarrow \infty} \text{Var}[u(X, Y')]$. Meanwhile, by applying assumption A2,

$$\begin{aligned}
 \lim_{d \rightarrow \infty} I(X; Y) &= \lim_{d \rightarrow \infty} \int p(x, y) u(x, y) dx dy = \lim_{d \rightarrow \infty} \int p(x) p(y) e^{u(x, y)} u(x, y) dx dy \\
 &= \lim_{d \rightarrow \infty} \mathbf{E}[e^{u(X, Y')} u(X, Y')] \\
 &= \int_{\mathbb{R}} e^z z \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z+\sigma^2/2)^2}{2\sigma^2}} dz \quad (\text{applying A2}) \\
 &= \int_{\mathbb{R}} z \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-\sigma^2/2)^2}{2\sigma^2}} dz \\
 &= \frac{1}{2} \sigma^2.
 \end{aligned}$$

Therefore,

$$\sigma^2 = 2\iota,$$

and

$$\lim_{d \rightarrow \infty} \mathbf{E}[u(X', Y)] = -\iota.$$

Once again by applying A2, we get

$$\begin{aligned}
 \lim_{d \rightarrow \infty} \text{Var}[u(X, Y)] &= \lim_{d \rightarrow \infty} \int (u(x, y) - \iota)^2 p(x, y) dx dy \\
 &= \lim_{d \rightarrow \infty} \int (u(x, y) - \iota)^2 e^{u(x, y)} p(x) p(y) dx dy \\
 &= \lim_{d \rightarrow \infty} \mathbf{E}[(u(X, Y') - \iota)^2 e^{u(X, Y')}] \\
 &= \int (z - \iota)^2 e^z \frac{1}{\sqrt{4\pi\iota}} e^{-\frac{(z+\iota)^2}{4\iota}} dz \quad (\text{applying A2}) \\
 &= \int (z - \iota)^2 \frac{1}{\sqrt{4\pi\iota}} e^{-\frac{(z-\iota)^2}{4\iota}} dz \\
 &= 2\iota.
 \end{aligned}$$

We now proceed to derive the limiting moments. We have

$$\begin{aligned}
 \lim_{d \rightarrow \infty} \mathbf{E}[Z] &= \lim_{d \rightarrow \infty} \mathbf{E}[\log p(X|Y') - \log p(X|Y)] \\
 &= \lim_{d \rightarrow \infty} \mathbf{E}[u(X, Y') - u(X, Y)] = -2\iota.
 \end{aligned}$$

Also,

$$\begin{aligned}
 \lim_{d \rightarrow \infty} \text{Var}[Z] &= \lim_{d \rightarrow \infty} \text{Var}[u(X, Y') - u(X, Y)] \\
 &= \lim_{d \rightarrow \infty} \text{Var}[u(X, Y')] + \text{Var}[u(X, Y)] \quad (\text{using assumption A4}) \\
 &= 4\iota,
 \end{aligned}$$

and similarly

$$\begin{aligned}
 \lim_{d \rightarrow \infty} \text{Cov}[Z_i, Z_j] &= \lim_{d \rightarrow \infty} \text{Var}[u(X, Y)] \quad (\text{using assumption A4}) \\
 &= 2\iota.
 \end{aligned}$$

This concludes the proof. \square .

5.3 Examples

We demonstrate the application of our proposed estimators, \hat{I}_{Ident} , which was introduced in the previous chapter, and the estimator \hat{I}_{HD} based on high-dimensional asymptotics, as defined in (5.2).

We compare our methods to a number of existing methods for mutual information. For any k -class classification task, define the $k \times k$ *confusion matrix* M as the matrix whose i, j counts how many times a output in the i th class was classified to the j th class.

The estimator \hat{I}_{Fano} is based on Fano's inequality, which reads

$$H(Z|Y) \leq H(1 - \text{BA}) + (1 - \text{BA}) \log ||\mathcal{Z}| - 1|$$

where BA is Bayes accuracy, and $H(e)$ is the entropy of a Bernoulli random variable with probability e . Replacing $H(Z|Y)$ with $H(X|Y)$ and replacing BA with the test accuracy TA, we get the estimator

$$\hat{I}_{Fano} = \log(K) - (1 - \text{TA})\log(K - 1) + (1 - \text{TA})\log(1 - \text{TA}) + \text{TA}\log(\text{TA}).$$

Meanwhile, the confusion matrix estimator computes

$$\hat{I}_{CM} = \frac{1}{k^2} \sum_{i=1}^k \sum_{j=1}^k \log \frac{M_{ij}}{r/k},$$

which is the empirical mutual information of the discrete joint distribution $(Z, f(Y))$.

The estimator \hat{I}_0 is the so-called ‘naive’ estimate of mutual information, based on the identity

$$I(X; Y) = H(Y) - H(Y|X).$$

Given any nonparametric estimator of entropy \hat{H} , we obtain

$$\hat{I}_0 = \hat{H}(Y) - \sum_{i=1}^k \hat{H}(Y|X^{(i)}).$$

We use the polynomial-based entropy estimator introduced by Jiao et al. 2015 as the estimator of entropy.

It is known that \hat{I}_0 tends to underestimate the mutual information. Therefore, Gastpar et al. introduced the anthropic correction estimator \hat{I}_α which adds a bias-correction term ²to \hat{I}_0 .

5.3.1 Simulation

We compare the estimators \hat{I}_{CM} , \hat{I}_{Fano} , \hat{I}_{HD} with a nonparametric estimator \hat{I}_α (Gastpar, Gill, and Theunissen 2009) in the following simulation, and the correctly specified parametric estimator \hat{I}_{MLE} . We generate data according to a multiple-response logistic regression model, where $Y \sim N(0, I_p)$, and X is a binary vector with conditional distribution

$$X_i|Y = y \sim \text{Bernoulli}(y^T B_i)$$

where B is a $p \times q$ matrix. We generate training data with r responses per stimulus $Y^{(i)}$. One application of this model might be modeling neural spike count data X

²However, without a principled approach to choose the parameter $\alpha \in (0, 1]$, \hat{I}_α could still vastly underestimate or overestimate the mutual information.

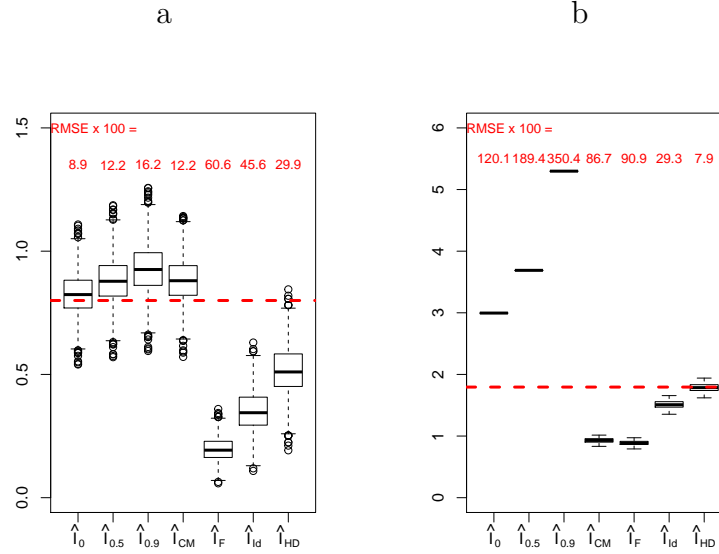


Figure 5.2: Simulation for inferring mutual information in a gaussian random classification model

arising in response to environmental stimuli Y (Banerjee, Dean, and Pesaran 2012). We choose the naive Bayes for the classifier: it is consistent for estimating the Bayes rule.

Figure 5.2 displays the simulation results for two cases:

- (a). A low-dimensional example with $\{p = 3, q = 3, B = \frac{4}{\sqrt{3}}I_3, K = 20, r = 40\}$
- (b). A high-dimensional example with $\{p = 50, q = 50, B = \frac{4}{\sqrt{50}}I_{50}, K = 20, r = 8000\}$

We see that in the low-dimensional example, the naïve estimator \hat{I}_0 achieves the best performance, with both of our proposed estimators \hat{I}_{Ident} and \hat{I}_{HD} biased downwards. However, in the high-dimensional example, which approximately satisfies the assumptions of our high-dimensional theory, we see that all estimators of mutual information become highly concentrated. Therefore, it is the amount of bias in the estimator that mainly determines estimation performance. We see heavy bias in estimators except for \hat{I}_{HD} , which is to be expected since the nonparametric estimators \hat{I}_α scale poorly in

high dimensions, while \hat{I}_{CM} , \hat{I}_{Fano} and \hat{I}_{Ident} are based on lower bounds and therefore should have a downward bias.

5.3.2 Real data example

Kay et al. 2008 employed a randomized stimuli design in their 2008 paper, “Identifying Natural Images from Human Brain Activity.” The experiment was designed in order to investigate how visual information from natural images is encoded in the V1 through V3 brain regions. The stimulus space, \mathcal{X} , consists of 128×128 -pixel grayscale photographs. The response data consists of BOLD response in regions V1, V2, and V3 from a single subject. The raw time series were processed to yield a single averaged response vector $y^{(i)}$ for each stimulus $x^{(i)}$, for $i = 1, \dots, 1870$. The dimensionality of $y^{(i)}$ varies depending on which regions of interest we are discussing, and whether we consider a subset of the voxels in those regions. Let v denote the dimensionality (number of voxels) of y .

Let $x^{(i)}$ denote the native 128×128 -pixel representation of the image (i.e. a 16384-dimensional vector with entries between 0 and 1.) One of the goals of the Kay et al. paper is to evaluate competing encoding models. In the context of the study, an *encoding model* is a vector-valued function from the stimulus to a p -dimensional space,

$$\vec{g}(x) = (g_1(x), \dots, g_p(x)).$$

One of the encoding models studied by Kay et al. is the Gabor wavelet pyramid, \vec{g}_{Gabor} , with $p = 10921$. Using a *training* subset of the stimulus-response pairs (x_i, y_i) , $i = 1, \dots, 1750$, Kay et al. fit a regularized linear model

$$y^{(i)} \approx B^T \vec{g}(x^{(i)})$$

where B is a $10921 \times v$ -dimensional matrix, which is to be fit to the data. Then they computed the identification accuracy for k (number of stimulus classes) ranging from 2 to 1000.

Using the Kay et al. data, we sort voxels in V1 according to estimated signal-to-noise ratio³ (Benjamini and Yu 2013). Then, using cross-validated elastic net on the top $p = \{100, 200, 300, 400, 500\}$ voxels, we compute the identification accuracy for $k = \{20, \dots, 240\}$ classes. The test identification accuracy was used to estimate mutual information using both \hat{I}_{Ident} and \hat{I}_{HD} . Due to the dimensionality of the data, we expected \hat{I}_{HD} to yield more accurate estimates of mutual information.

The estimated mutual information using $k = 240$ is displayed in Figure 5.3. Both estimators indicate a large jump in the mutual information when going from 100 to 200 voxels, almost no increase from 200 to 400, then a slight increase in information from 400 to 500. The two estimators \hat{I}_{Ident} and \hat{I}_{HD} differ significantly for all numbers of voxels. We see from this that it is important to choose the estimator which is best suited for the application: if the high-dimensional assumption can be justified, \hat{I}_{HD} can yield a much better estimate; but if not justified, then it could be a significant overestimate of the mutual information, and it would be preferable to use \hat{I}_{Ident} .

Next, let us examine the sensitivity of the point estimates to the choice of the tuning parameter k . Figure 5.4 shows how the estimates displayed in the previous figure depend on the choice of k . For all numbers of voxels, we see that k does have an appreciable affect on the resulting estimate of mutual information. This is predicted from the theory for \hat{I}_{Ident} , but under the high-dimensionality assumptions \hat{I}_{HD} should not significantly depend on k . Therefore, the sensitivity of \hat{I}_{HD} we see, especially for the 500-voxel case, indicates a detectable violation of the asymptotic assumptions.

Given such evidence for violation of assumptions, the conservative approach would be to abandon \hat{I}_{HD} in this situation and choose \hat{I}_{Ident} as the estimator. Since \hat{I}_{Ident} is an underestimate for all k , one should choose the k which maximizes the estimated mutual information. However, even then, as we see from the earlier simulation, we can expect \hat{I}_{Ident} to significantly underestimate the true mutual information.

³We are using the same data for variable selection and for training the classifier: therefore, the test error estimates are biased. However, the sample size is sufficiently large that the bias should be minor, and we neglect this effect.

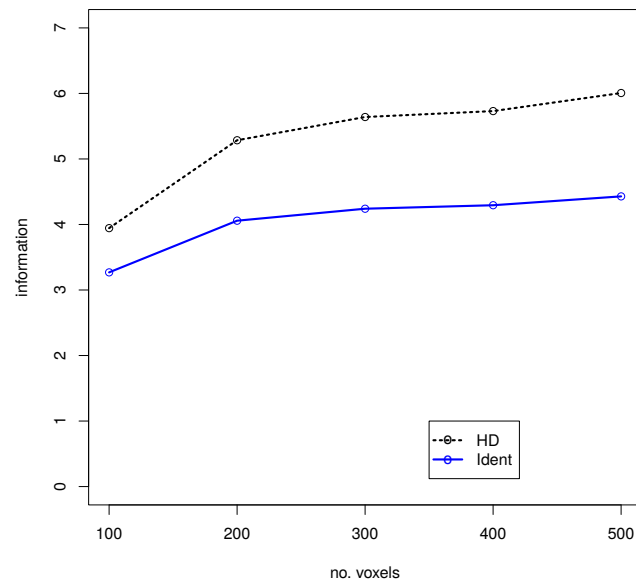


Figure 5.3: Estimated mutual information for different subsets of V1 voxels

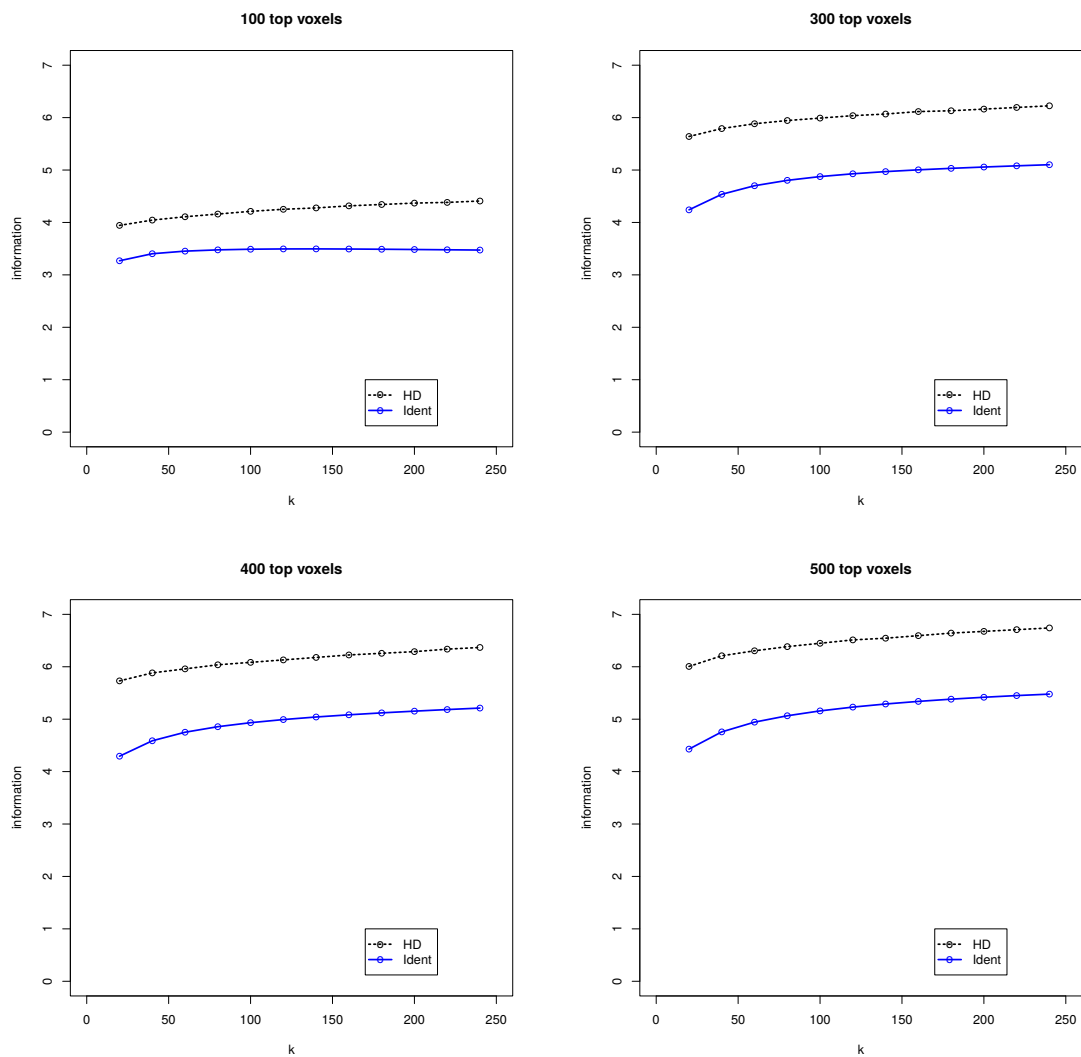


Figure 5.4: Dependence of estimated mutual information on k

Chapter 6

Discussion

In this thesis, we considered the problem of evaluating the quality of a feature embedding, or representation $\vec{g}(\vec{x})$ using side-information from a response y . As we showed in the introduction, the problem of supervised evaluation of representations occurs in both machine learning and neuroscience; therefore, our work is relevant for both areas, but also yields tools that may be useful for other disciplines as well.

We investigated three closely related approaches for doing so: mutual information $I(\vec{g}(\vec{X}); Y)$, average accuracy for randomized classification, and identification accuracy. The identification accuracy is based on the same fundamental prediction task as randomized classification, which is to assign a feature vector X to one of k discrete classes. Therefore, the average Bayes accuracy, defined in Chapter 2, is an upper bound for both average accuracy for randomized classification and identification accuracy. The mutual information can be linked, via Shannon’s coding theorem, to the decoding accuracy achievable using a random codebook, and therefore it is not surprising that we can link mutual information to average Bayes accuracy for randomized classification, as we showed in Chapter 4.

Thus, by combining all of these results, we can convert estimates of average accuracy or estimates of identification accuracy into estimates of mutual information. This provides a useful tool for several reasons. One is that for the purpose of assessing representations, both the average accuracy and the identification accuracy have a dependence on an arbitrary tuning parameter k . To eliminate the dependence, we

can convert either to an estimate of mutual information, which can be interpreted without knowing k . Of course, the estimation procedure itself might still depend on the choice of k , as we saw in the example of section 5.3.2—but it is possible that future work may provide a remedy.

Secondly, for users interested in estimating the mutual information $I(X; Y)$, the possibility of deriving new estimators of mutual information from average accuracy or identification accuracy greatly expands the arsenal of mutual information estimators. This is because one can exploit a variety of different types of problem-specific assumptions, such as sparsity, in the training of the regression or classification model used to compute average accuracy or identification risk.

On the other hand, because there is always some loss when converting one statistic to another non-equivalent statistic, one may choose to work with the average accuracy or identification accuracy directly. Rather than trying to eliminate the parameter k , we can try to understand its effect. For this purpose, the theory of average accuracy (which also applies to identification accuracy) developed in Chapter 3, is invaluable. We see that the k -class average accuracy is simply a $k - 1$ th moment of a *conditional accuracy* distribution; and this can be used to estimate average accuracy at arbitrary k . Understanding how the average accuracy scales with k also yields practical benefits: it allows us to understand how recognition systems might scale under larger problems, and it allows us to compare two different feature representations on the basis of the entire average accuracy curve rather than the average accuracy at a single k .

While we showed mathematically how mutual information is connected to average accuracy and identification accuracy, from an intuitive perspective, they can all be seen to correspond to some notion of information-theoretic volume in the embedding space. We already saw how mutual information can be related to metric entropy in Chapter 1; but the average accuracy (or equivalently, the identification accuracy) can also be used to measure volume, if we define the ϵ -*capacity* of an embedding to be the maximal number of classes k where the average accuracy is above some threshold ϵ . This concept of volume differs slightly from metric entropy or covering numbers because it refers to the spacing properties of random sets of points, rather than an optimized set of points. Still, we expect that the different ways of defining volume

can all be connected in a more formal sense, and we anticipate that lower or upper bounds of metric entropy can be stated in terms of average Bayes accuracy or mutual information.

Two obvious areas of development for our methods is to obtain (i) formal checks of the assumptions (e.g. the high-dimensionality assumption in Ch. 5) which our methods depend on, and (ii) to characterize the quality of the estimators in a decision-theoretic framework, and (iii) to develop interval estimates of mutual information or ϵ -capacity. We have made initial progress towards developing the understanding necessary to develop these methods, e.g. the variance bounds on Bayes accuracy in Chapter 2.

Yet, another important research direction would be to apply these methods to more practical real-life examples. Applications always reveal surprising insights about what kind of models are needed, or what kind of properties are most desirable for a statistical procedure. In this thesis, we discussed the use of our methods in predicting the performance of face-recognition systems, understanding the generalizability of neuroscience experiments, and estimating mutual information in high-dimensional settings; further exploration of any of these particular applications could motivate new refinements and extensions of our methods, but there may also be many additional applications where our ideas can be applied.

Appendix A

Appendix for Chapter 4

A.1 Proofs

Lemma A.1.1 *Let $f(t)$ be an increasing function from $[a, b] \rightarrow \mathbb{R}$, where $a < b$, and let $g(t)$ be a bounded continuous function from $[a, b] \rightarrow \mathbb{R}$. Define the set*

$$A = \{t : f(t) \neq g(t)\}.$$

Then, we can write A as a countable union of intervals

$$A = \bigcup_{i=1}^{\infty} A_i$$

where A_i are mutually disjoint intervals, with $\inf A_i < \sup A_i$, and for each i , either $f(t) > g(t)$ for all $t \in A_i$ or $f(t) < g(t)$ for all $t \in A_i$.

Proof of Lemma A.1.1.

The function $h(t) = f(t) - g(t)$ is measurable, since all increasing functions are measurable. Define $A^+ = \{t : f(t) > g(t)\}$ and $A^- = \{t : f(t) < g(t)\}$. Since A^+ and A^- are measurable subsets of \mathbb{R} , they both admit countable partitions consisting of open, closed, or half-open intervals. Let \mathcal{H}^+ be the collection of all partitions of A^+ consisting of such intervals. There exists a least refined partition \mathcal{A}^+ within \mathcal{H}^+ .

Define \mathcal{A}^- analogously, and let

$$\mathcal{A} = \mathcal{A}^+ \cup \mathcal{A}^-$$

and enumerate the elements

$$\mathcal{A} = \{A_i\}_{i=1}^\infty.$$

We claim that the partitions \mathcal{A}^+ and \mathcal{A}^- have the property that for all $t \in A^\pm$, the interval $I \in \mathcal{A}^\pm$ containing t has endpoints $l \leq u$ defined by

$$l = \inf_{x \in [a, b]} \{x : \text{Sign}(h([x, t])) = \{\text{Sign}(h(t))\}\}$$

and

$$u = \sup_{x \in [a, b]} \{x : \text{Sign}(h([t, x])) = \{\text{Sign}(h(t))\}\}.$$

We prove the claim for the partition \mathcal{A}^+ . Take $t \in A^+$ and define l and u as above. It is clear that $(l, u) \in A^+$, and furthermore, there is no $l' < l$ and $u' > u$ such that $(l', x) \in A^+$ or $(x, u') \in A^+$ for any $x \in I$. Let \mathcal{H} be any other partition of A^+ . Some disjoint union of intervals $H_i \in \mathcal{H}$ necessarily covers I for $i = 1, \dots$, and we can further require that none of the H_i are disjoint with I . Since each H_i has nonempty intersection with I , and I is an interval, this implies that $\cup_i H_i$ is also an interval. Let $l'' \leq u''$ be the endpoints of $\cup_i H_i$. Since $I \subseteq \cup_i H_i$, we have $l'' \leq l \leq u \leq u''$. However, since also $I \in A^+$, we must have $l \leq l'' \leq u'' \leq u$. This implies that $l'' = l$ and $u'' = u$. Since $\cup_i H_i = I$, and this holds for any $I \in \mathcal{A}^+$, we conclude that \mathcal{H} is a refinement of \mathcal{A}^+ . The proof of the claim for \mathcal{A}^- is similar.

It remains to show that there are not isolated points in \mathcal{A} , i.e. that for all $I \in \mathcal{A}$ with endpoints $l \leq u$, we have $l < u$. Take $I \in \mathcal{A}$ with endpoints $l \leq u$ and let $t = \frac{l+u}{2}$. By definition, we have $h(t) \neq 0$. Consider the two cases $h(t) > 0$ and $h(t) < 0$.

If $h(t) > 0$, then $t' = g^{-1}(h(t)) > t$, and for all $x \in [t, t']$ we have $h(x) > 0$. Therefore, it follows from definition that $[t, t'] \in I$, and since $l \leq t < t' \leq u$, this implies that $l < u$. The case $h(t) < 0$ is handled similarly. \square

Lemma A.1.2 *Let $f(t)$ be a measurable function from $[a, b] \rightarrow \mathbb{R}$, where $a < b$. Then there exists sets \mathcal{B}_0 and \mathcal{B}_1 , satisfying the following properties:*

- $\mathcal{B} = \mathcal{B}_0 \cup \mathcal{B}_1$ is countable partition of $[a, b]$,
- $f(t)$ is constant on all $B \in \mathcal{B}_0$, but not constant on any proper superinterval $B' \supset B$, and
- $B \in \mathcal{B}_1$ contains no positive-length subinterval where $f(t)$ is constant.

Proof of Lemma A.1.2. To construct the interval, define

$$l(t) = \inf\{x \in [0, 1] : f([x, t]) = \{f(t)\}\}$$

$$u(t) = \sup\{x \in [0, 1] : f([t, x]) = \{f(t)\}\},$$

Let B_0 be the set of all t such that $l(t) < u(t)$, and let B_1 be the set of all t such that $l(t) = t = u(t)$. For all $t \in B_0$, define

$$I(t) = (l(t), u(t)) \cup \{x \in \{l(t), u(t)\} : f(x) = f(t)\}.$$

Then we claim

$$\mathcal{B}_0 = \{I(t) : t \in B_0\}$$

is a countable partition of B_0 . The claim follows since the members of \mathcal{B}_0 are disjoint intervals of nonzero length, and B_0 has finite length. It follows from definition that for any $B \in \mathcal{B}_0$, that f is not constant on any proper superinterval $B' \supset B$.

Meanwhile, let \mathcal{B}_1 be a countable partition of B_1 into intervals.

Next, we show that for all $I \in \mathcal{B}_1$, I does not contain a subinterval I' of nonzero length such that f is constant on I' . Suppose to the contrary, we could find such an interval I and subinterval I' . Then for any $t \in I'$, we have $t \in B_0$. However, this implies that $t \notin B_1$, a contradiction.

Since $t \in [a, b]$ belongs to either B_0 or B_1 , letting $\mathcal{B} = \mathcal{B}_0 \cup \mathcal{B}_1$ yields the desired partition of $[a, b]$. \square .

Lemma A.1.3 *Define an exponential family on $[0, 1]$ by the density function*

$$q_\beta(t) = \exp[\beta t^{k-1} - \log Z(\beta)]$$

where

$$Z(\beta) = \int_0^1 \exp[\beta t^{k-1}] dt.$$

Then, the negative entropy

$$I(\beta) = \int_0^1 q_\beta(t) \log q_\beta(t) dt$$

is decreasing in β on the interval $(-\infty, 0]$. and increasing on the interval $[0, \infty)$.

Furthermore, for any $\iota \in (0, \infty)$, there exist two solutions to $I(\beta) = \iota$: one positive and one negative.

Proof of Lemma A.1.3.

Define $\beta(\mu)$ as the solution to

$$\mu = \int_0^1 t q_\beta(t) dt.$$

By [Wainwright and Jordan 2008], the function $\beta(\mu)$ is well-defined. Furthermore, since the sufficient statistic t^{k-1} is increasing in t , it follows that $\beta(\mu)$ is increasing.

Define the negative entropy as a function of μ ,

$$N(\mu) = \int_0^1 q_{\beta(\mu)}(t) \log q_{\beta(\mu)}(t) dt.$$

By Theorem 3.4 of [Wainwright and Jordan 2008], $N(\mu)$ is convex in μ . We claim that the derivative of $N(\mu) = 0$ at $\mu = \frac{1}{2}$. This implies that $N(\mu)$ is decreasing in μ for $\mu \leq \frac{1}{2}$ and increasing for $\mu \geq \frac{1}{2}$. Since $I(\beta(\mu)) = N(\mu)$, β is increasing in μ , and $\beta(\frac{1}{2}) = 0$, this implies that $I(\beta)$ is decreasing in β for $\beta \leq 0$ and increasing for $\beta \geq 0$.

We will now prove the claim. Write

$$\left. \frac{d}{d\mu} N(\mu) \right|_{\mu=1/2} = \left. \frac{d}{d\beta} I(\beta(\mu)) \right|_{\beta=0} \left. \frac{d\beta}{d\mu} \right|_{\mu=1/2}.$$

We have

$$\frac{d}{d\beta} I(\beta) = \beta \int q_\beta t^{k-1} dt - \log Z(\beta).$$

Meanwhile, $Z(0) = 1$ so $\log Z(0) = 0$. Therefore,

$$\left. \frac{d}{d\beta} I(\beta) \right|_{\beta=0} = 0.$$

This implies that $\left. \frac{d}{d\mu} N(\mu) \right|_{\mu=1/2} = 0$, as needed.

For the final statement of the lemma, note that $I(0) = 0$ since q_0 is the uniform distribution. Meanwhile, since q_β tends to a point mass as either $\beta \rightarrow \infty$ or $\beta \rightarrow -\infty$, we have

$$\lim_{\beta \rightarrow \infty} I(\beta) = \lim_{\beta \rightarrow -\infty} I(\beta) = \infty.$$

And, as we can check that $I(\beta)$ is continuous in β , this means that

$$I((-\infty, 0]) = I([0, \infty)) = [0, \infty)$$

by the mean-value theorem. Combining this fact with the monotonicity of $I(\beta)$ restricted to either the positive and negative half-line yields the fact that for any $\iota > 0$, there exists $\beta_1 < 0 < \beta_2$ such that $I(\beta_1) = I(\beta_2) = \iota$. \square .

Lemma A.1.4 *For any measure G on $[0, \infty]$, let G^k denote the measure defined by*

$$G^k(A) = G(A)^k,$$

and define

$$E[G] = \int x dG(x).$$

$$I[G] = \int x \log x dG(x)$$

and

$$\psi_k[G] = \int x d(G^k)(x).$$

Then, defining Q_c and c_t as in Theorem 1, we have

$$\sup_{G: E[G]=1, I[G] \leq \iota} \psi_k[G] = \int_0^1 Q_{c_t}(t) t^{k-1} dt.$$

Furthermore, the supremum is attained by a measure G that has cdf equal to Q_c^{-1} , and thus has a density g with respect to Lebesgue measure.

Proof of Lemma A.1.4.

Consider the quantile function $Q(t) = \inf_{x \in [0,1]} : G((-\infty, x]) \geq t$. $Q(t)$ must be a monotonically increasing function from $[0, 1]$ to $[0, \infty)$. Let \mathcal{Q} denote the collection of all such quantile functions.

We have

$$E[G] = \int_0^1 Q(t) dt$$

$$\psi_k[G] = \int_0^1 Q(t) x^{k-1} dt.$$

and

$$I[G] = \int_0^1 Q(t) \log Q(t) dt.$$

For any given ι , let P_ι denote the class of probability distributions G on $[0, \infty]$ such that $E[G] = 1$ and $I[G] \leq \iota$. From Markov's inequality, for any $G \in P_\iota$ we have

$$G([x, \infty]) \leq x^{-1}$$

for any $x \geq 0$, hence P_ι is tight. From tightness, we conclude that P_ι is closed under limits with respect to weak convergence. Hence, since ψ_k is a continuous function, there exists a distribution $G^* \in P_\iota$ which attains the supremum

$$\sup_{G \in P_\iota} \psi_k[G].$$

Let \mathcal{Q}_ι denote the collection of quantile functions of distributions in P_ι . Then, \mathcal{Q}_ι

consists of monotonic functions $Q : [0, 1] \rightarrow [0, \infty]$ which satisfy

$$E[Q] = \int_0^1 Q(t) dt = 1,$$

and

$$I[Q] = \int_0^1 Q(t) \log Q(t) dt \leq \iota.$$

Let \mathcal{Q} denote the collection of *all* quantile functions from measures on $[0, \infty]$. And letting Q^* be the quantile function for G^* , we have that Q^* attains the supremum

$$\sup_{Q \in \mathcal{Q}_\iota} \phi_k[Q] = \sup_{Q \in \mathcal{Q}_\iota} \int_0^1 Q(t) t^{k-1} dt. \quad (\text{A.1})$$

Therefore, there exist Lagrange multipliers $\lambda \geq 0$ and $\nu \geq 0$ such that defining

$$\mathcal{L}[Q] = -\phi_k[Q] + \lambda E[Q] + \nu I[Q] = \int_0^1 Q(t) (-t^{k-1} + \lambda + \nu \log Q(t)) dt,$$

Q^* attains the infimum of $\mathcal{L}[Q]$ over *all* quantile functions,

$$\mathcal{L}[Q^*] = \inf_{Q \in \mathcal{Q}} \mathcal{L}[Q].$$

The global minimizer Q^* is also necessarily a stationary point: that is, for any perturbation function $\xi : [0, 1] \rightarrow \mathbb{R}$ such that $Q^* + \xi \in \mathcal{Q}$, we have $\mathcal{L}[Q^*] \leq \mathcal{L}[Q^* + \xi]$. For sufficiently small ξ , we have

$$\mathcal{L}[Q + \xi] \approx \mathcal{L}[Q] + \int_0^1 \xi(t) (-t^{k-1} + \lambda + \nu + \nu \log Q(t)) dt. \quad (\text{A.2})$$

Define

$$\nabla \mathcal{L}_{Q^*}(t) = -t^{k-1} + \lambda + \nu + \nu \log Q(t). \quad (\text{A.3})$$

The function $\nabla \mathcal{L}_{Q^*}(t)$ is a *functional derivative* of the Lagrangian. Note that if we were able to show that $\nabla \mathcal{L}_{Q^*}(t) = 0$, this immediately yields

$$Q^*(t) = \exp[-1 - \lambda \nu^{-1} + \nu^{-1} t^{k-1}]. \quad (\text{A.4})$$

At this point, we know that the right-hand side of (A.4) gives a stationary point of \mathcal{L} , but we cannot be sure that it gives the global minimizer. The reason is because the optimization occurs on a constrained space. We will show that (A.4) indeed gives the global minimizer Q^* , but we do so by showing that the set of points t where $\nabla \mathcal{L}_{Q^*}(t) \neq 0$ is of zero measure. Since sets of zero measure don't affect the integrals defining the optimization problem (A.1), we conclude there exists a global optimal solution with $\nabla \mathcal{L}_{Q^*}(t) = 0$ everywhere, which is therefore given explicitly by (A.4) for some $\lambda \in \mathbb{R}$, $\nu \geq 0$.

We will need the following result: that for $\iota > 0$, any solution to (A.1) satisfies $\phi_k[Q] < 1$. This follows from the fact that

$$E[Q] - \phi_k[Q] = \int_0^1 (1 - t^{k-1})Q(t)dt,$$

where the term $(1 - t^{k-1})$ is negative, except for the one point $t = 1$. Therefore, in order for $\phi_k[Q] = 1 = E[Q]$, we must have $Q(t) = 0$ for $t < 1$. However, this yields a contradiction since $Q(t) = 0$ for $t < 1$ implies that $E[Q] = 0$, a violation of the hard constraint $E[Q] = 1$.

Let us establish that $\nu > 0$: in other words, the constraint $I[Q] = \iota$ is tight. Suppose to the contrary, that for some $\iota > 0$, the global optimum Q^* minimizes a Lagrangian with $\nu = 0$. Let $\phi^* = \phi_k[Q^*] < 1$. However, if we define $Q_\kappa(t) = I\{t \geq 1 - \frac{1}{\kappa}\}\kappa$, we have $E[Q_\kappa] = 1$, and also for some sufficiently large $\kappa > 0$, $\phi_k[Q_\kappa] > \phi^*$. But since the Lagrangian lacks a term corresponding to $I[Q]$, we conclude that $\mathcal{L}[Q_\kappa] < \mathcal{L}[Q^*]$, a contradiction.

The rest of the proof proceeds as follows. We will use Lemmas A.1.1 and A.1.2 to define a decomposition $A = D_0 \cup D_1 \cup D_2$, where D_2 is of measure zero. First, we show that assuming the existence of $t \in D_0$ yields a contradiction, and hence $D_0 = \emptyset$. Then, again using argument from contradiction we establish that $D_1 = \emptyset$. Finally, since D_2 is a set of zero measure, this allows us to conclude that the $Q^*(t) = 0$ on all but a set of zero measure.

We will now apply the Lemmas to obtain the necessary ingredients for constructing the sets D_i . Since $\nabla \mathcal{L}_{Q^*}(t)$ is a difference between an increasing function and a

continuous strictly increasing function, we can apply Lemma A.1.1 to conclude that there exists a countable partition \mathcal{A} of the set $A : \{t \in [0, 1] : \nabla \mathcal{L}_{Q^*}(t) \neq 0\}$ into intervals such that for all $J \in \mathcal{A}$, $|\text{Sign}(\nabla Q^*(J))| = 1$ and $\inf J < \sup J$. Applying Lemma A.1.2 we get a countable partition $\mathcal{B} = \mathcal{B}_0 \cup \mathcal{B}_1$ of $[0, 1]$ so that each element $J \in \mathcal{B}_0$ is an interval such that $\nabla \mathcal{L}_{Q^*}(t)$ is constant on J , and furthermore is not properly contained in any interval with the same property, and each element $J \in \mathcal{B}_1$ is an interval, such that J contains no positive-length subinterval where $\nabla \mathcal{L}_{Q^*}(t)$ is constant. Also define B_i as the union of the sets in \mathcal{B}_i for $i = 0, 1$.

Note that B_0 is necessarily a subset of A . That is because if $\nabla \mathcal{L}_{Q^*}(t) = 0$ on any interval J , then that $Q^*(t)$ is necessarily not constant on the interval.

We will construct a new countable partition of A , called \mathcal{D} . The partition \mathcal{D} is constructed by taking the union of three families of intervals,

$$\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1 \cup \mathcal{D}_2.$$

Define D_i to be the union of intervals in \mathcal{D}_i for $i = 0, 1, 2$.

Define $\mathcal{D}_0 = \mathcal{B}_0$, Define a countable partition \mathcal{D}_1 by

$$\mathcal{D}_1 = \{J \cap L : J \in \mathcal{A}, L \in \mathcal{B}_1, \text{ and } |L| > 1\},$$

in order words, \mathcal{D}_1 consists of positive-length intervals where $\nabla Q^*(t)$ is entirely positive or negative and is not constant. Define

$$\mathcal{D}_2 = \{J \in \mathcal{B}_1 : J \subset A \text{ and } |J| = 1\},$$

i.e. \mathcal{D}_2 consists of isolated points in A .

One verifies that \mathcal{D} is indeed a partition of A by checking that $D_0 = B_0$, $D_1 \cup D_2 = B_1 \cap A$, so that $D_0 \cup D_1 \cup D_2 = A$: it is also easy to check that elements of \mathcal{D} are disjoint. Furthermore, as we mentioned earlier, the set D_2 is indeed of zero measure, since it consists of countably many isolated points.

Now we will show that the existence of $t \in D_0$ implies a contradiction. Take $t \in D$

for $D \in \mathcal{D}_0$, and let $a = \inf D$ and $b = \sup D$. Define

$$\xi^+ = I\{t \in D\}(Q^*(b) - Q^*(t))$$

and

$$\xi^- = I\{t \in D\}(Q^*(a) - Q^*(t)).$$

Observe that $Q + \epsilon\xi^+ \in \mathcal{Q}$ and $Q + \epsilon\xi^- \in \mathcal{Q}$ for any $\epsilon \in [0, 1]$. Now, if $\nabla\mathcal{L}_{Q^*}(t)$ is strictly positive on D , then for some $\epsilon > 0$ we would have $\mathcal{L}[Q^* + \epsilon\xi^-] < \mathcal{L}[Q^*]$, a contradiction. A similar argument with ξ^+ shows that $\nabla\mathcal{L}_{Q^*}(t)$ cannot be strictly negative on D either. From this perturbation argument, we conclude that $\nabla\mathcal{L}_{Q^*}(t) = 0$. Since this argument applies for all $t \in D_0$, we conclude that $D_0 = \emptyset$: therefore, on the set $[0, 1] \setminus (D_1 \cup D_2)$, we have $\nabla\mathcal{L}_{Q^*}(t) = 0$.

The following observation is needed for the next stage of the proof. If we look at the function $Q^*(t)$, then up to sets of negligible measure, it is given by the expression (A.4) on the set $[0, 1] \setminus D_1$, and it is piecewise constant in-between. But since (A.4) gives a strictly increasing function, and since Q^* is increasing, this implies that Q^* is discontinuous at the boundary of D_1 .

Now we are prepared to show that $\nabla\mathcal{L}_{Q^*}(t) = 0$ for $t \in D_1$. Take $t \in D$ for $D \in \mathcal{D}_1$, and let $a = \inf D$ and $b = \sup D$. From the previous argument, there is a discontinuity at both a and b , so that $\lim_{u \rightarrow a^-} Q(u) < Q(t) < \lim_{u \rightarrow b^+} Q(u)$. Therefore, for any $\xi(t)$ which is increasing on D and zero elsewhere, there exists $\epsilon > 0$ such that $\nabla Q^* + \epsilon\xi \in \mathcal{Q}$. It remains to find such a perturbation ξ such that $\mathcal{L}[Q + \epsilon\xi] < \mathcal{L}[Q]$.

Also, since by definition $\nabla\mathcal{L}_{Q^*}(t)$ is constant on D , follows from (A.3) that ∇Q^* is strictly decreasing, and thus either

- Case 1: $\nabla\mathcal{L}_{Q^*}(t) \geq 0$ on D ,
- Case 2: $\nabla\mathcal{L}_{Q^*}(t) \leq 0$ on D , or
- Case 3: $\nabla\mathcal{L}_{Q^*}(t) \geq 0$ for all $t \in D \cap [a, t_0]$ and $\nabla\mathcal{L}_{Q^*}(t) \leq 0$ for all $t \in D \cap [t_0, b]$.

Depending on the case, we construct a suitable perturbation ξ :

- Case 1: Construct $\xi(t) = -I\{t \in D\}$.
- Case 2: Construct $\xi(t) = I\{t \in D\}$
- Case 2: Construct

$$\xi(t) = \begin{cases} -1 & \text{for } t \in D \cap [a, t_0], \\ 0 & \text{otherwise.} \end{cases}$$

In all three cases, given the corresponding construction for $\xi(t)$ we get

$$\int_0^1 \xi(t) \nabla \mathcal{L}_{Q^*}(t) dt < 0.$$

Therefore, from (A.2), there exists some $\epsilon > 0$ such that $\mathcal{L}[Q + \epsilon\xi] < \mathcal{L}[Q]$, a contradiction. Again, since the contradiction applies for all $t \in D_1$, we conclude that $D_1 = \emptyset$.

By now we have established that a global optimum for (A.1) exists, and is given by (A.4) for some $\lambda \in \mathbb{R}$, $\nu > 0$. It remains to determine the values of λ and ν .

Reparameterize $\alpha = \exp[-1 - \lambda\nu^{-1}]$ and $\beta = \nu^{-1}$. Therefore,

$$Q^*(t) = \alpha \exp[\beta t^{k-1}]$$

for $\alpha > 0$, $\beta > 0$. There is a one-to-one mapping from $(\alpha, \beta) \in (0, \infty)^2$ to $(\lambda, \nu) \in \mathbb{R} \times (0, \infty)$.

Now, from the constraint

$$1 = E[Q^*] = \int_0^1 \alpha \exp[\beta t^{k-1}] dt.$$

we conclude that

$$\alpha = \frac{1}{\int_0^1 \exp[\beta t^{k-1}] dt}.$$

Therefore, we have reduced the set of possible solutions Q^* to a one-parameter family,

$$Q^*(t) = \frac{\exp[\beta t^{k-1}]}{Z(\beta)}.$$

where

$$Z(\beta) = \int_0^1 \exp[\beta t^{k-1}] dt.$$

Next, note that

$$I[Q^*] = \int_0^1 Q^*(t) \log Q^*(t) = \beta \mu_\beta - \log Z(\beta),$$

as a function of β , is completely characterized by Lemma A.1.3. Let us define c_ι as the unique positive solution to the equation

$$c_\iota \mu_{c_\iota} - \log Z(c_\iota) = \iota$$

given by Lemma A.1.3. We therefore have

$$Q^*(t) = \frac{\exp[c_\iota t^{k-1}]}{\int_0^1 \exp[c_\iota t^{k-1}]},$$

as needed. \square

Lemma A.1.5 *The map*

$$\iota \rightarrow \int_0^1 Q_{c_\iota}(t) t^{k-1} dt$$

is concave in $\iota > 0$.

Proof of Lemma A.1.5. It is equivalent to show that the inverse function

$$C_k^{-1}(p) = \inf_{G: E[G]=1, \phi_k[G]=p} I[G]$$

is convex. Let $p_1, p_2 \in [0, 1]$. From lemma A.1.4, we can find measures G_1, G_2 on $[0, \infty)$ which minimize $I[G_i]$ subject to $E[G_i] = 1$ and $\phi_k[G_i] = p_i$. Define the measure

$$H = \frac{G_1 + G_2}{2}.$$

Since ϕ_k is a linear functional,

$$\phi_k[H] = \frac{\phi_k[G_1] + \phi_k[G_2]}{2} = \frac{p_1 + p_2}{2}.$$

But since I is a convex functional,

$$I[H] \leq \frac{I[G_1] + I[G_2]}{2}.$$

Therefore,

$$C_k^{-1}\left(\frac{p_1 + p_2}{2}\right) \leq I[H] = \frac{I[G_1] + I[G_2]}{2} = \frac{C_k^{-1}(p_1) + C_k^{-1}(p_2)}{2}.$$

□.

Bibliography

- Achanta, Rakesh and Trevor Hastie (2015). “Telugu OCR Framework using Deep Learning”. In: *arXiv preprint arXiv:1509.05962*.
- Adler, Robert J and Jonathan E Taylor (2009). *Random fields and geometry*. Springer Science & Business Media.
- Amari, Shun-ichi and Hiroshi Nagaoka (2007). *Methods of information geometry*. Vol. 191. American Mathematical Soc.
- Amos, Brandon, Bartosz Ludwiczuk, and Mahadev Satyanarayanan (2016). *OpenFace: A general-purpose face recognition library with mobile applications*. Tech. rep. CMU-CS-16-118, CMU School of Computer Science.
- Baldi, Pierre (2012). “Autoencoders, unsupervised learning, and deep architectures.” In: *ICML unsupervised and transfer learning* 27.37-50, p. 1.
- Banerjee, Arpan, Heather L Dean, and Bijan Pesaran (2012). “Parametric models to relate spike train and LFP dynamics with neural information processing”. In: *Frontiers in computational neuroscience* 6, p. 51.
- Bengio, Yoshua, Aaron Courville, and Pascal Vincent (2013). “Representation learning: A review and new perspectives”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8, pp. 1798–1828.
- Benjamini, Yuval, Bin Yu, et al. (2013). “The shuffle estimator for explainable variance in fmri experiments”. In: *The Annals of Applied Statistics* 7.4, pp. 2007–2033.
- Cioffi, John M. (2014). *Signal processing and detection*.

- Clarkson, Philip and Pedro J Moreno (1999). “On the use of support vector machines for phonetic classification”. In: *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*. Vol. 2. IEEE, pp. 585–588.
- Cover, Thomas M. and Joy A. Thomas (2006). *Elements of Information Theory*. 2nd ed. Wiley-Interscience. ISBN: 978-0471241959.
- 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>.
- Deng, Jia et al. (2010). “What does classifying more than 10,000 image categories tell us?” In: *European conference on computer vision*. Springer, pp. 71–84.
- Duygulu, Pinar et al. (2002). “Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary”. In: *European conference on computer vision*. Springer, pp. 97–112.
- Fano, Robert M (1961). *Transmission of information: A statistical theory of communications*.
- Fisher, Ronald A (1936). “The use of multiple measurements in taxonomic problems”. In: *Annals of eugenics* 7.2, pp. 179–188.
- Frey, Peter W and David J Slate (1991). “Letter recognition using Holland-style adaptive classifiers”. In: *Machine learning* 6.2, pp. 161–182.
- Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2010). “Regularization Paths for Generalized Linear Models via Coordinate Descent”. In: *Journal of Statistical Software* 33.1, pp. 1–22. URL: <http://www.jstatsoft.org/v33/i01/>.
- Gastpar, Michael C., Patrick R. Gill, and Frédéric E. Theunissen (2009). “Anthropic correction of information estimates”. In: *Proceedings - 2009 IEEE Information Theory Workshop on Networking and Information Theory, ITW 2009* 56.2, pp. 152–155. ISSN: 00189448. DOI: 10.1109/ITWNIT.2009.5158561.
- Grother, Patrick J (1995). “NIST special database 19”. In: *Handprinted forms and characters database, National Institute of Standards and Technology*.

- Haghighat, Mohammad, Saman Zonouz, and Mohamed Abdel-Mottaleb (2015). “CloudID: Trustworthy cloud-based and cross-enterprise biometric identification”. In: *Expert Systems with Applications* 42.21, pp. 7905–7916.
- 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>.
- Hinton, Geoffrey E and Ruslan R Salakhutdinov (2006). “Reducing the dimensionality of data with neural networks”. In: *science* 313.5786, pp. 504–507.
- Huang, Gary B. et al. (2007). *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. rep. 07-49. University of Massachusetts, Amherst.
- Jiao, Jiantao et al. (2015). “Minimax estimation of functionals of discrete distributions”. In: *IEEE Transactions on Information Theory* 61.5, pp. 2835–2885.
- 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.
- Kriegeskorte, Nikolaus, Marieke Mur, and Peter A Bandettini (2008). “Representational similarity analysis-connecting the branches of systems neuroscience”. In: *Frontiers in systems neuroscience* 2, p. 4.
- Kumar, Neeraj et al. (2009). “Attribute and simile classifiers for face verification”. In: *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, pp. 365–372.
- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). “Deep learning”. In: *Nature* 521.7553, pp. 436–444.
- LeCun, Yann and M Ranzato (2013). “Deep learning tutorial”. In: *Tutorials in International Conference on Machine Learning (ICML13)*.
- 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.

- Mickalstd, RS (1980). “LEARNING BY BEING TOLD AND LEARNING FROM EXAMPLES: AN EXPERIMENTAL COMPARISON OF THE TWO METHODS OF KNOWLEDGE ACQUISITION”. In:
- Mnatsakanov, R. M. et al. (2008). “K n -nearest neighbor estimators of entropy”. In: *Mathematical Methods of Statistics* 17.
- Partalas, Ioannis et al. (2015). “LSHTC: A benchmark for large-scale text classification”. In: *arXiv preprint arXiv:1503.08581*.
- Quián Quiroga, Rodrigo and Stefano Panzeri (2009). “Extracting information from neuronal populations: information theory and decoding approaches.” In: *Nature reviews. Neuroscience* 10.3, pp. 173–185. ISSN: 1471-003X. DOI: 10.1038/nrn2578.
- Schroff, Florian, Dmitry Kalenichenko, and James Philbin (2015). “Facenet: A unified embedding for face recognition and clustering”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823.
- Taigman, Yaniv et al. (2014). “Deepface: Closing the gap to human-level performance in face verification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708.
- Tebbe, D and S Dwyer (1968). “Uncertainty and the probability of error (Corresp.)” In: *IEEE Transactions on Information Theory* 14.3, pp. 516–518.
- 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>.