Supervised Evaluation of Representations

Author:
Charles ZHENG

Supervisor:
Dr. Trevor HASTIE and Dr.
Jonathan TAYLOR

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

in the

Department of Statistics

April 2, 2017

Contents

Chapter 1

Introduction

1.1 Finding the correct representation

A fundamental question in the cognitive sciences is how humans and other organisms percieve 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 perciever 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 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 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)
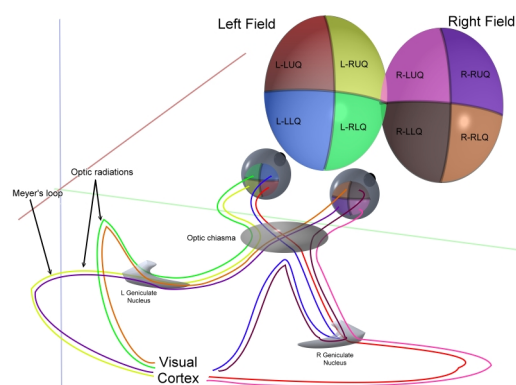

FIGURE 1.1: Visual pathway in humans. Image credit to Ratznium under CC 2.5 license.
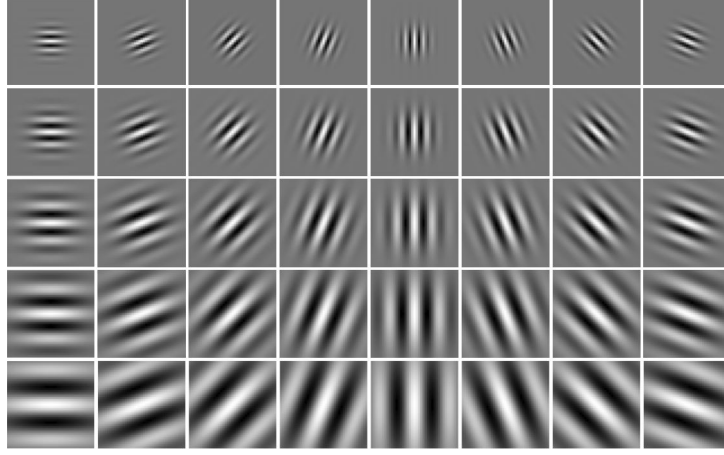
FIGURE 1.2: Examples of Gabor filters of varying size and orientation.
From Haghighat, Zonouz, and Abdel-Mottaleb 2015.

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 *representations*, which is a family $\vec{g} = (g_1, \ldots, g_m)$ of linear or nonlinear transformations of the visual stimulus $\vec{z}$. Let z_j denote the intensity of the jth pixel in the visual stimulus, and let $\ell_j = (r_j, c_j)$ indicate the row and column coordinates of the jth pixel. Under the retinotopic 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

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

where $||\cdot||^2$ is the squared modulus of a complex number, θ is a free parameter which

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

FIGURE 1.3: Face recognition problem

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

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

where $\boldsymbol{B}$, 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 significantly 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 jth labelled photograph in the database belonging to the ith individual. One way to study the problem is to fit it into the multi-class classification framework, where the label set $\mathcal{Y}$ consists of all individuals in the data set, $\{y^{(1)}, \ldots, y^{(k)}\}$.

Decades of research into facial recognition has confirmed that careful *feature-engineering* or *representation-learning* is the key to acheiving 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, 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.
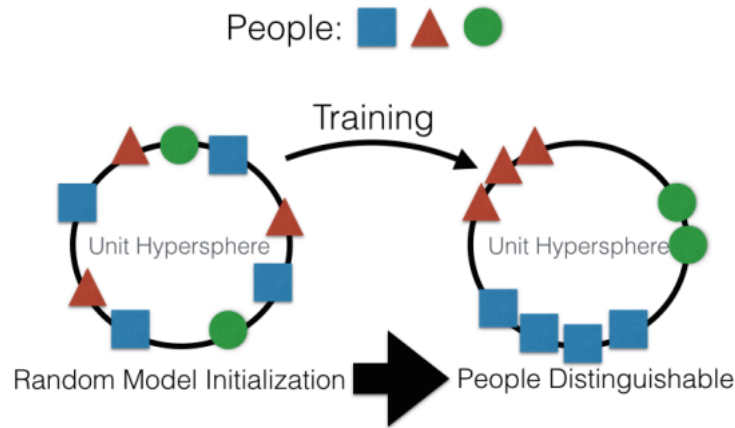
FIGURE 1.4: Triplet loss function for training face representations.
From Amos, Ludwiczuk, and Satyanarayanan 2016

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, acheived 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 hierarichical 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 [CITE] 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 and therefore find a good representation.

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,

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

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 representationBengio, 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.

If we look at the two examples of 'representation evaluation' that we just presented, one in neuroscience and one from facial recognition, then we see an intuition common to both cases which is that *a good representation* captures the 'signal', or meaningful information, in the raw input, while discarding the 'noise'–which can either take the form of actual noise, or simply information that is not relevant to the task.

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 accesories such as sunglasses. One might imagine that an idealized representation for facial recognition is one that estimates the latent, persistent traits, such as the shape and size of facial features, while discarding the effects of 'nuisance' parameters such as pose and lighting. 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 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).

In both examples, we have not 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 give us information as to what V1 neurons "care about" in the image.

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. As we will explain further in the ensuring chapters, we consider three different methods for evaluating the quality of the representation.

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 risk.*

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 risk*, which was introduced by the same functional MRI study of natural images (Kay et al. 2008) that we have been discussing. To our knowledge, we are the first to investigate the properties of the identification risk from a theoretical perspective.

All three of these methods enable *supervised evaluation of representations* because they define a quality metric which depends on a response variable Y. As in *supervised learning*, the response Y gives us a means of judging the quality of the representation $\vec{g}(\vec{z})$.

Comparing these methods, the advantage of the k-class average classification error or identification risk is that they are relatively easy to compute, even in high-dimensional data, because one can make use of machine-learning methods for computing these quantities. 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 risk, are highly related. In particular, we establish methods for lower-bounding the mutual information from either the k-class average classification accuracy or identification risk.

1.1.4 Related Work

As we hoped to convey in the introduction, the problem of finding and evaluating representations is an extremely 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 $\vec{z}$ [CITE]. The representations are obtained as summary statistics of the latent variables in the model, and the quality of the representation is assessed via tha *likelihood* of the parametric model. *Auto-encoders* form another family of methods [CITE]. 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 $i = 1, \ldots, n$, 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 pariwise 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   Recognition tasks

One of the key capabilities of an intelligent system, natural or artificial, is the ability to recognize objects, agents, and signs in the environment based on input data. 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 X decades, only very recent advances in machine learning, such as deep neural networks, have allowed algorithmic recognition algorithms to approach or exceed human performance [CITE].

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 success of the classification rule h is measured by the *expected loss* or *risk*, which in the case of zero-one loss takes the form

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

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; same 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 [CITE].

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.

An additional set of applications of the randomized classification framework lies in its connection to information theory. Randomized classification is the natural analogue of the *random code* models first studied by Claude Shannon. Furthermore, it becomes possible to prove extensions of Fano's inequality to the case of continuous X and Y by means of randomized classification. Therefore, randomized classification can be used as a means of inferring mutual information.

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.

1.3 Information and Discrimination

In studying the problem of recognition, 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, information theory, theoretical statistics, and machine learning are based on the same foundation: measure-theoretic probability theory; one could even 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 reciever's recieved 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 the problem is again to *discriminate* between two (or more) different hypotheses regarding the data-geenrating mechanism.

The concepts of *information* and *discrimination* are quite distinct from an intuitive standpoint; however, they are linked at a fundamental level. This link can be seen throughout statistics and machine learning, and in the way we think about statistical problems. 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 reciever 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 discrimination[1].

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 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 abut 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 discrmination 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.

---

[1]Supposing 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 $\omega \in \Omega$.
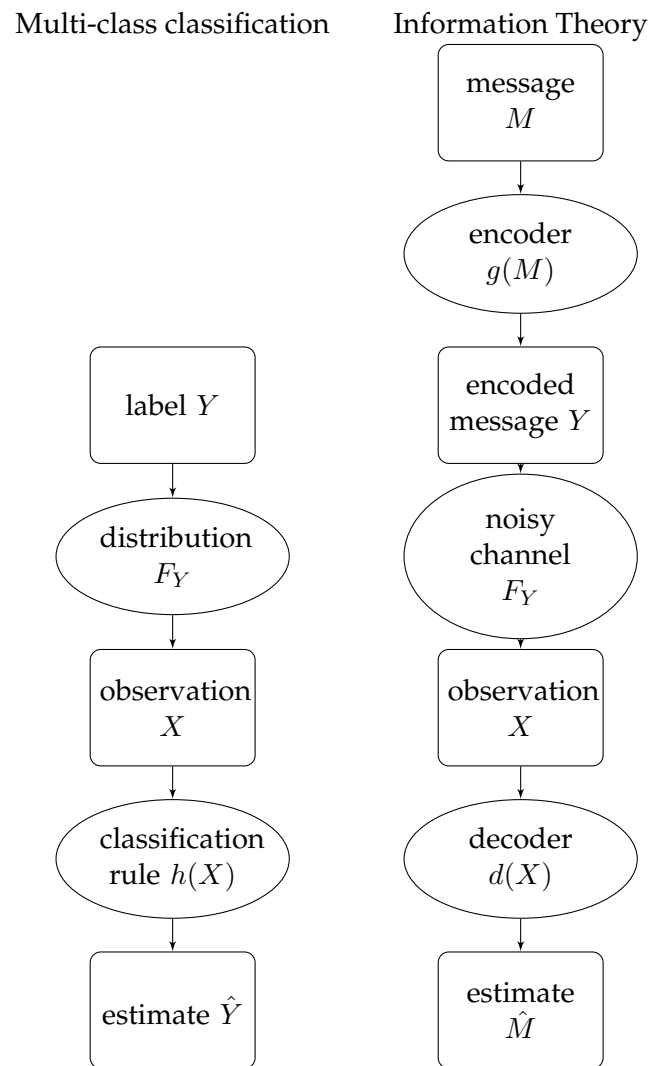
FIGURE 1.5: Comparing the discrimination tasks in multi-class classification and information theory.

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.

1.3.1 Supervised learning

Up until now we have been discussing *classification*, which is a particular type of *prediction task*. However, the most general recipe for a prediction task involves:

- 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, $C : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. 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 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 $\mathcal{Y} = \mathcal{Y}_1 \times \mathcal{Y}_2 \times \cdots \mathcal{Y}_\ell$. That is to say, that the response $\vec{Y}$ consists of a categorical vector, $\vec{Y} = (Y_1, \ldots, Y_\ell)$. 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 $h : \mathcal{X} \to \mathcal{Y}$ 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 $h : \mathcal{X} \to \mathcal{Y}$, the classification rule, is also a function! How learning algorithms are implemented in practice can very 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

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

  or equivalently,

$$\vec{X} \sim N((1, 0, \ldots, 0), \Sigma_X)$$
$$Y|\vec{X} \sim N(\vec{X}^T \beta, \Sigma_\epsilon).$$

  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: maximimum 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,

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

  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 is a penalized generative model which finds β minimizing the objective function

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

and then constructs the prediction rule

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

Here, the L1-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*)

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

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.

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

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

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

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

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 $\hat{\mathrm{sd}}$ 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 $\sigma : n \to n$, 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 $i = 1, \ldots, k$, we take the ith partition as the test set, and merge the 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 $\text{TestRisk}^{(1)}, \ldots, \text{TestRisk}^{(k)}$. 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 ith 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,

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

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.

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

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, $m_y : \mathcal{X} \to \mathbb{R}$ 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 ith label. The classification rule h, therefore, assigns points to the label having the highest margin for $\vec{x}$,

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

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* [CITE].

- *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

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

1.3.2 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 reciever, a *channel* that the sender can use in order to communicate to the reciever, 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 reciever can *decode* the signals recieved 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 reciever, 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 symbol output X is drawn from a distribution F_y that depends on y[2]. 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, $\vec{Y} = (Y_1, Y_2, \ldots, Y_L)$. The reciever will observe the output $\vec{X} = (X_1, X_2, \ldots, X_L)$, where each X_i is drawn from F_{Y_i} independently of the previous $X_1, \ldots, X_{i-1}$.

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 $\mathcal{M} = \{1, \ldots, m\}$. The message must be encoded into

---

[2]Note 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.

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* $g : \{1, \ldots, m\} \to \mathcal{Y}^L$ 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 $\{1, \ldots, m\}$ from the recieved signal $\vec{X}$. Theoretically speaking[3], a reasonable decoding scheme is the *maximum likelihood decoder*,

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

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

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

where $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ are the marginal densities with respect to μ_x and μ_y[4]. When the reference measure $\mu_x \times \mu_y$ is unambiguous, note that $\mathrm{I}(\boldsymbol{X}; \boldsymbol{Y})$ is simply a functional of the joint density $p(\boldsymbol{x}, \boldsymbol{y})$. Therefore, we can also use the *functional* notation

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

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

1. The channel input $\boldsymbol{X}$ and output $\boldsymbol{Y}$ can be random vectors of arbitrary dimension, and the mutual information remains a scalar functional of the joint distribution P of $(\boldsymbol{X}, \boldsymbol{Y})$.

2. When $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent, $\mathrm{I}(\boldsymbol{X}; \boldsymbol{Y}) = 0$; otherwise, $\mathrm{I}(\boldsymbol{X}; \boldsymbol{Y}) > 0$.

3. The data-processing inequality: for any vector-valued function $\vec{f}$ of the output space,
$$\mathrm{I}(\boldsymbol{X}; \vec{f}(\boldsymbol{Y})) \leq \mathrm{I}(\boldsymbol{X}; \boldsymbol{Y}).$$

4. Symmetry: $\mathrm{I}(\boldsymbol{X}; \boldsymbol{Y}) = \mathrm{I}(\boldsymbol{Y}; \boldsymbol{X})$.

---

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

[4]Note that the mutual information is invariant with respect to change-of-measure.

5. Independent additivity: if $(\boldsymbol{X}_1, \boldsymbol{Y}_1)$ is independent of $(\boldsymbol{X}_2, \boldsymbol{Y}_2)$, then

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

Three additional consequences result from the data-processing inequality:

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

$$\mathrm{I}(\boldsymbol{X}; \vec{f}(\boldsymbol{Y})) \le \mathrm{I}(\boldsymbol{X}; \boldsymbol{Y}).$$

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

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

  Then, by the data-processing inequality,

$$\mathrm{I}(\boldsymbol{X}; \boldsymbol{Y}) = \mathrm{I}(\boldsymbol{X}; (\boldsymbol{Y}, W)) \ge \mathrm{I}(\boldsymbol{X}; \vec{g}(\boldsymbol{Y}, W)) = \mathrm{I}(\boldsymbol{X}; \vec{f}(\boldsymbol{Y})).$$

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

$$\mathrm{I}(\boldsymbol{X}; \vec{f}(\boldsymbol{Y})) \le \mathrm{I}(\boldsymbol{X}; \boldsymbol{Y}) = \mathrm{I}(\boldsymbol{X}; \vec{f}^{-1}(\vec{f}(\boldsymbol{Y}))) \le \mathrm{I}(\boldsymbol{X}; \vec{f}(\boldsymbol{Y})),$$

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

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

$$\mathrm{I}(\boldsymbol{X}; \boldsymbol{Y}_1) \le \mathrm{I}(\boldsymbol{X}; (\boldsymbol{Y}_1, \boldsymbol{Y}_2)).$$

  Intuitively, if we observe both $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$, this can only *increase* the information we have about $\boldsymbol{X}$ compared to the case where we only observe $\boldsymbol{Y}_1$ 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:

$$\mathrm{H}_\mu(\boldsymbol{X}) = -\int p(\boldsymbol{X}) \log p(\boldsymbol{X}) d\mu(\boldsymbol{X}),$$

and the *conditional entropy*:

$$\mathrm{H}_\mu(\boldsymbol{X}|\boldsymbol{Y}) = -\int p(\boldsymbol{Y}) d\mu_y(\boldsymbol{Y}) \int p(\boldsymbol{X}|\boldsymbol{Y}) \log p(\boldsymbol{X}|\boldsymbol{Y}) d\mu_x(\boldsymbol{X}).$$

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 Lesbegue measure.) Intuitively, entropy measures an observer's uncertainty of the random

variable $\boldsymbol{X}$, supposing the observer has no prior information other than the distribution of $\boldsymbol{X}$. Conditional entropy measures the *expected uncertainty* of $\boldsymbol{X}$ supposing the observer observes $\boldsymbol{Y}$.

The following identities characterize mutual information in terms of entropy:

$$\mathrm{I}(\boldsymbol{X};\boldsymbol{Y}) = \mathrm{H}_{\mu_x \times \mu_y}((\boldsymbol{X},\boldsymbol{Y})) - \mathrm{H}_{\mu_x}(\boldsymbol{X}) - \mathrm{H}_{\mu_y}(\boldsymbol{Y}).$$

$$\mathrm{I}(\boldsymbol{X};\boldsymbol{Y}) = \mathrm{H}_{\mu}(\boldsymbol{Y}) - \mathrm{H}_{\mu}(\boldsymbol{Y}|\boldsymbol{X}). \tag{1.1}$$

The second identity (1.1) is noteworthy as being practically important for estimation of mutual information. Since the entropies in question only depend on the marginal and conditional distributions of $\boldsymbol{Y}$, the problem of estimating $\mathrm{I}(\boldsymbol{X};\boldsymbol{Y})$ can be reduced from a $\dim(\boldsymbol{X}) + \dim(\boldsymbol{Y})$-dimensional nonparametric estimation problem to a $\dim(\boldsymbol{Y})$-dimensional problem: hence this identity is a basis of several methods of estimation used in neuroscience, such as Gastpar (2014).

However, by symmetry, we also have the flipped identity

$$\mathrm{I}(\boldsymbol{X};\boldsymbol{Y}) = \mathrm{H}_{\mu}(\boldsymbol{X}) - \mathrm{H}_{\mu}(\boldsymbol{X}|\boldsymbol{Y}). \tag{1.2}$$

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

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

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, it may *never* be possible for the reciever to deterministically infer the original message. 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 $M = 2$.

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

The reciever should guess $M = 1$ if she recieves 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 mth 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 maximimum error probability over all messages $1, \ldots, m$ 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:

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

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

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

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 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 vanishing decoding error as $M \to \infty$[5], 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 $g(i) = (Y_1^{(i)}, \ldots, Y_L^{(i)})$ where $Y_j^{(i)}$ are iid draws from p_y^* for $i = 1, \ldots, M$ and $j = 1, \ldots, L$. Shannon proved that average decoding error, taken over the distribution of random codebooks, goes to zero as $M \to \infty$. This implies the existence of a deterministic sequence of codebooks with the same property, hence establishing (i).

---

[5]Shannon'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.

1.3.3 Comparisons

We see that in 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 were 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, Mickalstd 1980, $K = 15$ classes

- The NIST handwritten digits data set, Grother 1995, $K = 10$ classes

- Phoneme recognition on the TIMIT datset, 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 $K \to \infty$ 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 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.

1.4 Neuroscience applications

Both supervised learning and information theory are routinely used as tools in the study of human and animal brains. While Shannon's theory of information was motivated by the problem of designing communications system, the applicability of mutual information was quickly recognized by neuroscientists. Only four years after Shannon's seminal paper in information theory (1948), McKay and McCullough (1952) inaugurated the application of mutual information to neuroscience. Since then, mutual information has enjoyed a celebrated position in both experimental and theoretical neuroscience.

Supervised learning also has an extensive history of interaction with neuroscience. Notably, the development of artificial neural networks in supervised learning was motivated by models of biological neural networks. Examining the hierarchical nature of the visual system inspired the development of convolutional neural networks [CITE]. More recently, neuroscientists have begun to explore the possibility of using supervised learning models to model brain functionality. This approach is especially valuable for the problem of understanding specialization in the brain.

We now illustrate these use cases of information theory and supervised learning in a number of specific applications in neuroscience.

1.4.1 Selecting decoding models.

Neurons carry information via *spike trains*, which are temporal point processes. In response to stimulus $\boldsymbol{X}$, the neuron produces a spike train $Y(t)$ where $Y(t) = 1$ indicates a spike at time t and $Y(t) = 0$ indicates no spiking, for $t \in [0, T]$.

An open question in neuroscience that of how information is encoded in the spike train. Put loosely, what is 'signal' in the spike train $Y(t)$ and what is 'noise'? Presumably there exists some *decoder*–some function $\vec{g}$ of the time series, which compresses $Y(t)$ to a small dimension while preserving most of the information about $\boldsymbol{X}$.

Nelken et. al. (2005) investigated the neural code in the A1 auditory cortex of a cat, in response to recorded birdsongs. The stimulus $\boldsymbol{X} = (X_1, X_2)$ takes the form of 15 different auditory recordings presented in 24 spatial directions: $X_1 \in \{1, \ldots, 15\}$ indexes the recording and $X_2 \in \{1, \ldots, 24\}$ indexes the direction of presentation. The response $Y(t)$ takes the form of a spike train.

Nelkin et. al. compare the following *decoders* $\vec{g}$ in terms of the information $\mathrm{I}(\boldsymbol{X}; \vec{g}(Y(t)))$.

- The total spike count $\vec{g}_1(Y(t)) = \sum_t Y(t)$.

- The mean response time $\vec{g}_2(Y(t)) = \frac{1}{\sum_t Y(t)} \sum_t tY(t)$.

- The combination of the two codes: $\vec{g}_{1+2}(Y(t)) = (\vec{g}_1(Y(t)), \vec{g}_2(Y(t)))$.

The information of each decoder is compared to the full information of the signal, $\mathrm{I}(\boldsymbol{X}; Y(t))$, which is estimated via binning.

Nelkin et al. find that while the decoder $\vec{g}_1$ reduces the mutual information by 20 to 90 percent, the information loss from $\vec{g}_2$ is much less, and barely any information

at all is lost when both decoders are used jointly in $\vec{g}_{1+2}$. The scientific conclusion that can be drawn is that since $I(\boldsymbol{X}; \vec{g}_{1+2}(Y(t)))$ is not much smaller than $I(\boldsymbol{X}; Y(t))$, the "signal" in the spike train is mostly captured by the spike counts and response times: beyond that, the detailed temporal pattern of spiking is likely to be "noise." Of course, an important caveat to their conclusions is only *individual* neurons are considered: the analysis did not rule out the possibility that the temporal spiking pattern could yield information within an *ensemble* of neurons.

1.4.2   Inferring functional specialization

Historically, the earliest discoveries of specialized modules in the brain were due to lesion studies, where patients who had parts of their brain destroyed due to injury or clinical surgeries also lost cognitive or motor functionality as a result. The loss of functionality established a causal pathway between the lesioned area and the affected behavior–as, for example, in the case of the discovery of Broca's area, which was established in this way to be critical for speech production [CITE]. However, ethical and practical limitations restrict the use of lesioning as an experimental technique, and furthermore, lesion studies cannot be applied to exhaustively isolate the regions of the brain which are specialized for a given task.

Therefore, an alternative method of studying specialization is to acquire an image of the brain activity, and then to assess the discriminative performance of classifiers on either the whole brain, the brain minus a number of "lesioned" areas, or isolated regions of the brain by themselves. The classifier may achieve a certain accuracy using the whole brain image, and reduced accuracies using "lesioned" images– and, given the removal of the task-critical parts of the brain in the image, may be reduced to chance accuracy levels. Therefore, similar to lesion studies, these classification experiments can reveal specialization in the brain by establishing which parts of the brain image contain *information* related to the task. This type of classification study was first introduced to the neuroimaging community by Haxby (1999); approaches of these types are known as *multivariate pattern analyses* (MVPA) in the neuroscience literature. However, an important distinction between MVPA studies and actual lesion studies is that MVP analyses can only establish association, rather than causality, because actual lesions to the brain affect the activity patterns of other parts of the brain, and therefore the effect of lesions cannot actually be completely simulated by merely masking parts of the brain image.

Data Setup

While multivariate pattern analysis can be applied in a wide variety of brain imaging modalities, its dominant use has been in functional magnetic resonance imaging (fMRI.) Therefore, we will briefly describe the experimental setup and initial data analysis protocol for a typical task-fMRI experiment. For more details, the interested reader is invited to consult a standard reference such as Poldrack, Mumford, and Nichols 2011.

In task-fMRI experiments, a subject is presented with a timed sequence of behaviorial tasks to be performed, while the MRI scanner uses magnetic fields to measure blood oxygenation levels in the subject's brain. The raw imaging data is obtained in Fourier space. Initial preprocessing converts the data into a spatial time series consisting of one time dimension and three spatial dimensions, with measure of BOLD (blood-oxygenation level dependent) signal at each point in the brain. Further preprocessing steps reduce noise due to head motion, physiologically induced artifacts,

Task	Stimulus		Activation Map	
1	$y_1 = 1$		$\vec{x}_1 =$	
2	$y_2 = 2$		$\vec{x}_2 =$	
3	$y_3 = 1$		$\vec{x}_3 =$	
4	$y_4 = 2$		$\vec{x}_4 =$	

FIGURE 1.6: Data output of a typical analysis of a task fMRI experiment. The raw data is processed to obtain a parametric map of inferred activation levels to each stimulus. These activation levels are further utilized in downstream analyses.

and electromagnetically induced artifacts. A generalized linear model (GLM) is fitted to the spatial time series in order to infer an aggregated activity level for each specific task. We do not go into details of the GLM model; however, the output of the GLM model is an inferred activity level map for each task.

Figure 1.6 illustrates a cartooon schematic of the output of the GLM fitting stage. There are four tasks in the illustrated experiment. Each task y_i belongs to two different categories, coded as 1 and 2. Task 1 is a visual task, where the subject is presented an picture of an eagle, and asked to focus on the image. Task 2 is the same type of visual task, but with a picture of a bear instead. For each task $i = 1, \ldots, 4$, the GLM produces a parametric map of inferred activation levels, $\vec{x}_i$. The map $\vec{x}_i$ is divided into uniformly sized cubical regions in the brain, called *voxels*. We denote the number of voxels in the map by q; for typical resolutions ($2\text{mm} \times 2\text{mm} \times 2.5\text{mm}$), the map has on the order of $q = 20,000$ voxels. Each voxel has an inferred activity level. Therefore, the map $\vec{x}_i$, when represented as a numerical vector, is a vector in $\mathbb{R}^q$ where each component is the activity level of a particular voxel in the brain.

A number of statistical analyses can be performed on the task-activation pairs $(y_i, \vec{x}_i)$. The most common type of analysis tests for significant *peaks* of task-correlated activity. These analyses involve *global* tests of the null hypothesis of no correlation between task and activity. However, MVPA analyses tend to begin with local tests of independence between a given region-of-interest (a cluster of neighboring voxels) and the task.

Classification-based test of independence

A region of interest (ROI) is a collection of neighboring voxels, chosen based on anatomical or geometric considerations. To give a concrete example, one can choose an arbitrary voxel in the brain, and define an ROI as the set of all voxels within a given radius (say, 5mm) of the chosen voxel. Recalling that $\vec{x}_i$ denotes the activity level of the entire brain, let us write $\vec{x}_i^{ROI}$ for the subset of voxels in the brain map belonging to the ROI.

We would like to test the hypothesis of independence between the task Y and the joint activity level of the voxels belonging to the ROI, considered as a random vector, $\vec{X}^{ROI}$. While there are numerous methods in the statistical literature for testing for

independence between a categorical random variable Y and a random vector $\vec{X}$, the approach taken in MVPA is typically to use classification accuracies as the test statistic.

Recall that a classification rule is any (possibly stochastic) mapping $f : \mathcal{X} \to \{1, \dots, k\}$, where k is the number of levels of Y. Let us assume that the k different levels of Y have the same number of repeats within the data. The *generalization accuracy* of the classification rule is

$$\mathrm{GA}(f) = \frac{1}{k} \sum_{i=1}^{k} \Pr[f(\vec{X}^{ROI}) = i | Y = i].$$

A trivial classification rule which outputs the result of a k-sided die roll for all inputs y would achieve a generalization accuracy of $\mathrm{GA} = \frac{1}{k}$.

Recall that the generalization accuracy of a data-dependent classification rule f can be obtained by using *data-splitting*, provided that the rule f is constructed using only the *training data*. Supposing that the original data consists of pairs $(y_i, \vec{x}_i^{ROI})$ for $i = 1, \dots, n$, let $j_1, \dots, j_{n_1}$ denote the n_1 randomly drawn training indices, and $\ell_1, \dots, \ell_{n_2}$ denote the n_2 randomly drawn test indices. The training data $(y_{j_i}, \vec{x}_{j_i}^{ROI})_{i=1}^{n_1}$ is used to construct a classification rule f, while the test data is used to obtain a test accuracy T:

$$T = \frac{1}{n_2} \sum_{i=1}^{n_2} I(f(\vec{x}_{\ell_i}^{ROI}) = y_{\ell_i}).$$

Under the null hypothesis $H_0 : Y \perp \vec{X}^{ROI}$, the generalization accuracy of any classification rule f is equal to $1/k$. Therefore, we can write the null hypothesis as

$$H_0 : \mathrm{GA}(f) = \frac{1}{k}$$

and the alternative hypothesis as

$$H_1 : \mathrm{GA}(f) > \frac{1}{k}.$$

We will describe two different methods for testing H_0 versus H_1. The first approach is to use the fact that the distribution of T is known under the null hypothesis. The second approach is a permutation test[6]. The permutation test is more computationally expensive, but has important advantages in terms of family-wise error control, as we will describe in the sequel.

First approach. Under the null hypothesis, using the fact that the test observations $(y_{\ell_i}, \vec{x}_{\ell_i})$ are identical and independently distributed from the population distribution of stimulus-activity pairs, we have

$$n_2 T \sim_{H_0} \mathrm{Bernoulli}(n_2, \frac{1}{k})$$

Therefore, let c_α be the $(1 - \alpha)$ quantile of a $\mathrm{Bernoulli}(n_2, \frac{1}{k})$ distribution. To perform a hypothesis test at level α, reject the null H_0 if $n_2 T > c_\alpha$ and accept otherwise. Equivalently, define the p-value as the area under the tail of the $\mathrm{Bernoulli}(n_2, \frac{1}{k})$

---

[6]Permutation tests are widely used in statistical applications: for a good introduction to the subject, the reader is invited to consult Efron and Tibshirani 1994.
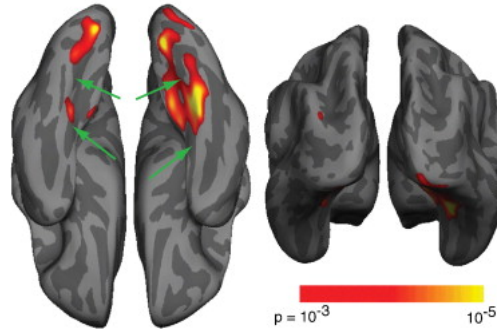
FIGURE 1.7: p-value map obtained from linear SVM-based searchlight applied to object perception data, using spherical neighborhoods with 9mm radius (Chen et al. 2011).

distribution to the right of T,

$$p = \Pr[X > n_2 T | X \sim \text{Bernoulli}(n_2, \frac{1}{k})],$$

and reject if and only if $p < \alpha$.

Second approach (permutation test). Under the null hypothesis of independence between Y and $\vec{X}^{ROI}$, the test statistic T (the test accuracy on n_2 test observations) has a distribution that is invariant under permutation of the labels $Y_1, \ldots, Y_n$. Therefore, obtain B samples of permutation test statistics $T^{(1)}, \ldots, T^{(B)}$ by the following procedure:

1. For $j = 1, \ldots, B$, draw a random permutation $\sigma : n \to n$.

2. Use the pairs $(y_{\sigma_i}, \vec{x}_{\sigma_i}^{ROI})_{i=1}^{n_1}$ to construct classification rule $f^{(j)}$ (using the same classifier as the one used to construct f).

3. Evaluate the test accuracy of $f^{(j)}$,

$$T^{(j)} = \frac{1}{n_2} \sum_{i=1}^{n_2} I(f^{(j)}(\vec{x}_{\sigma_i}^{ROI}) = y_{\sigma_i}).$$

The permutation p-value is calculated based on the rank of T within the sample of permuation test statistics:

$$p = \frac{\sum_{j=1}^{B} I(T^{(j)} < T) + 1}{B + 1},$$

and reject the null hypothesis if and only if $p < \alpha$.

Searchlight analysis

Classifier-based tests of independence are commonly combined with the *multivariate searchlight* procedure (Kriegeskorte, Goebel, and Bandettini 2006, Kriegeskorte and Bandettini 2007) in order the scan the entire brain for informative clusters of voxels. The procedure produces a 3D map of p-values in the brain, with significant p-values indicative of regions in the brain associated with the task. Figure 1.7 displays one example of a p-value map obtained from an object perception task (Chen et al. 2011).

A *searchlight* refers to a local region-of-interest centered around a particular *central voxel*. A variety of methods can be used to define the neighborhood. The most straightforward is the *spherical searchlight*, which fixes a radius r, and to include all voxels within a distance r of the central voxel. However, more sophisticated approaches may account for the geometry of the brain surface (Chen et al. 2011.)

Having chosen a method for obtaining searchlights, one generates a searchlight centered around each voxel in the fMRI image, $\vec{x}$. Let $\{ROI_1, \ldots, ROI_v\}$ denote the collection of such searchlights, with $\vec{X}^{ROI_i}$ consisting of the voxels centered around the ith voxel.

For each ROI_i, one conducts a hypothesis test for independence between Y and $\vec{X}^{ROI_i}$ using the local test of independence described earlier. of p-values, $p_1, \ldots, p_v$, for each voxel in the fMRI image. Clusters of small p-values can be informally interpreted as indicative of brain regions associated with the task. Alternatively, to test the global null hypothesis that all ROIs are independent of the task, one can apply the permutation testing approach simultaneously to all voxels in the brain. The global permuation approach is carried out as follows:

0. (Non-permuted data): Draw a random permutation $\sigma : n \to n$. For $i = 1, \ldots, v$, use the pairs $(y_{\sigma_\ell}, \vec{x}^{ROI_i}_{\sigma_\ell})^{n_1}_{\ell=1}$ to construct classification rule $f^{(0,i)}$

1. For $j = 1, \ldots, B$, draw a random permutation $\sigma : n \to n$.

2. For $i = 1, \ldots, v$, use the pairs $(y_{\sigma_\ell}, \vec{x}^{ROI_i}_{\sigma_k})^{n_1}_{\ell=1}$ to construct classification rule $f^{(j,i)}$ (using the same classifier as the one used to construct f).

3. Evaluate the test accuracy $T^{(j,i)}$ of $f^{(j,i)}$.

4. Find the maximum accuracy $T^{(j)} = \max_{i=1}^{v} T^{(j,i)}$.

5. Define the p-value for each ROI by

$$p_i = \frac{\sum_{j=1}^{B} I(T^{(j)} < T^{(0,i)}) + 1}{B + 1},$$

Since the permutations are shared across the ROIs, familywise control is attained, in the sense that $p^* = \min_i p_i$ is a valid p-value.

1.4.3 Other applications

Experimentally, mutual information has been used to detect strong dependencies between stimulus features and features derived from neural recordings, which can be used to draw conclusions about the kinds of stimuli that a neural subsystem is designed to detect, or to distinguish between signal and noise in the neural output. Theoretically, the assumption that neural systems maximize mutual information between salient features of the stimulus and neural output has allowed scientists to predict neural codes from signal processing models: for instance, the center-surround structure of human retinal neurons matches theoretical constructions for the optimal filter based on correlations found in natural images [cite].

1.4.4 Can classification accuracy quantify information?

Due to the analogy between the nervous system and a communications network, it was a natural choice for early neuroscientists to adopt Shannon's *mutual information*

as a quantitative measure of dependence, as we saw in the application of mutual information to decoding models (Section 1.4.1). But as new technologies enabled the recording of neural data at larger scales and resolution, the traditional reductionist goals of neuroscience were supplemented by increasingly ambitious attempts within neuroscience to understand the dynamics of neural ensembles, and by efforts originating within psychology and medicine to link the structure and function of the entire human brain to behavior or disease. The scale of the data, and the complexity of the models studied posed an obstacle to traditional information-theoretic approaches. Therefore, it was of considerable interest when Haxby (2001) introduced the usage of *supervised learning* (classification tasks) for the purpose of quantifying stimulus information in task fMRI scans, as we discussed in section 1.4.2.

Judging from the language used by the practioners themselves, it is intuitively clear to them how classification accuracies can be used to quantify information in brain scans. We agree with the intuition behind this practice, and go further to say that the use of classification accuracies as a measure of information can be formally justified. However, having said this, it is important to remark that Shannon's *mutual information* need not be taken as the canonical quantification of "information" as a measure of multivariate dependence. Instead, we take a step back and attempt to dissect the intuitive properties of "information" and why the concept is useful in science. We will argue that given these intuitions, both mutual information and classification accuracy can be seen as useful measures of information.

- *Intuition 1: Information is a measure of dependence.* If X and Y are statistically independent, then X gives no information about Y, and vice-versa.

This intuition is employed when researchers test the null hypothesis of chance accuracy for classification. If the null is accepted, the researcher concludes that there is no information in the predictors about the response.

- *Intuition 2a: Monotonicity with respect to inclusion of outputs.* If X_1 and X_2 are ensembles of neurons (or individual neurons), then the combined ensemble (X_1, X_2) has equal or more information about Y than either component by itself.

- *Intuition 2b: Noise adds no information.* If, in the previous example, X_2 is independent of Y, then X_2 adds no information to the ensemble.

The non-informativity of noise is vital for the purpose of *localizing* information within fMRI voxels (as is done in searchlight analysis.) Since noise voxels fail to improve classification performance (and indeed, sometimes harm empirical performance,) the optimal searchlight radius will concentrate on clusters of signal voxels, and minimize the inclusion of noise voxels.

- *Intuition 3:* Information can be used as a basis of model selection. Among multiple encoding/decoding models, a more accurate model should tend to have greater information relative to less accurate models.

Compared to the first two, this third intuition is somewhat less obvious, but nevertheless appears as an important use-case for mutual information, as seen in the application of mutual information to choose encoding models.

It is well-known that the mutual information $I(X; Y)$ satisfies these three intuitions. However, the *Bayes classification accuracy*–the accuracy of the *optimal* classifier, also satifies these three intuitions.

1. When X and Y are independent, then the Bayes accuracy falls to chance, $1/k$, indicating 'no information.'

2. As additional predictors are made available, the Bayes accuracy can only improve. As noise predictors are addded, the Bayes accuracy remains unchanged.

3. Supposing that there exists a 'true encoding' with Y only depending on some function basis $\vec{b}(X)$, then the Bayes accuracy for predicting Y from the basis $\vec{g}(X)$ is maximized when $\vec{g} = \vec{b}$.

However, an important caveat is that classification accuracies attained in practice are at best an *underestimate* of Bayes accuracy, and therefore do not perfectly inherit all of the above properties. For instance, while the Bayes accuracy satisfies Intuition 2b, classification accuracy generally does not: due to the phenomenon of overfitting, the inclusion of noise predictors reduces classification accuracy.

On the other hand, given sufficient training observations and well-specification of models, classification accuracies begin to approach Bayes accuracies, and therefore can be justifiable used as measures of information which satisfy Intuitions 1-3. This link between empirical classification accuracies, Bayes accuracies, and information forms the basis for much of the work in this thesis.

That said, a deeper understanding of the conditions which allow attained classification accuracies to mimic the properties of Bayes classification accuracy would be greatly useful for understanding the practical limitations of some of the methods that will be proposed in the later chapters. However, we must leave this important work for future research.

1.5   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.

Chapter 2

Randomized classification

2.1 Motivation

However, the formalism of classification is inadequate for studying many practical questions related to the generalizability of the facial recognition system. We can define a population risk (also called generalization error) for the classification rule, and make inferences about the population risk based on the test performance. However, the population risk and associated inferences apply only to the particular collection of individuals $\{y^{(1)}, \ldots, y^{(k)}\}$. 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 $\{y^{(1)}, \ldots, y^{(k+1)}\}$ defines a different response space than the old set of labels $\{y^{(1)}, \ldots, y^{(k)}\}$. Yet, these two classification problems are clearly linked. To take another example, a client might want to run the facial recognition system on their own database of individuals. In this case, there might be no overlap between the first set of labels (the people in my database) and the second set of labels (the people in the client's database.) And yet, the client might still expect the performance of the system on our database to be informative of how well it will do on the second set of labels!

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.

Our solution to the problems (a) and (b) is to study the *average risk* of a learning algorithm on a *randomized classification task*. We will motivate and define the concepts of *average risk* and *randomized classification task* shortly. In any case, we formalize the problem of generalizability between related classification tasks by formalizing it as the question of how to *infer* average risk for a target regime given data from a different regime. However, yet another application of our concepts is to understand the properties of *identification risk*, which was introduced as a model-selection criteria for encoding models in a functional MRI study (Kay et al. 2008). We introduce and analyze the identification risk using our concepts 2.6.

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.); for now, let us take the loss function as zero-one loss. 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 *risk* 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,

$$\mathrm{Risk}_P(\Lambda) = \mathbf{E}[\mathrm{Risk}(h)]$$

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 an extremely 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 $\mathrm{Risk}_{P_1}(\Lambda)$ and $\mathrm{Risk}_{P_2}(\Lambda)$? 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 a 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 availible 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.

2.2.1 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. Furthermore, there exists a prior distribution π on $\mathcal{Y}$. Also, suppose for now that we are concerned with the zero-one cost function

$$C(y; y') = I(y \neq y').$$

A random classification task P can be generated as follows. The label set $\mathcal{S} = \{Y^{(1)}, \ldots, Y^{(k)}\}$ is generated by drawing labels $Y^{(1)}, \ldots, Y^{(k)}$ 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 $X^{(i)}_{j,train}$ i.i.d. from $F_{Y^{(i)}}$ for $j = 1, \ldots, r_1$. The test set is likewise obtained by sampling r_2 observations $X^{(i)}_j$ i.i.d. from $F_{Y^{(i)}}$ for $j = 1, \ldots, r_2$. For notational convenience, we represent the training set as the set of empirical distributions $\{\hat{F}_{Y^{(i)}}\}_{i=1}^k$ where

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

Figure 2.1 illustrates the sampling scheme for generating the training set.
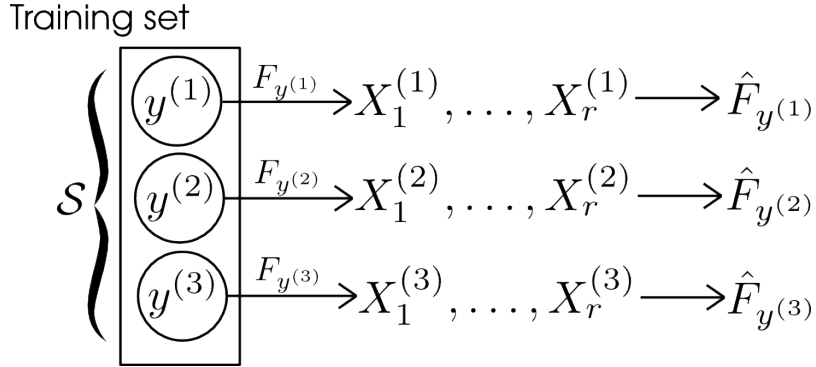
FIGURE 2.1: Training set

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*

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

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

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

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

   Comments:

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

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

   The resulting empirical approximation for the Bayes classifier (further assuming a uniform prior π) would be

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

2. Both the Quadratic Discriminant Analysis and the naive Bayes classifiers can be seen as specific instances of an estimated Bayes classifier [1]. For QDA, the margin function is given by

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

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

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

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

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

   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.

---

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

Classification Rule

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

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

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

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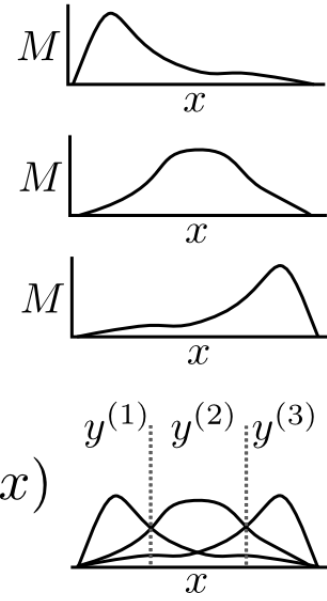

FIGURE 2.2: Classification rule

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 prediction risk of the classifier is deterministic:

$$\text{Risk}(h) = \Pr[Y \neq h(X) | Y \sim \text{Unif}(\mathcal{S}), \mathcal{S}, \{\hat{F}_{y^{(i)}}\}_{i=1}^{k}]$$

$$= \frac{1}{k} \sum_{i=1}^{k} \Pr[m_{y^{(i)}}(x) \neq \max_{j} m_{y^{(j)}}(x) | X \sim F_{y^{(i)}}, \mathcal{S}, \{\hat{F}_{y^{(i)}}\}_{i=1}^{k}].$$

$$= \frac{1}{k} \sum_{i=1}^{k} I(m_{y^{(i)}}(x) \neq \max_{j} m_{y^{(j)}}(x)) dF_{y^{(i)}}(x).$$

If we *fix* a particular realization of the random label set $\mathcal{S} = \{y^{(1)}, \dots, y^{(k)}\}$, then we can define the *classification risk* 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 risk of the classifier $\mathcal{M}$ on label set $\mathcal{S}$ is given by

$$\text{Risk}_{\mathcal{S}}(\mathcal{M}) = \Pr[Y \neq h(X) | Y \sim \text{Unif}(\mathcal{S}), \hat{F}_{y^{(i)}} \sim \Pi_{y^{(i)}, r_1}]$$

$$= \frac{1}{k} \sum_{i=1}^{k} \int I(\mathcal{M}(\hat{F}_{y^{(i)}})(x) \neq \max_{j} \mathcal{M}(\hat{F}_{y^{(j)}})) dF_{y^{(i)}}(x) \prod_{\ell=1}^{k} d\Pi_{y^{(\ell)}, r_1}(\hat{F}_{y^{(\ell)}}).$$

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

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 risk* of a marginal classifier $\mathcal{M}$
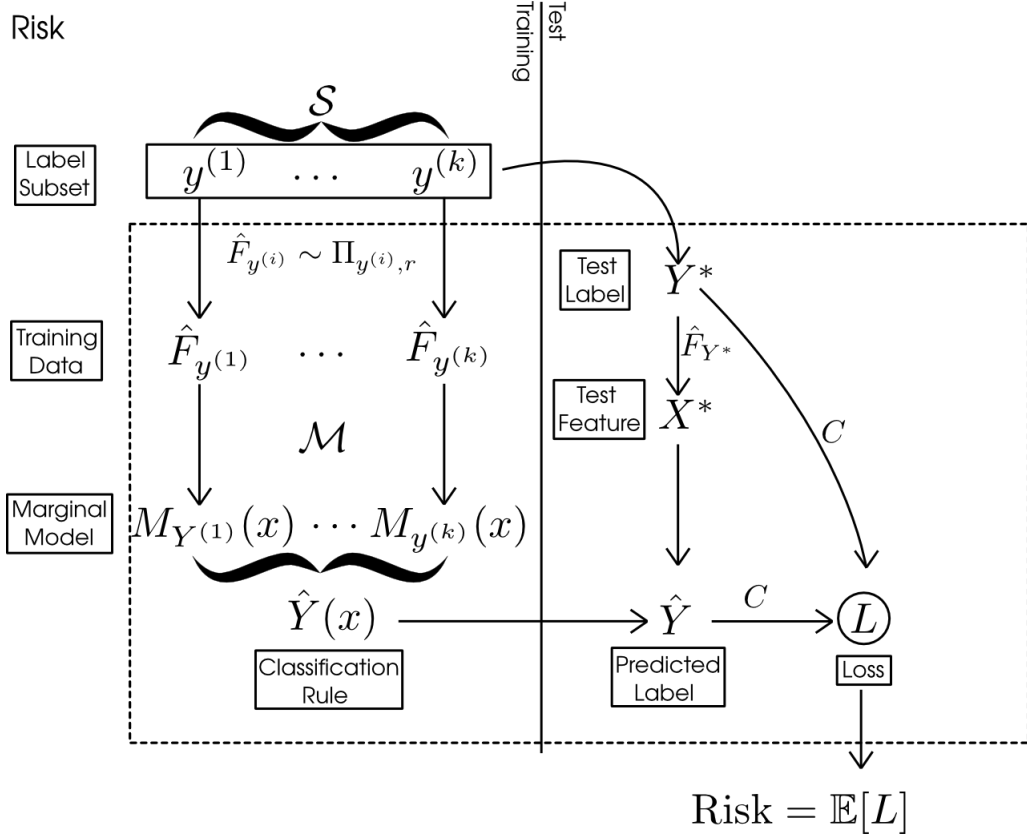
FIGURE 2.3: Classification risk

is defined as

$$\text{AvRisk}_{k,r_1}(\mathcal{M}) = \mathbf{E}[\text{Risk}_{\mathcal{S}}(\mathcal{M})|Y^{(1)},\dots,Y^{(k)} \sim \pi]$$

$$= \frac{1}{k}\sum_{i=1}^{k}\int I(\mathcal{M}(\hat{F}_{y^{(i)}})(x) \neq \max_{j}\mathcal{M}(\hat{F}_{y^{(j)}}))dF_{y^{(i)}}(x)\prod_{\ell=1}^{k}d\Pi_{y^{(\ell)},r_1}(\hat{F}_{y^{(\ell)}})d\pi(y^{(\ell)}).$$

The definition of average risk is illustrated in Figure 2.4.

Having defined the average risk for the randomized classification task, we begin to develop the theory of how to *estimate* the average risk 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 $(x_1^{(i)},\dots,x_{r_{test}}^{(i)})$ for $i = 1,\dots,k_1$. How can we estimate the k,r-average risk 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 risk of h is an unbiased estimator of $\text{AvRisk}_{k,r}$.

This follows from definition. Observe that AvRisk_{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
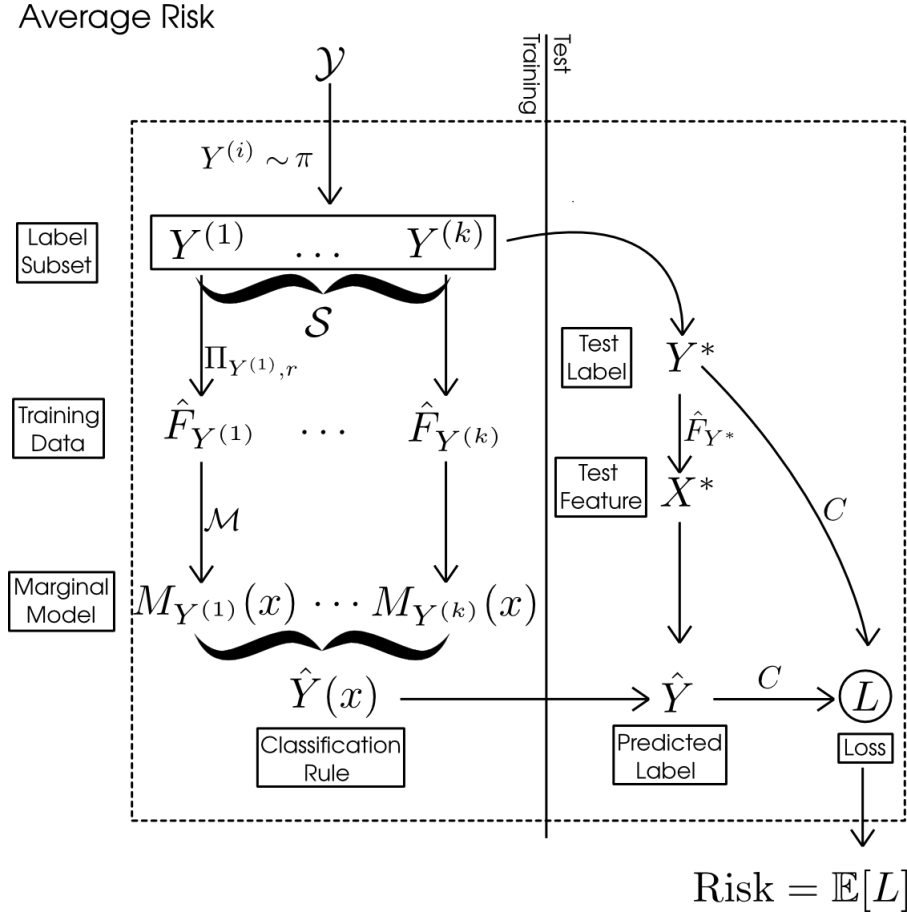
Average Risk

FIGURE 2.4: Average risk

been given is a random drawn from the dersired distribution of random classification problems. Therefore, the prediction risk of h constructed from P_1 is unbiased for AvRisk_{k_1, r_1}, and since test risk is unbiased for prediction risk, it follows that the test risk of h is an unbiased estimator of $\text{AvRisk}_{k,r}$, 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 risk.

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

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

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

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

The test risk is the average cost over test observations,

$$\text{Test Risk} = \frac{1}{r_{test} k} \sum_{i=1}^{k} \sum_{j=1}^{r_2} C(\hat{y}_j^{(i)}; y^{(i)}). \tag{2.1}$$

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 ℓ if and only if $R_{i,j,\ell} = k$. Thus, an equivalent expression for test risk is

$$\text{Test Risk} = \frac{1}{r_2 k} \sum_{i=1}^{k} \sum_{\ell=1}^{k} \sum_{j=1}^{r_2} C_{ij} I\{R_{ij\ell} = k\}. \tag{2.2}$$

where

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

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 risk for a classification rule h constructed using the training data in P_2 provides an unbiased estimate of AvRisk_{k,r_1}.

However, we can get a much better unbiased estimate of AvRisk_{k,r_1} by averaging over the randomization of $\mathcal{S}_2$. Na ively, 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.)

The reason why the efficient computation is possible is because the test risk for each label subset $\mathcal{S}_2$ can be determined by looking at the margins $M_{i,j,\ell}$, which remain the same as long as both i and ℓ are included in the subsample $\mathcal{S}_2$.

The computational trick is to look at each combination of test observation $x_j^{(i)}$ and class label $y^{(\ell)}$, and to count the number of subsets $N_{i,j,\ell}$ where (i) both i and ℓ are included in $\mathcal{S}_2$, and (ii) $\hat{y}_j^{(i)} = y^{(\ell)}$. Then it should be clear that the average test risk over all subsets is equal to

$$\text{AvTestRisk}_{k,r_1} = \frac{1}{\binom{k_1}{k}} \frac{1}{r_2 k} \sum_{i=1}^{k_1} \sum_{\ell \neq i} \sum_{j=1}^{r_2} C_{i\ell} N_{i,j,\ell}. \tag{2.3}$$

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

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

$$N_{i,j,\ell} = I\{R_{i,j,\ell} > R_{i,j,i}\} \binom{R_{i,j,\ell} - 2}{k_2 - 2} \tag{2.4}$$

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 risk 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. To make the problem of analyzing the variance of estimates $\hat{\text{AvRisk}}$ easier, we make the unrealistic assumption of *infinite* training and test data. This simplifies the problem because then we can neglect the variability in the tdata, and isolate the contribution of the variability due to label subsampling.

The effect of assuming infinite training and test data is equivalent to supposing that we have access to the *Bayes classification rule* and its risk for any given classification task. Therefore, our task in the following section is the understand the variability of the *Bayes* risk, or equivalently, its, complement, the *Bayes accuracy.*

2.4 Average Bayes accuracy

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

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

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

2.4.1 Definitions

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

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

average Bayes accuracy is defined as

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

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

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

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

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

as stated in the following theorem.

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

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

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

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

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

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

To show this claim, write

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

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

We therefore have

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

Therefore, we can write

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

$$= \frac{1}{k}\int p_X(x_1)\ldots p_X(x_k)p(y|x_{f(x_1,...,x_k,y)})dx_1\ldots dx_k dy.$$

$$= \frac{1}{k}\int p_X(x_1)\ldots p_X(x_k)\max_{i=1}^{k} p(y|x_i)dx_1\ldots dx_k dy.$$

2.5 Variability of Bayes Accuracy

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

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

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

We have the following upper bound on the variability.

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

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

Proof. According to the Efron-Stein lemma,

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

which is the same as

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

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

Note the following trivial results

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

This implies

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

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

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

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

2.5.1 Inference of average Bayes accuracy

2.5.2 Classification without model selection

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

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

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

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

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

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

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

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

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

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

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

Therefore,

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

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

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

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

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

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

Combining these facts, we get the following result.

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

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

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

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

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

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

and hence

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

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

2.5.3 Classification with model selection

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

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

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

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

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

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

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

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

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

2.6 Identification task

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

2.6.1 Experimental design

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

2.6.2 Data splitting

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

the $N \times p$ matrix of test stimuli and $\boldsymbol{Y}^{te}$ as the $N \times q$ matrix of corresponding test responses.

2.6.3   Probabilistic encoding model

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

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

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

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

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

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

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

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

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

where

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

2.6.4   Converting the encoding model to a decoding model

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

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

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

probability of each candidate stimulus $\vec{x}^{(i)}$ is given by

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

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

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

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

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

2.6.5 Computation of identification accuracy curve

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

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

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

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

Furthermore, define

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

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

$$\text{EmpAcc}_k = 1 - \frac{1}{\binom{M}{k}} \frac{1}{k} \sum_{i=1}^{k} \sum_{\ell \neq i} C_{i\ell} N_{i,\ell}. \tag{2.5}$$

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

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

$$N_{i,\ell} = I\{R_{i,\ell} > R_{i,i}\} \binom{R_{i,\ell} - 2}{k - 2} \tag{2.6}$$

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

Chapter 3

Extrapolating average accuracy

3.1 Introduction

In this chapter, we address the question of how to estimate the k-class average risk, $\text{AvRisk}_{k,r}$, based on data from a problem P_1 with $k_1 < k$ classes, and $r = r_1$-repeat training data. We continue with the framework introduced in Chapter 2, including the definition of the average risk and the marginal property of classifiers. However, we generalize the cost function $C(y; \hat{y})$ from the zero-one cost function to a general nonnegative cost function.

   We begin by continuing to analyze the average risk $\text{AvRisk}_{k,r}$, which results in an explicit formula for the average risk. The formula reveals that all of the information needed to compute the average risk is contained in a one-dimensional function $\bar{D}(u)$, and therefore that estimation of the average risk 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 will see the comparison of 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 risk $\text{AvRisk}_{k,r}$ as the weighted average of a function $\bar{D}(u)$, where $\bar{D}(u)$ is independent of k, and where k only changes the weighting. The result is stated as follows.

Theorem 3.2.1 *Suppose π, $\{F_y\}_{y \in \mathcal{Y}}$ and marginal classifier $\mathcal{F}$ satisfy the tie-breaking condition. Then, under the definitions* (3.3), (3.6), *and* (3.7), *we have*

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

   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 $\mathcal{M}$.

Let us revisit the definition of the average risk, under a general cost function C. The k-class, r-repeat *average risk* of a marginal classifier $\mathcal{M}$ is defined as

$$\text{AvRisk}_{k,r_1}(\mathcal{M}) = \mathbf{E}[\text{Risk}_{\mathcal{S}}(\mathcal{M})|Y^{(1)},\dots,Y^{(k)} \sim \pi]$$

$$= \frac{1}{k}\sum_{i=1}^{k}\int I(\mathcal{M}(\hat{F}_{y^{(i)}})(x) \neq \max_{j}\mathcal{M}(\hat{F}_{y^{(j)}}))dF_{y^{(i)}}(x)\prod_{\ell=1}^{k}d\Pi_{y^{(\ell)},r_1}(\hat{F}_{y^{(\ell)}})d\pi(y^{(\ell)}).$$

The definition of average risk is illustrated in Figure 3.1.

FIGURE 3.1: Average risk

As we can see from Figure 3.1, the average risk is obtained by averaging over four randomizations:

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

A2. Drawing the training dataset.

A3. Drawing Y^* uniformly at random from $\mathcal{S}$.

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

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

point (X^*, Y^*). Formally, we write

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

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

$$\text{AvRisk}_{k,r,\nu}(\mathcal{F}) = \mathbf{E}[L] = \mathbf{E}[C(\mathcal{F}(\{\hat{F}_y\}_{y \in \mathcal{S}})(X^*), Y^*)]. \tag{3.2}$$

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

Our strategy is to analyze the average risk (3.2) by means of *conditioning on* the true label and its training sample, $(y^*, \hat{F}_{y^*})$, and the test feature x^* while *averaging* over all the other random variables. Define the *conditional average risk* $\text{CondRisk}_k((y^*, \hat{F}_{y^*}), x^*)$ as

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

Figure 3.2 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 $\mathcal{S} = \{Y^*, Y^{(1)}, \ldots, Y^{(k-1)}\}$. Note that due to independence, $Y^{(1)}, \ldots, Y^{(k-1)}$ are still i.i.d. from π even conditioning on $Y^* = y^*$. Therefore, the conditional risk can be obtained via the following alternative order of randomizations:

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

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

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

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

and hence

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

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

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

$$U_{x^*}(y^*, \hat{F}_{y^*}) = \Pr[M_{y^*}(x^*) > \mathcal{M}(\hat{F}_Y)(x^*)] \tag{3.3}$$

$$= \int_{\mathcal{Y}} I\{M_{y^*}(x^*) > \mathcal{M}(\hat{F}_y)(x)\} d\Pi_{y,r}(\hat{F}_y) d\pi(y). \tag{3.4}$$

See also figure 3.3 for an graphical illustration of the definition.

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

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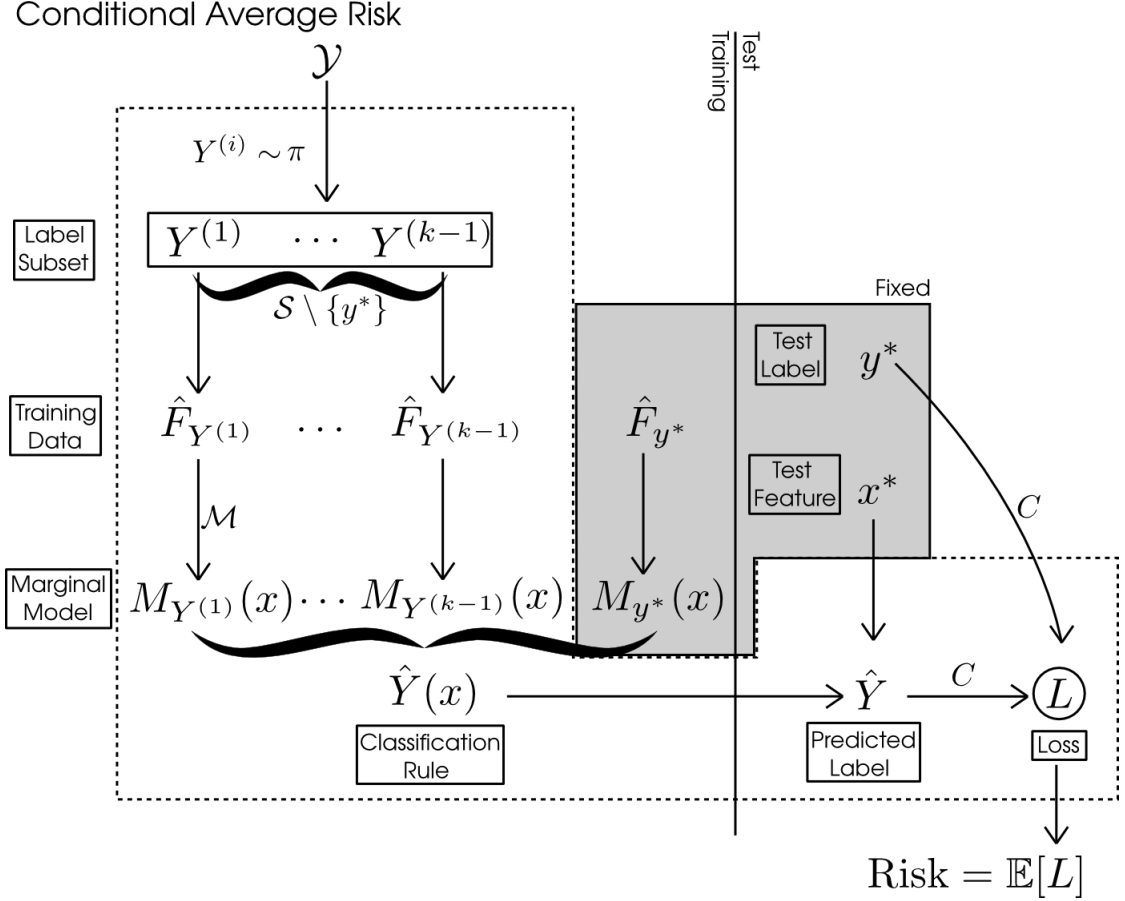
Conditional Average Risk

FIGURE 3.2: Conditional average risk

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

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

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

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

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

$$U_{max,k-1} \sim \text{Beta}(k-1, 1). \tag{3.5}$$

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

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

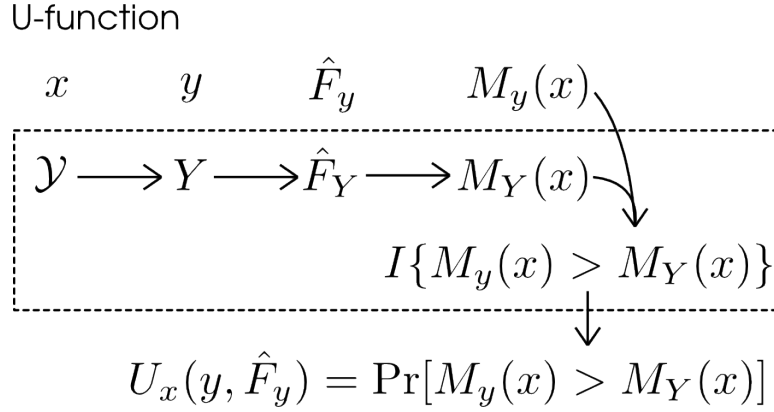$$\text{CondRisk}_k((y^*, \hat{F}_{y^*}), x^*) = \Pr[U_{max} > u^*] = \int_{u^*}^1 (k-1)u^{k-2} du.$$

U-function

$$U_x(y, \hat{F}_y) = \Pr[M_y(x) > M_Y(x)]$$

FIGURE 3.3: U-functions

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

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

Or equivalently,

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

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

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

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

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

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

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

joint density of $Y^{(1)}, \ldots, Y^{(k-1)}$ given $U_{max} = u$ can be written

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

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

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

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

$$D((y^*, \hat{F}_{y^*}), x^*, u) = \begin{cases} 0 \text{ if } u < u^* \\ \mathbf{E}[C(\hat{Y}, y^*)|U_{max} = u, x^*, y^*, \hat{F}_{y^*}] \text{ otherwise.} \end{cases} \tag{3.6}$$

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

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

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

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

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

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

where

$$\bar{D}(u) = \int D((y^*, \hat{F}_{y^*}), x^*, u) \pi(y^*) dy dF_{y^*}(x^*) d\Pi_{y^*,r}(\hat{F}_{y^*}). \tag{3.7}$$

This is the key result behind our estimation method, which was stated in theorem 3.2.1. The proof is given in the appendix.

Having this theoretical result allows us to understand how the expected k-class risk scales with k in problems where all the relevant densities are known. However, applying this result in practice to estimate Average Risk$_k$ requires some means of estimating the unknown function $\bar{D}$–which we discuss in the following.

3.3 Estimation

Now we address the problem of estimating AvRisk$_{k_2, r_{train}}$ from data. As we have seen from Theorem 3.2.1, the k-class average risk of a marginal classifier $\mathcal{M}$ is a functional of a object called $\bar{D}(u)$, which depends marginal model $\mathcal{M}$ of the classifier,

the joint distribution of labels Y and features X when Y is drawn from the sampling density ν.

Therefore, the strategy we take is to attempt to estimate $\bar{D}$ for then given classification model, and then plug in our estimate of $\bar{D}$ into the integral (3.1) to obtain an estimate of $\text{AvRisk}_{k_2, r_{train}}$.

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

Let us assume the linear model

$$\bar{D}(u) = \sum_{\ell=1}^{m} \beta_\ell h_\ell(u), \tag{3.8}$$

where $h_\ell(u)$ are known basis functions, and β are the model parameters to be estimated. We can obtain *unbiased* estimation of $\text{AvRisk}_{k_2, r_{train}}$ via the unbiased estimates of k-class average risk obtained from (2.5).

If we plug in the assumed linear model (3.8) into the identity (3.1), then we get

$$\text{AvRisk}_{k, r_{train}} = (k-2) \int \bar{D}(u) u^{k-2} du \tag{3.9}$$

$$= (k-2) \int_0^1 \sum_{\ell=1}^{m} \beta_\ell h_\ell(u) u^{k-2} du \tag{3.10}$$

$$= \sum_{\ell=1}^{m} \beta_\ell H_{\ell, k} \tag{3.11}$$

where

$$H_{\ell, k} = (k-2) \int_0^1 h_\ell(u) u^{k-2} du. \tag{3.12}$$

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

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

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

is unbiased for β, under any choice of positive weights w_k. The estimate of $\text{AvRisk}_{k_2, r_{train}}$ is similarly obtained from (3.11), via

$$\widehat{\text{AvRisk}_{k_2, r_{train}}} = \sum_{\ell=1}^{m} \hat{\beta}_\ell H_{\ell, k_2}. \tag{3.13}$$

3.3.1 Large-Sample Theory

How good are the estimated average risks (3.13)? Let us investigate the accuracy of the estimates in the limit where $k_1 \to \infty$, first in the case where the model (3.8) is correctly specified, and then considering possible model misspecification.

If we fix the number of classes k_2 which defines the estimation target, then we need not use the estimator (3.13), since once $k_1 > k_2$, we can use the AvTestRisk_{k_2}

as an estimator instead, which can easily be shown to a have a convergence rate of $O(1/\sqrt{k_1})$ to the true average risk. Therefore, if we want to quantify the performance of the regression-based estimator (3.13), it does not make sense to look at asymptotic settings where k_2 is fixed. One approach is to specify a setting where k_2 changes as a function of k_1. However, the approach we will take is to look at the minimax error: that is, to look at the maximum discrepancy between the estimate and the true average risk over all k_2 simultaneously. The performance criterion is the minimax error, defined

$$\text{MinimaxError} = \sup_{k_2 > 2} |\widehat{\text{AvRisk}_{k_2, r_{train}}} - \text{AvRisk}_{k_2, r_{train}}|. \qquad (3.14)$$

Well-specified case.

Let us first assume that the parametric model (3.8) is correct. Then

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

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

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

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

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

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

Theorem 3.3.1 *Consider a sequence of problems where the model $\mathcal{M}$, r_{train}, r_{test}, joint distribution $\{F_y\}_{y \in \mathcal{Y}}$, and class sampling distribution η are fixed as $k_1 \to \infty$. Further assume that the function $\bar{D}(u)$ defined by $\{F_y\}_{y \in \mathcal{Y}}$, η, and $\mathcal{M}$ satisfies*

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

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

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

Then, defining

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

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

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

Proof. Note that the statistics $AvTestRisk_k$ are U-statistics of the k_1 pairs of test and training samples. Therefore, by Hoeffding 1948, it follows that $(AvTestRisk_2, \ldots, AvTestRisk_{k_0})$

is asymptotically normal with covariance satisfying

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

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

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

It follows that defining

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

we have

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

$\square$.

Misspecified case.

Now consider the more realistic setting where the model (3.8) is misspecified. We quantify the degree of misspecification by the ℓ_∞ error on [0,1]. Define

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

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

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

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

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

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

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

Define the maximum bias as

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

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

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

3.4 Examples

Chapter 4

Inference of mutual information

4.1 Motivation

4.1.1 Gene expression dataset example

4.2 Identification loss

4.3 Average Bayes accuracy and Mutual information

4.3.1 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:

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

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

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

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

Conversely we have

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

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

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

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 4.3.1 *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.

4.3.2   Reduction

Let $p(x, y)$ be a density supported on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is a subset of $\mathbb{R}^{d_1}$ and $\mathcal{Y}$ is a subset of $\mathbb{R}^{d_2}$, 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

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

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

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

This is accomplished via the following lemma.

Lemma 4.3.1 *(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 Lesbegue measure, and a subset $\mathcal{Y} \subset \mathbb{R}^{d_y}$ with finite volume with respect to d_y-dimensional Lesbegue 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

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

Proof. Recall the definition of the Shannon entropy H:

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

It is a well-known in information theory that

$$\text{I}[p(x,y)] = \text{H}[p(x)] + \text{H}[p(y)] - \text{H}[p(x,y)].$$

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 $\lim_{i\to\infty}\mathcal{X}_i = \mathbb{R}^{d_x}$, $\lim_{i\to\infty}\mathcal{Y}_j$. Define

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

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

$$\left|\int_{x\notin\mathcal{X}_i}p(x)\log p(x)dx\right| < \frac{\epsilon}{6}$$

$$\left|\int_{y\notin\mathcal{Y}_i}p(y)\log p(y)dy\right| < \frac{\epsilon}{6}$$

$$\left|\int_{(x,y)\notin\mathcal{X}_i\times\mathcal{Y}_i}p(x,y)\log p(x,y)dxdy\right| < \frac{\epsilon}{6}$$

and also such that

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

Then, it follows that

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

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,

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

Define

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

by continuity of probability we have $\lim_i \epsilon_i \to 0$. 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

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

We have

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

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 $(X^{(1)}, ..., X^{(k)}, Y) \in \mathcal{X}_i^k \times \mathcal{Y}_i$. Therefore,

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

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 $|\mathrm{ABA}_k[\tilde{p}_i] - \mathrm{ABA}_k[p]| \leq \epsilon_i$, as claimed. $\square$

One can go from bounded-volume sets to uniform distributions by adding auxillary variables. To illustrate the intution, 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 4.3.1, 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 4.3.2 *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,

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

□

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

Theorem 4.3.2 *(Reduction.)*

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

The proof is trivial given the previous two lemmas.

4.3.3 Proof of theorem

Proof of theorem 4.3.1

Using Theorem 4.3.2, we have

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

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

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 $\mathrm{I}[p_\iota] = \iota$, and

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

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 $\mathrm{I}[p] \leq \iota$, that $\mathrm{ABA}_k[p] \leq \mathrm{ABA}_k[p_\iota]$.

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

$$\mathrm{I}[p(x, y)] = \mathbf{E}(Z(Y) \log Z(Y))$$

while

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

Letting G_y be the distribution of $Z(y)$, we have

$$E[G_y] = 1$$

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

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

where the expectation is taken over $Y \sim \mathrm{Unif}[0, 1]$ and where $E[G]$, $I[G]$, and $\psi_k[G]$ are defined as in Lemma **??**.

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

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

Now, since f is concave by Lemma **??**, we can apply Jensen's inequality to conclude that

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

which completes the proof. $\square$

4.4   Lower confidence bound

Get lower confidence bound for ABA then plug into result.

4.5   Example

Chapter 5

High-dimensional inference of mutual information

5.1 Motivation

5.1.1 Quantifying precision of decoding models

Both computational and cognitive neuroscience are concerned with understanding brain function: while computational neuroscience is concerned with understanding functionality at the level of the spiking behavior of individual neurons and small neural populations, cognitive neuroscience tends to emphasize functionality at the level of macroscale regions of the interest in the brain. While the recording technologies, motivating questions, and analytical methodologies differ between the two subdisciplines, the conceptualization of brain functionality in terms of *encoding* and *decoding* models has been widely applied in both areas Quian Quiroga and Panzeri 2009Naselaris et al. 2011. In computational neuroscience, cell recording experiments are conducted to determine whether spike trains have a temporal and/or correlational code Nelken et al. 2005Hatsopoulos et al. 1998, to examine how the neural code adapts to changes in stimulus distribution Fairhall et al. 2001 and whether downstream neurons make use of higher-order correlations for decoding Oizumi et al. 2010. Meanwhile, in neuroimaging studies, functional MRI experiments are employed to model the receptive fields of early visual areas in the human brain Kay et al. 2008, to examine the semantic encoding of words Mitchell et al. 2008 or objects Huth et al. 2012.

The dual perspectives of encoding and decoding originate naturally from the fact that in examining the link between brain activity and function, one can either start with brain activity on one end, or with external stimulation or behavioral observation on the other end. Starting by exposing the subject to sensory stimuli or prompting the subject to engage in particular motor tasks, one can search for areas in the brain which respond to the task: in other words, one can test to see which areas of the brain *encode* the given stimulus. In the other direction, one seeks to understand the functionality of a given brain region: in other words, how to *decode* brain activity in that region.

Formulation of encoding models is relatively straightforward, since one needs only to characterize the observed brain response to a given stimulus. One can further ask how to distinguish between signal and noise in the encoding mechanism Nelken et al. 2005, or in complex stimuli, seek a linearizing feature set which reveals the nature of the brain representation Naselaris et al. 2011. However, the establishment of complete decoding models is much less amenable to experimental manipulation, since to exhaustively characterize the functionality of a neuron, one would

have to know in advance the type of information it encodes. Early advances in decoding often depended on strokes of luck: Hubel Hubel 1982 originally discovered the existence of neurons with orientation-sensitive receptive fields due to the vigorous response of a cell to the perfectly angled shadow of a glass slide that they were inserting into the ophthalmoscope. Yet, even now, the goal of completely characterizing the function of a given brain region remains a difficult task, with the most promising approach being a *reverse inference* procedure Poldrack 2006 which aggregates information from the literature about activity-functionality relationships.

A more feasible goal is to establish the *precision* with which a neuron can decode a particular type of feature. This can be accomplished by first training an encoding model, and then inverting the encoding model using Bayes' rule to obtain a decoding model Oram et al. 1998Quian Quiroga and Panzeri 2009Naselaris et al. 2011.

By decoding *precision*, we mean the specificity which we can identify or reconstruct the stimulus based on the neural response. As such, in our view, the term decoder *precision* is more or less synonymous with terms such as decoder *performance* or decoder *accuracy* as they are used in the literature. However, we choose the word *precision* in particular, because it communicates the idea that the essential quality of a good decoder is that it allows one to confidently and precisely infer the stimulus.

Measures of decoding precision can be used to support several different kinds of scientific inferences. When there exist multiple plausible encoding models–for instance, a model where stimulus information is encoded solely by average firing rate versus a model where inter-spike timings also carry information–the precision of the decoder can be used as a basis for deciding the best encoding model. For two encoding models with equal complexity, such as comparing two different types of receptive field models, the model with better decoding precision could be considered the more plausible model. In the case where a more complex encoding model is compared to a strictly simpler model–such as comparing a model with a temporal code versus a model only incorporating average firing rate, a substantial improvement in decoding precision for the more complex model is needed to demonstrate its validity, since in the null hypothesis where the simpler model is correct, the more complex model should still have approximately equal decoding performance.

Yet another application of decoding precision is to track the adaptivity of the neural code. Fairhall Fairhall et al. 2001 recorded the output of a motion-sensitive neuron in a fly in response to a visual stimulus with changing angular velocity. Changing the variance of the stimulus results in rapid adaptation: the neural code starts adapting to the change in stimulus distribution within tens of milliseconds, which is reflected by an increased or decreased precision (as measured by mutual information) in resolving angular velocity to match the variance of the stimulus. More generally, comparisons of decoding precisions between different conditions can show how the encoded information increases or decreases across experimental conditions. Kayser Kayser, Logothetis, and Panzeri 2010 demonstrated how the mutual information between a sound stimulus and neurons in the auditory cortex increased when the subjects were also presented a matching visual stimulus (e.g. showing a picture of a lion roaring while playing the sound of a lion's roar.)

Differing types and parameterizations of stimuli naturally lead to differing measures of decoding precision. For stimuli which can be parameterized by a scalar x, the precision can be measured by the squared correlation coefficient R^2 Abbott 1994. However, the resulting measure of precision is not invariant to scaling of the parameterization: for instance, the choice of whether to parameterize volume on an absolute scale or a logarithmic scale. The mutual information Shannon 1948 between the stimulus and the predicted stimulus is invariant to the parameterization of the

stimulus. Due to its invariance and a number of other properties, the mutual information is widely used to measure the precision of the neural code in cell recording studies, both for single-neuron decoding models Borst and Theunissen 1999 and for population coding models Quian Quiroga and Panzeri 2009Ince et al. 2010.

However, the difficulties of estimating mutual information in small samples has been widely recognized, with a large literature on bias correction methods Panzeri et al. 2007Paninski 2003. Methods for bias correction have been developed for three different sample size regimes: the moderate-sample regime, where the number of observations is larger than the number of stimulus-response pairsMiller 1955Strong et al. 1998Treves and Panzeri 1995, the undersampled regime, where the number of observations is less than the number of stimulus-response pairs Nemenman, Bialek, and de Ruyter van Steveninck 2004, and a *stimulus-undersampled* regime, where only a small fraction of possible stimuli are sampled, but with a large number of observations for each of the sampled stimuli Gastpar, Gill, and Theunissen 2009. Nevertheless, even the bias-corrected estimates may be unusably inaccurate in problems of moderate dimensionality, since the cardinality of response space grows exponentially with the dimensionality. In such cases, alternative approaches for estimating the mutual information include the assumption of a parametric model Brunel and Nadal 1998Gastpar, Gill, and Theunissen 2009Yarrow, Challis, and Seriès 2012, or usage of the maximum entropy principle to obtain bounds on the mutual information subject to the empirical moments of a certain order Ince et al. 2009Globerson et al. 2009.

Perhaps due to the technical difficulties of estimating mutual information in high dimensions, mutual information has never, to our knowledge, been used as a measure of decoding precision in neuroimaging studies, although it has been proposed for the purpose of bypassing the modelling of the hemodynamic response function for single-voxel analyses Fuhrmann Alpert et al. 2007. Instead, a variety of methods are employed to characterize the precision of decoding models, depending on the nature of the stimulus and the experimental setup.

In task fMRI experiments where stimuli are drawn from a number of disjoint semantic categories– for instance, 'birds', 'insects', and 'mammals' as in Connolly et al. 2012, it is natural to construct a decoder which outputs the predicted category of a stimulus as a function of the response. Such a decoder is known as a *classifier* in the machine learning literature Hastie, Tibshirani, and Friedman 2009, and a natural measure of classifier precision is the probability that the decoder outputs the correct category on a new, randomly drawn test example, which is the *classification accuracy*.

In experiments where the subject is presented a number of parameterized stimuli are drawn from a continuous distribution (such as natural images or sounds), there are two types of decoders which can be constructed. In the first case, one constructs a decoder which estimates the parameters of the stimulus which we call a *reconstructor*: the precision of such a decoder is measured by the correlation between the estimated and true parameter vector Pasley et al. 2012 Nishimoto et al. 2011Naselaris et al. 2009. In the second case, one constructs a decoder which picks the most likely stimulus from a finite library of examples *which includes the true stimulus* Kay et al. 2008Mitchell et al. 2008. Since the true stimulus is included in the library, the task is to 'identify' the correct stimulus from the library. A natural measure of decoder performance is therefore the probability of correct identification. However, note that this probability is dependent on the arbitrary choice of the size of the exemplar library: a different choice for library size therefore results in a different measure of precision. We refer to the probability of correct classification for a library of k exemplars as the *k-example identification accuracy*.

In their respective domains, these different measures of precision suffice to make inferences on many interesting scientific questions: to list a few examples, showing the superiority of a Gabor filters versus center-surround filters for modeling the receptive fields of V1 and V2 neurons Kay et al. 2008, or demonstrating that brain activity in response to viewing an Engish noun can be predicted from word association frequencies Mitchell et al. 2008.

A commonality to all applications of decoding models in neuroimaging is the pairwise comparison of two decoding models (Gabor vs. retinotopic) or the comparison of a single decoding model to chance accuracy. Looking ahead to anticipate what kinds of analyses might be employed in the future based on neuroimaging data, it is suggestive to note that the earliest decoding studies in the cell recording literature also involved comparisons between two or three different decoders Eckhorn et al. 1976. However, as neuroscientists began to consider questions of population coding, analyses of the redundancy between neurons started to make use of comparisons between large numbers of decoders: for a population of N neurons, one might compare the precision of a decoder (mutual information) based on the entire ensemble, compared to the precisions of decoders based on each of the N individual neurons. Furthermore, one can make the same comparison for a range of different ensemble sizes N. As questions about the redundancy of the neural code are relevant on both the micro scale (the domain of cell recording studies) and the macro scale (the domain of neuroimaging), it is safe to assume that similar analyses, requiring comparisons of large numbers of decoders, will emerge in neuroimaging studies. Already in the functional MRI literature, we see similar decompositions of decoding accuracy versus ensemble size Kay et al. 2008, but another possible type of decomposition would be to compare decoding performance as the number of stimulus features is varied, rather than the number of voxels.

The scaling properties of mutual information are highly advantageous when comparing multiple decoders, which could potentially span a wide range of decoding precision: for instance, a single neuron versus an ensemble of thousands of neurons. In contrast, classification accuracy, k-class identification accuracy and reconstruction accuracy all suffer from the issue of *limited dynamic range*: that is, they are only effective at measuring precision within a certain range.

Let us illustrate with the example of identification accuracy. A low precision decoder, such as a decoder based on a single voxel, may have an accuracy which is so close to chance accuracy, $1/k$, as to be statistically indistinguishable from chance based on the data. On the other hand, a sufficiently high-precision decoder may face the opposite problem, where it achieves perfect classification on the limited number of test examples. Any empirical estimate of identification accuracy can only be used to accurately rank decoders which have accuracies sufficiently bounded away from both $1/k$ and 1. The same issue applies to reconstruction accuracy (bounded between 0 and 1) and classification accuracy (bounded between $1/k$ and 1, where k is the number of classes): any bounded measure of precision is ineffective at comparing decoders which are too close to either the upper bound or lower bound of achievable precision.

In practice, the solution to this issue is to find a measure of precision which is well-suited for all of the decoders that needed to be compared. If there are two encoding models which both achieve perfect classification on the test set, then perhaps the more demanding measure of reconstruction accuracy can be used to distinguish them. However, this strategy begins to become impractical as the number of decoders to be compared increases. One wishes to relate the decoding precision of an N-voxel ensemble for N spanning from 1 to 10000: however, any bounded measure

of precision which is suitably stringent for distinguishing $N = 9999$ from $N = 10000$ would fail for comparing $N = 1$ to $N = 2$, and vice-versa.

We have seen that one solution to this predicament is to use an unbounded measure of precision which can remain sensitive to variations in precision across a large dynamic range: for instance, the mutual information. Yet, given the difficulty of estimating the mutual information in high-dimensional settings, one might consider another approach: to develop a systematic means for comparing decoders by using multiple (easily estimated) precision measures, each of which may only capture a limited range of precisions, but which collectively span a sufficiently large range of precisions to include all of the decoders being compared.

Our contribution in this paper is to show that both of these approaches–the estimation of mutual information, and the comparison of decoders based on a range of decoding metrics, turn out to be the very same problem in high-dimensional settings. The *identification accuracy curve*, which we define as the collection of all k-class identification accuracies for $k \geq 2$, can be used to compare a collection of decoders over a large span of precisions. Yet, a recent theoretical resultZheng and Benjamini 2016 shows that the identification accuracy curve for the Bayes decoder (the optimal decoder) is determined by the mutual information in a certain high-dimensional regime. While it is generally not feasible to approximate the Bayes decoder in high-dimensional settings, we use this result to define the *implied information* for a non-Bayes (suboptimal) decoder. The implied information, $I_{implied}$, is not the true mutual information between the stimulus and response, but it provides a means of comparing two accuracy curves (estimate the implied information from each, and then compare the estimates), as well as providing an unbounded measure of decoding precision which, similar to mutual information, has desirable scaling properties for the purpose of comparing decoders spanning a range of precisions.

5.1.2 Kay et al. example

5.2 Setup

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

A1. $\lim_{d\to\infty} I(X^{[d]}; Y^{[d]}) = \iota < \infty$.

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 $i \neq k$,

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

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 $d = 1, \ldots,$ 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.

5.3 Theory

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 (**??**).

Let us start by rewriting the average Bayes error:

$$e_{ABE,k} = \Pr[p(Y|X_1) \le \max_{j \ne 1} p(Y|X_j)|X = X_1].$$

Defining the statistic $Z_i = \log p(Y|X_i) - \log p(Y|X_1)$, where $Y \sim p(y|X_1)$, we obtain $e_{ABE} = \Pr[\max_{j>1} Z_i > 0]$. The key assumption we need is that $Z_2, \ldots, Z_k$ are asymptotically multivariate normal. If so, the following lemma allows us to obtain a formula for the misclassification rate.

Lemma 1.   *Suppose $(Z_1, Z_2, \ldots, Z_k)$ are jointly multivariate normal, with $\mathbf{E}[Z_1 - Z_i] = \alpha$, $Var(Z_1) = \beta \ge 0$, $Cov(Z_1, Z_i) = \gamma$, $Var(Z_i) = \delta$, and $Cov(Z_i, Z_j) = \epsilon$ for all $i, j = 2, \ldots, k$, such that $\beta + \epsilon - 2\gamma > 0$. Then, letting*

$$\mu = \frac{\mathbf{E}[Z_1 - Z_i]}{\sqrt{\frac{1}{2} Var(Z_i - Z_j)}} = \frac{\alpha}{\sqrt{\delta - \epsilon}},$$

$$\nu^2 = \frac{Cov(Z_1 - Z_i, Z_1 - Z_j)}{\frac{1}{2} Var(Z_i - Z_j)} = \frac{\beta + \epsilon - 2\gamma}{\delta - \epsilon},$$

we have

$$\Pr[Z_1 < \max_{i=2}^{k} Z_i] = \Pr[W < M_{k-1}]$$

$$= 1 - \int \frac{1}{\sqrt{2\pi\nu^2}} e^{-\frac{(w-\mu)^2}{2\nu^2}} \Phi(w)^{k-1} dw,$$

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.

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

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

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

$$Z_i = \sum_{m=1}^{d} \log p_m(y_m|x_m^{(i)}) - \log p_m(y_m|x_1^{(m)})$$

where $x_{i,j}$ is the ith component of x_j. The d terms $\log p_m(y_m|x_{m,i}) - \log p_m(y_m|x_{m,1})$ are independent across the indices m, but dependent between the $i = 1, \ldots, k$. Therefore, the multivariate central limit theorem can be applied to conclude that the vector $(Z_2, \ldots, Z_k)$ 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 $(Z_2, \ldots, Z_k)$ holds under more general conditions, and appears reasonable in practice.

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

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

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 the theorem.

Theorem 1. *Let $p^{[d]}(x,y)$ be a sequence of joint densities for $d = 1, 2, \ldots$. Further assume that*

A1. $\lim_{d \to \infty} I(X^{[d]}; Y^{[d]}) = \iota < \infty$.

A2. *There exists a sequence of scaling constants $a_{ij}^{[d]}$ and $b_{ij}^{[d]}$ such that the random vector $(a_{ij}\ell_{ij}^{[d]} + b_{ij}^{[d]})_{i,j=1,\ldots,k}$ 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 $i \neq k$,*
$$\lim_{d \to \infty} Cov[u^{[d]}(X^{(i)}, Y^{(j)}), u^{[d]}(X^{(k)}, Y^{(j)})] = 0.$$

Then for $e_{ABE,k}$ as defined above, we have

$$\lim_{d \to \infty} e_{ABE,k} = \pi_k(\sqrt{2\iota})$$

where

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

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

5.4　Estimator

Define the Bayes risk as the identification risk of the optimal decoder. The result of ZB 2016 says that under certain regularity conditions, for for sufficiently high-dimensional $p(\vec{x}, \vec{y})$, we have

$$\text{BayesAcc}_k \approx \bar{\pi}_k(\sqrt{2I(\vec{x}; \vec{y})})$$

where $\bar{\pi}_k$ is the function

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

and where $I(\vec{x}; \vec{y})$ is the Shannon information

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

This is an important result because it implies that the entire identification accuracy curve can be summarized by a single parameter–the mutual information. This means that in the asymptotic regime specified by ZB 2016, (i) any portion of the curve can be used to estimate the mutual information and therefore reconstruct the entire curve, and (ii) that there exists a strict ordering over identification accuracy curves: for any two curves A_k and A'_k, one dominates the other for all k: either $A_k \geq A'_k$ for all $k \geq 2$, or $A'_k \geq A_k$ for all $k \geq 2$.

However, the result in ZB 2016 only applies to the optimal decoder, or *Bayes decoder*. Yet, it is impossible to obtain the Bayes decoder in practice, since constructing the Bayes decoder requires knowing $p(\vec{x}, \vec{y})$. Therefore, we propose that under similar conditions to those stipulated in ZB 2016, for a certain class of classifiers[1], we have

$$\text{IdAcc}_k \approx \bar{\pi}_k(\sqrt{2I_{implied}})$$

where IdRisk_k is the k-class identification risk for a given classifier trained from the training set, and where $I_{implied}$ is a real-valued attribute of the classifier called the *implied information*. Furthermore, since $\text{IdAcc}_k \leq \text{BayesAcc}_k$ by definition (as BayesAcc_k is the best achievable accuracy), we have

$$I_{implied} \leq I(\vec{X}; \vec{Y}).$$

In order to estimate the implied information, we can rely on the fact that the empirical identification accuracy curve EmpAcc_k is an unbiased estimate of the true identification accuracy curve IdAcc_k. Therefore, we can estimate $I_{implied}$ by finding the theoretical curve which gives the best fit to the empirical accuracies in terms of mean-squared error. Thus, define $\hat{I}_{implied}$ as the nonlinear least-squares estimator

$$\hat{I}_{implied} = \text{argmin}_{\iota \geq 0} \sum_{k=2}^{M} (\text{EmpAcc}_k - \bar{\pi}_k(\sqrt{2\iota}))^2.$$

5.5　Examples

---

[1] We leave it to future work to specify the conditions on the joint density and classifiers needed to formally establish the desired property.

Appendix A

Frequently Asked Questions

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

The color of links can be changed to your liking using:
    `\hypersetup{urlcolor=red}`, or
    `\hypersetup{citecolor=green}`, or
    `\hypersetup{allcolor=blue}`.
If you want to completely hide the links, you can use:
    `\hypersetup{allcolors=.}`, or even better:
    `\hypersetup{hidelinks}`.
If you want to have obvious links in the PDF but not the printed text, use:
    `\hypersetup{colorlinks=false}`.

Bibliography

Abbott, Larry F. (1994). "Decoding neuronal firing and modelling neural networks." In: *Quarterly reviews of biophysics* 27.3, pp. 291–331. ISSN: 0033-5835. URL: http://www.ncbi.nlm.nih.gov/pubmed/7899551.

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.

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.

Borst, Alexander and Frédéric E. Theunissen (1999). "Information theory and neural coding". In: *Nature Neuroscience* 2.11, pp. 947–957. ISSN: 10976256. DOI: 10.1038/14731. URL: http://www.nature.com/doifinder/10.1038/14731.

Brunel, Nicolas and J P Nadal (1998). "Mutual information, Fisher information, and population coding." In: *Neural computation* 10.7, pp. 1731–57. ISSN: 0899-7667. DOI: 10.1162/089976698300017115. arXiv: arXiv:1011.1669v3. URL: http://www.ncbi.nlm.nih.gov/pubmed/9744895.

Chen, Yi et al. (2011). "Cortical surface-based searchlight decoding". In: *Neuroimage* 56.2, pp. 582–592.

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.

Connolly, Andrew C. et al. (2012). "The Representation of Biological Classes in the Human Brain". In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 32.8, pp. 2608–2618. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.5547-11.2012. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3532035/{\%}5Cnhttp://www.ncbi.nlm.nih.gov/pmc/articles/PMC3532035/pdf/nihms361453.pdf.

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.

Eckhorn, R et al. (1976). "Efficiency of Different Neuronal Codes: Information Transfer Calculations for Three Different Neuronal Systems". In: *Biol. Cybernetics* 22, pp. 49–60.

Efron, Bradley and Robert J Tibshirani (1994). *An introduction to the bootstrap*. CRC press.

Fairhall, Adrienne L. et al. (2001). "Efficiency and ambiguity in an adaptive neural code". In: *Nature* 412.23, pp. 787–792. ISSN: 0028-0836. DOI: 10.1038/35090500. URL: http://www.nature.com/doifinder/10.1038/35090500.

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.

Fuhrmann Alpert, Galit et al. (2007). "Spatio-temporal information analysis of event-related BOLD responses". In: *NeuroImage* 34.4, pp. 1545–1561. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2006.10.020.

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.

Globerson, Amir et al. (2009). "The minimum information principle and its application to neural code analysis." In: *Proceedings of the National Academy of Sciences of the United States of America* 106.9, pp. 3490–5. ISSN: 1091-6490. DOI: 10.1073/pnas.0806782106. URL: http://www.ncbi.nlm.nih.gov/pubmed/19218435http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2651257.

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.

Hatsopoulos, N G et al. (1998). "Information about movement direction obtained from synchronous activity of motor cortical neurons." In: *Proceedings of the National Academy of Sciences of the United States of America* 95.26, pp. 15706–11. ISSN: 0027-8424. URL: http://www.ncbi.nlm.nih.gov/pubmed/9861034http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC28108.

Hubel, David H. (1982). "Evolution of ideas on the primary visual cortex, 1955–1978: A biased historical account". In: *Bioscience Reports* 2.7, pp. 435–469.

Huth, Alexander G. et al. (2012). "A Continuous Semantic Space Describes the Representation of Thousands of Object and Action Categories across the Human Brain". In: *Neuron* 76.6, pp. 1210–1224. ISSN: 08966273. DOI: 10.1016/j.neuron.2012.10.014. arXiv: NIHMS150003.

Ince, Robin A A et al. (2009). "Python for information theoretic analysis of neural data." In: *Frontiers in neuroinformatics* 3, p. 4. ISSN: 1662-5196. DOI: 10.3389/neuro.11.004.2009. URL: http://www.ncbi.nlm.nih.gov/pubmed/19242557http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2647335.

Ince, Robin A.A. et al. (2010). "Information-theoretic methods for studying population codes". In: *Neural Networks* 23.6, pp. 713–727. ISSN: 08936080. DOI: 10.1016/j.neunet.2010.05.008.

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`.

Kayser, Christoph, Nikos K. Logothetis, and Stefano Panzeri (2010). "Visual Enhancement of the Information Representation in Auditory Cortex". In: *Current Biology* 20.1, pp. 19–24. ISSN: 09609822. DOI: `10.1016/j.cub.2009.10.068`.

Kriegeskorte, Nikolaus and Peter Bandettini (2007). "Analyzing for information, not activation, to exploit high-resolution fMRI". In: *Neuroimage* 38.4, pp. 649–662.

Kriegeskorte, Nikolaus, Rainer Goebel, and Peter Bandettini (2006). "Information-based functional brain mapping". In: *Proceedings of the National academy of Sciences of the United States of America* 103.10, pp. 3863–3868.

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.

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:

Miller (1955). "Note on the bias of information estimates". In: *Information Theory in Psychology: Problems and Methods*.

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

Naselaris, Thomas et al. (2009). "Bayesian Reconstruction of Natural Images from Human Brain Activity". In: *Neuron* 63.6, pp. 902–915. ISSN: 08966273. DOI: `10.1016/j.neuron.2009.09.006`. URL: `http://dx.doi.org/10.1016/j.neuron.2009.09.006`.

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

Nelken, Israel et al. (2005). "Encoding Stimulus Information by Spike Numbers and Mean Response Time in Primary Auditory Cortex". In: *Journal of Computational Neuroscience* 19, pp. 199–221.

Nemenman, Ilya, William Bialek, and Rob de Ruyter van Steveninck (2004). "Entropy and information in neural spike trains: progress on the sampling problem." In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 69.5 Pt 2, p. 056111. ISSN: 1539-3755. DOI: `10.1103/PhysRevE.69.056111`. arXiv: `0306063 [physics]`. URL: `http://www.ncbi.nlm.nih.gov/pubmed/15244887http://link.aps.org/doi/10.1103/PhysRevE.69.056111`.

Nishimoto, Shinji et al. (2011). "Reconstructing visual experiences from brain activity evoked by natural movies". In: *Current Biology* 21, pp. 1641–1646. ISSN: 09609822. DOI: `10.1016/j.cub.2011.08.031`.

Oizumi, Masafumi et al. (2010). "Mismatched Decoding in the Brain". In: *Journal of Neuroscience* 30.13, pp. 4815–1826.

Oram, Mike W. et al. (1998). "The 'Ideal Homunculus': decoding neural population signals". In: *Trends in Neurosciences* 21.6, pp. 259–265. ISSN: 01662236. DOI: `10.1016/S0166-2236(97)01216-2`.

Paninski, Liam (2003). "Estimation of Entropy and Mutual Information". In: *Neural Computation* 15.6, pp. 1191–1253. ISSN: 0899-7667. DOI: `10.1162/089976603321780272`.

arXiv: `0402594v3 [arXiv:cond-mat]`. URL: `http://www.mitpressjournals.org/doi/abs/10.1162/089976603321780272`.

Panzeri, Stefano et al. (2007). "Correcting for the Sampling Bias Problem in Spike Train Information Measures". In: *Journal of Neurophysiology* 98.3.

Partalas, Ioannis et al. (2015). "LSHTC: A benchmark for large-scale text classification". In: *arXiv preprint arXiv:1503.08581*.

Pasley, Brian N. et al. (2012). "Reconstructing speech from human auditory cortex". In: *PLoS Biology* 10.1. ISSN: 15449173. DOI: `10.1371/journal.pbio.1001251`.

Poldrack, Russell A. (2006). "Can cognitive processes be inferred from neuroimaging data?" In: *Trends in Cognitive Sciences* 10.2, pp. 59–63. ISSN: 13646613. DOI: `10.1016/j.tics.2005.12.004`.

Poldrack, Russell A, Jeanette A Mumford, and Thomas E Nichols (2011). *Handbook of functional MRI data analysis*. Cambridge University Press.

Quian 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.

Shannon, C. E. (1948). "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3, pp. 379–423. ISSN: 00058580. DOI: `10.1002/j.1538-7305.1948.tb01338.x`. URL: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6773024`.

Strong, S. P. et al. (1998). "Entropy and Information in Neural Spike Trains". In: *Physical Review Letters* 80.1, pp. 197–200. ISSN: 0031-9007. DOI: `10.1103/PhysRevLett.80.197`.

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.

Treves, Alessandro and Stefano Panzeri (1995). "The Upward Bias in Measures of Information Derived from Limited Data Samples". In: *Neural Computation* 7.2, pp. 399–407. ISSN: 0899-7667. DOI: `10.1162/neco.1995.7.2.399`.

Yarrow, Stuart, Edward Challis, and Peggy Seriès (2012). "Fisher and Shannon Information in Finite Neural Populations". In: *Neural Computation* 24.7, pp. 1740–1780. ISSN: 0899-7667. DOI: `10.1162/NECO_a_00292`.

Zheng, Charles Y. and Yuval Benjamini (2016). "Estimating mutual information in high dimensions via classification error". In: arXiv: `1606.05229`. URL: `http://arxiv.org/abs/1606.05229`.

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`.