Carnegie Mellon University
Dietrich College of Humanities and Social Sciences

Dissertation

Submitted in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy

Title: Accounting for Changes in the Distribution of Data

Presented by: Ciaran Evans

Accepted by: Department of Statistics and Data Science

Readers:

Max G'Sell, Advisor

Christopher Genovese, Advisor

Christina Patterson (UPMC)

Aaditya Ramdas

Chad Schafer

Cosma Shalizi

Valerie Ventura

Approved by the Committee on Graduate Degrees:

Richard Scheines, Dean Date

Carnegie Mellon University

Accounting for Changes in the Distribution of Data

A Dissertation Submitted to the Graduate School
in Partial Fulfillment of the Requirements for the degree

Doctor of Philosophy

in

Statistics

by

Ciaran Evans

Department of Statistics and Data Science
Carnegie Mellon University
Pittsburgh, PA 15213

Carnegie Mellon University

June 2021

To Sam.

Acknowledgements

I would like to thank my advisors, Max G'Sell and Chris Genovese, for working with me over the last five years and mentoring me in research. Their expertise and advice has been invaluable. I would also like to thank my collaborators – Zara Weinberg, Manoj Puthenveedu, Christina Patterson, and Ira Bergman – for working with me on interesting problems applying statistics to the sciences. I am also grateful for the support and guidance from my committee members – Aaditya Ramdas, Chad Schafer, Cosma Shalizi, and Valerie Ventura – and the staff and faculty of the CMU Statistics and Data Science department. Finally, thank you to all my friends and family for their support through this process.

Contents

List of Tables

List of Figures

Chapter I

Introduction

Accomodating changes to the distribution of data in classification and inference procedures is an important problem for statistics. For instance, statistical predictions often rely on the assumption that future data will follow the same distribution as past data, and predictions may fail when there is a change to the data generating process. In other cases, detecting changes to the distribution of data is of scientific interest itself. For example, when monitoring a patient's brain activity, a change may have serious health implications; and in cellular biology, characterizing differences between biological conditions is a key goal for experiments.

We focus on three aspects of distributional change, and particularly their relationship to classification and changepoint detection. First, we consider the problem of using classifier predictions for inference with unlabeled data under dataset shift. As data collection requires an expensive manual labelling step, we seek to automate labelling by applying a classifier constructed on labelled training data. However, classification must account for differences between the training and test sets, which requires that predictions be generalizable to new domains. We investigate this problem in an application to cellular transport in which experiments are performed to analyze different receptors on the cell surface, and the goal is to characterize differences between receptors.

We then consider the problem of detecting changes in sequentially observed classification data. When applying a classifier to new data, shifts in the overall base rate of class labels in the population – known as label shift – can lead to miscalibration of the classifier scores and significant decreases in performance. Such a change could occur when a disease becomes more prevalent in a population, but the symptoms used to diagnose the disease remain the same. Knowing that a change has occurred is also essential for making interventions.

To detect label shift, we propose a nonparametric detection procedure that directly leverages the label shift assumption and the classifier setting. At their most stringent, classical sequential detection procedures require both the pre- and post-change distributions to be known, with the changepoint the only unknown. While these assumptions allow for optimal change detection, many modern applications involve multivariate data with complex changes that do not fit the classical paradigm. We provide conditions under which our nonparametric label shift detector is asymptotically optimal, without assuming the pre- and post-change distributions are known.

In addition to detecting classifier label shift, we apply nonparametric changepoint detection to flag changes to brain activity with multivariate electroencephalogram (EEG) data. For example, pediatric neurology patients under continuous EEG monitoring sometimes experience a catastrophic decline in brain function that is difficult to discern in real time, but more evident post-hoc in the EEG record. Detecting changes in brain activity as quickly as possible can be essential to the patients outcomes.

I.1 Aims and significance

This thesis will focus on developing tools for performing inference when the distribution of data changes, and quickly detecting changes in sequentially observed data. In all settings we consider, the data is multivariate and may be high-dimensional. A key feature of the work in this thesis is to leverage structure in the problem to handle the data dimensionality and account for changes in distribution. For analysis of cellular transport, we carefully construct features that capture the inherent geometry of transport over time, thereby creating features which are preserved across training and test data. For label shift, the problem reduces first to detecting a change in the univariate distribution of classifier predictions, then to detecting change to a single parameter - the rate of positive cases. For EEG data, we transform the data into the frequency domain, as changes in the distribution of frequencies are easier to characterize and frequencies are commonly used to describe brain behavior.

Aim 1: Generalizable classifier predictions for inference on unlabeled data

The first part of this thesis focuses on performing inference on unlabeled test data that comes from different distributions than existing training data. Under common assumptions for the nature of dataset shift, such as covariate shift or label shift, classifier predictions can be used in place of ground truth labels for constructing confidence intervals and testing hypotheses. In Chapter II, we present bootstrap procedures which use these assumptions, while accounting for variability in both the classifier predictions and the test data. In

Chapter III, we present study design guidelines that help promote generalizability. This work was done in collaboration with Max G'Sell (Carnegie Mellon University), Zara Weinberg (University of California, San Francisco), and Manoj Puthenveedu (University of Michigan).

Aim 2: Detecting label shift

The second part of this thesis focuses on label shift in binary classification. While classical changepoint detection methods require strong assumptions to achieve optimality, we show that near-optimal nonparametric methods exist when the label shift assumption is made. Our approach requires minimal assumptions on the data, and leverages the existing classifier predictions. Our results allow for quick changepoint detection when a classifier is applied to online data. This work was done in collaboration with Max G'Sell (Carnegie Mellon University), and is presented in Chapter IV.

Aim 3: Detecting EEG changes

In the third part of this thesis, we describe a pilot study using nonparametric changepoint detection with EEG data, to assist in the early detection of negative changes to brain activity. Currently, patient EEGs are monitored by highly trained clinicians who observe the raw and filtered signals, as well as frequency-domain representations such as the spectrogram. However, our collaborators at UPMC believe that it can be hard to detect subtle changes by hand, and quicker interventions could save patient lives. We aim to expand existing detection techniques to capture complex changes and improve detection time.

Experienced practitioners can also identify different types of changes to the EEG. Some changes may be benign, perhaps simply resulting from movement, while other changes have more serious implications. When monitoring patients for a change to the EEG signal, we would like to know what type of change occurs in addition to whether a change occurs. A long term goal of this research, beyond the scope of this thesis, is to learn over time which types of changes are important to detect, which will also help inform practitioners of possible outcomes of a change. Our work on detecting changes in EEG data will help support this future work. This work is in collaboration with Chris Genovese (Carnegie Mellon University), Christina Patterson (UPMC), Ira Bergman (UPMC), and their colleagues at UPMC Children's Hospital of Pittsburgh.

Chapter II

Inference with classifier predictions

II.1 Introduction

This chapter studies statistical issues that arise when classifiers are used to automate labor-intensive data collection in scientific pipelines. With data collection often requiring laborious human labels of objects or events, it has become common to apply automated labeling techniques, in which a classifier is trained on labeled examples to predict labels in new data (Norouzzadeh et al., 2018; Christiansen et al., 2018; Caicedo et al., 2019). These classifier predictions, rather than ground truth labels, are then used for statistical inference across groups and experimental conditions.

However, unless the classifier is perfect in all experimental settings, any inference based on the classifier predictions must incorporate the additional variability introduced by the classifier. Furthermore, to make valid comparisons across experimental units or conditions, the classifier must exhibit the same performance across those units and conditions. Since the purpose of the study is often to show that two conditions are actually different, this requirement is often unsatisified unless explicitly designed for.

In this chapter, we present methodology for carrying out inference based on classifier-labeled data, with a particular focus on accounting for differences in data distribution between conditions. We outline considerations in designing a classifier beyond simple accuracy, and define the necessary assumptions and models to perform valid statistical inference. Our work is inspired by the following case study from cellular biology.

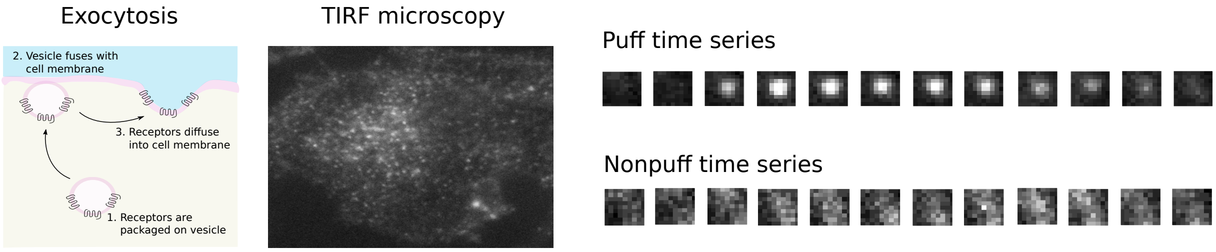
Exocytosis
2. Vesicle fuses with cell membrane
3. Receptors diffuse into cell membrane
1. Receptors are packaged on vesicle
TIRF microscopy
Puff time series
Nonpuff time series

Figure II.1: Fluorescent proteins are used to study exocytosis with TIRF microscopy. Left: Exocytosis regulated the concentration of receptor proteins on the cell surface by adding receptors through vesicle fusion. Center: The surface of a cell in a TIRF microscopy image; bright spots correspond to concentrated clusters of receptors on the cell surface. Right: Consecutive frames showing behavior of an exocytic event over time, in a 50Hz microscopy video. True exocytic events ("puffs") have a characteristic pattern of diffusion over time. Other bright spots on the cell surface are not puffs, and do not show puff behavior.

Motivating example: Studying cellular transport through exocytosis. Receptors on the cell surface play a crucial role in a cell's response to external stimuli. These receptors—and thus the corresponding responsiveness—are regulated in part by a process called *exocytosis*, which brings new receptors to the surface by packaging them on bubbles of membrane, which then merge with the outer membrane of the cell (Figure II.1) (Yu et al., 1993; Pippig et al., 1995). Biologists can observe and measure exocytosis using *total internal reflection fluorescence (TIRF) microscopy*, in which individual exocytotic events manifest as bright "puffs" of fluorescence on the cell surface, as flourescence-tagged receptors are deposited and then diffuse (Figure II.1) (Axelrod, 1981; Sankaranarayanan et al., 2000; Rappoport et al., 2003). For simplicity, we'll refer to exocytic events as *puffs* for this reason. However, measuring these events is complicated by the fact that other objects and processes in the cell can also manifest as bright spots in these TIRF videos. Indeed, each cell typically has 5000 - 10000 detected events, and only about 5% are puffs.

A typical experiment compares puff behavior for several different surface receptors or experimental conditions, resulting in a hierarchical structure common to biological experiments, shown in Figure II.2. When TIRF microscopy images are labeled by hand, statistical inference between conditions is straightforward:

1. TIRF microscopy is performed for each cell
2. Researchers use characteristic patterns of puff appearance to identify puffs (Logan et al., 2017; Kou et al., 2019)
3. Features describing puff behavior (e.g., how long diffusion takes, what diffusion looks like, etc.) are recorded (Yudowski et al., 2006; Bowman et al., 2015; Bohannon et al., 2017)
4. These features are compared across conditions

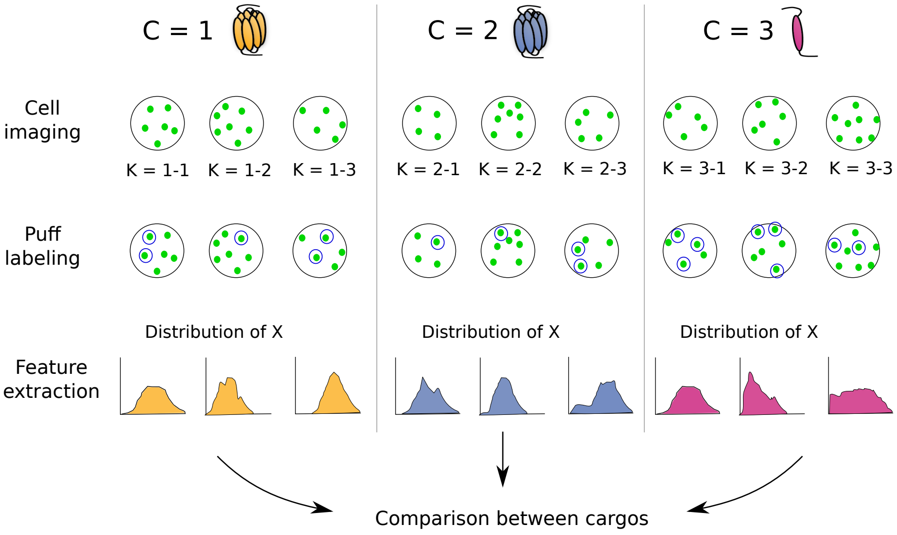

Figure II.2: Overview of an experiment to compare exocytic events between three different conditions ($C = 1, 2, 3$). For each condition, several cells K are imaged with TIRF microscopy, and bright spots are identified. These detected events are then labeled as puffs or nonpuffs, and a feature X is recorded for each puff. The distribution of X is then compared across conditions.

However, the volume of detected events in each cell makes manual labeling challenging. Automated labeling is therefore valuable, using a classifier trained to predict whether a bright spot is a puff or nonpuff. These predictions are then used for inference in place of hand labels.

Problem statement. Inspired by this motivating example, we consider data with the following structure, notated $\{(V_i, Y_i, C_i, K_i)\}_{i=1}^{n}$, where $Y \in \mathcal{Y}$ is the *unobserved* true label, V is a set of observed covariates, $C \in \mathcal{C}$ is experimental condition, and $K \in \mathcal{K}$ is a grouping variable nested within C that captures the hierarchical structure of the data. For instance, in the TIRF microscopy example, $Y \in \{0, 1\}$ denotes whether each event is a puff, V is a set of features derived from the microscopy images (either specially designed, or created by methods like convolutional neural networks), C denotes condition/receptor type, and K denotes the cell (Figure II.2). More generally, C may indicate an experimental condition, and K a repetition of that experiment. We present the problem in this hierarchical setting because it is most common, and note that our methods can be simplified when the hierarchical structure does not apply.

Comparing experimental conditions $C \in \mathcal{C}$ often involves asking one or both of the following questions:

1. How does label prevalence $P(Y = y|C = c)$ vary for conditions $c \in \mathcal{C}$?

2. How does the conditional distribution of $X|Y = y, C = c$ vary for conditions $c \in \mathcal{C}$, where $X \in \mathbb{R}$, $X \subset V$, denotes a feature of interest?

Crucially, Y is unobserved in new experimental data. To perform inference with unlabeled data, we have access to *labeled* training data $\{(V'_j, Y'_j, C'_j, K'_j)\}_{j=1}^m$, with the same set of labels $Y'_j \in \mathcal{Y}$, but different conditions $C'_j \in \mathcal{C}'$ and $K'_j \in \mathcal{K}'$. As new studies and experiments typically investigate different conditions, training and test data necessarily come from different experimental conditions and groups ($\mathcal{C}' \cap \mathcal{C} = \emptyset$ and $\mathcal{K}' \cap \mathcal{K} = \emptyset$). Therefore, when making predictions with the training data, we must consider possible differences in the data distribution between training $\{(V'_j, Y'_j, C'_j, K'_j)\}_{j=1}^m$ and test $\{(V_i, Y_i, C_i, K_i)\}_{i=1}^n$.

Contributions. This chapter makes the following contributions:

1. We explicitly define the statistical task of downstream inference about label prevalence $P(Y = 1|C = c)$ and the class-conditional means $\mathbb{E}[X|Y = y, C = c]$ using classification predictions. We codify the different sets of assumptions required to enable meaningful inference in this setting.
2. We develop semiparametric bootstrap methods for making downstream inference with classifier predictions, which properly incorporate variance when the required generalizability assumptions hold.
3. We demonstrate the use and performance of these methods in both simulation and in a detailed case study with TIRF microscopy data, where we describe the process of constructing a generalizable classifier, checking assumptions for valid statistical inference, and creating bootstrap confidence intervals.
4. Through our case study, we provide practical advice on feature and classifier construction in order to satisfy the assumptions necessary for downstream inference.

In Section II.2, we describe previous literature on automatic labeling and generalizable classifiers. In Section II.3, we present our bootstrap methods for inference with unlabeled data, and the different assumptions required. We demonstrate the performance of our methods on simulated data in Section II.4, and discuss possible modifications of our algorithms for different scenarios. Finally, in Section II.5, we perform inference in a case study with real TIRF microscopy data.

II.2 Background

II.2.1 Automated labeling in scientific studies

Automated labeling is useful in a wide range of applications, often with large image datasets. For example, Norouzzadeh et al. (2018) developed a classifier to identify animal species in camera trap images, as well as the number of animals in each picture and a description of their actions. The classifier was trained on millions of images from the Snapshot Serengeti dataset (Swanson et al., 2015), using convolutional neural networks. In ecology, large-scale classifiers have also been used to label deforestation (Maretto et al., 2020) and pest infestations (Rammer and Seidl, 2019). Automated labeling is also common in cell biology, where microscopy can produce thousands of images a day, which need to be annotated to identify nuclei (Caicedo et al., 2019), cell state and type (Christiansen et al., 2018), cell health phenotypes (Way et al., 2021), and protein localization (Kraus et al., 2017).

In each of these examples, it is important that classifiers generalize, or *transfer*, to new data. If predictions are not robust to changes in the distribution of input data, classifiers can fail when applied to new settings (Pan and Yang, 2009; Quiñonero-Candela et al., 2009). It has therefore become common for researchers constructing automated labeling systems to design classifiers which they expect to transfer. For example, Norouzzadeh et al. (2018) describe how their method can be updated for new camera trap locations with only a small amount of additional data.

II.2.2 Generalizable classifiers

A typical assumption in supervised learning is that training and test data come from the same distribution, which allows classifier predictions to be meaningful on new data. In practice, however, training and test data often come from different distributions, and so assumptions on the nature of distributional change are needed to understand how classifiers to generalize to new data.

Typically, we say that a classifier trained on data from condition $C = c'$, using the covariates V, generalizes to a new condition $C = c$ if $P(Y = y|V = v, C = c) = P(Y = y|V = v, C = c')$ (Subbaswamy et al., 2019). In this case, the features V satisfy the *covariate shift* assumption (Bickel et al., 2009; Gretton et al., 2009): the marginal distribution of V may change, but the conditional distribution of $Y|V$ remains the same. Under covariate shift, predictions can be applied directly to new data, or the classifier can be re-trained on weighted training data to be more efficient at risk minimization on test data (Shimodaira, 2000; Sugiyama et al., 2008). However, not all features typically satisfy the covariate shift assumption, as we expect systematic differences between conditions to appear in some features. Therefore, we write $V = \{X, Z, U\}$, where $X \in \mathbb{R}$ is a feature of interest for inference, U is a set of unused features, and $Z \in \mathbb{R}^d$ is a subset of covariates which satisfy the covariate shift assumption:

$$P(Y = y|Z = z, C = c) = P(Y = y|Z = z, C = c'), \quad \text{for all } c, c'. \tag{II.1}$$

Ideally, Z is also *sufficient* to capture the label information, so that $P(Y = y|X, U, Z, C) = P(Y = y|Z)$ (Peters et al., 2016; Kuang et al., 2018).

The strategy of identifying features that allow generalizability is common, and there are a variety of techniques. For example, Magliacane et al. (2017) and Rojas-Carulla et al. (2018) use variable selection techniques to identify a subset of predictors for which covariate shift holds, and Peters et al. (2016) performs hypothesis tests on the relationship between the predictors and the response. These methods are inspired by causal inference and causal discovery, as are Subbaswamy et al. (2019), who represent the data generating process explicitly with a causal graph and use the graph to identify stable predictors. Kuang et al. (2018) also use ideas from causal inference, in particular balancing models which use weights to account for differences in covariate distributions across environments, to identify a subset of covariates which generalize. More broadly, other authors have developed regularized regression methods to learn features which are common to multiple environments (Argyriou et al., 2007, 2008).

In some cases, however, it may not be possible to identify an appropriate subset of features Z for which covariate shift holds (Subbaswamy et al., 2019). For example, if $Z|Y = y, C = c \overset{d}{=} Z|Y = y, C = c'$, but the prevalence $P(Y = y|C = c) \neq P(Y = y|C = c')$, then predicted probabilities will not be calibrated for all $c \in \mathcal{C}$. This is the *label shift* scenario, in which the marginal distribution of labels changes, but the conditional distributions of features remain the same. Fortunately, there are a variety of methods for detecting and correcting for label shift, which allow predictions to be easily adjusted in this setting (Saerens et al., 2002; Storkey, 2009; Lipton et al., 2018; Garg et al., 2020).

To accomodate situations like label shift, in this chapter we consider a classifier trained on features Z to *generalize* if label probabilities $P(Y = y|Z = z, C = c)$ for each condition $c \in \mathcal{C}$ can be estimated from labeled training data $\{(Z'_j, Y'_j, C'_j, K'_j)\}_{j=1}^m$ and unlabeled test data $\{(Z_i, C_i, K_i)\}_{i=1}^n$—either by directly applying a classifier, in the case of covariate shift, or by correcting classifier predictions, like in the case of label shift. The existence of appropriate features Z is assumed; in practice, there are a variety of methods for identifying Z, as discussed above, and in our case study on TIRF microscopy data we use simple exploratory techniques.

II.3 Inference with generalizable predictions

We are interested in two inference questions using the classifier labels. First, how label prevalence, $P(Y = y|C = c)$, differs across conditions C. Second, how the conditional distribution of a feature of interest X, $X|Y = y, C = c$, differs across conditions C. As the labels Y are unobserved for the new data, we use a classifier $\mathcal{A}$ trained on labeled training data $\{(Z'_j, Y'_j, C'_j, K'_j)\}_{j=1}^m$. For simplicity, we'll assume that

$Y \in \{0,1\}$, so $\mathcal{A}(z) = \widehat{P}(Y' = 1|Z' = z)$, but the same methods can be used when labels belong to more than two classes. (Notation remark: training data is typically comprised of multiple conditions $C' \in \mathcal{C}'$, which may have different prevalences $P(Y' = 1|C')$. Probabilities which don't condition on C', e.g. $P(Y' = 1)$, are understood to refer to the specific combination of conditions in the observed training data). To be able to make predictions on new data from new conditions $c \in \mathcal{C}$, we assume that our classifier generalizes, as discussed above in Section II.2.2. In particular, we assume that the covariates Z satisfy the following assumptions:

(A1) $P(Y = 1|Z, C)$ can be estimated using the classifier $\mathcal{A}$, labeled training data $\{(Z'_j, Y'_j, C'_j, K'_j)\}_{j=1}^m$, and unlabeled test data $\{(Z_i, C_i, K_i)\}_{i=1}^n$.

(A2) $\mathcal{A}(z) = \widehat{P}(Y' = 1|Z' = z)$ is consistent for $P(Y' = 1|Z' = z)$.

If the features Z satisfy the covariate shift assumption, (A1) is straightforward, with $\widehat{P}(Y = 1|Z, C) = \mathcal{A}(Z)$. In other scenarios, classifier predictions $\mathcal{A}(Z)$ may need to be corrected on new data. For example, in the label shift setting, conditionwise prevalence $P(Y = 1|C = c)$ can be estimated for each condition $c \in \mathcal{C}$ using label shift correction methods (see Appendix A.1), and then $P(Y = 1|Z, C)$ is estimated via Bayes theorem:

$$\widehat{P}(Y = 1|Z, C) = \mathcal{A}_L(Z, C) := \frac{\frac{\widehat{P}(Y=1|C)}{\widehat{P}(Y'=1)}\mathcal{A}(Z)}{\frac{\widehat{P}(Y=1|C)}{\widehat{P}(Y'=1)}\mathcal{A}(Z) + \frac{1-\widehat{P}(Y=1|C)}{1-\widehat{P}(Y'=1)}(1 - \mathcal{A}(Z))}, \tag{II.2}$$

where $\mathcal{A}_L(Z, C)$ denotes the label shift-corrected predictions for condition C. For the purpose of this chapter, we will focus on the label shift setting.However, we note that our work can be applied to other settings as well.

II.3.1 Inference for prevalence

Our first goal is to construct a confidence interval for the conditionwise prevalence, $P(Y = 1|C = c)$. Given estimated probabilities $\widehat{P}(Y = 1|Z, C)$ that are close to the true probabilities $P(Y = 1|Z, C)$, point estimation of this quantity is straightforward: $\widehat{P}(Y = 1|C = c) = \frac{1}{\#\{i:C_i=c\}} \sum_{i=1}^{n} \widehat{P}(Y_i = 1|Z_i, C_i)\mathbb{1}\{C_i = c\}$. Alternatively, in the label shift setting, $\widehat{P}(Y = 1|C = c)$ is estimated separately by leveraging the label shift assumption (Appendix A.1). In either case, a simple binomial confidence interval for the prevalence $P(Y = 1C = c)$ does not suffice, because $\widehat{P}(Y = 1|C = c)$ relies on both training and test data. We therefore propose a bootstrap procedure which resamples both training and test data at each step. In particular, for bootstrap samples $s = 1, ..., B$ we

1. Resample the training data, $(Z_i'^*, Y_i'^*)$

2. Retrain the classifier, $\mathcal{A}^*$, on the bootstrap training data

3. Resample the test data (Z_i^*, C_i^*)

4. Re-estimate the prevalence $\widehat{P}(Y^* = 1|C^* = c)$

The full procedure, applied to the label shift setting, is described in Algorithm 2 in Appendix A.3. Algorithm 2 can also be easily modified for other forms of distributional change. For instance, in the case of covariate shift, we simply remove the label shift estimation and correction steps.

Similar bootstrap approaches are used below, for inference on $X|Y = y, C = c$. Here we make several remarks that apply to all the bootstrap procedures discussed in this chapter.

Remark: Retraining a classifier on bootstrapped training data may be time-consuming. For certain classifiers, it may be possible to sample a new classification function without re-fitting the full model. For example, for a logistic GAM, penalizing the spline fit is equivalent to placing a prior distribution on the spline coefficients (Krivobokova et al., 2010; Wood, 2017). This results in a posterior distribution for the classifier function, given the training data, and this posterior distribution has good frequentist properties (Krivobokova et al., 2010). Then, a bootstrapped classifier $\mathcal{A}^*$ can be sampled from this posterior distribution rather than by re-fitting on bootstrapped training data. Further details are provided in Appendix A.2.

Remark: Because estimates depend on both training and test data, our bootstrap procedure resamples both training and test. This also means that coverage for the resulting bootstrap confidence intervals is defined over pairs of training and test data $\{(Z_i', Y_i', C_i', K_i')\}$, $\{(Z_i, C_i, K_i)\}$. For example, if we construct a 95% confidence interval, in the long run 95% of training/test pairs $\{(Z_i', Y_i', C_i', K_i')\}, \{(Z_i, C_i, K_i)\}$ will produce an interval that captures the true parameter. It is *not* true that for any training set $\{(Z_i', Y_i', C_i', K_i')\}$, 95% of future test sets $\{(Z_i, C_i, K_i)\}$ will yield a confidence interval containing the true parameter.

Remark: Algorithm 2 (Appendix A.3) describes a bootstrap procedure for confidence intervals. Here our bootstrap intervals are first order, such as bootstrap z-intervals, bootstrap percentile intervals, or bootstrap pivotal intervals. The same approach can be used for more accuracte intervals, such as calibrated intervals, bootstrap t-intervals, and BC$_a$ intervals (DiCiccio et al., 1996). However, these more accurate intervals require a second level of sampling at each bootstrap iteration, which is likely to be computationally infeasible with classifier retraining inside the bootstrap.

II.3.2 Inference for feature distributions

Our second question is how the conditional distribution of a feature of interest $X|Y = y, C = c$ differs across conditions c. We will focus on confidence intervals for the class-conditional mean $\mathbb{E}[X|Y = y, C = c]$, though other summaries of the conditional distribution could be used instead. Given the hierarchical nature of the data, with grouping variables C and K, it is natural to model the conditional mean using a mixed effects model:

$$\mathbb{E}[X|Y = y, C = c, K = k] = \beta_{c,y} + b_k, \tag{II.3}$$

where $\beta_{c,y}$ is a fixed effect and $b_k \sim N(0, \omega^2)$ is a random effect.

The labels Y are unobserved, but we note that

$$\begin{aligned} \mathbb{E}[X|Y = y, C = c, K = k] &= \int x f_{X|Y=y,c,k}(x)dx = \int x \frac{P(Y = y|X = x, c, k)}{P(Y = y|c, k)} f_{X|c,k}(x)dx \\ &= \mathbb{E}\left[X \frac{P(Y = y|X, C = c, K = k)}{P(Y = y|C = c, K = k)} \middle| C = c, K = k \right]. \end{aligned} \tag{II.4}$$

Therefore, we can estimate $\beta_{c,y}$ in (II.3) using a weighted mixed effects model, where

$$X_i \sim N\left(\beta_{c_i,y} + b_{k_i}, \frac{\sigma_y^2}{w_{i,y}}\right), \qquad b_k \sim N(0, \omega^2), \tag{II.5}$$

with weights $w_{i,y} = P(Y_i = y|X_i, C_i, K_i)$. The assumption of a parametric form for the random effect, which is used in bootstrapping, is necessary when we observe few levels of K for each condition C, which is common in many scientific studies. The assumption of conditional normality for the feature of interest X is used for maximum likelihood estimation (or restricted maximum likelihood estimation) of the model parameters, but is not required for inference. As we describe below, our approach to inference involves a semiparametric bootstrap which resamples residuals from the fitted model, and we see in simulations (Section II.4) that departures from conditional normality do not seem to harm the coverage of our confidence intervals.

Since the true label probabilities $P(Y = y|X, C, K)$ are unknown, we use estimated probabilities instead, yielding weights $w_{i,y} = \widehat{P}(Y_i = y|Z_i, C_i)$. This requires the assumption that the feature of interest X provides no additional information about the label Y, after accounting for the covariates Z and the condition C. Formally, we assume the following, which is similar to assumptions found in Peters et al. (2016) and Kuang et al. (2018):

(A3) $P(Y = y|X, Z, C, K) = P(Y = y|Z, C)$.

Fitting the model (II.5) with probability weights yields a point estimate $\widehat{\beta}_{c,y}$. To construct a confidence interval for $\beta_{c,y}$, we bootstrap the training and test data, as in Section II.3.1. When K has many levels for each $C = c$, then a hierarchical bootstrap may be employed to resample the test data. However, in many scientific studies, K often has few levels for each C, and so we instead create bootstrap test data by sampling random effects and residuals. In particular, we define residuals for each class $y \in \mathcal{Y}$ by $e_i = X_i - \widehat{b}_{k_i}$, which we combine with new random effects $b_k^* \sim N(0, \widehat{\omega}^2)$. In the context of label shift, for each bootstrap sample $s = 1, ..., B$ we

1. Resample the training data, $(Z_i'^*, Y_i'^*)$
2. Retrain the classifier, $\mathcal{A}^*$, on the bootstrap training data
3. Resample the test data:
   (a) Sample $b_k^* \sim N(0, \widehat{\omega}^2)$ for each group $k \in \mathcal{K}$
   (b) Sample $(Z_i^*, C_i^*, \mathcal{A}_L(Z_i^*, C_i^*), e_i^*)$ by resampling rows (to preserve any correlation between covariates Z and residuals e_i)
   (c) Sample $Y_i^* \sim \text{Bernoulli}(A_L(Z_i^*, C_i^*))$
   (d) Generate new observations X_i^* by $X_i^* = e_i^* + b_k^*$
4. Calculate the label shift correction on the bootstrap training and test data, and re-fit the weighted mixed effects model

The full details are provided in Algorithm 3 in Appendix A.3. As in Section II.3.1, Algorithm 3 can be modified for other forms of distributional change. For covariate shift, simply remove the label shift correction steps, and replace $\mathcal{A}_L(Z, C)$ with $\mathcal{A}(Z)$.

Remark: Rather than probability weights, inference may also be based directly on binary predictions $\widehat{Y}_i \in \{0, 1\}$. The procedure is similar, just with $\mathbb{E}[X|\widehat{Y} = 1, C = c, K = k]$. If the classifier predictions $\mathcal{A}_L(Z, C)$ are good, we generally expect probability weights to give better estimates than binary predictions. In particular, if there is a relationship between X and $\mathcal{A}_L(Z, C)$, then thresholding classifier predictions to produce binary labels will lead to biased estimates.

Remark: A consequence of assumption (A3) is that the random effect b_k in (II.3) does not depend on the label Y. If random effects are in fact label-dependent, which may be assessed with training data, separate random effects can be estimated when fitting (II.5) and in constructing bootstrap confidence intervals. However, label-dependent random effects violate (A3) and so may lead to a decrease in confidence interval coverage. We investigate this further, and suggest a potential adjustment to improve coverage in Section II.4.

II.3.3 Mixture models: an alternative to probability weighting

Assumption (A3) states that the covariates Z are sufficient for classification. This assumption can be checked on training data, but it may be challenging to find a subset of covariates Z which satisfies both (A3) and (A1), or even one which satisfies (A1) alone. In this case, estimating appropriate weights for the weighted mixed effects model (II.5) may be difficult. An alternative is to recognize that inference for the conditional feature distribution $X|Y = y, C = c$ naturally fits a mixture model approach, with the observed distribution of $X|C = c$ being a mixture of conditional distributions $X|Y = y, C = c$ over unobserved labels $y \in \mathcal{Y}$.

If $X|Y = y, C = c$ is assumed to follow a parametric distribution, then maximum likelihood estimation of the model parameters is possible. For example, we might assume a Gaussian hierarchical mixture where

$$X|Y = y, C = c, K = k \;\sim\; N(\beta_{c,y} + b_{k,y}, \sigma_y^2), \tag{II.6}$$

and $b_{k,y} \sim N(0, \omega_y^2)$. This replaces assumptions (A1) - (A3) with parametric assumptions on the conditional distribution of the feature of interest X; while we use Gaussian mixture models throughout this chapter, as in (II.6), other parametric distributions can be chosen based on the observed training data. Furthermore, by removing assumption (A3), it is straightforward to allow label-dependent random effects $b_{k,y}$ in (II.6). (*Note:* the parametric assumptions for the mixture model (II.5) are much more important for estimation than the parametric model used for mixed effect model estimation (II.6).)

However, mixture models can be difficult to estimate well, particularly when parametric assumptions are violated, and when the class distribution is unbalanced. To assist with mixture model estimation, we can use information from classifier predictions. In particular, we propose using a classifier to estimate $P(Y = y|C = c)$, and then using these class proportions to improve mixture model estimation. This approach relies on (A1) and (A2), but not (A3).

Thus we have two alternative assumptions for inference on the conditional mean $\mathbb{E}[X|Y = y, C = c]$: that Z is sufficient for classification (assumption (A3)), or that we know a parametric form for the conditional distribution $X|Y = y, C = c$ (as in (II.6)). Which is assumption is more appropriate is problem-specific. The bootstrap procedure is almost identical to the one described in Section II.3.2, the only difference is the model used for parameter estimation, and that residuals $e_{i,y}$ are calculated for each class to accomdate label-dependent random effects $b_{k,y}$. Algorithm 4, in Appendix A.3, describes the full procedure in detail in the context of label shift; as before, modifications are straightforward.

II.4 Simulations

In this section, we investigate the performance of the bootstrap procedures described in Section II.3, using simulated data. All of our methods rely on some subset of: assumpions (A1), (A2), (A3), parametric assumptions about a feature X, and assumptions about the relationship between labels and random effects. To evaluate the impact of assumptions on the performance of our bootstrap methods, we assess coverage of bootstrap confidence intervals when different assumptions are satisfied.

As discussed in Section II.3.1, inference for label prevalence requires assumptions (A1) and (A2), which allow label probabilities to be estimated on new data using a subset of covariates Z. For inference on a feature of interest X, we also require either assumption (A3) (if using the mixed effects approach of Section II.3.2) or a known parametric form for the class distributions (if using the parametric mixture model approach of Section II.3.3). To evaluate performance of our proposed confidence intervals, we consider six different simulation settings, varying the assumptions that are satisfied. For simplicity, we consider a single additional covariate Z, one test condition C ($|\mathcal{C}| = 1$), and 15 nested subgroups K ($|\mathcal{K}| = 15$). As in the rest of this chapter, we focus on label shift to illustrate our proposed procedures.

Scenarios. We consider three main scenarios, under which different combinations of (A1), (A2), and (A3) are satisfied. To evaluate the sensitivity of the mixture model approach to parametric assumptions, we use Gaussian mixture models and generate data from both Gaussian and skewed normal distributions for each scenario. The parameters of the skewed normal distributions are chosen so that variance and separation between the two class distributions are roughly equivalent to the Gaussian case. Full simulation details are provided in Table A.1 in Appendix A.5.

> **Scenario 1:** Assumptions (A1), (A2), and (A3) are satisfied. The covariate Z used for classification satisfies the label shift assumption between the training and test data (A1), and a logistic spline fit is used for classification (A2). Gaussian random effects b_k are generated for each $k \in \mathcal{K}$. The feature of interest X is given by $X_i = Z_i + b_{k_i} +$ noise, satisfying (A3).
>
> **Scenario 2:** Assumptions (A2) and (A3) are satisfied, but (A1) is not. Though label shift methods are employed for constructing confidence intervals, as in Algorithms 2, 3, and 4 (Appendix A.3), the

conditional distribution of $Z|Y = 0$ differs between training and test data. As in Scenario 1, a logistic spline fit is used for classification, and $X_i = Z_i + b_{k_i} + \text{noise}$.

Scenario 3: Assumptions (A1) and (A2) are satisfied, but (A3) is not. As in Scenario 1, the covariate Z satisfies the label shift assumption, and a logistic spline fit is used for classification. However, $\mathbb{E}[X|Z, Y = 1] \neq \mathbb{E}[X|Z, Y = 0]$, which violates (A3).

Comparisons. For each scenario, we calculate estimates and 95% confidence intervals for the prevalence $P(Y = 1|C = 1)$, and the class mean $\mathbb{E}[X|Y = 1, C = 1]$ (recall that for the test data, we consider only one test condition, i.e. $|\mathcal{C}| = 1$). Inference for prevalence is done as in Section II.3.1. Inference for $\mathbb{E}[X|Y = 1, C = 1]$ is done with both mixed effects models (Section II.3.2) and mixture models (Section II.3.3). Gaussian mixture models are used, and the mixing proportions are first estimated using label shift methods (Appendix A.1). As the random effect b_k in simulations does not depend on the class label Y, we modify Algorithm 4 to fit a single random effect for each $k \in \mathcal{K}$, as in Algorithm 3. We then compare the bias of the point estimates from each method, and the observed coverage of nominal 95% bootstrap pivotal intervals. Logistic splines were fit using the `mgcv` package (Wood, 2011) in `R`, while mixed effects models used the `lme4` package (Bates et al., 2015), and mixture models were implemented in `stan` using `rstan` (Stan Development Team, 2020).

Results. Average point estimates and confidence interval coverage are shown in Table II.1. Inference for the prevalence $P(Y = 1|C = 1)$ depends on the validity of assumptions (A1) and (A2), and Table II.1 shows that bias is small and coverage is close to the nominal level when (A1) and (A2) are satisfied, regardless of assumption (A3) and the parametric form of the data. The mixed effects model requires the additional assumption (A3) in order to perform inference on the feature of interest X. When (A1), (A2), and (A3) are satisfied, bias is close to 0 and coverage is close to 95%, and this holds for both the normal and skewed normal distributions. However, when (A1) or (A3) are violated, the point estimates become biased, leading to a decrease in coverage. In contrast, the Gaussian mixture model requires (A1) and (A2), along with a feature X which is conditionally normal given class $y \in \mathcal{Y}$.

When these assumptions are met, bias and coverage both behave well. However, departures from normality lead to biased estimates and lower coverage. Violation of the label shift assumption also results in poor performance, because the label shift assumption is used to estimate mixing proportions.

Assumptions	Normal?	Prevalence		Mixed Effects Model		Mixture Model	
		Mean	Coverage	Mean	Coverage	Mean	Coverage
(A1), (A2), (A3)	yes	0.4 (0.001)	0.93 (0.015)	2.99 (0.008)	0.93 (0.014)	3 (0.007)	0.94 (0.013)
	no	0.4 (0.002)	0.9 (0.018)	3.51 (0.009)	0.93 (0.014)	3.86 (0.010)	0.28 (0.026)
(A2), (A3)	yes	0.35 (0.001)	0.07 (0.015)	3.16 (0.008)	0.72 (0.026)	3.13 (0.007)	0.8 (0.023)
	no	0.45 (0.002)	0.71 (0.026)	3.44 (0.009)	0.91 (0.016)	3.74 (0.009)	0.54 (0.029)
(A1), (A2)	yes	0.4 (0.001)	0.93 (0.015)	3.87 (0.008)	0.86 (0.020)	4 (0.007)	0.94 (0.014)
	no	0.4 (0.002)	0.9 (0.018)	4.12 (0.010)	0.41 (0.028)	4.68 (0.012)	0.63 (0.028)

Table II.1: Coverage of bootstrap confidence intervals in simulated data.

II.4.1 Label-dependent random effects

As discussed in Section II.3.2, the weighted mixed effects model (II.5) relies on assumption (A3) to use classifier predictions as probability weights. This assumes that the feature of interest, X, adds no information to the covariates Z to distinguish between labels $Y = 0$ and $Y = 1$. In practice, this may be true on average across the population but not within groups $k \in \mathcal{K}$. In particular, we may observe that the random effect b_k depends on both the group k and also the label Y, which violates (A3).

Label dependence can be accomodated in the mixed effects model by modifying Algorithm 3 to estimate separate separate random effects $b_{k,y}$ for $Y = 0$ and $Y = 1$ within each group $k \in \mathcal{K}$, as in Algorithm 4. Table II.2 shows the results of mixed effects estimation and coverage. Here data is simulated as in Table A.1, except that label-specific random effects $b_{k,y}$ are simulated separately for each $y \in \{0, 1\}$ (that is, $X_i = Z_i + b_{k,0}\mathbb{1}(Y_i = 0) + b_{k,1}\mathbb{1}(Y_i = 1) + \text{noise}$), and the model is fit by estimating each $b_{k,y}$ separately. Comparing Table II.2 to Table II.1, we see that while our point estimates perform equivalently, label dependence for the random effects results in decreased coverage for the confidence intervals.

This decrease in coverage arises because the probability weights in (II.5) are slightly wrong, which causes the random effect variance ω_y^2 to be underestimated. Fortunately, this issue can be corrected by an additional variance calibration step in the bootstrap. Let $\widehat{\omega}_y^2$, $y \in \{0, 1\}$, be the initial variance estimates from fitting weighted mixed effects models. For bootstrap samples $s = 1, ..., B$, we sample bootstrap training and test data as described in Algorithm 4, using our initial estimates $\widehat{\omega}_y^2$. For each sample, we then calculate the observed variance estimate $\widehat{\omega}_{y,s}^{*2}$. The same process that results in $\widehat{\omega}_y^2$ being biased for ω_y^2 will cause $\widehat{\omega}_{y,s}^{*2}$ to be biased for $\widehat{\omega}_y^2$. Using the $\widehat{\omega}_{y,s}^{*2}$ and $\widehat{\omega}_y^2$, we estimate a variance correction, which we apply to $\widehat{\omega}_y^2$. The corrected variance estimate is then used for bootstrap simulation to construct a confidence interval. Full details are provided in Appendix A.4.

Table II.2 shows the estimates and coverage of the variance-adjusted mixed effects bootstrap. We can see that the bias remains the same, but adjusting the variance improves confidence interval coverage. For comparison, we also assess the Gaussian mixture model approach with label-dependent random effects (II.6). When all assumptions are satisfied, coverage of the mixture model confidence intervals is slightly worse than the variance-adjusted mixed effects approach, and is slightly worse than mixture model coverage in Table II.1. This is because allowing label-dependent random effects increases the number of quantities to estimate in the mixture model, which makes model fitting more challenging.

Assumptions	Normal?	Mixed Effects Model		Mixed Effects, Variance Adjustment		Mixture Model	
		Mean	Coverage	Mean	Coverage	Mean	Coverage
(A1), (A2), (A3)	yes	3 (0.007)	0.91 (0.016)	3 (0.007)	0.94 (0.013)	3.01 (0.009)	0.89 (0.018)
	no	3.48 (0.007)	0.83 (0.022)	3.49 (0.006)	0.93 (0.015)	3.86 (0.008)	0.13 (0.020)
(A2), (A3)	yes	3.16 (0.008)	0.6 (0.028)	3.15 (0.008)	0.71 (0.026)	3.12 (0.008)	0.81 (0.022)
	no	3.45 (0.006)	0.91 (0.017)	3.42 (0.007)	0.89 (0.018)	3.75 (0.007)	0.37 (0.028)
(A1), (A2)	yes	3.88 (0.007)	0.78 (0.024)	3.89 (0.008)	0.84 (0.021)	4.01 (0.008)	0.86 (0.020)
	no	4.12 (0.007)	0.18 (0.022)	4.13 (0.008)	0.25 (0.025)	4.7 (0.009)	0.5 (0.029)

Table II.2: Coverage of bootstrap confidence intervals in simulated data.

II.5 Case study: live cell microscopy data

In this section, we apply our methods for inference with classifier predictions to a large live cell microscopy dataset that was collected with TIRF microscopy and manually labeled. This will allow us to explore the process of assessing generalizability assumptions, engineering generalizable classifiers, and evaluating inference with unlabeled data.

Note: Aside from the issues we discuss in this chapter, this TIRF dataset is known to have some human label bias. For illustrative purposes, we will ignore that label bias here and take the human labels as ground truth.

II.5.1 Data, classifier, and assumptions

Data. Labeled data was collected on TIRF microscopy images under three different experimental conditions, which we will refer to as Condition 1, Condition 2, and Condition 3, denoted respectively by $C = 1, 2, 3$. The experiment recorded data for 18 different cells, with 5 cells from Condition 1, 6 from Condition 2, and 7 from Condition 3, yielding a total of 134127 events across the 18 cells, with approximately 2.7% of the events being puffs.

Table II.3 shows the breakdown of events in each cell. Conditions 1 and 2 have similar puff rates, while Condition 3 has a higher proportion of puffs, and furthermore it is hypothesized that puffs for Condition 3 may have different characteristics than puffs from Conditions 1 and 2. For this case study, we will therefore treat Conditions 1 and 2 as training data and Condition 3 as test data, allowing us to study generalization. For statistical inference, we want to construct confidence intervals for Condition 3. To reflect the process of

	Training data										
Condition	1	1	1	1	1	2	2	2	2	2	2
Cell	1-1	1-2	1-3	1-4	1-5	2-1	2-2	2-3	2-4	2-5	2-6
# Puffs	71	254	78	35	118	62	108	37	34	72	16
Puff prevalence	0.012	0.024	0.011	0.005	0.020	0.007	0.011	0.004	0.010	0.009	0.003

	Test data						
Condition	3	3	3	3	3	3	3
Cell	3-1	3-2	3-3	3-4	3-5	3-6	3-7
# Puffs	147	488	509	482	266	604	183
Puff prevalence	0.024	0.052	0.050	0.062	0.040	0.075	0.049

Table II.3: Breakdown of events in each cell of the TIRF microscopy data, showing the number of puffs and the prevalence.

classifier construction and assessment, we divide Conditions 1 and 2 into training data (7 cells) and validation data (4 cells). Note that the split between training and validation data is done by cell, to better capture cell-to-cell variability.

Features. We will investigate the hypothesis that the pattern of diffusion into the cell membrane after vesicle fusion differs between conditions. To capture these diffusion differences, we create a feature called *Smoothness*, which we expect may vary across conditions. The feature is construted using a functional PCA approach on the raw images (see Appendix A.6) to capture a smoothness aspect of the diffusion. In the notation from Section II.1, *Smoothness* is our feature of interest X.

We also construct a set of features Z for classification. These are carefully designed to capture the fundamental characteristics of puffs shown in Figure II.1, and are also created from a functional PCA representation of the images (see Appendix A.6). Because we expect common geometric characteristics of puffs to be preserved across conditions, but that the rate of puffs will differ, we will aim for generalizable prediction by constructing a set of features—and therefore a classifier—that obey the label shift assumption. We construct these by comparing the distributions of our designed features across conditions, selecting those features for which the label shift assumption appears to hold in exploratory data analysis. These are:

- $IntensityRatio$: the event's maximum intensity / minimum intensity (Δ_f)
- SNR: a measure of the signal-to-noise in the event
- $ConvexArea$ and $ConvexPerimeter$: measures of the amount of diffusion in the event's intensity over time
- $Randomness$: a measure of randomness in the event's time series

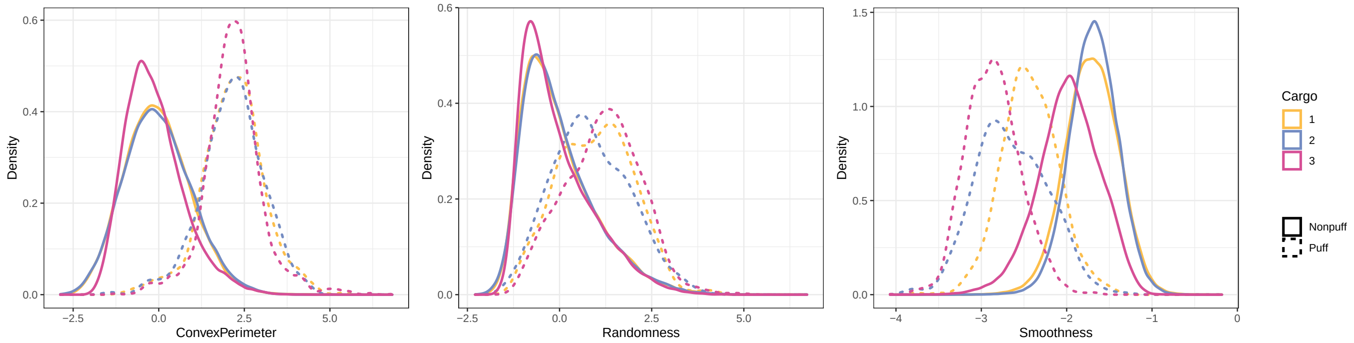

Figure II.3: Distribution of three features calculated on the TIRF events, broken down by condition and puff/nonpuff. The first two features show label shift between the different conditions, and will be used to construct a classifier. The distribution of the third feature, *Smoothness*, changes between conditions. Inference on *Smoothness* in new conditions is of interest, but the feature is not included in the classifier because of the change in distribution.

For example, Figure II.3 shows the distribution of $ConvexArea$ and $ConvexPerimeter$; as the feature distribution within each class (puff and nonpuff) is the same between conditions, the only difference is one of label shift, which supports (A1). In contrast, the distribution of X ($Smoothness$) does not satisfy label shift, as expected (Figure II.3).

To evaluate assumptions (A2) and (A3), we first need to construct a classifier.

Classifier. A logistic GAM (Hastie and Tibshirani, 1990; Wood, 2017) is trained on the seven training cells, using features $Z = IntensityRatio$, SNR, $ConvexArea$, $ConvexPerimeter$, and $Randomness$. We choose a logistic GAM because it provides a balance between simplicity and flexibility, and because sampling from the posterior (see Appendix A.2) makes our bootstrap procedures much more efficient. Performance of the classifier is assessed on the four validation cells, and the classifier is then applied to the test data. Use of validation data allows us to assess performance of classifier predictions on new data from the training distribution, which provides a benchmark for performace on new data from a different distribution. Figure II.4 shows calibration plots for the predicted probabilities on validation and test data. As expected, the predicted probabilities appear to be calibrated on the validation data from Conditions 1 and 2 (which supports (A2)), but are mis-calibrated—due to label shift—on the test data from Condition 3. Furthermore, because the label shift assumption is appropriate for our classifier features (Figure II.3), $P(Y = 1|C = 3)$ can be estimated as in the Appendix A.1. The resulting estimate is 0.043 using the method from Lipton et al. (2018), and 0.050 using the fixed point method based on Saerens et al. (2002) (Section II.5.2), compared to the true sample prevalence of 0.051. Using this estimate, we correct the classifier predictions as in (II.2) with Bayes theorem, and Figure II.4 shows the corrected predictions are much better calibrated, which again supports (A1).

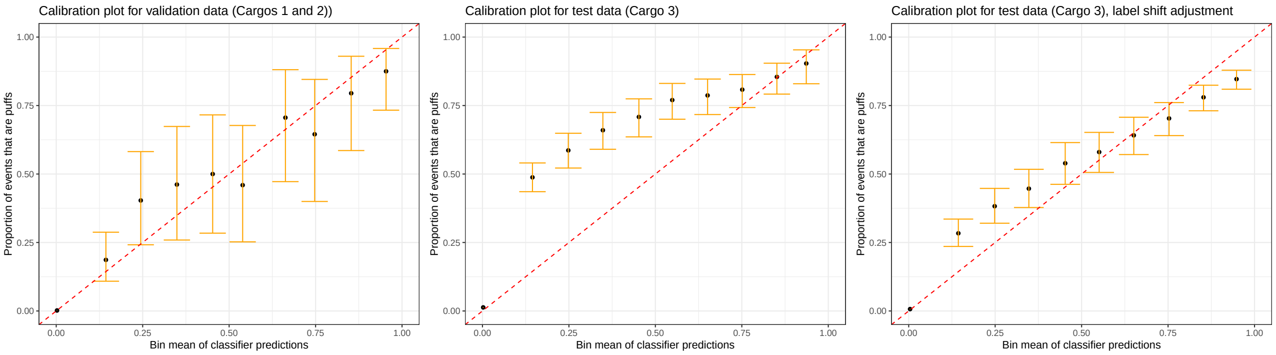

Figure II.4: Calibration plots for the classifier on validation (left panel) and test data (middle and right panels). Because TfR cells have a higher proportion of puffs (Table II.3), classifier predictions are mis-calibrated (middle panel), but they can be corrected with a label shift adjustment (right panel).

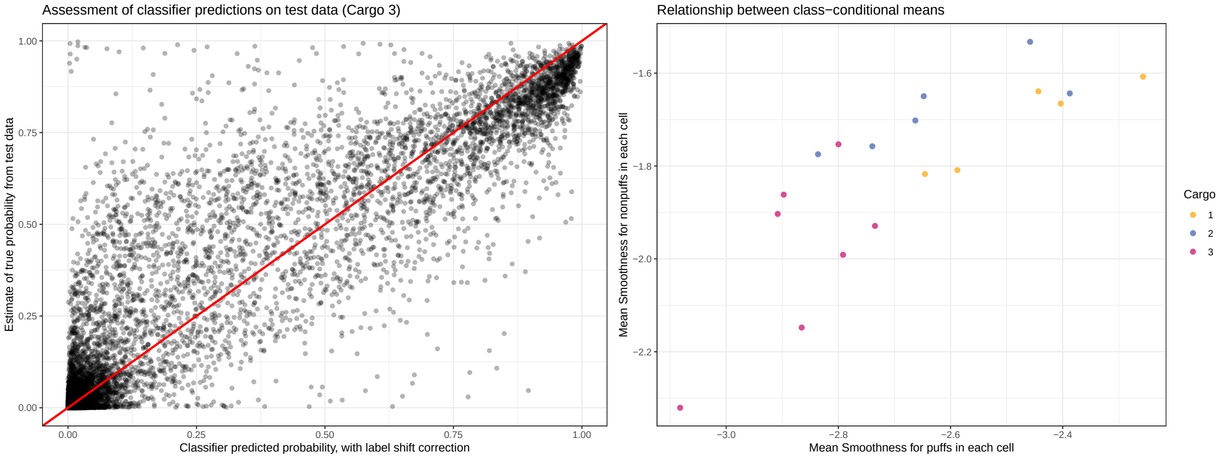

Figure II.5: Left: $\widehat{P}(Y_i = 1|X_i, Z_i)$ vs. $\mathcal{A}_L(Z_i, C_i)$ on test data. Right: Relationship between the class-conditional means $\mathbb{E}[X|Y = 1, C = c, K = k]$ and $\mathbb{E}[X|Y = 0, C = c, K = k]$ for each cell condition c and cell k.

Finally, to conduct inference about $X|Y = y, C = c$, we require (A3) to hold. To assess (A3), we compare $\mathcal{A}_L(Z, C)$ from (II.2) to an estimate of $P(Y = 1|X, Z, C)$ from regression on the test data. The resulting plot is shown in Figure II.5, which suggests that while not perfect, the corrected predictions $\mathcal{A}_L(Z, C)$ are a good estimate of the true probability $P(Y = 1|X, Z, C)$. As discussed in Section II.3 and Section II.4, a consequence of (A3) is that the per-cell random effect b_k should not depend on the label Y. Figure II.5 also shows the relationship between $\mathbb{E}[X|Y = 1, C = c, K = k]$ and $\mathbb{E}[X|Y = 0, C = c, K = k]$. As the relationship is roughly linear with a slope of approximately 1, the assumption of label-independent random effects is not unreasonable, but confidence intervals can also be constructed that model label-dependent random effects if desired, as described in Section II.4.

Cell	Puff mean	Weighted mean, raw	Weighted mean, label shift	Threshold mean, raw	Threshold mean, label shift	Nonpuff mean
3-1	-3.08	**-3.09**	-3.01	-3.19	-3.11	-2.32
3-2	-2.90	-2.97	**-2.84**	-3.10	-2.99	-1.86
3-3	-2.91	-2.95	-2.86	-3.05	**-2.94**	-1.90
3-4	-2.87	-2.99	**-2.92**	-3.10	-2.99	-2.15
3-5	-2.80	-2.92	**-2.82**	-3.06	-2.94	-1.75
3-6	-2.79	-2.92	**-2.85**	-3.04	-2.93	-1.99
3-7	-2.73	-2.78	-2.67	-2.90	**-2.77**	-1.93
Combined	-2.87	-2.95	**-2.85**	-3.06	-2.95	-1.99

Table II.4: Performance of estimates with classifier predictions on test data (Condition 3).

II.5.2 Inference with classifier predictions

Inference for prevalence. We begin with inference for $P(Y = 1|C = 3)$. Our point estimate, from label shift estimation, was 0.050 using the fixed point method (see Saerens et al. (2002) and Appendix A.1) and 0.043 using the method from Lipton et al. (2018). Using the procedure described in Algorithm 2, we construct a 95% confidence interval for $P(Y = 1|C = 3)$ with each label shift method, which are respectively (0.047, 0.055) and (0.039, 0.049). Both label shift estimates are a marked improvement on the point estimate from uncorrected classifier probabilities, which is 0.029.

To assess coverage of this interval, we perform a simulation using the real TIRF microscopy data. We simulate new training and test sets by resampling from the original training and test data, using Algorithm 2 to calculate a bootstrap confidence interval for the prevalence $P(Y = 1|C = 3)$ in each simulation. The simulated training data ($C = 1, 2$) has a prevalence of 0.01, while the simulated test data ($C = 3$) has a prevalence of 0.05. Using the fixed point method (Appendix A.1 and Saerens et al. (2002)), nominal 95% bootstrap pivotal intervals have a coverage of 95% in simulations, while coverage using the discretization method (Lipton et al., 2018) is about 19%. The lower coverage using the discretization method is due to bias, which may result because the prevalences are close to 0, or because the label shift assumption is not perfectly satisfied.

Inference for feature distributions. Next, we are interested in constructing a confidence interval for $\mathbb{E}[X|Y = 1, C = 3]$, where X is diffusion *Smoothness*. We first examine estimates of mean puff *Smoothness* in each test cell, using classifier predictions. Table II.4 shows estimates using both probability weights and binary predictions from thresholding, and compares the classifier with and without label shift correction. We can see that the estimated means are close to the true sample means in each cell, and as expected the label shift correction produces better estimates. The weighted mean generally does better than the thresholding-based mean, which is biased because classifier predictions are negatively associated with *Smoothness*.

The point estimate for $\mathbb{E}[X|Y = 1, C = 3]$, using the weighted mixed effects model (II.5), is -2.85. Using the procedure described in Algorithm 3, we construct a confidence interval, which is (-3.00, -2.73). To assess coverage of our confidence interval on TIRF microscopy data, we simulate new training and test sets from the real data. Training data is sampled by bootstrapping from the original training data, and test data is simulated by adding a per-cell random effect to *Smoothness* in bootstrap samples from the original test data. For each simulated training and test pair, the procedure in Algorithm 3 is used to construct a confidence interval, and coverage is assessed across training/test pairs. In these simulations, nominal 95% bootstrap pivotal intervals have a coverage of about 90%, with similar numbers for other first-order intervals like bootstrap percentile intervals. This is close to the coverage seen in our simulations in Section II.4.

Using a modification of Algorithm 3 to accomodate label-dependent random effects, a 95% bootstrap confidence interval for $\mathbb{E}[X|Y = 1, C = 3]$ in the test data is (-2.93, -2.78). In simulations with label-dependent random effects, these nominal 95% intervals again have a coverage of about 90%. The difference in width between the two confidence intervals likely results from different estimates of the random effect variance: as the nonpuff cell means vary slightly more than the puff cell means in the observed data, bootstrap puff data varies less when we allow label-dependent random effects.

Mixture models. As an alternative to probability weighting, we employ the mixture model approach described in Section II.3.3. We use a two-component Gaussian mixture, fitting the hierarchical mixture described in (II.6). To improve estimation of the mixture model, we pre-specify the prevalence of puffs in the test data, using the label shift estimate that we calculated above. Our point estimate for $\mathbb{E}[X|Y = 1, C = 3]$ is then $\widehat{\beta}_{3,1} = -2.95$, and the confidence interval from Algorithm 4 is (-2.99, -2.85). To assess coverage, we simulate from training/test pairs as with the mixed effects model above. Even though pre-specifying the puff prevalence improves the mixture estimation, the nominal 95% bootstrap percentile intervals have a coverage of approximately 15%. The poor coverage here is due largely to bias resulting from skewness in the data, which matches the simulation results for the skewed normal distribution in Section II.4. We experimented with other parametric models beyond a Gaussian mixture, but estimation remained challenging. Figures showing the fitted Gaussian mixture model in each cell (with and without pre-specifying puff prevalence) can be found in Appendix A.7, in addition to the estimated means in each cell.

II.6 Discussion

Scientific studies often require painstakingly labeling large volumes of unlabeled data, causing labeling to be a key limiting factor in data analysis. If data can be automatically labeled with predictive models, manual labeling costs can be dramatically reduced and much higher through-put science can be enabled. However, rigorous scientific analysis with predicted labels requires generalizable classifier predictions for valid inference.

We have described methods for valid inference two common downstream targets of inference: the label prevalence $P(Y = y|C = c)$ and class-conditional feature means $\mathbb{E}[X|Y = y, C = c]$. Inspired by our motivating example from TIRF microscopy, we focus on the case where the classifier used for automatic labeling is trained on data that differs in distribution from the new, unlabeled data. As this dataset shift may prevent classifiers from generalizing to the new data, and therefore prevent valid statistical inference, we rely on identification of a subset of features that enable construction of a generalizable classifier. These features can be designed from training data, and a variety of methods exist to construct features which satisfy the covariate shift assumption (Peters et al., 2016; Magliacane et al., 2017; Kuang et al., 2018; Rojas-Carulla et al., 2018; Subbaswamy et al., 2019). In our TIRF microscopy case study, a label shift assumption is more appropriate than covariate shift, and we show that exploratory data analysis and careful feature engineering can construct generalizable covariates.

While we focus on the label shift setting in our algorithms, simulations, and case study, our methods can be easily modified for other types of dataset shift. Furthermore, our methods are designed to accomodate a flexible hierarchical data structure, which is common to scientific experiments in which multiple repetitions of the experiment are performed for each experimental condition. Through simulations and a case study with TIRF microscopy data, we show that the tools presented in this chapter allow statistical inference with unlabeled, classifier-scored data, and that a generalizable classifier is crucial for valid analysis.

For inference with the class-conditional means $\mathbb{E}[X|Y = y, C = c]$, we describe two approaches for using classifier predictions for estimation and confidence intervals: weighted mixed effects models that use classifier predictions as probability weights, and hierarchical mixture models that use classification to estimate mixing proportions. The specific assumptions required by these two approaches are detailed in Section II.3.2 and Section II.3.3. As seen in simulations and the TIRF microscopy case study, the weighted mixed effects method is particularly sensitive to violations of assumption (A3), that the feature of interest X does not help distinguish between the classes $y \in \mathcal{Y}$ once we condition on the classifier features Z. On the other hand, the mixture model method is particularly sensitive to departures from the assumed parametric class distributions. Which assumption is more appropriate is problem dependent; fortunately, though assumptions typically cannot be checked on test data, they can be assessed on training data. As a result, collecting a large and diverse training set, with observations from many domains, is an important part of data analysis.

The methods we propose in this chapter allow valid statistical inference under appropriate assumptions, but there are still limitations to inference on unlabeled data. Even if classifier generalizability assumptions hold for inference on some features of interest, they may fail for others. In other cases, it may be impossible to construct features which satisfy the necessary conditions. Interpretation of inference results is also more nuanced: since confidence intervals must account for variability in the training data, coverage applies to joint training/test pairs, rather than to all new test datasets. Even more subtly, we often rely on humans to provide "ground truth" labels for classifier training; however, if researchers are accustomed to labeling observations under only some experimental conditions, then manual labels may suffer the same generalizability problems as automated predictions. While the methods in this chapter may not help generalize human predictions, we believe that the explicit assumptions discussed above can still help researchers reflect on their own manual labeling system.

Chapter III

Study design for inference with classifier predictions

III.1 Introduction

In this chapter, we consider pitfalls in data collection which can prevent classifiers generalizing to new data, and show how to mitigate these issues with careful study design. Many authors have already suggested best practices for statistical learning, including data splitting (training, validation, and test sets) to avoid overfitting, accounting for imbalanced classes, appropriately incorporating variability, and accounting for correlations and hierarchical relationships between observations (Kass et al., 2016; Chicco, 2017; Makin and de Xivry, 2019). Such guidelines can be seen in practice in the case study in Chapter II, and apply to a broad range of statistical learning applications. Here we focus on issues which particularly affect the use of classifier predictions on new, unlabeled data from a different distribution to the training data. Practitioners must carefully design their studies to avoid these potential pitfalls, in addition to heeding existing guidance, when performing inference with classifier predictions.

Contributions: In this chapter, we make the following contributions:

1. We identify potential sources of bias which can violate the generalizability assumptions discussed in Chapter II.
2. We describe steps to mitigate biases and promote classifier generalizability.

3. We show the effects of bias on inference with classifier predictions, using the results of real TIRF microscopy experiments before and after implementing our proposed steps. This expands on the case study used in Chapter II.

In Section III.2, we identify potential sources of error in classifier training which can violate the generalizability assumptions. In Section III.3, we then describe steps which can be taken to mitigate these errors. Throughout, we illustrate our advice with multiple datasets from a TIRF microscopy study, in which the initial data suffered from biases, and subsequent experiments were run to correct these problems.

III.2 Sources of bias

Using the notation from Chapter II, a classifier trained on a set of features $Z \in \mathbb{R}^d$, under conditions $C \in \mathcal{C}'$, generalizes to new conditions $C \in \mathcal{C} \neq \mathcal{C}'$ if for each $c \in \mathcal{C}$, $P(Y = y|Z = z, C = c)$ can be estimated from labeled training data $\{(Z'_j, Y'_j, C'_j, K'_j)\}_{j=1}^m$ and unlabeled test data $\{(Z_i, C_i, K_i)\}_{i=1}^n$. For example, under the covariate shift assumption, $P(Y = y|Z = z, C = c') = P(Y = y|Z = z, C = c)$ and so classifier predictions can be applied directly to new data, while under the label shift assumption $P(Y = y|Z = z, C = c)$ is calculated from $P(Y = y|Z = z, C = c')$ via Bayes rule. Provided the classifier predictions $\mathcal{A}(z) = \widehat{P}(Y = 1|Z = z)$ are close to the true probabilities (assumption (A2)) and can be generalized to new data (assumption (A1)), then inference on the class prevalence $P(Y = y|C = c)$ is possible. For inference on the conditional mean $\mathbb{E}[X|Y = y, C = c]$ of a feature of interest X, additional assumptions are needed – for example, (A3) is used for probability-weighted models, while a mixture-model approach typically requires parametric assumptions.

Violations of these assumptions prevent meaningful inference, and there are many ways in which the assumptions may be violated. Based on our experiences, we have identified what we believe are the most likely sources of bias that would prevent a classifier from generalizing:

1. **Changes in experimental settings.** When experiments are performed under different experimental settings, the observed data may have different covariate distributions. For example, if training data (Z'_j, Y'_j, C'_j, K'_j) were collected at different times, and there is a relationship between time and cargo C' and/or cell K', we may observe systematic differences in the feature distributions $Z'|C', K'$ or the label distributions $Y'|C', K'$. These systematic differences may prevent the use of features which would otherwise satisfy covariate shift or label shift assumptions, unless features which are robust to experimental settings can be identified.

2. **Labeling bias.** Inference with unlabeled data is particularly useful when ground-truth labels are hard to collect. Due to labeling difficulties, training labels may suffer from errors and biases. Letting Y denote the ground-truth label and S the assigned training label, label bias is present when $P(S = 1|Z, C) \neq P(Y = 1|Z, C)$, which affects estimates of prevalence and probability-weighted inference. This can occur if, for example, $P(S = 1|Y, Z, C)$ is a function of Y, Z, or C.

It is important to note that even in a perfectly designed study with ground-truth labels, inference on new, unlabeled data can still fail: it may be impossible to generalize classifier predictions (e.g., no features satisfying (A1) exist), or the feature of interest fails to satisfy (A3) or parametric assumptions. However, addressing study design and labeling biases gives us the best chance of successful inference.

III.2.1 Examples: TIRF microscopy data

In the Chapter II case study, we examined a TIRF micrsoscopy dataset with three different experimental conditions. The data used in Chapter II is actually the mid-point of this study, collected after taking some steps to address the bias described above. Here we examine a prior TIRF microscopy dataset on the same cargos, and show why it was necessary to run further experiments.

Data. The original data for this study was collected with TIRF microscopy experiments for three different cargos: the β2-adrenergic receptor (B2), the μ-opioid receptor (MOR), and transferrin (TfR). As a large training set is desirable for classifier training, particularly as the prevalence of puffs is low, the original data combined experiments conducted at different times in the lab, and by different lab members. In total, 17 different cells were analyzed with TIRF microscopy (5 for B2, 6 for MOR, and 6 for TfR). The labels from the original experiments were used, and the number and prevalence of puffs in each of the 17 cells is shown in Table III.1. TfR cells tend to have higher numbers of puffs, while MOR cells tend to have fewer puffs, though there is high intra-cargo variability.

Features. Based on previous studies and expertise, our collaborators constructed a number of features to measure the behavior of each detected event, and to discriminate between puffs and nonpuffs. Many of these features are based on intensity profiles, the time series of intensities for each frame in an event. For example, $IntegratedDensity$ captures the area under the curve of the intensity profile, while $NPeaks$ measures the number of peaks in the time series. Other features include the $Lifetime$ of an event (the number of frames it exists for), and $DeltaF$ (the ratio of an event's maximum intensity to minimum intensity).

Many of these features show differences between puffs and nonpuffs, and could potentially be used for classification. However, as discussed in Chapter II, additional assumptions are required for a *generalizable*

Original data

Cargo	Cell	Number of puffs	Puff prevalence
B2	B2-1	17	0.021
B2	B2-2	18	0.007
B2	B2-3	31	0.031
B2	B2-4	217	0.050
B2	B2-5	40	0.043
MOR	MOR-1	60	0.030
MOR	MOR-2	17	0.017
MOR	MOR-3	61	0.030
MOR	MOR-4	45	0.045
MOR	MOR-5	17	0.016
MOR	MOR-6	18	0.017
TfR	TfR-1	15	0.021
TfR	TfR-2	152	0.079
TfR	TfR-3	18	0.018
TfR	TfR-4	29	0.016
TfR	TfR-5	12	0.010
TfR	TfR-6	120	0.055

Table III.1: Puffs by cell and cargo, original data.

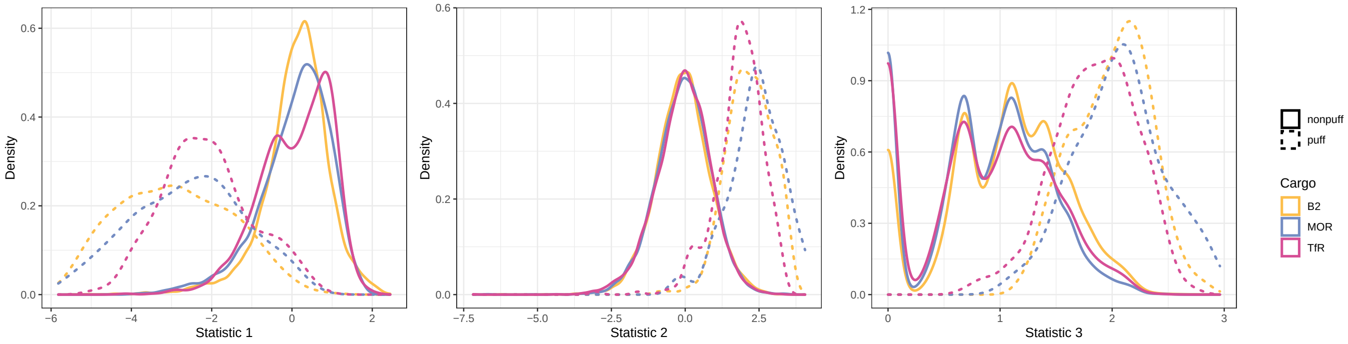

Figure III.1: Three covariates for label shift in the original data.

classifier which would allow inference on new data. While there are differences in puff prevalences between cargos, it is hoped that there will be characteristics common to all puffs; as in Chapter II, the label shift assumption for generalizing classifier predictions therefore may be appropriate.

Checking assumptions. Exploratory data analysis with the training set can be used to identify a set of covariates which appear to satisfy the label shift assumption. Using visualizations, we look for features which have the same class-conditional distributions across cargos. Figure III.1 shows exploratory plots for three statistics derived from the intensity time series, which appear to satisfy the label shift assumption in our original data.

However, as discussed in Chapter II, we are interested in using our classifier predictions for inference on unlabeled data, in a way that accounts for cell-to-cell variability. This requires that our classifier predictions be valid for each cell within a cargo, not just marginally for the cargo as a whole. In consequence, the label shift assumption should hold between cells in a cargo. Unfortunately, it appears that in our original data, there are some substantial differences in feature distributions between cells. Figure III.2 shows the

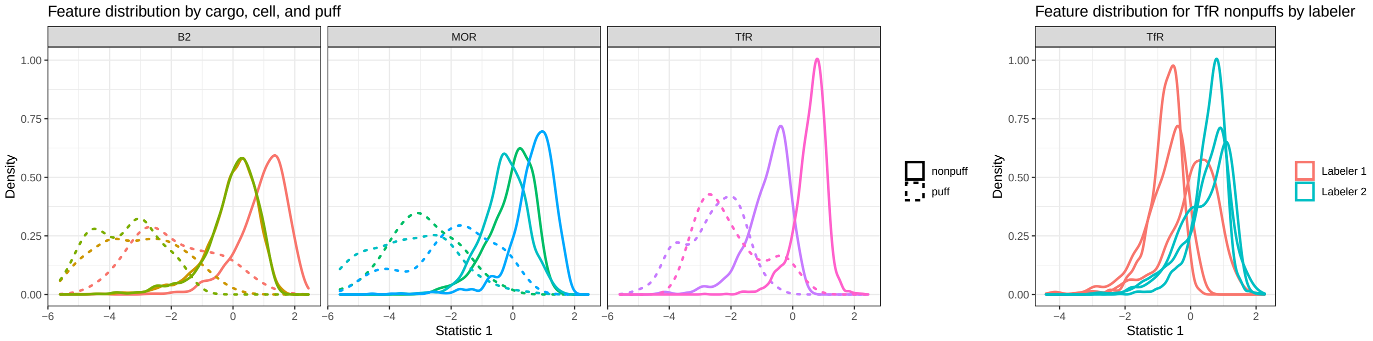

Figure III.2: Variation between cells in the original data.

distribution of one of the features from Figure III.1, but this time grouped by cell as well as cargo. Intra-cargo differences are particularly notable in TfR cells.

Changes in experimental settings. A potential explanation for this variability is that experiments were performed by different researchers at different times in the lab. As a result, there may be unintended shifts in experimental settings or changes to equipment which result in different feature distributions. For example, the TfR cells come from separate experiments by two lab members, which may explain the different distributions (Figure III.2). Of course, some amount of variation between runs of an experiment is expected, even if all settings are held constant. But without careful study design, it is unclear if differences between cells are simply due to this variability, or are systematic biases. Constructing a generalizable classifier requires either identifying features which are robust to experimental and labeler variability, or minimizing this variability with careful design. In the next section, we discuss steps to mitigate potential sources of error in data collection.

Remark: The data shown in Figure III.1 is not used for classification, due to the possibility of bias from changes to experimental settings. However, if the features shown in Figure III.1 were used for classification, we would also need to check that the label shift assumption is satisfied jointly, not just marginally. This can be done with dimension reduction (such as tSNE or PCA), or by training a classifier on part of the data and evaluating it on the holdout data, as in Chapter II.

Labeling bias. Labeling bias can arise from the strategy used to label detected events, and the potential for bias can be identified by carefully enumerating the labeling strategy. Consider two different labeling schemes:

1. **Scheme 1:** For each cell, a researcher scans the TIRF microscopy videos for puffs on the cell surface. These events are labeled puffs, while every event not spotted by the researcher is assumed to be a nonpuff.

2. **Scheme 2:** For each cell, each detected event is presented to the researcher in turn. The researcher labels each event as a puff or nonpuff, before moving on to the next event.

These two labeling schemes are subtly different. In Scheme 2, we trust that the labels are good proxies for the ground truth. In Scheme 1, however, an unknown number of puffs may be missed.

Formally, let $Y_i \in \{0, 1\}$ denote the ground-truth label for event i, let $S_{1,i}$ denote the label assigned by Scheme 1, and let $S_{2,i}$ denote the label assigned by Scheme 2. In the absence of assays to empirically determine Y_i, expert judgment is required to identify puffs, and so we will assume that $P(S_2 = Y) = 1$. However, Scheme 1 is not guaranteed to recover the same labels: unless the labeler can pick out all puffs in a video, $P(S_1 = 1|Y = 1) < 1$. Furthermore, the events for which $S_{1,i} = 1$ are unlikely to be a random sample of the events for which $Y_i = 1$, as puffs which look similar to nonpuffs are more likely to be missed. Therefore, we expect that $P(S_1 = 1|Y, X, Z, C)$ is a function of Y, X, and/or Z, and that $X, Z|Y = 1, C = c \overset{d}{\neq} X, Z|S_1 = 1, C = c$.

In practice, Scheme 2 is time-consuming to implement. As approximately 95% of detected events are nonpuffs, a sufficiently large training set requires carefully labeling thousands of events. In contrast, Scheme 1 requires only scanning for puffs, without making explicit judgments on each nonpuff. Scheme 1 was used when collecting the original TIRF microscopy data shown in Table III.1, which allows the possibility of labeling bias.

The extent of this bias cannot be assessed without gathering data under both schemes. In the next section, we examine new data collected to minimize changes in experimental settings, and labeled under both Schemes 1 and 2 for comparison.

III.3 Steps to mitigate bias

With our collaborators, we identified several key steps in study design to prevent and assess systematic biases in the labeled training data:

1. Perform all experiments close together in time, to minimize differences in laboratory equipment and experimental settings.
2. Perform experiments in a random order, to avoid systematic differences between cargos.
3. When labeling puffs, blind the labeler to cargo identity, to avoid preconceived biases about differences between cargos.

4. Label all events with Scheme 1. If feasible, label all events with Scheme 2 as well. If not, select a random subset of events to label with Scheme 2.

Our collaborators then collected a new training set, from new experiments with the same three cargos (B2, MOR, and TfR), using these steps. We analyze the new training set in the next sections.

III.3.1 New training data

50 cells across the three cargos were analyszed in TIRF microscopy experiments, of which 18 were randomly selected for labeling and classifier training. These 18 training cells (5 from B2, 6 from MOR, and 7 from TfR) comprise 134217 total events. All 134217 events were labeled with Scheme 1, and the number and prevalence of puffs in each cell – under Scheme 1 – are shown in Table II.3 in Chapter II (condition $C = 1$ corresponds to B2, $C = 2$ to MOR, and $C = 3$ to TfR).

A full analysis of the Scheme 1 labeling is found in the case study in Chapter II. In Section II.5 and Appendix A.6, we construct features for classification which satisfy the label shift assumptions under Scheme 1, and which permit inference for puff $Smoothness$ in new data.

III.3.2 Changes in experimental settings

Figure II.4 provides evidence for the label shift assumption with covariates used for classification in Section II.5, under Scheme 1. We also assess intra-cargo variability in covariate distributions, as was seen in Figure III.2. In the new data under Scheme 1, there appears to be less variation between cells within each cargo. Figure III.3 shows the distribution of two features used in classification under Scheme 1, $ConvexPerimeter$ and *Randomness*, broken down by cell, cargo, and Scheme 1 label. There is still some variation between cells for these features, but the distributions appear more similar than in Figure III.2. Furthermore, because the data was collected to minimize biases, we can be confident that differences between cells are due to natural variation rather than any systematic bias.

III.3.3 Labeling bias

Under the assumption that Scheme 2 provides ground-truth labels, we compare labels from Scheme 1 and Scheme 2 to assess labeling bias. If no bias is noted, then Scheme 1 labels can be used for classifier training. However, if bias is present then classifier predictions under Scheme 1 must be corrected.

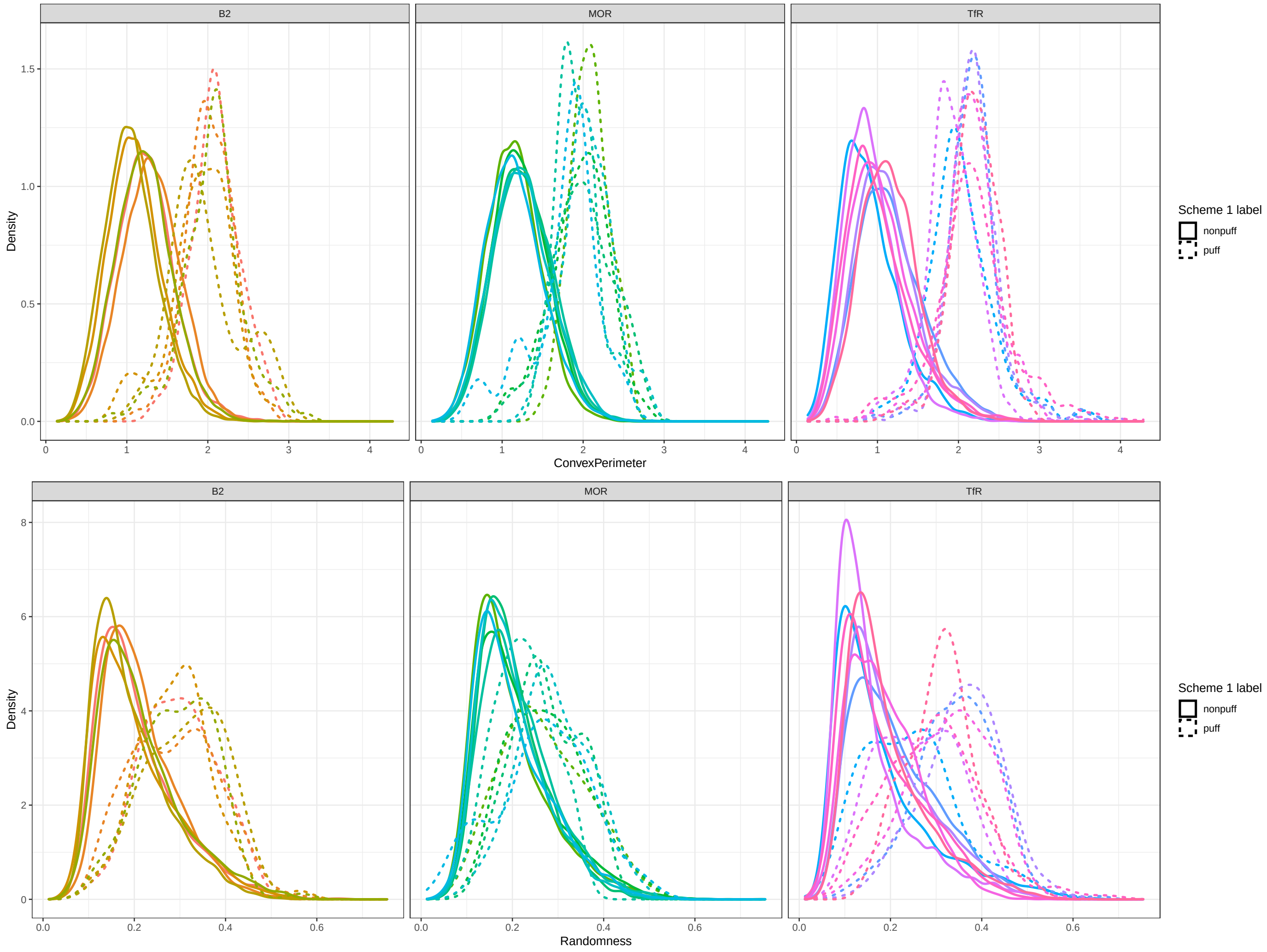

Figure III.3: Assessing inter-cell variability within each cargo, under Scheme 1 labels. Data were collected following steps to mitigate bias, so variability between cells is due to natural variation rather than systematic changes. Each density curve corresponds to a different cell.

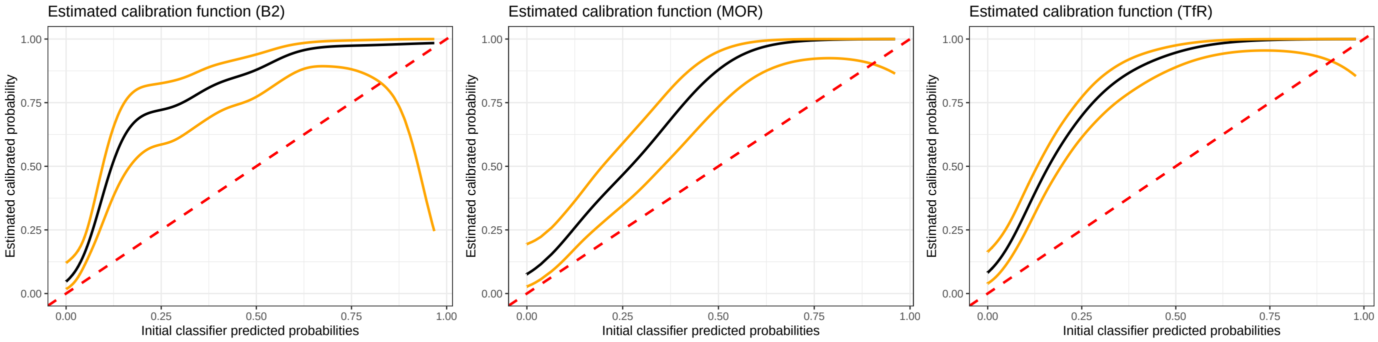

Figure III.4: Assessing the extend of label bias under Scheme 1. A stratified sample of events under Scheme 1 is re-labeled under Scheme 2, and a logistic spline is fit to assess the change in puff frequency.

Labeling under Scheme 2. Because Scheme 2 labels are labor-intensive, we aim to minimize the number of events labeled under Scheme 2. To efficiently estimate the prevalence of puffs, we first label a stratified sample of events. To stratify, we train a classifier on each cargo under Scheme 1 labels, and divide events into 10 evenly-spaced bins based on classifier predicted probability (0 to 0.1, 0.1 to 0.2, etc.). We then take a random sample from each bin, and label the random sample with Scheme 2. Finally, we fit a logistic spline to estimate the frequency of puffs in each bin. Figure III.4 shows the change from Scheme 1 puff probabilities to Scheme 2 puff probabilities, demonstrating that the prevalence of puffs is higher under Scheme 2 than under Scheme 1. Using the spline fit, we estimate that at least 5% of events in B2 and MOR cells are puffs, and at least 10% of events in TfR cells are puffs. For comparison, approximately 1-2% of B2 and MOR events, and 5% of TfR events, were initially labeled as puffs under Scheme 1.

The predicted probabilities used for stratification in Figure III.4 were not designed for generalizability, and the stratified sample is not representative of the population of events. We therefore take a larger, random sample to be labeled under Scheme 2. Using our estimates of the lower bound on puff prevalence (5%), we randomly sample 5000 events from each cargo, which we expect to provide approximately 250 puffs in B2 and MOR cells, and approximately 500 puffs in TfR cells. These 15000 randomly sampled events are labeled under Scheme 2 in a random order, with no information on cell, cargo or initial (Scheme 1) label provided.

Scheme 1 vs. Scheme 2. Table III.2 shows the breakdown of puffs in each cell in the Scheme 2 sample. As estimated from Figure III.4, B2 and MOR cells have puff prevalence around 5%, while TfR cells have puff prevalence around 10%. Table III.3 compares Scheme 1 scores to Scheme 2 scores on the random sample. Most events labeled as puffs under Scheme 1 are truly puffs, but a large number of puffs were missed under Scheme 1.

Unsurprisingly, it appears that the events which were mis-labeled as nonpuffs under Scheme 1 are puffs which look more like nonpuffs. Figure III.5 shows the distributions of two features, $Lifetime$ and $Smoothness$,

Training sample, Scheme 2 labels

Cargo	Cell	Number of puffs	Puff prevalence
B2	1-1	52	0.061
B2	1-2	104	0.074
B2	1-3	57	0.056
B2	1-4	31	0.031
B2	1-5	25	0.034
MOR	2-1	44	0.046
MOR	2-2	62	0.053
MOR	2-3	52	0.048
MOR	2-4	14	0.038

Training sample, Scheme 2 labels

Cargo	Cell	Number of puffs	Puff prevalence
MOR	2-5	38	0.045
MOR	2-6	14	0.025
TfR	3-1	64	0.116
TfR	3-2	113	0.124
TfR	3-3	137	0.141
TfR	3-4	85	0.111
TfR	3-5	54	0.085
TfR	3-6	128	0.163
TfR	3-7	38	0.102

Table III.2: Breakdown of events by cell and cargo under Scheme 2. The data in this table are a random sample of 5000 events from each cargo. For comparison, Table II.3 shows the breakdown of labels for the same cells under Scheme 1, in a superset of 134217 total events.

B2

Scheme: 1 \2	Nonpuffs	Puffs
Nonpuffs	4728	202
Puffs	3	67

MOR

Scheme: 1 \2	Nonpuffs	Puffs
Nonpuffs	4766	181
Puffs	10	43

TfR

Scheme: 1 \2	Nonpuffs	Puffs
Nonpuffs	4371	360
Puffs	10	259

Table III.3: Scheme 1 vs. Scheme 2 labels, compared on a random sample of 5000 events from each cargo. We assume that Scheme 2 labels are ground truth.

under Scheme 1 and Scheme 2. We can see that Scheme 2 puffs are more similar to nonpuffs. Features like lifetime might even explain the difference in labeling: short-lived puffs are hard to spot in Scheme 1, so are more likely to be missed.

Remark: $Lifetime$ is commonly used to compare different cargos. In Figure III.5, there appear to be differences in the distribution of $Lifetime$ between B2, MOR, and TfR under Scheme 1. However, it is interesting to note that the distributions are much more similar once data is rescored under Scheme 2.

The change in covariate distributions can also be seen in features used for classification. In Chapter II, we fit a logistic GAM on $DeltaF$, SNR, $ConvexArea$, $ConvexPerimeter$, and $Randomness$, using Scheme 1 labels as the response variable. These features appeared to satisfy the label shift assumption under Scheme 1, and were useful for distinguishing puffs and nonpuffs. Under Scheme 2, however, puffs and nonpuffs look more similar, and the label shift assumption no longer appears appropriate for all covariates, particularly $ConvexArea$ and $ConvexPerimeter$ (Figure III.6). A generalizability classifier thus should not use exactly the same covariates as in Scheme 1. (Note: while Figure III.5 and Figure III.6 show only univariate plots, the same conclusions hold for joint distributions of the covariates). In the next section, we consider options for inference with the Scheme 2 data.

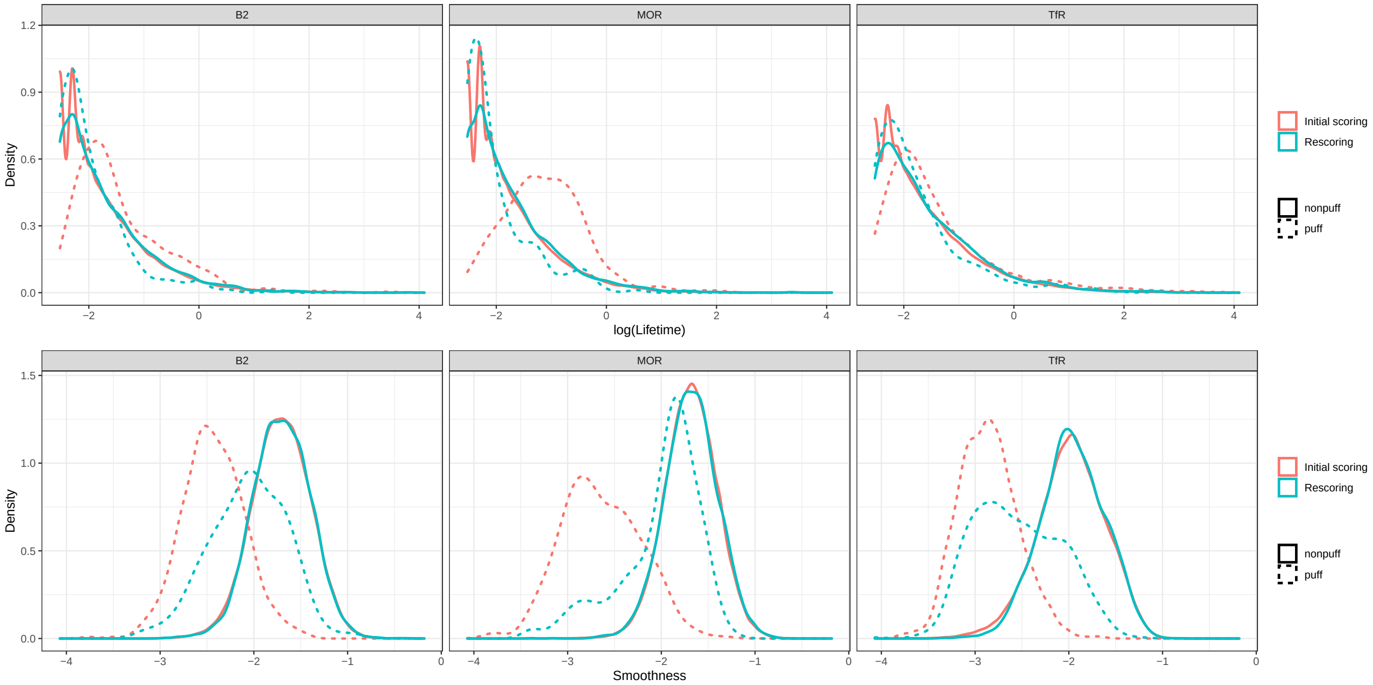

Figure III.5: Feature distributions under Scheme 1 vs. Scheme 2 labels, showing changes in characteristics of puffs.

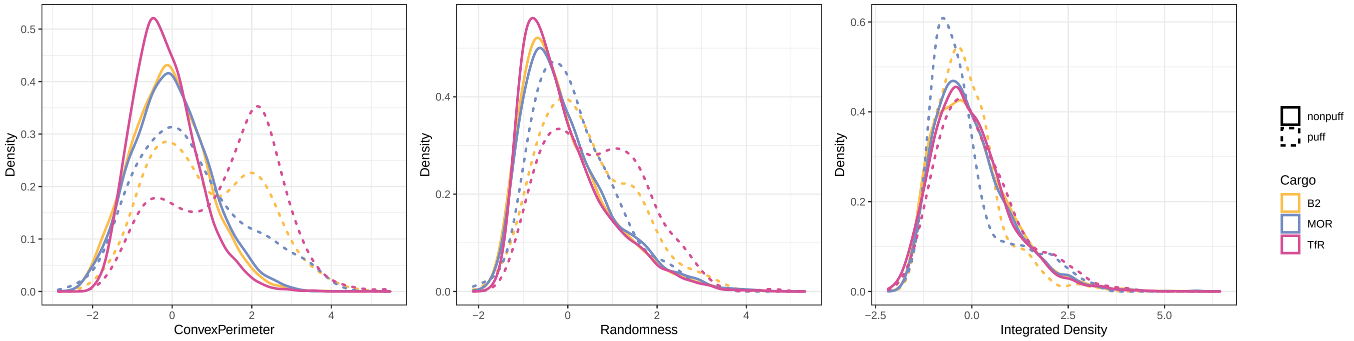

Figure III.6: Changes to classifier covariates under Scheme 2. Here we show the distribution of two of the covariates used for classification in the case study in Chapter II. The label shift assumption is no longer appropriate for $ConvexPerimeter$, though may still be reasonable for $Randomness$ (and $DeltaF$ and SNR, not shown). The label shift shift assumption may also be appropriate for $IntegratedDensity$. However, puffs are much more similar to nonpuffs under the Scheme 2 labels.

III.4 Inference after correcting labeling bias

In Chapter II, we provide a full discussion of inference with TIRF microscopy data under Scheme 1, where label shift methods can be used to as probability weights for inference on *Smoothness*. However, Scheme 1 labels are biased, and inference with Scheme 1 predictions is not appropriate for Scheme 2 (true labels). There are then two options for Scheme 2 labels: train a new classifier, or try to calibrate Scheme 1 classifier predictions.

III.4.1 Training a new classifier

One option for inference is to train a new classifier on the 15000 events labeled under Scheme 2. While this subset is smaller than the initial training set of 134217 events, it is a random sample from the population of detected events, and so the same techniques from Chapter II can be used. Provided that covariates satisfying generalizability assumptions can be identified (they need not be the same covariates as for Scheme 1 labels), classifier predictions may be generalizable.

Example: Based on exploratory data analysis (such as in Figure III.6), we identify features which appear to approximately satisfy the label shift assumption under Scheme 2 labels. These are $DeltaF$, SNR, $Randomness$, and $IntegratedDensity$. Denote these features by Z. We will train a classifier $\mathcal{A}_2$ to estimate $P(Y = 1|Z = z)$, with label shift corrected predictions $\mathcal{A}_{2,L}(z, c) = \widehat{P}(Y = 1|Z = z, C = c)$. To assess the performance of the classifier on heldout data, we train the classifier three different times, each time using two of the three cargos as training data and the remaining cargo as test data, in round-robin fashion. For each classifier, we use the label shift assumption to estimate the prevalence $P(Y = 1|C = c)$ in the heldout cargo.

The true sample prevalences for each cargo, under Scheme 2, are 0.054 for B2, 0.045 for MOR, and 0.124 for TfR. After round-robin classifier training and label shift estimation, the estimated prevalences are respectively 0.052, 0.041, and 0.132. Figure III.7 shows a calibration plot for the label shift corrected predictions $\mathcal{A}_{2,L}(z, c)$ for each cargo c, with any miscalibrated predictions suggesting deviations from the label shift assumption. Overall, predictions appear reasonably well-calibrated, though assessment is challenging for B2 and MOR due to few observations in several of the bins, and some predictions for TfR are underestimates.

For inference with a feature of interest X, the predictions $\mathcal{A}_{2,L}(Z, C)$ need to capture any information in X that distinguishes puffs and nonpuffs. That is, assumption (A3) implies that the predicted probabilities $\mathcal{A}_{2,L}(Z, C)$ are only useful as weights when $\mathbb{E}[X|Y = 1, \mathcal{A}_{2,L}(Z, C)] = \mathbb{E}[X|Y = 0, \mathcal{A}_{2,L}(Z, C)]$. We assess this

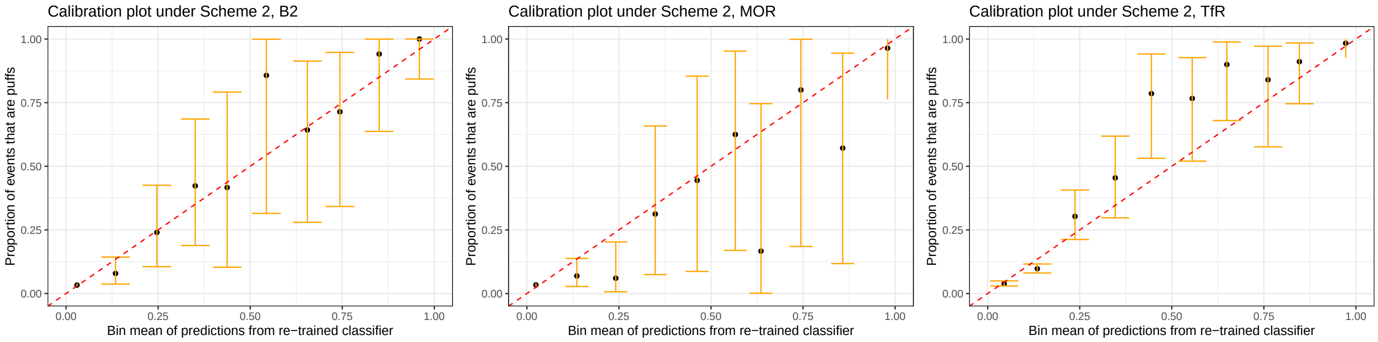

Figure III.7: Performance of retrained classifiers under Scheme 2 labels. For each of the three cargos in turn, a logistic GAM with covariates $DeltaF$, SNR, $Randomness$, and $IntegratedDensity$ is trained on the other two cargos to predict Scheme 2 label. Here the label shift corrected predictions ($\mathcal{A}_{2,L}(Z, C)$) are plotted on the horizontal axis, and the true frequency of puffs on the vertical axis.

assumption for $Smoothness$ (the feature of interest used in Chapter II) in Figure III.8, which suggests that $Smoothness$ does contain some additional information to distinguish puffs and nonpuffs, but the conditional means $\mathbb{E}[X|Y = 1, \mathcal{A}_{2,L}(Z, C)]$ and $\mathbb{E}[X|Y = 0, \mathcal{A}_{2,L}(Z, C)]$ are generally close. In contrast, assumption (A3) appears less appropriate if we are instead interested in inference on $Lifetime$: differences in the conditional means, particularly in areas of high density, will impact weighted estimates.

The resulting estimated puff means $\widehat{\mathbb{E}}[X|Y = 1, C = c, K = k]$ in each cell k, using classifier predictions $\mathcal{A}_{2,L}(Z, C)$ as weights, are compared to the true sample means in Figure III.9. Estimates for $Smoothness$ are close to the true sample means for B2 and MOR, but are biased for TfR. This matches Figure III.8, in which the conditional means $\mathbb{E}[Smoothness|Y = y, \mathcal{A}_{2,L}(Z, C)]$ are more different for TfR than for B2 and MOR. Likewise, as expected from Figure III.8, mean estimates for $Lifetime$ perform worse than for $Smoothness$.

III.4.2 Calibrating classifier predictions

Instead of training a new classifier $\mathcal{A}_2$ on the Scheme 2 labels, we could try to calibrate the Scheme 1 classifier $\mathcal{A}$ for use with Scheme 2 data. Now let Z denote features $DeltaF$, SNR, $Randomness$, $ConvexArea$, and $ConvexPerimeter$. In Chapter II, we trained a classifier $\mathcal{A}$ with raw predictions $\mathcal{A}(z) = \widehat{P}(S_1 = 1|Z = z)$ and label shift corrected predictions $\mathcal{A}_L(z, c) = \widehat{P}(S_1 = 1|Z = z, C = c)$. As shown in Table III.3, $P(S_1 = 1|Z = z, C = c) \neq P(Y = 1|Z = z, C = c)$. However, suppose we make the following assumption:

(A4) $P(S_1 = 1|Z = z, C = c) = g(P(Y = 1|Z = z, C = c))$, where g is a smooth, invertible function.

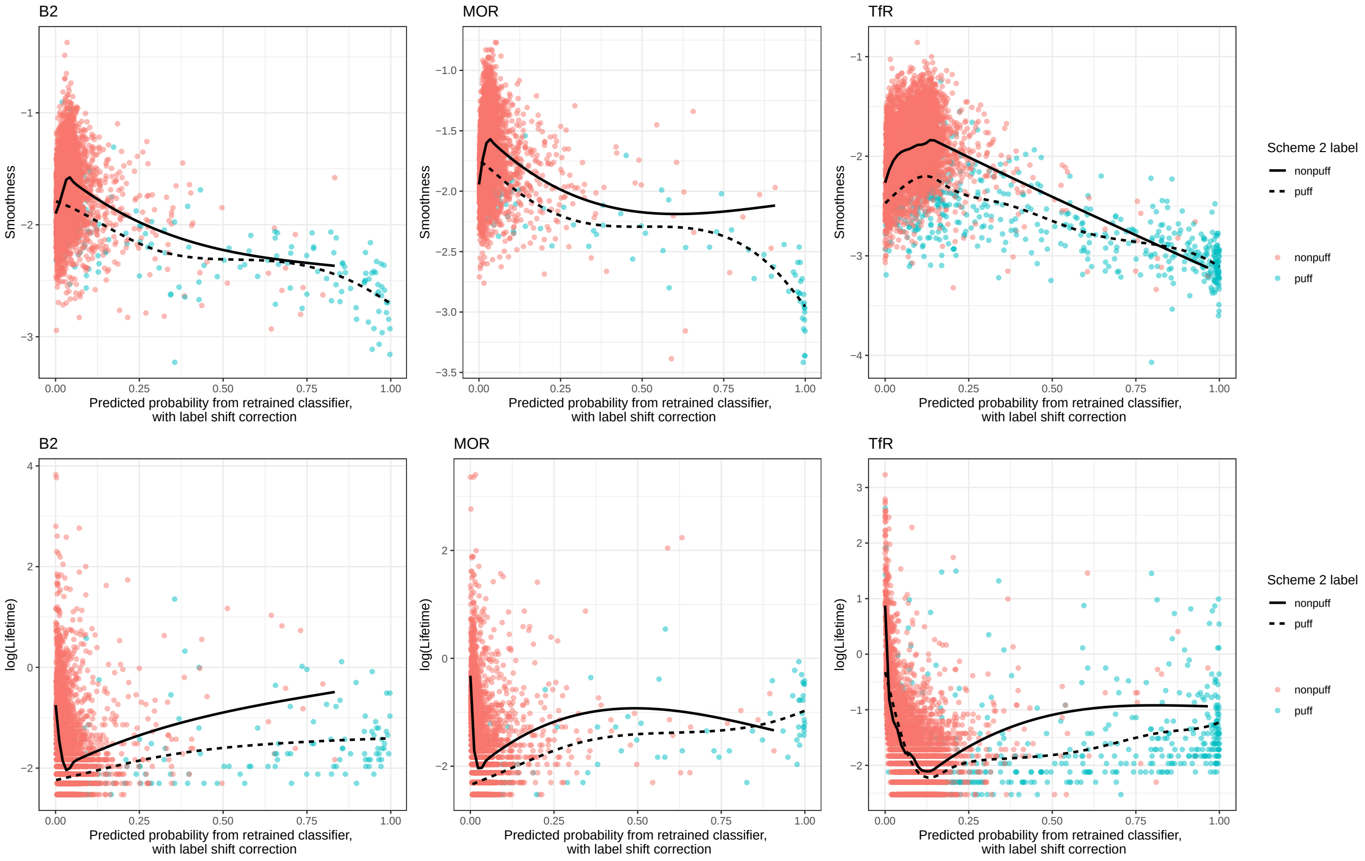

Figure III.8: Checking assumptions for feature inference with a retrained classifier, $\mathcal{A}_2$. If the classifier predictions can be used as weights for inference under Scheme 2 labels, we need $\mathbb{E}[X|Y=1,\mathcal{A}_{2,L}(Z,C)] = \mathbb{E}[X|Y=0,\mathcal{A}_{2,L}(Z,C)]$. The new label shift corrected predictions, $\mathcal{A}_{2,L}(Z,C)$, are plotted on the horizontal axis, and the feature of interest ($X = Smoothness$ or $log(Lifetime)$) on the vertical axis. A smoothing spline fit for each group (representing $\mathbb{E}[X|Y=y,\mathcal{A}_{2,L}(Z,C)]$) is included for comparison between puffs and nonpuffs.

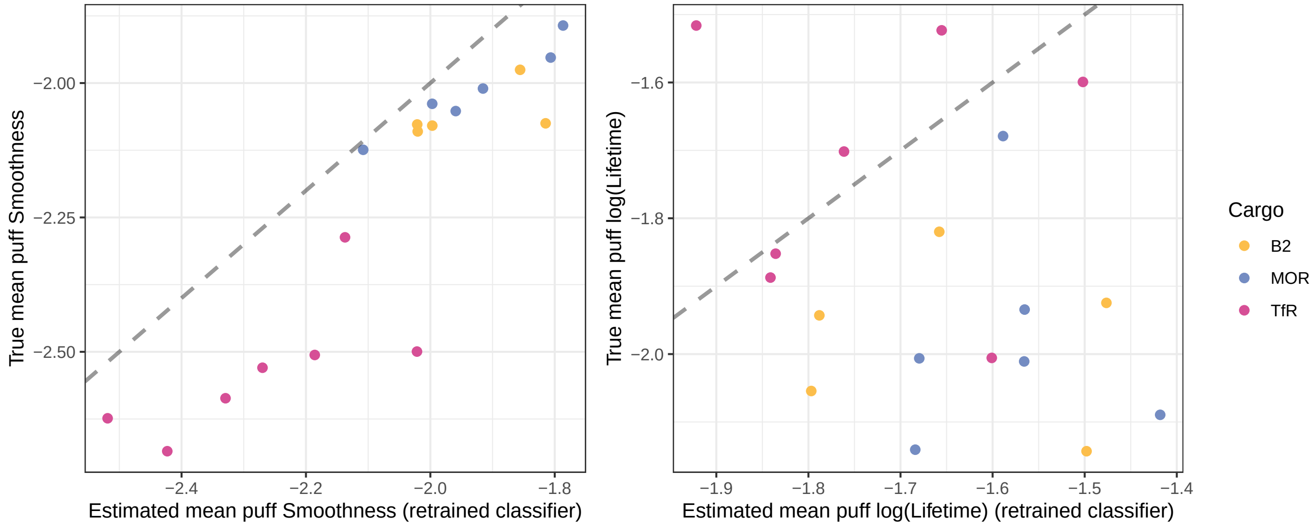

Figure III.9: Estimated feature means for puffs in each cell, under Scheme 2. The true sample mean in each cell (using true labels) is plotted on the vertical axis, while the weighted mean (using label shift corrected predictions $\mathcal{A}_{2,L}(Z, C)$ with a new classifier trained on the Scheme 2 data) is plotted on the horizontal axis.

Then, if (A1) and (A2) hold for initial labels S_1 (rather than the true labels Y), and (A4) holds too, we can estimate $P(Y = 1|Z = z, C = c)$ as follows:

1. On training data $(Z'_i, S'_{1,i}, Y'_i, C'_i)$, train a classifier $\mathcal{A}$ with $\mathcal{A}(z') = \widehat{P}(S'_1 = 1|Z' = z')$.
2. For each training cargo $c' \in \mathcal{C}'$, use the observed prevalence $P(S'_1 = 1|C' = c')$ and Bayes theorem to estimate $\mathcal{A}_L(z', c') = \widehat{P}(S'_1 = 1|Z' = z', C' = c')$, as in Equation (II.2).
3. Use a logistic spline to regress Y' on $\mathcal{A}_L(Z', C')$. Let $\widehat{h}$ be the spline fit, so under (A4), $\widehat{P}(Y' = 1|Z' = z', C' = c') = \widehat{h}(\mathcal{A}_L(z', c'))$.
4. On unlabeled test data (Z_i, C_i), apply classifier $\mathcal{A}$ to get predictions $\mathcal{A}(Z_i)$.
5. Use the label shift assumption to estimate prevalence $\widehat{P}(S_1 = 1|C = c)$ for each test cargo c, and calculate label shift corrected predictions $\mathcal{A}_L(Z_i, C_i) = \widehat{P}(S_{1,i} = 1|Z_i, C_i)$ as in Equation (II.2).
6. Finally, $\widehat{P}(Y = 1|Z = z, C = c) = \widehat{h}(\mathcal{A}_L(z, c))$, and prevalence can be estimated by averaging these predicted probabilities.

Assumption (A4) holds if mislabeling $(S_1 \neq Y)$ depends on the features Z and cargo C only through the fraction of events which are puffs. This is a strict assumption, but potentially plausible: it is easier to spot puffs when there are more of them.

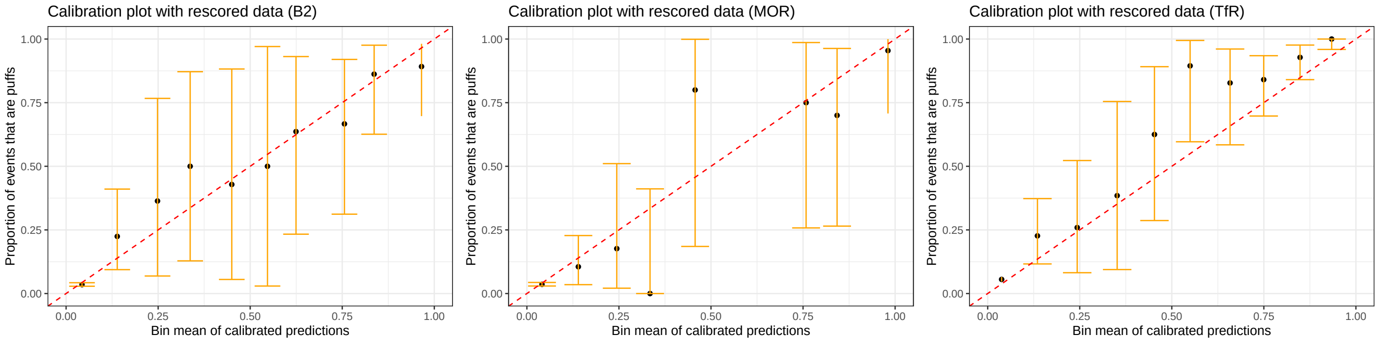

Figure III.10: Performance of calibrated classifiers under Scheme 2 labels. For each of the three cargos in turn, a logistic GAM with covariates $DeltaF$, SNR, $ConvexArea$, $ConvexPerimeter$, and $Randomness$ is trained on the other two cargos to predict Scheme 1 label. A logistic spline is then used to calibrate label shift corrected predictions, to estimate Scheme 2 labels. Finally, calibrated predictions are applied to the heldout cargo. Here the final predictions ($\widehat{h}(\mathcal{A}_L(Z, C))$) are plotted on the horizontal axis, and the true frequency of puffs on the vertical axis.

For inference on class-conditional feature means $\mathbb{E}[X|Y = y, C = c]$, (A4) must be expanded to incorporate the feature of interest X:

(A5) $P(S_1 = 1|Z = z, C = c, X = x) = g(P(Y = 1|Z = z, C = c, X = x))$, where g is a smooth, invertible function.

Then, if (A1), (A2), and (A3) hold for the initial labels S_1, and (A5) holds as well, $\widehat{h}(\mathcal{A}_L(z, c)) = \widehat{P}(Y = 1|Z = z, C = c, X = x)$, and these predicted probabilities can be used as weights in inference.

Example: As before, we assess performance of the classifier calibration approach in round-robin fashion. Each time, the classifier and calibration function are fit on two of the cargos, and applied to the heldout third cargo.

First, the true sample prevalences for each cargo, under Scheme 2, are 0.054 for B2, 0.045 for MOR, and 0.124 for TfR. After round-robin classifier training and calibration, the estimated prevalenced are respectively 0.061, 0.051, and 0.099. Figure III.10 shows the performance of the final predictions $\widehat{h}(\mathcal{A}_L(z, c))$. As with the retrained classifier predictions $\mathcal{A}_{2,L}(z, c)$ in Figure III.7, the predictions $\widehat{h}(\mathcal{A}_L(z, c))$ perform reasonably well, but some predictions for TfR are underestimates, and we don't have enough data in some B2 and MOR bins.

For inference with a feature of interest X, the final predictions $\widehat{h}(\mathcal{A}_L(Z, C))$ also need to capture any information in X that distinguishes puffs and nonpuffs. That is, assumption (A3) and (A5) together imply that the predictions $\widehat{h}(\mathcal{A}_L(Z, C))$ are only useful as weights when $\mathbb{E}[X|Y = 1, \widehat{h}(\mathcal{A}_L(Z, C))] = \mathbb{E}[X|Y = 0, \widehat{h}(\mathcal{A}_L(Z, C))]$. As in Figure III.8, we assess this assumption for $Smoothness$ and $Lifetime$ in Figure

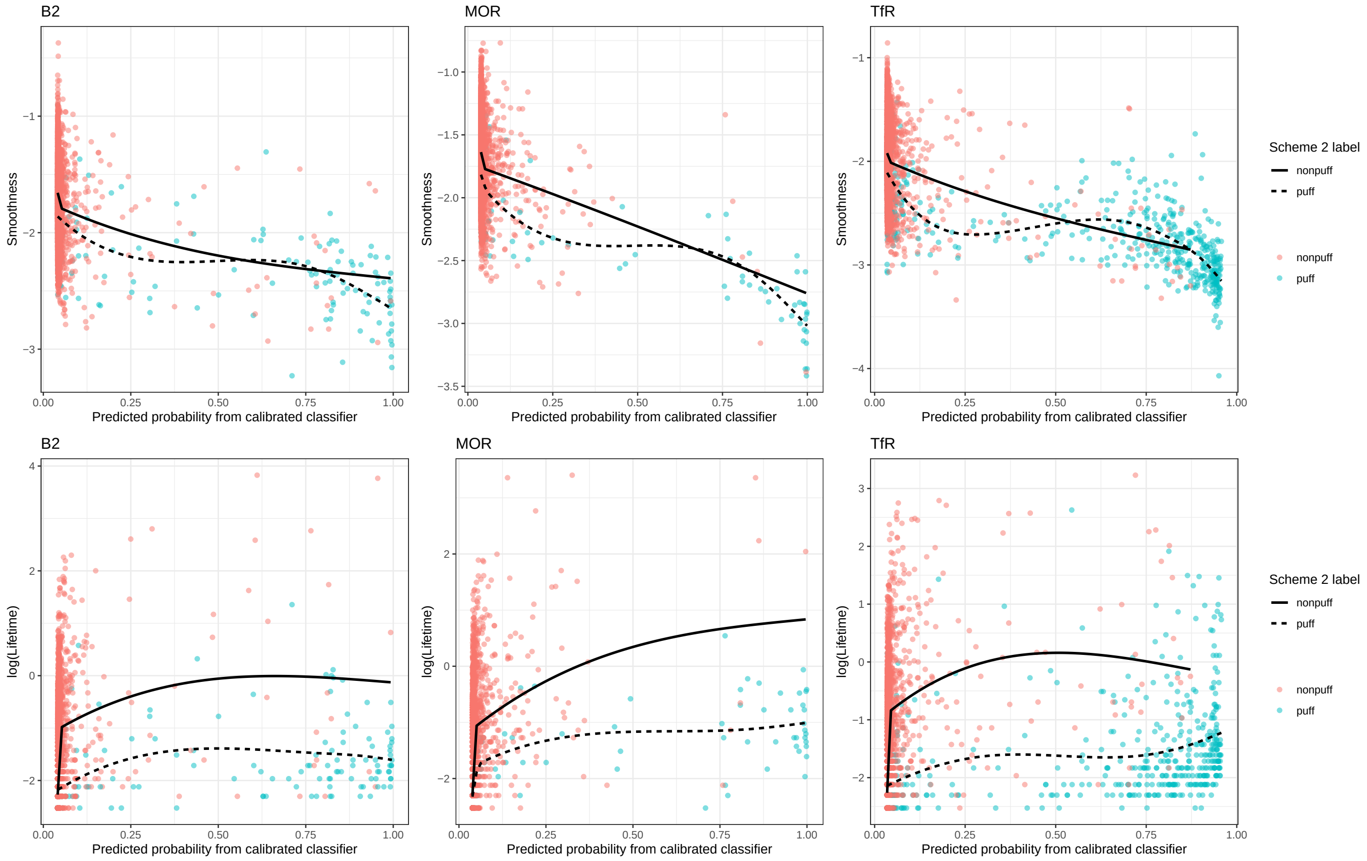

Figure III.11: Checking assumptions for feature inference with calibrated classifiers. If the final predicted probabilities can be used as weights for inference under Scheme 2 labels, we need $\mathbb{E}[X|Y = 1, \widehat{h}(\mathcal{A}_L(Z, C))] = \mathbb{E}[X|Y = 0, \widehat{h}(\mathcal{A}_L(Z, C))]$. The final predictions, $\widehat{h}(\mathcal{A}_L(Z, C))$, are plotted on the horizontal axis, and the feature of interest ($X = Smoothness$ or $log(Lifetime)$) on the vertical axis. A smoothing spline fit for each group is included for comparison between puffs and nonpuffs.

III.11. It appears that *Smoothness* does contain some additional information to distinguish puffs and nonpuffs. In general, however, the puff conditional mean $\mathbb{E}[X|Y = 1, \widehat{h}(\mathcal{A}_L(Z, C))]$ is close to the nonpuff conditional mean $\mathbb{E}[X|Y = 0, \widehat{h}(\mathcal{A}_L(Z, C))]$. It is most important that these two means are close in areas of high density, which appears to be true. As a result, the calibrated classifier predictions $\widehat{h}(\mathcal{A}_L(Z, C))$ may be useful for inference with *Smoothness*. Figure III.12 shows the true mean puff *Smoothness* in each cell, under Scheme 2 labeling, compared to the estimated weighted mean with the calibrated predictions $\widehat{h}(\mathcal{A}_L(Z, C))$. Estimates are generally close to the true sample means, with a slight bias. In contrast, Figure III.11 and Figure III.12 demonstrate that the predictions $\widehat{h}(\mathcal{A}_L(Z, C))$ are not sufficient for inference with *Lifetime*.

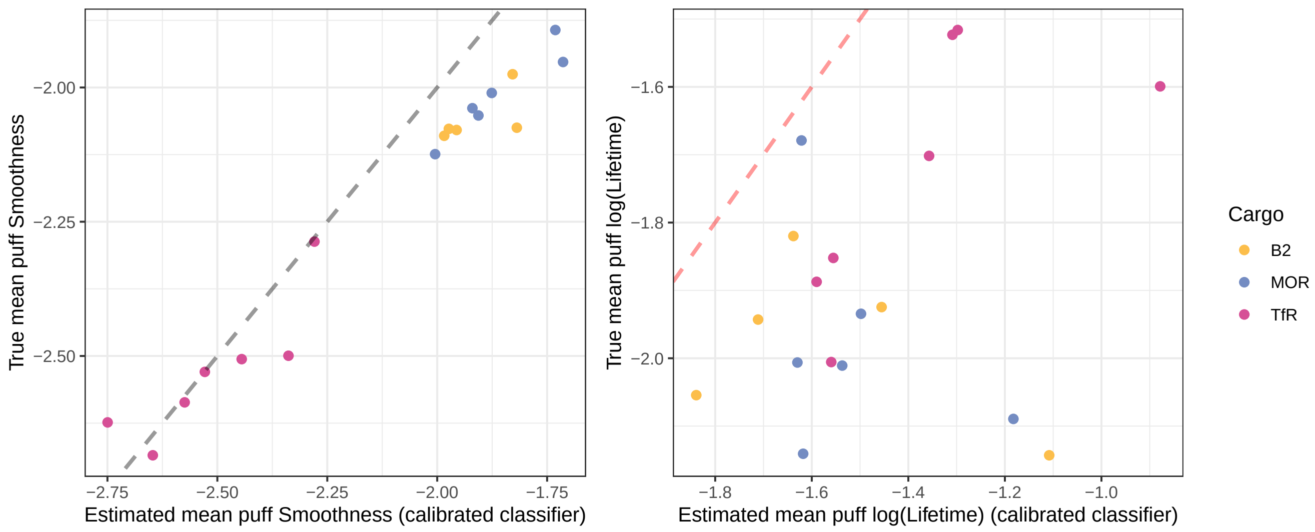

Figure III.12: Estimated feature means for puffs in each cell, under Scheme 2. The true sample mean in each cell (using true labels) is plotted on the vertical axis, while the weighted mean (using calibrated classifier predictions $\widehat{h}(\mathcal{A}_L(Z, C))$) is plotted on the horizontal axis.

III.5 Discussion

In Chapter II, we show how inference for prevalence $P(Y = 1|C = c)$ and conditional feature means $\mathbb{E}[X|Y = y, C = c]$ can be performed using classifier predictions on unlabeled data. However, generalizability assumptions are key for valid inference. In this chapter, we describe two potential sources of systematic bias that would prevent statistical inference– changes in experimental settings, and labeling bias – and propose concrete steps to assess and mitigate this bias.

While changes in experimental settings can be mitigated by steps such as performing experiments close together in time, and in a random order, we recognize that these steps may not always be feasible. For example, new laboratory equipment may be purchased, research projects are conducted over a period of years, and collaborations often involve experiments performed by different researchers. In these cases, valid inference requires 1) covariates which are robust to changes in experimental settings (or can be appropriately normalized, but normalization is often a challenging topic in itself), and 2) that differences in experimental settings do not cause spurious differences in the targets of inference (such as prevalence and conditional feature means).

Labeling bias is a concern in the exact settings where automated labeling is useful: labels are difficult and time-consuming to collect. If existing labels are biased, new labels need to be assigned to assess and correct the problem. If new labels cannot be gathered, valid predictions may still be possible under further assumptions on the labels and covariates. For example, the special case where $P(Y = 1|S_1 = 1) = 1$ is knownin the literature as the *positive-unlabeled* (PU) setting, and a variety of approaches exist for classifier training with PU data (see Bekker and Davis (2020) for a survey). These approaches typically require assumptions about the selection procedure for the positive cases $S = 1$, or the conditional distributions $Z|Y = 1$ and $Z|Y = 0$ (such as separability). However, without a sample of ground truth labels Y, it is difficult to assess the validity of these additional assumptions. We focus instead on the setting where a random sample of true labels can be provided.

Of the two options presented here for classification with new labels – retrain a classifier or calibrate classifier predictions – the former seems the natural choice at first glance. Indeed, calibrating a classifier that was trained on biased labels, to work with the true labels, requires more assumptions ((A4) or (A5)) than just retraining. However, there are reasons why calibrating a biased classifier may be preferred, if the additional assumptions can be satisfied:

- Recalibrating is regression with a single predictor, whereas retraining typically requires multiple predictors. Recalibration can therefore be performed with a smaller number of ground truth labels than required by retraining. Indeed, if we truly believe the recalibration assumptions, the recalibration sample need not even be random – stratified samples, such as in Figure III.4, could be used instead. This is beneficial when collecting ground truth labels is laborious, and when the class prevalences are imbalanced.
- If biased labels are easier to collect, or have been used extensively in previous studies, then existing research may have spent considerable time developing useful features for distinguishing classes under these biased labels. In consequence, these features may satisfy generalizability assumptions under biased labeling, but not under ground truth labels. If so, recalibration could allow classification with existing features, rather than requiring new features to be engineered for generalizability.

However, we note that recalibration does add an additional step to bootstrap methods for inference. While we have focused on visualizations and point estimates in this chapter, bootstrap estimates of variability would require an additional resampling step for the recalibration function.

Ultimately, the choice of approach depends on carefully assessing assumptions on the available labeled data. It is therefore crucial that the training data be as rich and diverse as possible. Even then, inference may not be possible, if generalizability assumptions fail. Careful study design and labeling provides the best chance for valid inference, but as we saw with $Lifetime$, not all inferences are possible.

In situations where good labels are expensive to collect, active learning approaches may allow for more efficient data collection. However, it is unclear under what conditions active learning is compatible with generalizability assumptions, and we plan to study this question in future work. We also plan to investigate how to best combine labels from multiple sources while preserving generalizability.

Chapter IV

Sequential changepoint detection for label shift in classification

IV.1 Introduction

We consider the problem of rapid, online detection of a change in the distribution of classification data, without access to the true classification labels of those data points. This problem is of importance both because such a change impacts the performance of a classification algorithm, and because it often reflects an interesting change in the generating process itself. The general problem of classification under a changed generating distribution has been extensively studied, as has the general problem of sequential changepoint detection; see Section IV.2.4. However, the intersection of these topics is relatively unexplored, and yields interesting structure and methodological improvements. In this chapter, we restrict ourselves to the particular case of a *label shift* change in the distribution (Lipton et al., 2018), where the distribution of classification labels changes without changing the conditional distribution of the covariates; we illustrate this setting in the following motivating example.

Motivation – Dengue outbreaks Dengue, a viral infection transmitted by mosquitoes, is found in tropical and sub-tropical regions around the world, and affects up to 400 million people a year (WHO, 2020). Early treatment of dengue is important for improving prognosis, and so it is key to correctly diagnose patients with the disease. However, dengue cases are commonly mis-diagnosed (WHO, 2020); while gold-standard diagnostic tests and rapid antigen tests exist, these may not always be available to healthcare providers. To assist healthcare workers in diagnosis and early detection of dengue, Tuan et al.

> (2015) developed a classifier based on simple diagnostic and laboratory measurements, such as temperature, vomiting, and white blood cell count. The authors recommend deploying the classifier to help diagnose dengue in patients, which entails sequentially applying the classifier to make a prediction for each new patient.
>
> However, the prevalence of dengue in a community may change quickly, due to both seasonal trends and outbreaks (Wiwanitkit, 2006; Garg et al., 2011; Hsu et al., 2017). When a sudden change in dengue prevalence occurs, it is vital to raise an alarm, for two reasons: because a change in community prevalence shifts the posterior probabilities underlying the classifier and requires we update our classifier predictions, and also as a matter of public health. As noted by Hsu et al. (2017), "strategies are needed to respond quickly to unexpected incidents."

Consider the above case when there is a dengue outbreak: the proportion of patients with dengue will increase, but we might expect that the symptoms used by Tuan et al. (2015) will remain the same. Denote by $X \in \mathbb{R}^d$ the set of covariates for classification (such as temperature, vomiting, white blood cell count, etc.), and by $Y \in \{0, 1\}$ a patient's true disease status. An outbreak implies that $P(Y = 1)$ changes, but the distributions of $X|Y = 0$ and $X|Y = 1$ do not. This example – introduced by Lipton et al. (2018) – constitutes a *label shift* in the distribution.

Contributions. In this chapter, we focus on nonparametric methods for detecting label shift in sequential classification data. The major contributions are as follows:

1. We construct a nonparametric procedure for detecting label-shift changepoints in the data distribution, and prove that it is asymptotically optimal under assumptions on the performance of the underlying classifier. We also demonstrate in simulations that the procedure performs well under mild violations of both the classifier assumptions and the label shift assumpton.

2. More generally, we provide new theoretical results about the performance of any nonparametric changepoint detection procedures that are based on likelihood ratio estimates. Our results guarantee asymptotic optimality for changepoint detection when the likelihood ratio estimate converges in total variation distance to the true likelihood ratio. These results are applicable beyond the label shift setting considered in our chapter.

3. We demonstrate significantly improved performance of the proposed procedures over the current state-of-the-art in both simulation and real data. For the latter, we apply our procedure to detect changes in dengue prevalence using real data from Tuan et al. (2015).

In Section IV.2, we formally develop the problem and provide relevant background on sequential detection and label shift. In Section IV.2.3, we describe a simple nonparametric detection statistic based on underlying classifier scores. Intuitively, performance of the proposed procedure will depend on performance of the classifier. In Section IV.3, we make this relationship clear by developing new theoretical results on the performance of a broader class of nonparametric detection procedures. We demonstrate the efficiency of the proposed procedure in both simulation and in an application to real dengue data in Sections IV.4 and IV.5, in comparison to other nonparametric detection procedures.

IV.2 Problem and method

IV.2.1 Problem statement and notation

We consider a sequential classification setting with unobserved labels, where feature vectors $X_1, X_2, X_3, ... \in \mathbb{R}^d$ arrive sequentially, but the associated labels $Y_i \in \{0, 1\}$ are *unobserved*. In our dengue example, X_i represents diagnostic measurements like temperature and white blood cell count, while Y_i represents true dengue status. We assume that a classifier, $\mathcal{A}(\cdot)$, has been trained on a separate set of training observations $(X'_1, Y'_1), ..., (X'_m, Y'_m)$ and is used to predict the unobserved labels Y_i.

At some time $\nu \geq 0$ in this sequence, called the *changepoint*, the distribution of (X_i, Y_i) changes. We notate the pre-change distribution as $\mathbb{P}_\infty$ and the post-change distribution as $\mathbb{P}_0$, such that (X'_1, Y'_1),...,(X'_m, Y'_m), (X_1, Y_1),...,$(X_\nu, Y_\nu) \overset{iid}{\sim} \mathbb{P}_\infty$ and $(X_{\nu+1}, Y_{\nu+1}), (X_{\nu+2}, Y_{\nu+2}), ... \overset{iid}{\sim} \mathbb{P}_0$. Our aim is to detect the change in the distribution of (X_i, Y_i) as quickly as possible, using the observed sequence X_i.

Remark: Throughout the chapter we use the subscripts ∞ and 0 for pre- and post-change quantities respectively, to be consistent with the sequential changepoint detection literature. The motivation is that $\nu = \infty$ indicates the change never occurs, so data is from the pre-change distribution, while $\nu = 0$ indicates the change occurs before we observe any data, so data is from the post-change distribution. With some abuse of notation, when context is clear we will let $\mathbb{P}_\infty$ and $\mathbb{P}_0$ denote general pre- and post-change distributions, so for example $(X_i, Y_i) \sim \mathbb{P}_\infty$ and $X_i \sim \mathbb{P}_\infty$ both indicate data drawn before a change occurs.

The general problem of classification under a changed distribution has been studied extensively in the literature. Because arbitrary changes to high-dimensional classification data may be impossible to correct or detect, it is standard to make additional assumptions on the nature of the change. We will focus on the **label shift** setting (Saerens et al., 2002; Storkey, 2009), which has received recent attention in the machine learning literature (Ackerman et al., 2020; Azizzadenesheli et al., 2019; Lipton et al., 2018; Rabanser et al., 2019). Label shift assumes that the marginal distribution of Y_i changes, but the conditional distribution of $X_i|Y_i$ does not:

Definition (label shift). Let $f_{\infty,X,Y}$, $f_{\infty,Y}$, and $f_{\infty,X|Y=y}$ denote the densities/mass functions of (X_i, Y_i), Y_i, and $X_i|Y_i = y$ respectively, under $\mathbb{P}_\infty$. Similarly define $f_{0,X,Y}$, $f_{0,Y}$, and $f_{0,X|Y=y}$. The label shift assumption is that $f_{0,X|Y=y} \equiv f_{\infty,X|Y=y}$ for all y, so

$$f_{0,X,Y}(x,y) = f_{0,Y}(y) f_{0,X|Y=y}(x) = f_{0,Y}(y) f_{\infty,X|Y=y}(x) \qquad \forall x, y. \tag{IV.1}$$

Label shift is simply a change in the mixing proportion for the class distributions $X|Y = 0$ and $X|Y = 1$. Since the conditional distribution of $X|Y = y$ remains the same, the conditional distribution of the classifier predictions, $\mathcal{A}(X)|Y = y$, does as well. This has inspired previous research to perform changepoint detection directly on the univariate sequence of classifier predictions $\mathcal{A}(X_1), \mathcal{A}(X_2), ...$ (Ackerman et al., 2020). As we show in the following sections, however, the classifier predictions $\mathcal{A}(X_i)$ can be further leveraged to improve changepoint detection in the label shift setting.

IV.2.2 Motivation

A common approach to sequential changepoint detection assumes that the changepoint ν is the only unknown, and constructs a detection statistic using the likelihood ratio between the pre- and post-change distributions. Let $f_{\infty,X}$ and $f_{0,X}$ denote the densities or mass functions under $\mathbb{P}_\infty$ and $\mathbb{P}_0$ respectively, and $\lambda(x) = f_{0,X}(x)/f_{\infty,X}(x)$. Many popular changepoint detection procedures are defined by a recursive detection statistic $R_t^x = \Psi(R_{t-1}^x)\lambda(X_t)$ (Polunchenko and Tartakovsky, 2012), with an initial value $R_0^x = x \in [0, A]$, and a stopping time $T^x(A) = \inf\{t \geq 1 : R_t^x \geq A\}$, where A is a pre-specified threshold (Figure IV.1(a)). For example, the classical CUSUM procedure has $\Psi(r) = \max\{1, r\}$ and $x = 1$, while the Shiryaev-Roberts procedure has $\Psi(r) = 1 + r$ and $x = 0$. Other variations on the Shiryaev-Roberts procedure choose a different starting point x, such as x sampled randomly from the quasi-stationary distribution (Tartakovsky et al., 2009).

In general, the pre- and post-change densities $f_{\infty,X}$ and $f_{0,X}$ are unknown, and so a variety of nonparametric procedures which estimate the likelihood ratio λ have been proposed, which we summarize in Section IV.2.4. However, existing likelihood ratio estimates do not leverage information from the label shift assumption, so we propose using the predictions $\mathcal{A}(X_i)$ to directly estimate the likelihood ratio λ. Notice that under the label shift assumption, we can rewrite the likelihood ratio as

$$\frac{f_{0,X}(X_i)}{f_{\infty,X}(X_i)} = \left(\frac{\pi_0}{\pi_\infty} - \frac{1-\pi_0}{1-\pi_\infty}\right) \mathbb{P}_\infty(Y_i = 1|X_i) + \frac{1-\pi_0}{1-\pi_\infty}. \tag{IV.2}$$

Similarly, if the true labels Y_i were observed, the likelihood ratio would be

$$\frac{f_{0,X,Y}(X_i, Y_i)}{f_{\infty,X,Y}(X_i, Y_i)} = \left(\frac{\pi_0}{\pi_\infty} - \frac{1-\pi_0}{1-\pi_\infty}\right) Y_i + \frac{1-\pi_0}{1-\pi_\infty}. \tag{IV.3}$$

If our classifier predictions $\mathcal{A}(X_i)$ are predicted probabilities in $[0, 1]$ and are a good approximation to $\mathbb{P}_\infty(Y_i = 1|X_i)$, or are binary predictions in $\{0, 1\}$ and are close to Y_i, then the likelihood ratio can be estimated directly by a linear function of the classifier scores $\mathcal{A}(X_i)$.

Furthermore, the performance of sequential changepoint detection procedures typically depends on the KL divergence, $KL(f_{0,X}, f_{\infty,X})$, between the pre- and post-change distributions (Lorden, 1971), with a greater KL divergence corresponding to improved detection ability. By the label shift assumption (IV.1) and Pinsker's inequality, we have

$$KL(f_{0,X}, f_{\infty,X}) \geq 2(\pi_0 - \pi_\infty)^2 \; TV^2\left(f_{\infty,X|Y=1}, f_{\infty,X|Y=0}\right), \tag{IV.4}$$

where $TV\left(f_{\infty,X|Y=1}, f_{\infty,X|Y=0}\right)$ denotes the total variation distance between the class distributions $X|Y = 1$ and $X|Y = 0$. Since classifiers are essentially constructed to maximize the difference between these two distributions, using classifier scores $\mathcal{A}(X_i)$ for detection should naturally lead to good changepoint detection performance.

IV.2.3 Proposed method

Let $(X_1', Y_1'), ..., (X_m', Y_m') \overset{iid}{\sim} \mathbb{P}_\infty$ denote our labeled training set, used to train the classifier $\mathcal{A}(\cdot)$, and assume for now that the pre- and post-change proportions π_∞ and π_0 are known (below, we consider the case of unknown parameters). Classifiers typically predict either the probability of a positive case $\mathbb{P}(Y_i = 1|X_i)$ or the label Y_i, and so we assume $\mathcal{A}(X_i) \in [0, 1]$ is a predicted probability, or $\mathcal{A}(X_i) \in \{0, 1\}$ is a predicted label. Define the estimated likelihood ratio $\widehat{\lambda}_{\mathcal{A},m}$ by

$$\widehat{\lambda}_{\mathcal{A},m}(x) = \left(\frac{\pi_0}{\pi_\infty} - \frac{1-\pi_0}{1-\pi_\infty}\right) \mathcal{A}(x) + \frac{1-\pi_0}{1-\pi_\infty}, \tag{IV.5}$$

where the subscripts $\mathcal{A}$ and m denote dependence on the classifier and the size of the training set. We sequentially observe unlabeled data $X_1, X_2, ...$ and wish to raise an alarm soon after the time ν at which the change from π_∞ to π_0 occurs. The nonparametric detection procedure is defined by a detection statistic $\widetilde{R}_t^x$ and a stopping time $\widetilde{T}^x(A)$ with threshold $A > 0$, where $\widetilde{R}_0^x = x$ and $\widetilde{R}_t^x = \Psi(\widetilde{R}_{t-1}^x)\widehat{\lambda}_{\mathcal{A},m}(X_t)$, and $\widetilde{T}^x(A) = \inf\{t \geq 1 : \widetilde{R}_t^x \geq A\}$ (Figure IV.1(d)).

Changepoint detection with unknown π_0

As part of training our classifier $\mathcal{A}$, we have access to labeled pre-change training data $(X_1', Y_1'), ..., (X_m', Y_m')$, and so the assumption that π_∞ is known is reasonable. However, it is less common to have a sample of post-change data, and so the post-change parameter π_0 is often unknown. One approach to overcome an unknown π_0 is to mix over a set Π_0 of potential values for the post-change parameter, with a weight distribution w. Here we are inspired by the work of Lai (1998), who deals with the computational complexity involved in the integration by considering a window-limited approach that uses only a fixed number of the most recent observations. Let Π_0 be the set of possible values for π_0, and let $w(\pi_0)$ be a density on Π_0. Each potential π_0 results in a different likelihood ratio function λ_{π_0}. Lai defines a CUSUM-type mixture stopping rule with detection statistic $R_{t,w}$ and stopping time $T_w(A)$ (Lai, 1998):

$$R_{t,w} = \max_{t-m_\alpha \leq k \leq t} \int_{\Pi_0} \prod_{i=k}^{t} \lambda_{\pi_0}(X_i) w(\pi_0) d\pi_0 \qquad T_w(A) = \inf\{t \geq 1 : R_{t,w} \geq A\}, \tag{IV.6}$$

where m_α is the window size. In our label shift setting, we have

$$\lambda_{\pi_0}(x) = \frac{\pi_0 f_{\infty,X|Y=1}(x) + (1-\pi_0) f_{\infty,X|Y=0}(x)}{\pi_\infty f_{\infty,X|Y=1}(x) + (1-\pi_\infty) f_{\infty,X|Y=0}(x)}. \tag{IV.7}$$

For each π_0, we can replace λ_{π_0} with its estimate $\widehat{\lambda}_{\pi_0,\mathcal{A},m}$ from (IV.5), yielding the detection statistic $\widetilde{R}_{t,w}$ and stopping time $\widetilde{T}_w(A)$:

$$\widetilde{R}_{t,w} = \max_{t-m_\alpha \leq k \leq t} \int_{\Pi_0} \prod_{i=k}^{t} \widehat{\lambda}_{\pi_0,\mathcal{A},m}(X_i) w(\pi_0) d\pi_0 \qquad \widetilde{T}_w(A) = \inf\{t \geq 1 : \widetilde{R}_{t,w} \geq A\}. \tag{IV.8}$$

An alternative to mixing over Π_0 is to maximize over possible values of π_0 at each time step. This is the generalized likelihood ratio (GLR) approach, and has also been studied in previous research (see, e.g., Siegmund and Venkatraman (1995)). For exponential families, some optimality properties of the GLR have been shown, but it is typically harder to control the average run length to false alarm (Tartakovsky et al., 2014). Another option is to perform detection with a worst-case $\pi_0^* \in \Pi_0$ (Unnikrishnan et al., 2011), which provides a worst-case bound on detection delay.

As our proposed detection procedures replace the true likelihood ratio λ with an estimate $\widehat{\lambda}_m$, performance of the estimated detection procedure will depend on the quality of $\widehat{\lambda}_m$. As we increase the size of the training sample $(X_1', Y_1'), ..., (X_m', Y_m')$ used to estimate the likelihood ratio, we hope that $\widehat{\lambda}_m$ will converge to λ, and the expected stopping time of the estimated procedure will converge to the expected stopping time of the optimal procedure. In Section IV.3, we introduce new theoretical results that provide sufficient conditions for this convergence. For our proposed likelihood ratio estimate in (IV.5), the results in Section IV.3 demonstrate that detection performance depends directly on performance of the classifier $\mathcal{A}$, and our results can also be applied to other likelihood ratio estimates. An advantage of the label shift setting is that it supports a variety of of approaches to likelihood ratio estimation. For example, approaches like kernel mean matching (Gretton et al., 2009) and uLSIF (Kanamori et al., 2009) rely on both pre- and post-change data; under the label shift assumption, a post-change sample can be generated by re-sampling or re-weighting the training data $(X_1', Y_1'), ..., (X_m', Y_m')$ when π_∞ is known. We compare this approach to our classifier-based likelihood ratio estimate in Section IV.4.

IV.2.4 Related literature

Label shift testing. Non-sequential two-stample tests between training and test data have been proposed for detecting label shift between batches of data. Saerens et al. (2002) propose a likelihood ratio test, based on expectation-maximization. In Lipton et al. (2018), the authors note that label shift implies a change in the distribution of classifier predictions, and therefore use a two-sample test directly on the training and test set predictions to detecting the change. This is expanded by Rabanser et al. (2019), who recommend tests for label shift as a general method for detecting distributional changes, even if the label shift assumption is not met. For sequential detection of label shift, Ackerman et al. (2020) implement the nonparametric detection procedure in Ross and Adams (2012) based on repeated Cramer–von-Mises tests for a change in distribution, using the `cpm` package (Ross, 2015) in `R`. Like Ackerman et al. (2020), we consider the problem of detecting label shift in a sequential setting, rather than a batch setting. In the sequential setting, label shift detection is an example of the classic problem of sequential changepoint detection, and we apply nonparametric sequential detection tools to the label shift problem. In contrast to Ackerman et al. (2020), we use classifier predictions and the label-shift assumption directly in our detection procedures, which allows us to construct asymptotically optimal nonparametric label shift detection procedures.

Nonparametric detection procedures. Because the pre- and post-change data distributions are rarely known in practice, a variety of nonparametric detection procedures have been proposed. For example, several authors have adapted nonparametric hypothesis tests to the changepoint detection problem, such as Kolmogorov-Smirnov tests (Madrid Padilla et al., 2019), Cramer-von-Mises tests (Ross and Adams, 2012),

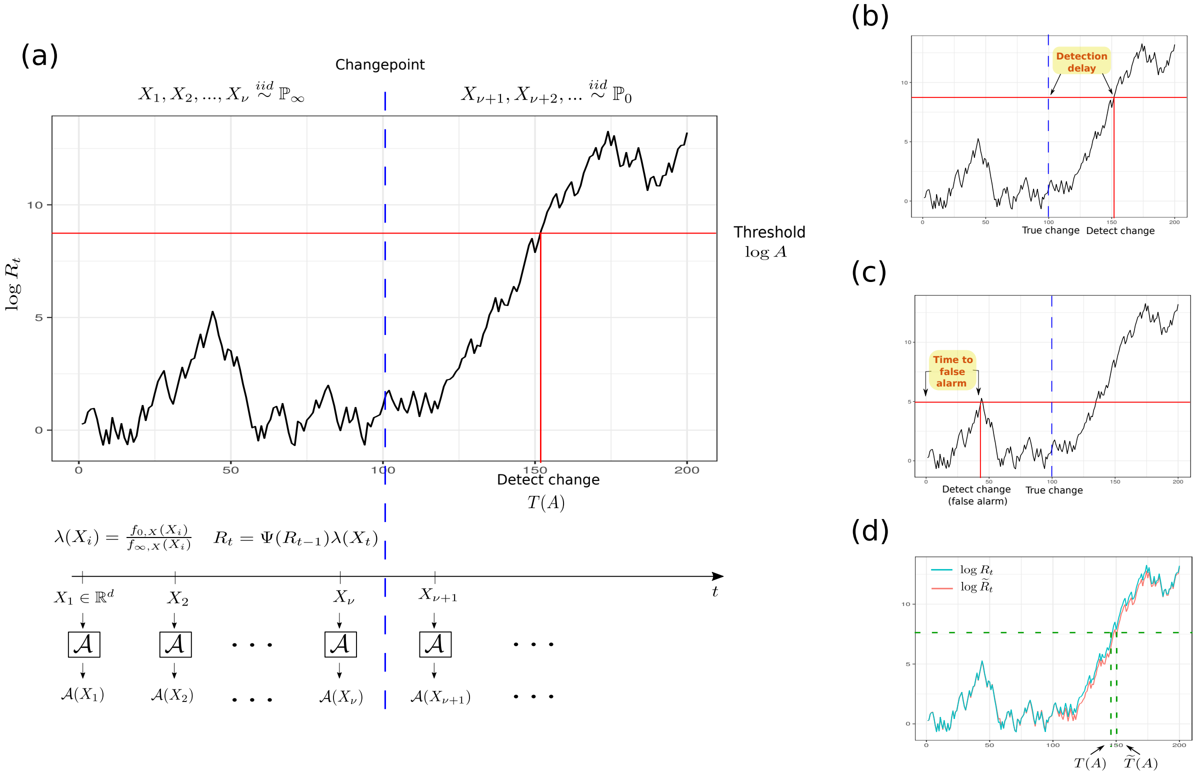

Figure IV.1: Overview of sequential changepoint detection in the classifier setting. **(a)** Data $X_1, X_2, ...$ is observed from the pre-change distribution $\mathbb{P}_\infty$ and the post-change distribution $\mathbb{P}_0$. At each time t, a prediction $\mathcal{A}(X_t)$ is made. If $f_{0,X}$ and $f_{\infty,X}$ are known, then a detection statistic R_t can be calculated using the likelihood ratio $\lambda(X_t)$. A change is detected when $R_t \geq A$ (or equivalently $\log R_t \geq \log A$). **(b)** When a change is detected after the true changepoint ν, then $T(A) - \nu$ is the detection delay. **(c)** When $T(A) < \nu$, then we have a false alarm, and $T(A)$ is the time to false alarm. **(d)** When the true likelihood ratio λ is unknown, we can use an estimate $\widehat{\lambda}$ instead; $\widetilde{R}_t = \Psi(\widetilde{R}_{t-1})\widehat{\lambda}(X_t)$ is the resulting detection statistic. When $\widehat{\lambda}$ is close to λ, the stopping times $\widetilde{T}(A)$ and $T(A)$ are also expected to be close.

and graph-based nearest-neighbors tests (Chen, 2019; Chu and Chen, 2018). Another common approach is to replace the likelihood ratio λ with an estimate $\widehat{\lambda}$. Often, this estimate $\widehat{\lambda}$ is constructed to detect specific types of expected changes, such as shifts in the mean or variance (Brodsky and Darkhovsky, 1993, 2000; Tartakovsky et al., 2012b, 2006a,b), or a change to a stochastically larger/smaller distribution (Bell et al., 1994; Gordon and Pollak, 1994, 1995; McDonald, 1990).

Other authors employ nonparametric density ratio estimates that utilize samples from both the pre- and post-change distributions $\mathbb{P}_\infty$ and $\mathbb{P}_0$. For example, Baron (2000) estimates the post-change distribution sequentially with a histogram density estimator, while another approach is to choose the ratio that maximizes an estimate of the divergence between the pre- and post-change distributions (Nguyen et al., 2010; Kanamori et al., 2009; Kawahara and Sugiyama, 2009; Sugiyama et al., 2008; Liu et al., 2013). Similarly, the kernel mean matching approach (Gretton et al., 2009; Yu and Szepesvári, 2012) estimates the ratio by matching moments after mapping into a reproducing kernel Hilbert space (RKHS), while Bickel et al. (2009) propose training a classifier to predict whether data comes from the pre- or post-change distribution.

In the label shift case, the difficulty of having samples from the post-change distribution is reduced to knowing the post-change parameter π_0 (see (IV.1)). In the following section, we propose a simple estimate of the likelihood ratio as a linear function of the classifier scores $\mathcal{A}(X_i)$. An advantage of this method is that the existing classifier can be used without additional estimation, and performance of the detection procedure is directly related to performance of the classifier. In addition, when π_0 is unknown it is easy to calculate the likelihood ratio over a range of potential values Π_0, without having to re-estimate the ratio each time (see Section IV.2.3).

Operating characteristics. The performance of sequential detection procedures, with stopping time $T^x(A)$ at threshold A, is typically assessed by two operating characteristics, the *average time to false alarm* $\mathbb{E}_\infty(T^x(A))$, and the *average detection delay* $\mathbb{E}_0(T^x(A))$, which are expected stopping times under the pre- and post-change distributions respectively (Figure IV.1(b) and IV.1(c)). The goal is to minimize the average detection delay, subject to a lower bound on the average time to false alarm, and the CUSUM and Shiryaev-Roberts procedures are known to be optimal or approximately optimal for this problem (Lorden, 1971; Moustakides, 1986; Tartakovsky et al., 2012a). We therefore compare average detection delay and average time to false alarm as a way to assess procedures in this manuscript.

IV.3 Operating characteristics of nonparametric detection procedures

As suggested in (IV.4) and (IV.5), the ability of the classifier $\mathcal{A}$ to discriminate between $X|Y=0$ and $X|Y=1$, and approximate either $\mathbb{P}_\infty(Y_i=1|X_i)$ or the labels Y_i, is linked to performance of the nonparametric changepoint detection procedure. Previous results on the performance of nonparametric detection procedures have typically focused on the relative efficiency of estimated procedures compared to optimal performance (see, e.g., Bell et al. (1994); Unnikrishnan et al. (2011)), by examining the limiting behavior of the ratio $\mathbb{E}_0(\widetilde{T}^x(\widetilde{A}))/\mathbb{E}_0(T^x(A))$ as $\widetilde{A}, A \to \infty$, where $\widetilde{A}$ is chosen so that $\mathbb{E}_\infty(\widetilde{T}^x(\widetilde{A})) = \mathbb{E}_\infty(T^x(A))$.

While relative efficiency is useful for comparing detection procedures, and it is natural to compare detection delay at the same ARL, formal results are typically asymptotic in the thresholds A and $\widetilde{A}$. As an alternative, in this section we introduce new results on the convergence of operating characteristics that are asymptotic in the size of the training sample, and consider a single fixed threshold A.

The differences $|\mathbb{E}_i(\widetilde{T}^x(A)) - \mathbb{E}_i(T^x(A))|$, $i \in \{0, \infty\}$, compare operating characteristics of the estimated, nonparametric procedure (with likelihood ratio estimate $\widehat{\lambda}$ and stopping time $\widetilde{T}^x$) and the true, optimal procedure (with likelihood ratio λ and stopping time T^x). To emphasize the dependence of $\mathbb{E}_i(\widetilde{T}^x(A))$ on $\widehat{\lambda}$, we will write $\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_m)$, where the subscript m is used to show the dependence of $\widehat{\lambda}_m$ on the size of the training set. Ideally, $\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_m) \xrightarrow{p} \mathbb{E}_i(T^x(A))$ as $m \to \infty$. In the following sections, we provide conditions under which this convergence in probability holds, and we provide upper bounds on the rate of convergence. Our main result is Theorem 1, which demonstrates convergence when $\lambda(X)$ and $\widehat{\lambda}_m(X)$ are continuous random variables; this is the typical case in the sequential detection literature (see, e.g., Tartakovsky et al. (2014)). We also prove a more general result in Theorem 2 which makes no assumptions on the distribution of $\lambda(X)$ or $\widehat{\lambda}_m(X)$ and can be applied to the mixture stopping rule in (IV.8) when the post-change prevalence π_0 is unknown.

In the context of classifier label shift, our results demonstrate the connection between detection performance and classifier performance. In the label shift setting, we observe a sequence of unlabeled data $X_1, X_2, X_3, \ldots$, for which a true likelihood ratio $\lambda(x) = f_{0,X}(x)/f_{\infty,X}(x)$ exists, given by (IV.2). The detection statistic $R_t = \Psi(R_{t-1})\lambda(X_t)$ which uses the true likelihood ratio can be considered optimal for detecting a change in the unlabeled data, and Theorem 1 and Theorem 2 provide conditions under which using $\widehat{\lambda}_m(X)$ in place of $\lambda(X)$ is asymptotically optimal. However, it is important to note that $\lambda(x) = f_{0,X}(x)/f_{\infty,X}(x)$ is typically *not* optimal for detecting a change in the *labeled* data $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \ldots$, in which case the true (optimal) likelihood ratio $\lambda(X_i, Y_i)$ is given by (IV.3). Detection with the marginal likelihood ratio $\lambda(X_i)$ is only optimal for labeled data (X_i, Y_i) when $\lambda(X_i) \equiv \lambda(X_i, Y_i)$, which requires $X|Y = 0$ and $X|Y = 1$ to be separable. In this special case, convergence of $\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_{\mathcal{A},m})$ depends on the sensitivity and specificity of the classifier $\mathcal{A}$, which we demonstrate in Corollary 1.

IV.3.1 Continuous likelihood ratios

In this chapter we consider detection statistics of the form $R_t = \Psi(R_{t-1})\lambda(X_t)$ and $\widetilde{R}_t = \Psi(\widetilde{R}_{t-1})\widehat{\lambda}_m(X_t)$, which involve the random variables $\lambda(X)$ and $\widehat{\lambda}_m(X)$ at each step. Convergence of $\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_m)$ to $\mathbb{E}_i(T^x(A))$ therefore requires that the distribution of $\widehat{\lambda}_m(X)$ be "close" to the distribution of $\lambda(X)$. The distribution of $\lambda(X)$ is generally continuous when X is continuous, such as the ratio for two normal distributions (see Example 1 below). Likewise, the estimate $\widehat{\lambda}_{\mathcal{A},m}$ from (IV.5) is typically continuous when $\mathcal{A}(X) \in [0, 1]$ is a predicted probability. When $\lambda(X)$ and $\widehat{\lambda}_m(X)$ are continuous, convergence of $|\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_m) - \mathbb{E}_i(T^x(A))|$ depends on the total variation distance between these two distributions, as shown in Theorem 1, under the following assumptions:

(B1) The detection procedure is defined by a detection statistic R_t^x and a stopping time $T^x(A) = \inf\{t \geq 1 : R_t^x \geq A\}$, with $R_t^x = \Psi(R_{t-1}^x)\lambda(X_t)$ and $R_0^x = x$. Likewise, the estimated detection procedure is defined by $\widetilde{T}^x(A) = \inf\{t \geq 1 : \widetilde{R}_t^x(A) \geq A\}$, with $\widetilde{R}_t^x = \Psi(\widetilde{R}_{t-1}^x)\widehat{\lambda}_m(X_t)$ and $\widetilde{R}_0^x = x$.

(B2) The distributions of $\lambda(X)$ and $\widehat{\lambda}_m(X)$ are continuous, with $X \sim \mathbb{P}_i$.

(B3) The densities f_λ^i and $f_{\widehat{\lambda}_m}^i$ of $\lambda(X)$ and $\widehat{\lambda}_m(X)$ under $\mathbb{P}_i$ satisfy the following assumptions:

(a) $\int\limits_0^A (f_\lambda^i(s))^2 ds \ < \infty$ and $\int\limits_0^A (f_{\widehat{\lambda}_m}^i(s))^2 ds \ < \infty$,

(b) $\lim\limits_{\varepsilon \to 0} \sup\limits_{x,z \in [0,A]; |x-z| < \varepsilon} \int\limits_0^A |f_\lambda^i\left(\frac{y}{\Psi(x)}\right) - f_\lambda^i\left(\frac{y}{\Psi(z)}\right)| dy \ = 0$ (and likewise for $f_{\widehat{\lambda}_m}^i$).

(B4) The functions $\Psi(r)$ and $1/\Psi(r)$ are Lipschitz continuous on $[0, A]$, and $\Psi(r) \geq 1$ for all r.

(B5) The total variation distance $TV(f_\lambda^i, f_{\widehat{\lambda}_m}^i) = \int |f_\lambda^i(s) - f_{\widehat{\lambda}_m}^i(s)| ds \ \xrightarrow{p} 0$ as $m \to \infty$.

Assumption (B1) holds for standard detection procedures like CUSUM, Shiryaev-Roberts, and variants. Note that the assumption that $\lambda(X)$ is continuous is common in the changepoint detection literature, and is relied on by many approaches to calculating or approximating the expected stopping time of a detection procedure. In general we are interested in $|\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_m) - \mathbb{E}_i(T^x(A))|$ for both $i = 0$ and $i = \infty$, which means (B2) requires that the support of X is the same under $\mathbb{P}_0$ and $\mathbb{P}_\infty$. Assumption (B4) holds for common Ψ, such as $\Psi(r) = \max\{1, r\}$ (CUSUM) and $\Psi(r) = 1 + r$ (Shiryaev-Roberts). Assumption (B3) can be thought of as a requirement that the distributions of $\lambda(X)$ and $\widehat{\lambda}_m(X)$ are not close to having any point masses. A sufficient condition for (B3) is that the densities f^i_λ and $f^i_{\widehat{\lambda}_m}$ are Lipschitz continuous.

Theorem 1. *Suppose that assumptions (B1) - (B5) hold. Then*

$$|\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_m) - \mathbb{E}_i(T^x(A))| \leq O_P(TV(f^i_\lambda, f^i_{\widehat{\lambda}_m})). \tag{IV.9}$$

If, in addition, f^i_λ is bounded, then

$$|\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_m) - \mathbb{E}_i(T^x(A))| \leq O_P(\mathbb{E}_i(|\widehat{\lambda}_m(X) - \lambda(X)|) + O_P(||F^i_{\widehat{\lambda}_m} - F^i_\lambda||_\infty). \tag{IV.10}$$

Proof. See Appendix B.1. □

The bound in Theorem 1 depends on the distribution of the likelihood ratio, which is intuitive as our detection procedure includes a random draw from this distribution at each step (see (B1)). As we can see from (IV.10), when f^i_λ is bounded, convergence depends directly on the L_1 rate of convergence of the likelihood ratio estimate $\widehat{\lambda}_m$. In the classifier setting, this is equivalent to the L_1 convergence of the classifier scores $\mathcal{A}(X)$ to the true probabilities $\mathbb{P}_\infty(Y = 1|X)$, illustrating that performance of the changepoint detection procedure depends on performance of the classifier. Note that without parametric assumptions, the optimal L_1 rate of convergence is $m^{-p/(2p+d)}$ if λ is a p-times differentiable function (Stone, 1982), though not all classifiers will converge. In general, results for nonparametric regression are more common for L_2 convergence rather than L_1 convergence, and the same is true for other methods of density ratio estimation – for example, Nguyen et al. (2010) provide conditions under which a density ratio estimate from divergence maximization converges in Hellinger distance.

Remark: Consider the classifier label shift setting, with unlabeled data, where the true likelihood ratio $\lambda(x) = f_{0,X}(x)/f_{\infty,X}(x)$ is given by (IV.2). Suppose that $\lambda(X)$ is continuous, but we use a binary classifier with $\mathcal{A}(X) \in \{0, 1\}$ (for example, by thresholding predicted probabilities), and estimate λ with $\widehat{\lambda}_{\mathcal{A},m}$ as in (IV.5). Because the binary predictions $\mathcal{A}(X)$ will never converge to the true probabilities $\mathbb{P}_\infty(Y = 1|X)$, then $\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_{\mathcal{A},m})$ won't converge to $\mathbb{E}_i(T^x(A))$. Therefore, if $\mathbb{P}_\infty(Y = 1|X)$ is expected to be a continuous function of X, it is better to use predicted probabilities, rather than binary predictions, for changepoint detection.

IV.3.2 Unknown π_0

Theorem 1 applies when $\lambda(X)$ and $\widehat{\lambda}_m(X)$ are continuous, and π_0 is known or can be consistently estimated. However, as discussed in Section IV.2.3, we typically expect π_0 to be unknown in practice. Here we explicitly consider the classifier label shift setting, with likelihood ratio estimate $\widehat{\lambda}_{\pi_0,\mathcal{A},m}$ for each potential π_0 given by (IV.5). Note that an advantage of this likelihood ratio estimate is that mixing over Π_0 is simple: we need only change π_0 in (IV.5). In contrast, two-sample ratio estimation procedures like those discussed in Gretton et al. (2009), Nguyen et al. (2010), and Kanamori et al. (2009) would require a new post-change sample and a new likelihood ratio estimate to be calculated for each π_0.

Since π_0 is unknown, we apply the CUSUM-type mixture stopping rule (IV.8) discussed in Section IV.2.3. As $\widetilde{R}_{t,w}$ cannot be written recursively, the proof techniques of Theorem 1 do not apply, but it is still possible to show that $\mathbb{E}_i(\widetilde{T}_w(A)|\mathcal{A})$ is consistent for $\mathbb{E}_i(T_w(A))$ as $m \to \infty$, though we lose the ability to provide a rate of convergence:

Theorem 2. *Let $\Pi_0 = [a, b]$ where $0 < a \leq b < 1$, and suppose there exist sets $\mathcal{S}_c$ indexed by $c > 0$, such that $\mathcal{S}_{c_1} \subseteq \mathcal{S}_{c_2}$ when $c_1 > c_2$, $\lim\limits_{c \to 0} \mathbb{P}_i(X \in \mathcal{S}_c) = 1$, and such that $\sup\limits_{\substack{x \in \mathcal{S}_c \\ \pi_0 \in \Pi_0}} |\widehat{\lambda}_{\pi_0,\mathcal{A},m}(x) - \lambda_{\pi_0}(x)| \xrightarrow{p} 0$ for all c. Furthermore, assume that $\mathbb{E}_i(T_w(A))$ is a continuous function of A. Then for all $A > 0$, $\mathbb{E}_i(\widetilde{T}_w(A)|\mathcal{A}) \xrightarrow{p} \mathbb{E}_i(T_w(A))$.*

Proof. See Appendix B.3. □

In practice, the mixture procedure in (IV.8) often performs quite close to the procedure with estimated likelihood ratio (IV.5), and performance improves when Π_0 can be narrowed. We will see in Section IV.5 that the mixture procedure outperforms other nonparametric detection procedures which don't require knowledge of π_0 for detecting a change in dengue prevalence.

IV.3.3 Separable class distributions

As discussed above, the true likelihood ratio and optimal detection procedure depend on whether the labels Y_i are observed. The exception is when the classes can be separated: if $X|Y = 0$ and $X|Y = 1$ are separable, then optimal changepoint detection with the unlabeled data $X_1, X_2, X_3, ...$ is equivalent to optimal changepoint detection with the labeled data $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), ...$ Assuming the classifier $\mathcal{A}$ can separate the two classes given enough training data, then the nonparametric detection procedure is asymptotically optimal, and convergence of $|\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_m) - \mathbb{E}_i(T^x(A))|$ depends on the sensitivity and specificit of $\mathcal{A}$:

Corollary 1. *Suppose that π_∞ and π_0 are known, and furthermore that $\log(\pi_0/\pi_\infty)$ and $\log((1-\pi_0)/(1-\pi_\infty))$ are rational. The optimal detection procedure observes $(X_1, Y_1), (X_2, Y_2), ...,$ with likelihood ratio $\lambda(X_i, Y_i)$ given by (IV.3), and detection statistic $R_t^x = \max\{1, R_{t-1}^x\}\lambda(X_t, Y_t)$. When the labels Y_i are unobserved, the nonparametric detection procedure uses a binary classifier $\mathcal{A}$, with $\mathcal{A}(X) \in \{0, 1\}$; the likelihood ratio $\widehat{\lambda}_{\mathcal{A},m}$ is given by (IV.5), and the detection statistic is $\widetilde{R}_t^x = \max\{1, \widetilde{R}_{t-1}^x\}\widehat{\lambda}_{\mathcal{A},m}(X_t)$. Then, if $\mathbb{P}_i(\mathcal{A}(X) = 1|Y = 1, \mathcal{A}) \xrightarrow{p} 1$ and $\mathbb{P}_i(\mathcal{A}(X) = 0|Y = 0, \mathcal{A}) \xrightarrow{p} 1$ as $m \to \infty$,*

$$|\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_m) - \mathbb{E}_i(T^x(A))| \leq O_P(\mathbb{P}_i(\mathcal{A}(X) = 0|Y = 1, \mathcal{A}) + \mathbb{P}_i(\mathcal{A}(X) = 1|Y = 0, \mathcal{A})). \tag{IV.11}$$

Proof. See Appendix B.2. □

The detection procedure in Corollary 1 is simply a Bernoulli CUSUM procedure, and the right hand side of (IV.11) depends on the convergence of the specificity and sensitivity of the classifier $\mathcal{A}$, which requires that the two distributions $X|Y = 0$ and $X|Y = 1$ can be separated. This is a stronger requirement than in Theorem 1: consistently estimating probabilities $\mathbb{P}_i(Y = 1|X)$ can be possible even when consistently estimating labels is impossible. In practice, $X|Y = 0$ and $X|Y = 1$ are rarely perfectly separable, but Corollary 1 is still helpful for seeing the relationship between classifier performance and changepoint performance.

The assumption that $\log(\pi_0/\pi_\infty)$ and $\log((1-\pi_0)/(1-\pi_\infty))$ are rational is needed to ensure that R_t^x and $\widetilde{R}_t^x$ are Markov processes on the same state space, as is the restriction of Ψ to the CUSUM $\Psi(x) = \max\{1, x\}$. In practice, Reynolds Jr and Stoumbos (1999) show that the expected stopping time is close when $\log \lambda(X)$ is not rational but a rational approximation is used.

IV.3.4 Examples

To help illustrate our theoretical results, we consider several specific change detection scenarios, which provide concrete examples of our results in action. First, we consider a simple univariate parametric setting. When we have a parametric model, we hope that the rate of convergence for our detection procedure is the same as the rate of convergence for the parameter estimates. In this example, we consider a shift in the mean of a normal distribution, which also demonstrates that our results extend beyond the label shift setting.

Example 1 (Normal Shift)**.** Suppose it is known that under $\mathbb{P}_\infty$, $X \sim N(0, 1)$, and under $\mathbb{P}_0$, $X \sim N(\mu, 1)$, with $\mu > 0$. Then, the likelihood ratio λ is given by $\lambda(x) = \exp\{\mu x - \mu^2/2\}$. Furthermore, we have a training sample $X_1', ..., X_m' \overset{iid}{\sim} N(\mu, 1)$, with which we estimate μ by $\widehat{\mu}_m = \frac{1}{m}\sum\limits_{i=1}^{m} X_i'$; the likelihood ratio estimate is then $\widehat{\lambda}_m(x) = \exp\{\widehat{\mu}_m x - \widehat{\mu}_m^2/2\}$ (this is similar to the procedures proposed in Tartakovsky et al. (2012b)). Then, the conditions for Theorem 1 hold with $TV(f^i_\lambda, f^i_{\widehat{\lambda}_m}) \leq O_P(|\widehat{\mu}_m - \mu|)$. Therefore, an upper bound on the rate of convergence for the estimated detection procedure is $|\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_m) - \mathbb{E}_i(T^x(A))| \leq O_P(|\widehat{\mu}_m - \mu|) = O_P(1/\sqrt{m})$. Full details and calculations can be found in Appendix B.4.

We now examine the case where we detect label shift with classifier predictions. To make the example clear, we consider an LDA classifier, for which the classifier scores and their distribution have a closed form.

Example 2 (LDA)**.** Suppose that under $\mathbb{P}_\infty$, $X \sim \pi_\infty N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + (1 - \pi_\infty) N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma})$, while under $\mathbb{P}_0$, $X \sim \pi_0 N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + (1 - \pi_0) N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma})$. Suppose that π_∞ and π_0 are known, and we have a training sample $X_1', ..., X_m' \overset{iid}{\sim} \mathbb{P}_\infty$ with which we construct estimates $\widehat{\boldsymbol{\mu}}_1, \widehat{\boldsymbol{\mu}}_0, \widehat{\boldsymbol{\Sigma}}$. Our likelihood ratio estimate $\widehat{\lambda}_{\mathcal{A},m}$ is given by (IV.5), where $\mathcal{A}$ is given by

$$\mathcal{A}(X_i) = \frac{\pi_\infty \text{MVN}(X_i; \widehat{\boldsymbol{\mu}_1}, \widehat{\boldsymbol{\Sigma}})}{\pi_\infty \text{MVN}(X_i; \widehat{\boldsymbol{\mu}_1}, \widehat{\boldsymbol{\Sigma}}) + (1 - \pi_\infty)\text{MVN}(X_i; \widehat{\boldsymbol{\mu}_0}, \widehat{\boldsymbol{\Sigma}})}, \tag{IV.12}$$

where $\text{MVN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate normal density with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The true likelihood ratio is similar, just replacing $\widehat{\boldsymbol{\mu}}_1, \widehat{\boldsymbol{\mu}}_0, \widehat{\boldsymbol{\Sigma}}$ with the true parameters. (B3) follows because f^i_λ and $f^i_{\widehat{\lambda}}$ can be shown to be Lipschitz, while (B5) follows from the strong consistency of $\widehat{\boldsymbol{\mu}}_i$ and $\widehat{\boldsymbol{\Sigma}}$. The rate of convergence in Theorem 1 depends on the rate of convergence for $||\widehat{\boldsymbol{\Sigma}}^{-1} - \boldsymbol{\Sigma}^{-1}||_F$, where $|| \cdot ||_F$ denotes Frobenius norm. Details are provided in Appendix B.5.

While the formal results in Theorem 1 and Corollary 1 depend on convergence of $\widehat{\lambda}(X)$ to $\lambda(X)$, detection performance will depend on classifier performance even if $\widehat{\lambda}(X)$ does not converge. Here we demonstrate that the relationship between detection delay and classifier performance still holds for mis-specified classifiers, and that the expected L_1 distance from Theorem 1, $\mathbb{E}_i(|\widehat{\lambda}(X) - \lambda(X)|)$, is a useful summary of classifier performance.

Example 3 (LDA vs. QDA)**.** Suppose we observe training data $(X_1', Y_1'), ..., (X_m', Y_m') \in \mathbb{R}^{150}$, with $X|Y = 0 \sim N(\boldsymbol{\mu_0}, \boldsymbol{\Sigma_0})$ and $X|Y = 1 \sim N(\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1})$, with $\boldsymbol{\Sigma_0} \neq \boldsymbol{\Sigma_1}$. Construct linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) classifiers $\mathcal{A}_L$ and $\mathcal{A}_Q$. Using (IV.5), perform CUSUM changepoint detection with each classifier. Figure IV.2 shows $\mathbb{E}_0(|\widehat{\lambda}(X) - \lambda(X)|)$ and $\mathbb{E}_0(\widetilde{T}(A))$ for the resulting LDA and QDA detection procedures; for each classifier, the threshold A is chosen so that $\mathbb{E}_\infty(\widetilde{T}(A)) \approx 180$. For large m, $\mathcal{A}_Q$ outperforms $\mathcal{A}_L$ because the QDA assumptions are met whereas the LDA assumptions are violated, but for small m we see that $\mathcal{A}_L$ does better due to the bias-variance trade-off. As

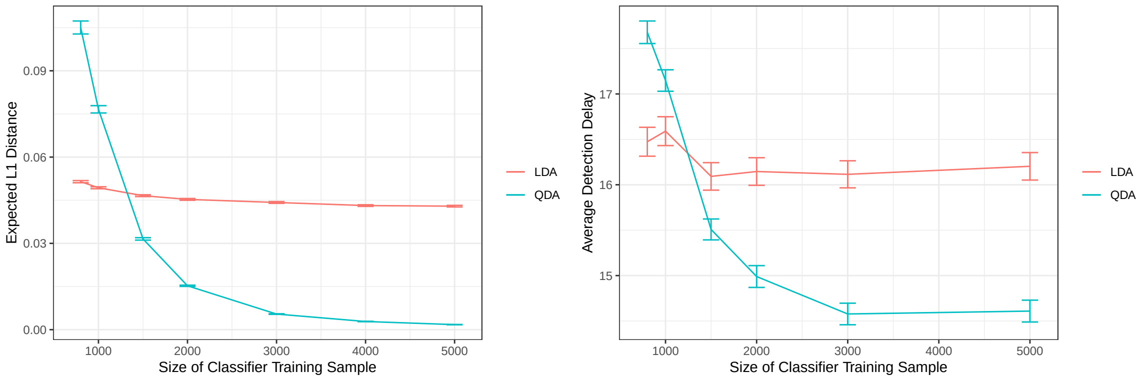

Figure IV.2: Comparison between changepoint detection with LDA and QDA classifier scores, when LDA assumptions are violated but QDA assumptions are satisfied. Left: Average classifier test error on new data, as a function of the size of the classifier training sample. Right: Each LDA and QDA classifier is used for KDE changepoint detection with the classifier scores. This plot shows the average detection delay when the average ARL is approximately 180 (averaged across classifiers) as a function of the size of the classifier training sample.

shown in Figure IV.2, the LDA detection procedure does better than the QDA detection procedure exactly when $\mathcal{A}_L$ outperforms $\mathcal{A}_Q$.

In Example 2, we used an LDA classifier to detect label shift in a mixture of normals with a common covariance matrix. Here we consider a similar procedure which uses nonparametric density estimation rather than an assumption of Gaussianity (see, e.g. Cipolli and Hanson (2017, 2019)).

Example 4 (Density Estimation Classifier)**.** Suppose that π_∞ and π_0 are known, and we have a training sample $X_1', ..., X_m' \overset{iid}{\sim} \mathbb{P}_\infty$ with which we construct density estimates $\widehat{f}_{\infty,X|Y=0}$ and $\widehat{f}_{\infty,X|Y=1}$. Our classifier $\mathcal{A}$ is given by

$$\mathcal{A}(x) = \frac{\pi_\infty \widehat{f}_{\infty,X|Y=1}(x)}{\pi_\infty \widehat{f}_{\infty,X|Y=1}(x) + (1-\pi_\infty)\widehat{f}_{\infty,X|Y=0}(x)}, \tag{IV.13}$$

and our likelihood ratio estimate is given by (IV.5). For $c > 0$, let $\mathcal{S}_c = \{x : f_{\infty,X}(x) > c\}$. Clearly, $\lim_{c\to 0} \mathbb{P}_i(X \in \mathcal{S}_c) = 1$. Furthermore, with appropriate choice of kernel and bandwidth, $||\widehat{f}_{\infty,X|Y=1} - f_{\infty,X|Y=1}||_\infty, ||\widehat{f}_{\infty,X|Y=0} - f_{\infty,X|Y=0}||_\infty \overset{p}{\to} 0$ (see, e.g., Giné and Guillou (2002)). Thus for each c, $\sup_{\substack{x\in\mathcal{S}_c \\ \pi_0\in\Pi_0}} |\widehat{\lambda}_{\pi_0,\mathcal{A},m}(x) - \lambda_{\pi_0}(x)| \overset{p}{\to} 0$.

IV.4 Simulation studies

We investigate the empirical performance of the classifier-based label shift detection procedure described in Section IV.2.3, with the likelihood ratio estimate in (IV.5). Our likelihood ratio estimate depends on a classifier, and for simplicity we will use an LDA classifier, since it is easy to control whether the LDA assumptions are satisfied, as in Example 3. For comparison, we consider several other detection procedures, which represent different approaches to changepoint detection. These procedures are summarized below and in Table IV.1. Because the proposed procedure from Section IV.2.3 is designed specifically for the classifier label shift setting, it leverages more information than the other nonparametric detection procedures. In particular, as summarized in Table IV.1, estimating the likelihood ratio with (IV.5) assumes that the label shift assumption holds, and the classifier $\mathcal{A}(\cdot)$ performs well. Through simulations, we show that detection with (IV.5) outperforms the other nonparametric procedures when these assumptions are met, and can still perform well when the assumptions are violated. While we use a simple setting for simulations, in Section IV.5 we apply the same methods to detect a change in dengue prevalence using the data and classifier from Tuan et al. (2015), with similar results to our simulations in this section.

We compare the following methods:

Classifier-based CUSUM This is the nonparametric method proposed in Section IV.2.3, with likelihood ratio estimate (IV.5). For the purposes of simulations, $\mathcal{A}$ in (IV.5) is an LDA classifier. Here we use a CUSUM procedure, so $\Psi(r) = \max\{1, r\}$.

Optimal CUSUM The optimal CUSUM procedure (Page, 1954) uses the true likelihood ratio, and can be implemented when the true likelihood ratio is known.

uLSIF CUSUM uLSIF (Kanamori et al., 2009) is a nonparametric method for estimating the likelihood ratio, by maximizing an empirical divergence. As described in Section IV.2.3, uLSIF can be used with training data under the label shift assumption by re-weighting or re-sampling training points, but it does not exploit the label shift structure of the likelihood ratio. A variety of similar density ratio estimation approaches exist, including KLIEP and kernel mean matching (Sugiyama et al., 2008; Gretton et al., 2009; Kanamori et al., 2012), and we take uLSIF as a representative. Here we used the `densratio` package (Makiyama, 2019) to implement uLSIF, and employ the resulting estimate in a CUSUM procedure.

CPM Ackerman et al. (2020) perform nonparametric label shift detection using the CPM framework described in Ross et al. (2011) and Ross (2015). The CPM framework detects changes in a sequence of univariate data using repeated nonparametric tests; Ackerman et al. (2020) applied repeated Cramer–von-Mises tests to a sequence of cosine divergences calculated between new data and training data. We evaluate CPM applied to both the **classifier** predictions and the cosine **divergences** used by Ackerman et al. (2020). CPM stopping times are calculated with the `cpm` package (Ross, 2015).

Information leveraged	Classifier CUSUM	Optimal CUSUM	uLSIF CUSUM	CPM (classifier)	CPM (divergence)	kNN
True labels		✓				
Label shift	✓		✓			
Good classifier	✓			✓		
Training data	✓		✓	✓	✓	✓

Table IV.1: Comparison of the information used by each changepoint detection procedure considered in simulations. CPM and kNN are more general than the classifier CUSUM procedure from Section IV.2.3, but as a result they leverage less information. If the label shift assumption holds and the classifier performs well, we expect the classifier-based CUSUM method to outperform these more general procedures.

kNN Chen (2019) and Chu and Chen (2018) propose a sequential graph-based kNN detection procedure, based on repeated nearest-neighbor two-sample tests in a sliding window. Note that while the kNN approach uses training data, only a fixed window of data is considered. Similar to some parameters in Chu and Chen (2018), we set the window size to 200 and the number of nearest neighbors to $k = 5$. Stopping times are calculated with the `gStream` package (Chen and Chu, 2019).

Metrics. Performance of each detection procedure is measured by detection delay, calculated as $\mathbb{E}_0[T]$ (for CUSUM procedures, this corresponds to Lorden's (Lorden, 1971) detection delay). As is standard, we compare detection delays with each method calibrated to have the same average run length $\mathbb{E}_\infty[T]$. Here we use $\mathbb{E}_\infty[T] = 500$, which is a common value in the sequential detection literature. Expected stopping times are estimated via Monte Carlo simulation.

Scenarios. Under the label shift assumption, the classifier-based CUSUM procedure uses classifier predictions $\mathcal{A}(X_i)$ to estimate the likelihood ratio. To compare performance of the different detection procedures, we use two different simulation scenarios. In the first scenario, we change the training sample size and the performance of the classifier (by changing the distribution of the data X_i and violating LDA assumptions). In the second scenario, we change the performance of the classifier and the suitability of the label shift assumption.

Scenario 1: Pre-change data is generated as $X \sim \pi_\infty N(\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1}) + (1 - \pi_\infty)N(\boldsymbol{\mu_0}, \boldsymbol{\Sigma_0})$, and post-change data is generated as $X \sim \pi_0 N(\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1}) + (1 - \pi_0)N(\boldsymbol{\mu_0}, \boldsymbol{\Sigma_0})$. In all simulations, $\pi_\infty = 0.4$, $\pi_0 = 0.7$, $\boldsymbol{\mu_0} = [0, 0]$, $\boldsymbol{\mu_1} = [1.5, 1.5]$, and $\boldsymbol{\Sigma_0} = \boldsymbol{I}$. Training data $(X'_1, Y'_1), ..., (X'_m, Y'_m)$ is simulated from the pre-change distribution, and used to train the LDA classifier, estimate the uLSIF likelihood ratio, and startup the CPM and kNN detection statistics. We consider $m \in \{200, 1000, 5000\}$, and $\boldsymbol{\Sigma_1} \in \left\{\boldsymbol{I}, \begin{bmatrix} 2 & 0.1 \\ 0.1 & 2 \end{bmatrix}, \begin{bmatrix} 4 & 0.5 \\ 0.5 & 4 \end{bmatrix}\right\}$.

$\mathbf{\Sigma_1}$	m	Detection delay when $\mathbb{E}_\infty[T] \approx 500$					
		Classifier CUSUM	Optimal CUSUM	uLSIF CUSUM	CPM (classifier)	CPM (divergence)	kNN
$\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$	200	28.8 (0.59)	29.0 (0.23)	33.8 (5.62)	46.9 (1.45)	51.5 (1.72)	$\geq$ 155 (3.46)
	1000	28.8 (0.26)		33.4 (1.51)	35.0 (0.81)	37.0 (0.85)	
	5000	28.8 (0.10)		31.5 (1.11)	32.6 (0.76)	33.7 (0.81)	
$\begin{bmatrix}2 & 0.1\\0.1 & 2\end{bmatrix}$	200	34.3 (0.96)	33.1 (0.28)	49.0 (11.4)	61.2 (2.24)	69.5 (2.67)	$\geq$ 168 (3.58)
	1000	34.0 (0.40)		43.3 (2.39)	42.4 (1.07)	46.6 (1.18)	
	5000	34.0 (0.16)		38.8 (2.44)	38.1 (0.94)	41.7 (1.03)	
$\begin{bmatrix}4 & 0.5\\0.5 & 4\end{bmatrix}$	200	41.9 (1.44)	33.4 (0.29)	164 (47.3)	74.6 (2.82)	94.8 (3.56)	$\geq$ 176 (3.59)
	1000	41.8 (0.60)		73.3 (7.05)	52.1 (1.46)	62.4 (1.74)	
	5000	41.9 (0.25)		54.2 (5.19)	51.2 (1.36)	58.5 (1.55)	

Table IV.2: Simulation results for Scenario 1. Performance of each procedure is measured by detection delay, calculated as $\mathbb{E}_0[T]$. The estimated detection delay from Monte Carlo simulation is reported, with the standard error in parentheses. For the kNN procedure, a window of size 200 is used, so only 200 training points are considered. In the case of kNN, if a change is not detected within the sliding window, windows after time point 200 will consist of only post-change observations, so for computational purposes a fixed number of post-change observations is simulated and we report a lower bound on the detection delay.

Scenario 2: Pre-change data is generated as $X \sim \pi_\infty N(\boldsymbol{\mu_{\infty,1}}, \mathbf{\Sigma_1}) + (1 - \pi_\infty) N(\boldsymbol{\mu_{\infty,0}}, \mathbf{\Sigma_0})$, and post-change data is generated as $X \sim \pi_0 N(\boldsymbol{\mu_{0,1}}, \mathbf{\Sigma_1}) + (1 - \pi_0) N(\boldsymbol{\mu_{0,0}}, \mathbf{\Sigma_0})$. In all simulations, $\pi_\infty = 0.4$, $\pi_0 = 0.7$, $\boldsymbol{\mu_{\infty,0}} = [0, 0]$, $\boldsymbol{\mu_{\infty,1}} = [1.5, 1.5]$, and $\mathbf{\Sigma_0} = \boldsymbol{I}$. Training data $(X_1', Y_1'), ..., (X_{1000}', Y_{1000}')$ is simulated from the pre-change distribution, and used to train the LDA classifier, estimate the uLSIF likelihood ratio, and startup the CPM and kNN detection statistics. We consider $\mathbf{\Sigma_1} \in \left\{ \boldsymbol{I}, \begin{bmatrix}2 & 0.1\\0.1 & 2\end{bmatrix}, \begin{bmatrix}4 & 0.5\\0.5 & 4\end{bmatrix} \right\}$ and the following pairs for $\boldsymbol{\mu_{0,0}}$ and $\boldsymbol{\mu_{0,1}}$: $\boldsymbol{\mu_{0,0}} = [0.5, 0.5]$ and $\boldsymbol{\mu_{0,1}} = [1, 1]$; $\boldsymbol{\mu_{0,0}} = [0.75, 0.75]$ and $\boldsymbol{\mu_{0,1}} = [0.75, 0.75]$; and $\boldsymbol{\mu_{0,0}} = [1, 1]$ and $\boldsymbol{\mu_{0,1}} = [0.5, 0.5]$.

Results. Table IV.2 shows the results for Scenario 1, when the label shift assumption holds. We can see that when the LDA assumptions are met (specifically $\mathbf{\Sigma_1} = \mathbf{\Sigma_0} = \boldsymbol{I}$), LDA performs very close to the optimal CUSUM procedure, as we would predict from Example 2. Performance of the LDA detection procedure relative to the optimal CUSUM procedure declines as the assumption that $\mathbf{\Sigma_1} = \mathbf{\Sigma_0}$ is violated, but is still better than the other nonparametric methods. This suggests that if the label shift assumption holds, the likelihood ratio estimate in (IV.5) is a good choice for detecting the change. Detection with the uLSIF procedure improves with training sample size m, as it becomes easier to estimate the likelihood ratio function and variability in the likelihood ratio estimate decreases. CPM also performs better as the sample size increases, as training data is used to construct the detection statistic. While the kNN method makes no assumptions about the change or the distribution of data, the cost of this flexibility is a decrease in detection performance.

$\mathbf{\Sigma_1}$	Post-change distribution	Detection delay when $\mathbb{E}_\infty[T] \approx 500$					
		Classifier CUSUM	Optimal CUSUM	uLSIF CUSUM	CPM (classifier)	CPM (divergence)	kNN
$\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$	$\boldsymbol{\mu}_{\mathbf{0,0}} = [0.5, 0.5]$ $\boldsymbol{\mu}_{\mathbf{0,1}} = [1, 1]$	70.2 (1.68)	24.1 (0.17)	55.8 (4.49)	64.4 (1.27)	66.4 (1.40)	$\geq$ 130 (3.22)
	$\boldsymbol{\mu}_{\mathbf{0,0}} = [0.75, 0.75]$ $\boldsymbol{\mu}_{\mathbf{0,1}} = [0.75, 0.75]$	184 (7.49)	22.9 (0.16)	117 (14.6)	92.1 (1.64)	96.0 (1.78)	$\geq$ 131 (3.30)
	$\boldsymbol{\mu}_{\mathbf{0,0}} = [1, 1]$ $\boldsymbol{\mu}_{\mathbf{0,1}} = [0.5, 0.5]$	541 (17.4)	28.9 (0.21)	282 (3.23)	177 (3.18)	193 (3.55)	$\geq$ 161 (3.57)
$\begin{bmatrix}2 & 0.1\\0.1 & 2\end{bmatrix}$	$\boldsymbol{\mu}_{\mathbf{0,0}} = [0.5, 0.5]$ $\boldsymbol{\mu}_{\mathbf{0,1}} = [1, 1]$	70.9 (1.66)	34.3 (0.27)	67.9 (6.17)	74.6 (1.78)	79.4 (1.92)	$\geq$ 173 (3.60)
	$\boldsymbol{\mu}_{\mathbf{0,0}} = [0.75, 0.75]$ $\boldsymbol{\mu}_{\mathbf{0,1}} = [0.75, 0.75]$	123 (3.82)	31.4 (0.26)	101 (11.5)	109 (2.69)	120 (3.03)	$\geq$ 177 (3.59)
	$\boldsymbol{\mu}_{\mathbf{0,0}} = [1, 1]$ $\boldsymbol{\mu}_{\mathbf{0,1}} = [0.5, 0.5]$	225 (8.56)	30.0 (0.24)	166 (20.4)	196 (5.42)	217 (6.15)	$\geq$ 177 (3.60)
$\begin{bmatrix}4 & 0.5\\0.5 & 4\end{bmatrix}$	$\boldsymbol{\mu}_{\mathbf{0,0}} = [0.5, 0.5]$ $\boldsymbol{\mu}_{\mathbf{0,1}} = [1, 1]$	73.8 (1.67)	29.8 (0.24)	87.6 (10.1)	77.0 (1.98)	88.7 (2.49)	$\geq$ 170 (3.57)
	$\boldsymbol{\mu}_{\mathbf{0,0}} = [0.75, 0.75]$ $\boldsymbol{\mu}_{\mathbf{0,1}} = [0.75, 0.75]$	103 (2.78)	22.9 (0.19)	101 (11.9)	88.9 (2.55)	101 (3.08)	$\geq$ 152 (3.50)
	$\boldsymbol{\mu}_{\mathbf{0,0}} = [1, 1]$ $\boldsymbol{\mu}_{\mathbf{0,1}} = [0.5, 0.5]$	146 (4.58)	18.1 (0.15)	120 (15.3)	107 (3.12)	122 (3.75)	$\geq$ 120 (3.20)

Table IV.3: Simulation results for Scenario 2. Performance of each procedure is measured by detection delay, calculated as $\mathbb{E}_0[T]$. The estimated detection delay from Monte Carlo simulation is reported, with the standard error in parentheses. For the kNN procedure, a window of size 200 is used, so only 200 training points are considered. In the case of kNN, if a change is not detected within the sliding window, windows after time point 200 will consist of only post-change observations, so for computational purposes a fixed number of post-change observations is simulated and we report a lower bound on the detection delay.

Table IV.3 shows the results for Scenario 2, when the label shift assumption is violated. When the label shift assumption is approximately true ($\boldsymbol{\mu}_{\mathbf{0,0}} = [0.5, 0.5]$ and $\boldsymbol{\mu}_{\mathbf{0,1}} = [1, 1]$), we can see that LDA detection is comparable to uLSIF and CPM. However, the LDA procedure is more sensitive to large departures from the label shift assumption, for which methods with fewer assumptions perform better. Overall, CPM with classifier predictions performs well, as the classifier predictions are a useful summary of the data even when label shift doesn't hold.

IV.5 Detecting a change in dengue prevalence

We apply our changepoint detection procedure to the problem of detecting a change in the prevalence of dengue, using data and classifier predictions from the work of Tuan et al. (2015). As the prevalence of dengue changes, but the symptoms are expected to stay the same, the label shift assumption is appropriate for this change.

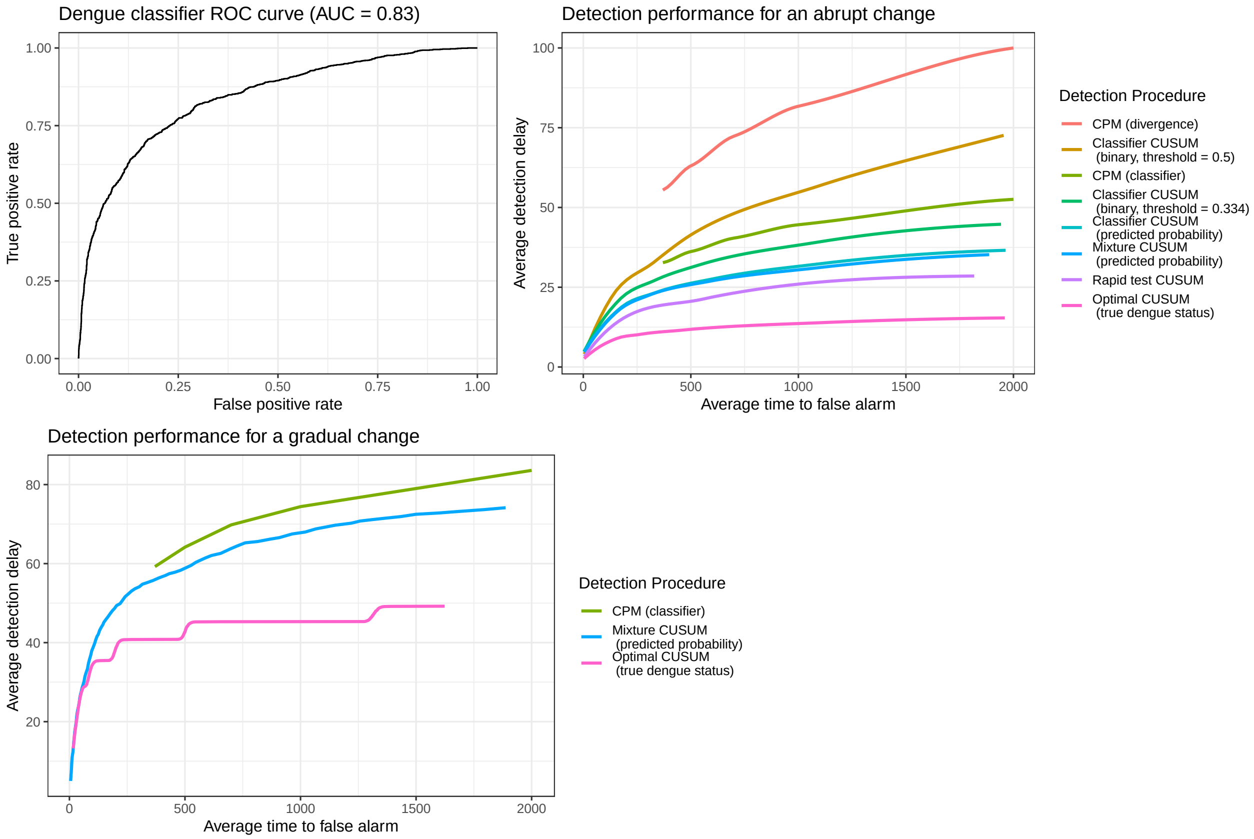

Figure IV.3: Top left: ROC curve for the logistic GAM classifier to predict dengue status, using covariates and data from Tuan et al. (2015). Top right: Comparison of detection performance for CUSUM procedures using different detection procedures, for a change in dengue prevalence from $\pi_\infty = 0.3$ to $\pi_0 = 0.68$. For ease, the method labels for the plot are displayed in descending order of detection delay. Bottom left: Comparison of detection performance when π_0 changes gradually from 0.3 to 0.68.

Data. Data comes from Tuan et al. (2015), who collected information on 5720 febrile patients aged 15 or younger in three Vietnamese hospitals. Of these patients, 30% had dengue. The authors recorded their true dengue status (using a gold-standard test), the results of an NS1 rapid antigen test, and a variety of physical measurements for classification with a logistic regression classifier.

Classifier. We use 1000 patients as training data for the classifier, and save the rest for evaluation our classifier and estimating changepoint detection performance. With the training set, we construct a logistic GAM classifier to predict true dengue status with the following covariates: vomiting (yes/no), skin bleeding (yes/no), BMI, age, temperature, white blood cell count, hematocrit, and platelet count. Figure IV.3 shows the ROC curve for the resulting classifier, which is very close to the ROC curve in Tuan et al. (2015), and has a similar AUC of approximately 0.8.

Scenarios. To assess change detection, we simulate a change in the prevalence of dengue by resampling the 4720 patients not used for training. As the group of patients in the study aims to represent the population of patients who would be tested for dengue, we take the sample proportion of 30% as our baseline dengue prevalence among patients who would be tested. The degree of change in this prevalence, when an outbreak occurs, depends on the magnitude of the outbreak and the baseline prevalence in the population. Magnitude of change varies; for example, Hanoi, Vietnam saw roughly a five-fold increase in 2009 and 2015 (Cuong et al., 2011; Cheng et al., 2020), while Kaohsiung City, Taiwan saw a 15-fold increase in 2014 (Hsu et al., 2017). Baseline prevalence in the full population varies depending on location – for example, Wiwanitkit (2006) shows approximately 1 in 1 million for certain areas of Thailand, whereas Hsu et al. (2017) show roughly 1 in 10000 on average in Taiwan. For Vietnam, Cuong et al. (2011) report roughly 1 in 10000 to 1 in 1000 in Hanoi, with a peak of 384 per 100000 in 2009. For our purposes, we consider two label shift changes in prevalence:

Abrupt change: We simulate an abrupt 5-fold increase, and take the baseline prevalence in the population to be roughly 1 in 10000. Applying Bayes rule, this gives a post-change prevalence of about 68% in our study population, and so we simulate a change from 30% to 68% and assess our ability to detect this shift.

Gradual change: When the change occurs, prevalence increases gradually, rather than abruptly. Here, prevalence in the study population changes smoothly from 30% to 68% over the course of 100 observations.

Methods. We compare the methods from Section IV.4 to detect the change in dengue prevalence. The classifier CUSUM detection procedure is implemented using (IV.5) with $\mathcal{A}(X)$ the predicted probabilities from the dengue classifier described above. We also compare CUSUM with binarized predictions, using both a threshold of 0.5 and the threshold, 0.33, which maximizes sensitivity + specificity. The optimal CUSUM procedure uses the true dengue status, which is observable if gold-standard tests are available, and we also include CUSUM with binary predictions from the NS1 rapid antigen tests, which again may not be available. The rapid test has a specificity of approximately 99% and a sensitivity of 70% Tuan et al. (2015), compared with a specificity and sensitivity of 82% and 70% for the binarized classifier at threshold 0.33. As in Section IV.4, we also compare CPM using the classifier predicted probabilities, and CPM with divergences. uLSIF failed to consistently estimate the likelihood ratio, and so was not considered, while kNN was not considered because it performed worse than the other methods in Section IV.4. Finally, as the post-change parameter is typically unknown, we include the mixing procedure described in (IV.8). We use $\Pi_0 = [0.6, 0.8]$, which corresponds to a 3.5-fold to 9-fold increase in prevalence.

Method	Detection delay for three values of $\mathbb{E}_\infty[T]$		
	$\mathbb{E}_\infty[T] = 500$	$\mathbb{E}_\infty[T] = 700$	$\mathbb{E}_\infty[T] = 1000$
Optimal CUSUM	11.73 (0.06)	12.56 (0.06)	13.62 (0.07)
Rapid test CUSUM	19.46 (0.15)	23.21 (0.20)	24.66 (0.20)
Mixture CUSUM	25.56 (0.68)	27.52 (0.70)	30.04 (0.75)
Classifier CUSUM (predicted probability)	26.28 (0.16)	29.06 (0.17)	31.67 (0.18)
Classifier CUSM (binary, threshold = 0.33)	30.54 (0.22)	33.58 (0.23)	37.54 (0.26)
CPM (classifier)	36.2 (0.28)	40.4 (0.30)	44.6 (0.32)
Classifier CUSUM (binary, threshold = 0.5)	41.22 (0.37)	49.72 (0.45)	56.04 (0.51)
CPM (divergence)	63.0 (0.54)	72.3 (0.60)	81.7 (0.67)
uLSIF CUSUM	1295 (88)	1745 (111)	2305 (141)

Table IV.4: Comparison of method performance for detecting an abrupt change in dengue prevalence. Performance of each procedure is assessed by average detection delay ($\mathbb{E}_0[T]$), calculated at three different values of average run length ($\mathbb{E}_\infty[T]$). The estimated detection delay from Monte Carlo simulation is reported, with the standard error in parentheses.

For the abrupt change scenario, all methods are compared. For the gradual change, we compare the mixture CUSUM procedure to CPM with classifier predictions, as these two methods perform well at detecting an abrupt change and do not require knowledge of the post-change parameter, and we include optimal CUSUM for reference.

Abrupt change. Figure IV.3 and Table IV.4 show the relationship between $\mathbb{E}_\infty[T]$ (average time to false alarm) and $\mathbb{E}_0[T]$ (average detection delay) for each method (uLSIF is not shown in Figure IV.3 because the detection delays are too large). As expected from Corollary 1, the true dengue status and the rapid antigen test give the best detection performance. The predicted probabilities outperform the binarized predictions, as binarization throws away information on the likelihood ratio. The two binarized predictions are close, but the optimal threshold – which maximizes sensitivity + specificity – performs better, as predicted by Corollary 1. Mixture CUSUM and CUSUM with the predicted probabilities perform equally well, likely because all $\pi_0 \in \Pi_0 = [0.6, 0.8]$ provide similar results. While CPM performs worse than CUSUM with predicted probabilities, it still provides a competitive alternative that requires no assumptions on the post-change prevalence. uLSIF has difficulty estimating the likelihood ratio, and performs substantially worse than the other methods.

Gradual change. Figure IV.3 shows the relationship between $\mathbb{E}_\infty[T]$ and $\mathbb{E}_0[T]$ for each method. Detection delays are longer for all methods under gradual change than abrupt change, because the magnitude of change is initially smaller. However, each method can raise an alarm reasonably quickly. This is valuable because real changes in prevalence are expected to be continuous, rather than an abrupt switch from one prevalence to another. While the classic CUSUM procedure, and the nonparametric methods discussed in this chapter, are designed to detect an abrupt change, Figure IV.3 demonstrates that these methods are sensitive to other changes too.

IV.6 Discussion

When a classifier is applied sequentially over time, it is important to detect any change in the distribution of classification data. First, distributional shifts can affect the validity of classifier predictions, and second, a change in distribution may suggest a problem like a disease outbreak. In this chapter, we consider procedures for detecting label shift, which can occur when the prevalence of a disease changes over time, but the symptoms of the disease remain the same.

As we focus on detecting changes in classification data, it is natural to use the classifier predictions in our detection procedure. Here we propose a simple, nonparametric sequential changepoint detection method that uses the classifier predictions to approximate the true likelihood ratio (IV.5). Our procedure requires no additional estimation or training, assuming only that a reasonable value of the post-change prevalence π_0 can be specified. Furthermore, when this post-change parameter is unknown, we combine our nonparametric procedure with Lai's mixture CUSUM approach (Lai, 1998), and mix over the unknown prevalence.

Performance of the detection procedure then depends directly on classifier performance. To demonstrate this, in Section IV.3 we introduce new convergence results for nonparametric sequential detection procedures with likelihood ratio estimates. Through simulations in Section IV.4, we illustrate that our proposed detection procedure outperforms other nonparametric methods when the label shift assumption holds, and still achieves comparable performance when the the label shift assumption is violated. The same holds true when these methods are applied to real dengue classification data in Section IV.5, in which we apply the classifier described in Tuan et al. (2015) to detect a simulated dengue outbreak. First, we see that improved classifier performance results in improved detection performance – if the gold standard dengue test is unavailable, only the NS1 rapid antigen test (which has better specificity than the classifier from Tuan et al. (2015)) outperforms our proposed procedure. Second, other nonparametric procedures respond more slowly to the outbreak, because they leverage less information about a change in prevalence.

Chapter V

Changepoint detection for EEG data

V.1 Introduction

Brain injury is a common cause of death in children, and it is important to monitor patients in the pediatric intensive care unit (PICU) to detect adverse neurological events (Grinspan et al., 2016). This continuous monitoring is performed with electroencephalography (EEG), in which electrodes on a patient's scalp are used to record electrical impulses from their unerlying brain function, and a large body of research exists on using EEG data to detect changes. In particular, many methods have been proposed for detecting epileptic seizures (see, e.g., Qu and Gotman (1997); Saab and Gotman (2005); Mohseni et al. (2006); Schröder and Ombao (2019)). In this chapter, however, we consider a broad class of adverse events characterized by sustained, catastrophic changes, in contrast to transient epileptic seizures. These adverse events, resulting from brain injury, must be detected as soon as possible, so that interventions can be made to save the patient's life.

As raw EEG data is multi-channel, noisy, and complex, techniques for detecting changes often rely on appropriate summaries and transformations which highlight important EEG features. For example, time-frequency transformations are often used to examine the distribution of power in different frequency bands, and changes to this distribution can be flagged (see, e.g., Kannathal et al. (2005); Schröder and Ombao (2019)). Ideally, the EEG features used for change detection are general enough to capture a broad class of adverse events, and are robust to inter-patient variation. In this chapter, we will focus on several general quantitative EEG (qEEG) features, which have had previous success in change identification for damage such as ischemic stroke (Finnigan et al., 2016). We describe a systematic detection procedure based on these features, which implements tools from nonparametric sequential changedpoint detection, and assess its performance on real data in a case study with UPMC ICU patients. The goal of this work is to provide an initial exploration of sequential changepoint detection for catastrophic changes in brain function, to characterize limitations of the initial approach, and to suggest directions for future improvement.

Section V.2 describes our proposed procedure and its connection to previous literature, and Section V.3 describes the result of our case study. An important challenge with automated brain monitoring is low specificity and a tendency to false alarms (Swisher et al., 2015; Goenka et al., 2018), which can lead to over-treatment and alarm fatigue. To assess false alarms with our approach, our case study includes both patients who have experienced catastrophic changes, and those who have not. We discuss the limitations of our initial results in Section V.4, and next research steps to improve detection ability.

V.2 Method and background

V.2.1 Proposed method

Let $X_1, X_2, ... \in \mathbb{R}$ be a sequence of observed values from an EEG summary statistic. For example X_t may measure the asymmetry in power between the left and right hemispheres of the brain, or may measure the relative power in different frequency bands. As there is substantial variability between patients, detecting abnormal brain activity requires a patient-specific pre-change baseline for comparisons. Let t_0 denote the number of baseline observations, so that we assume no adverse event occurs in the window $X_1, X_2, ..., X_{t_0}$.

Let the changepoint ν denote the uknown time at which a change occurs. In the simplest case, as described in Chapter IV, classic changepoint detection assumes that $X_1, ..., X_\nu \overset{iid}{\sim} \mathbb{P}_\infty$ and $X_{\nu+1}, X_{\nu+2}, ... \overset{iid}{\sim} \mathbb{P}_0$. However, due to normal fluctuations in brain behavior over time, the assumption of independence is unlikely to hold. Following previous literature which assumes X_t is a piecewise stationary process (Kaplan et al., 2005), we instead model X_t as $X_t = f_t(X_1, ..., X_{t-1}) + \varepsilon_t$, where f_t is a forecaster at time t and ε_t is a noise term. Letting $\widehat{f}_t$ be the fitted model and $e_t = X_t - \widehat{f}_t(X_1, ..., X_{t-1})$, we perform changepoint detection on the sequence of residuals $e_1, e_2, ...,$ assuming that $e_1, ..., e_\nu \overset{iid}{\sim} \mathbb{P}_\infty$ and $e_{\nu+1}, e_{\nu+2}, ... \overset{iid}{\sim} \mathbb{P}_0$.

The choice of model for the forecaster f_t, and the data used to estimate $\widehat{f}_t$, determines the changes which can be detected. For example, if $\widehat{f}_t(X_1, ..., X_{t-1}) = \frac{1}{t_0} \sum_{i=1}^{t_0} X_i$, then changepoint detection on the residuals e_t reduces to the classic setting which assumes the X_t are independent. More generally, if $\widehat{f}_t$ is fit only on the known baseline $X_1, ..., X_{t_0}$, changepoint detection will be sensitive to any change in the X_t process which is not explained by $\widehat{f}_t$. For example, if $\widehat{f}_t$ is an ARMA(p, q) model, then sustained changes in the mean or variance of the X_t will appear as changes in the distribution of the residuals e_t. In contrast, if $\widehat{f}_t$ is an ARIMA$(p, 1, q)$ model, then a mean shift will not be detectable but a trend may be.

In this chapter, we propose approximating X_t with an ARMA(p, q) model, with coefficients estimated on the known baseline $X_1, ..., X_{t_0}$. The choice of ARMA model is designed to provide sufficient flexibility to allow fluctuations and excursions of the summary measure X_t, while still remaining sensitive to systematic shifts. From previous literature (Finnigan et al., 2016) and conversations with our collaborators, we wish to be sensitive to location shifts in the residuals e_t; for example, brain injury can result in suppression of the overall signal power. We therefore perform changepoint detection on the scaled residuals $e'_t = e_t/s_{t-l+1,t}$, where $s^2_{t-l+1,t}$ denotes the sample variance of $e_{t-l+1}, ..., e_t$, and l is a window size for local variance estimation.

As it is unclear what parametric form EEG data should have, we use nonparametric methods for sequential changepoint detection. We propose using two different nonparametric detection techniques. First, we employ the Change Point Models (CPM) framework (Hawkins et al., 2003; Ross et al., 2011; Ross, 2015), in which repeated hypothesis tests are performed for each new observation e'_t, to test whether a change in distribution has occurred. In the CPM framework, a parameter α is chosen so that the probability of a type I error for each new test is α; the expected time to false alarm (a common operating characteristic in sequential changepoint detection (Tartakovsky et al., 2014)) is then $1/\alpha$, under the assumption that the residuals e'_t are independent and identically distributed before the change. To detect location shift without parametric assumptions, we use Mann-Whitney tests at each step.

In addition, we implement a simple CUSUM procedure, inspired by Volkhonskiy et al. (2017). Similar procedures have been used with EEG data in previous research, such as Gao et al. (2018). In this procedure, the baseline residuals $e'_l, e'_{l+1}, ..., e'_{t_0}$ are used to approximate the pre-change residual distribution. At time t, a p-value for e'_t is calculated with respect to this baseline distribution:

$$p_t = \frac{1}{t_0 - l + 1} \sum_{i=l}^{t_0} \mathbb{1}\{|e'_i| > |e'_t|\}. \tag{V.1}$$

Under the pre-change distribution, we expect that $p_t \approx U[0, 1]$, while the post-change distribution of p_t will be stochastically smaller if a location shift in the residuals occurs. Let g denote a density on $[0, 1]$ which is stochastically smaller than a $U[0, 1]$ – here we use $g(p) \propto \exp\{-p\}$, but other densities can be used as well. We construct a CUSUM statistic W_t defined by

$$W_{t_0} = 0, \qquad W_t = \max\{0, W_{t-1}\} + \log g(p_t). \tag{V.2}$$

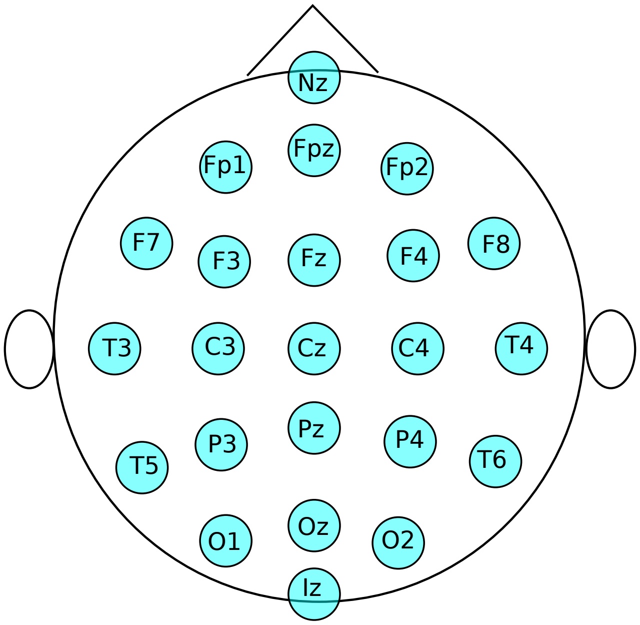

Figure V.1: International 10-20 system of EEG electrode placement.

An advantage of the Mann-Whitney CPM procedure is that under the null hypothesis, the average time to false alarm is $1/\alpha$. This allows the risk of false alarms to be well understood in advance. Furthermore, the CPM framework naturally allows for processing multiple changes, and also estimates the location of a changepoint in addition to the time at which the change was detected. However, updates to the CPM test statistic for a new observation e'_t depend on the previous sequence, making the behavior of the detection statistic challenging to interpret. In contrast, updates to the CUSUM statistic are independent. We find the behavior of the CUSUM statistic *after* a change is detected useful for providing information about the nature of the change, in particular whether it is transitory or persistent.

V.2.2 Background

EEG data. Electroencephalogram (EEG) signals record the input from d different channels, with each channel corresponding to an electrode placed on the patient's scalp, recording a local electrical signal in the brain. Electrodes are commonly placed using the Internation 10-20 System (Klem et al., 1999), as shown in the diagram in Figure V.1. Each electrode measures the voltage of the electrical signal in its respective region. While the underlying system is continuous, measurement occurs in discrete time, though with high frequency.

The raw EEG signal is noisy and multivariate, making analysis challenging. Therefore, quantitative EEG (qEEG) summaries have been proposed that capture important information in the original signal. We will focus on one of these summaries for illustration, the alpha/delta ratio. The alpha/delta ratio is based on an initial Fourier transform, which estimates the distribution of power across frequencies in the signal:

> **Alpha/Delta Ratio:** Practitioners often divide the range of frequencies into five bands: δ (0.5 - 4 Hz), θ (4 - 8 Hz), α (8 - 12 Hz), β (12 - 30 Hz), and γ (30 - 100 Hz), though there is variation in the band definitions in the literature (Newson and Thiagarajan, 2019). The alpha/delta ratio measures the relative amount of power in the α-band compared to the δ-band. A change in alpha/delta ratio has been shown to correlate with adverse events like ischemic stroke (Finnigan et al., 2016).

Detecting changes in EEG data. There is a large body of research that tackles the problem of detecting changes in EEG signals, typically focused on detecting epileptic seizures in EEG data. During an epileptic event, brain signals become much more chaotic as the electrical activity in the brain changes. Much existing work has been done on distinguishing between normal and epileptic EEG signals by leveraging these changes; the variety of approaches to characterizing these changes include (Mohseni et al., 2006) characterizations of dynamical systems (such as with the Maximal Lyapunov exponent, which is used to measure how chaotic a system is, or the Kolmogorov-Sinai entropy) (Hively et al., 1999; Kannathal et al., 2005; Palus et al., 1999); features calculated on the raw EEG signal within a time window (Altunay et al., 2010; Dingle et al., 1993; Gao et al., 2018; Qu and Gotman, 1997); features calculated from wavelet transforms (Saab and Gotman, 2005; Wang et al., 2011); and features calculated from the power spectrum (Brodsky et al., 1999; Chen et al., 2019; Kannathal et al., 2005; Lavielle, 2005; Schröder and Ombao, 2019; Shoeb and Guttag, 2010).

As noted by many authors (e.g., Kaplan et al. (2005)), the EEG is not a stationary process, but is often treated as a piece-wise stationary process, with abrupt changes between stationary segments. This has led researchers to use EEG characterizations for several related goals in detecting changes.

First, some authors describe procedures that are directly compatible with online change detection, in which we wish to raise an alarm as soon as a change occurs. These approaches define a measure of dissimilarity between windows of the EEG signal, such as L_1 and χ^2 distances in phase space (Hively et al., 1999); a conformal p-value based approach (Gao et al., 2018) that applies the conformal martingale approach from Vovk et al. (2003); prediction error from a linear prediction filter (Altunay et al., 2010); and posterior probabilities from a simple Bayesian model (Saab and Gotman, 2005). Other authors have approached seizure detection from a classification perspective: given a segment of EEG, does it come from a normal or epileptic seizure portion of the recording? For example, Kannathal et al. (2005) construct an entropy-based neuro-fuzzy classifier; Wang et al. (2011) use a k-NN classifier on coefficients from a wavelet packet transform;

and Shoeb and Guttag (2010) use an SVM with features from the power spectral density in each window. As these papers focus on classification, it is unclear whether such an implementation can necessarily detect early warning signs for seizures in the EEG, *before* the seizures occur. Finally, in contrast to online detection methods, other research has considered *retrospective* EEG changepoint detection. Here the goal is to segment a fixed, piecewise-stationary time series into its stationary segments. Such methods often rely on a contrast between the two windows immediately before and after a putative change point. For example, Lavielle (2005) contrasts the amount of energy in pre-specified frequency bands before and after the change point. Similarly, Schröder and Ombao (2019) measure the difference between the auto- or cross-spectrum in adjacent windows; and Preuss et al. (2015), while not explicitly applied to EEG data, consider the maximal difference in the spectral density matrix between windows. We note that a direct implementation of segmentation methods cannot be directly applied for the purpose of online change detection, but the features and statistics used in these retrospective procedures could be adapted for online use.

V.3 Case study

In this section, we apply the two sequential detection techniques from Section V.2 to a group of real ICU patients at UPMC Children's Hospital of Pittsburgh (CHP).

Data description. EEG monitoring of ICU patients at UPMC CHP has been performed over the course of several years. For the purposes of this study, we examine 17 patients who visited the CHP ICU. Six of these patients (numbered $1-6$) suffered a catastrophic failure in brain function, while the remaining 11 other patients (numbered $\text{N1}-\text{N11}$) suffered no adverse events. We note that changes may still occur in the EEGs for the 11 patients without adverse events, but any changes are not correlated to adverse events. Each patient was monitored during a portion of their stay, and for patients who suffered a catastrophic decline, the decline occurred during the EEG recording. Recordings for the patients with adverse events are several hours long, while to examine the behavior of EEG data in patients without adverse events, we selected recordings of around 6.5 hours. For patients with adverse events, experts reviewed the full EEG data and identified two times of interest: the first time in the recording at which an EEG change occurs, and the time at which the patient's state is clinically alarming.

Detection settings. For each patient, the first hour of data was used as the baseline for model fitting and changepoint detection. To choose an appropriate number of parameters for the ARMA time series model, we performed model selection on the full log(Alpha/Delta Ratio) time series for each non-adverse patient. These fitted models tend to have 3-6 autoregressive parameters, and 1-4 moving average parameters, so for each patient we fit an ARMA(6, 4) model on the baseline data. For CPM detection, we choose α so

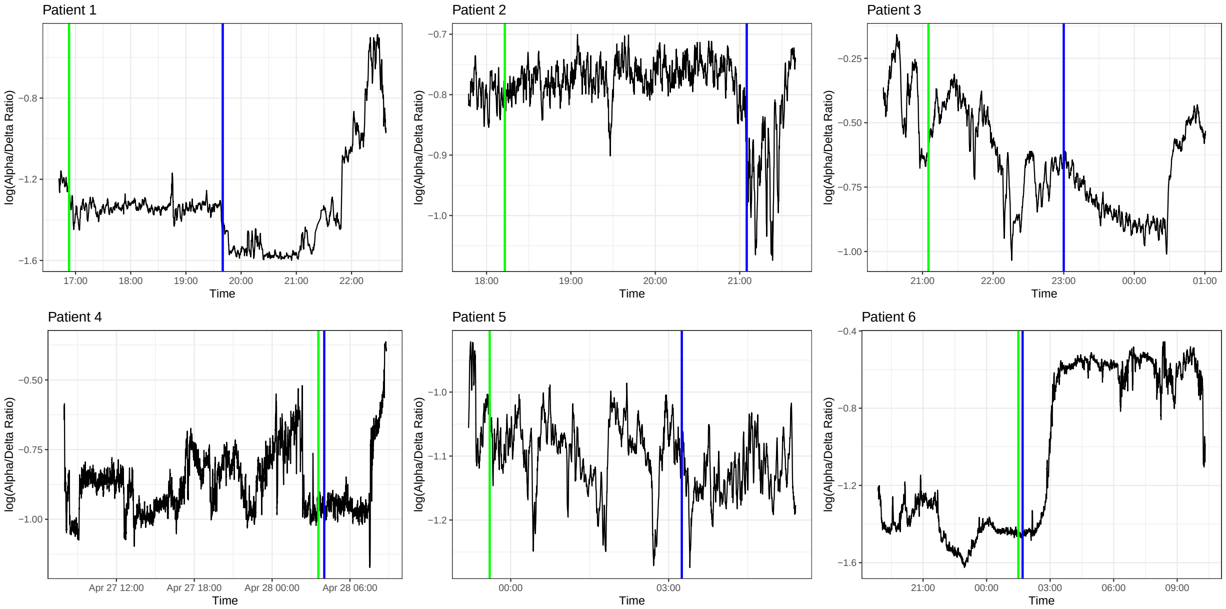

Figure V.2: Alpha/Delta ratio time series for six patients with adverse events. The green vertical lines denote the first EEG change noted in qEEG review, while the blue vertical lines denote clinical alarms.

that if the fitted ARMA model is appropriate and no change occurs, the average time between false alarms is approximately 100 hours. The CPM detection procedure was restarted after each detected change, so multiple changes could potentially be detected.

Initial time series. Figure V.2 shows the alpha/delta ratios for the six patients with adverse events, with the labeled changes noted on the plot. First, we notice that in several patients (1, 2, and 6), a change in the time series is clearly visible at the time of clinical change (blue lines). Moreover, these changes appear to be persistent, with the time series not returning to its previous behavior. In other patients, there is less clearly a definitive change in behavior at the time of clinical change. This may suggest that alpha/delta ratio is not an appropriate summary for each patient, or that changes in the EEG also occur before the clinical change.

For comparison, Figure V.3 shows the alpha/delta ratio for the 11 patients without adverse events. Clearly, the statistic fluctuates over time, which motivates the use of a time series model to capture some of these fluctuations.

Mann-Whitney CPM detection. For each patient, we train an ARMA(6, 4) model on the first hour of data, and then fit the model to the remaining time series. The scaled residuals are used for changepoint detection with the Mann-Whitney CPM method to detect a shift in mean. The results of changepoint detection for the patients with adverse events are shown in Figure V.4. First, we notice that for four patients (1, 2, 3, and 5), the first EEG change occurs during the baseline window. This first EEG change

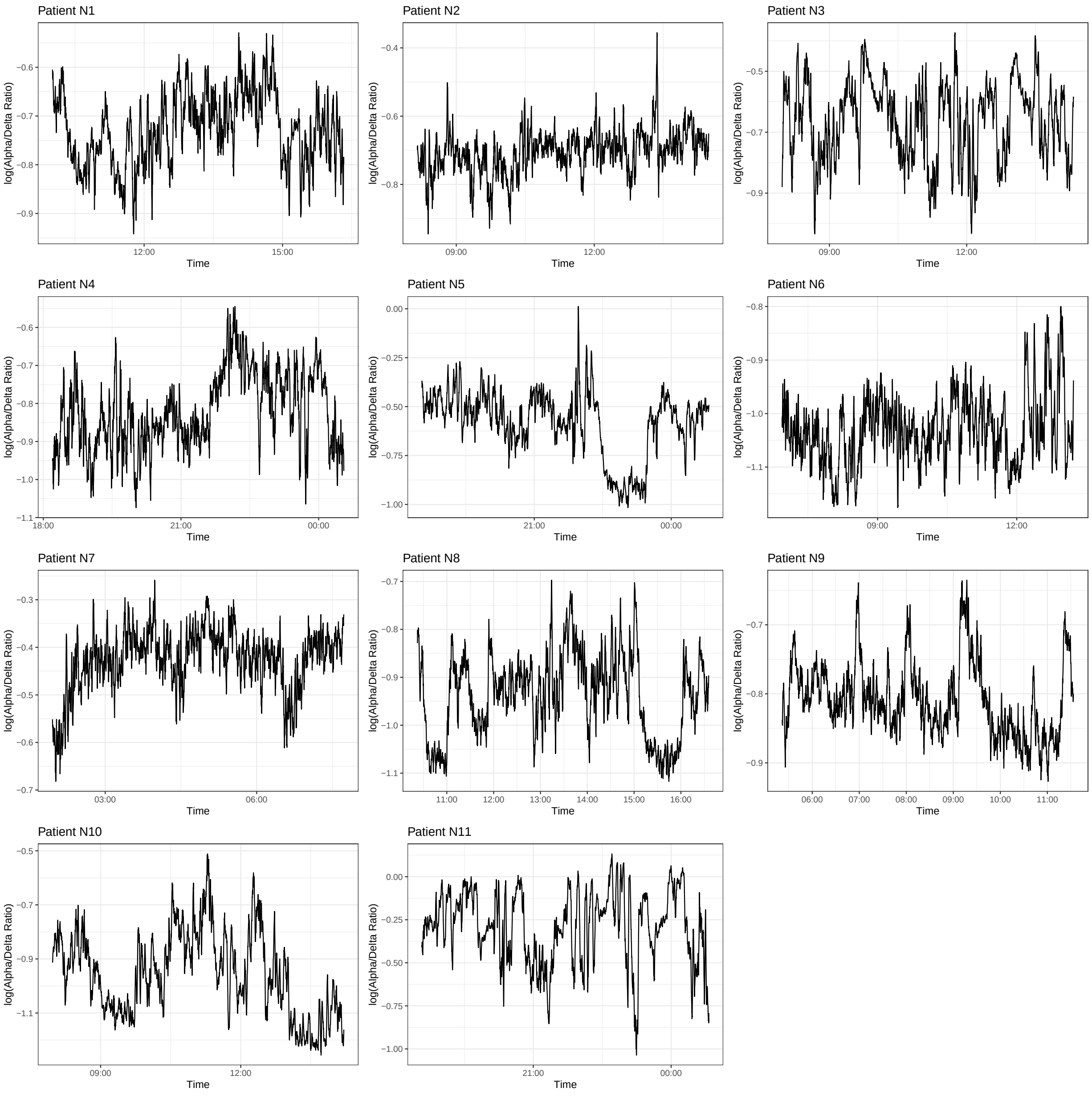

Figure V.3: Alpha/Delta ratio time series for 11 patients without adverse events.

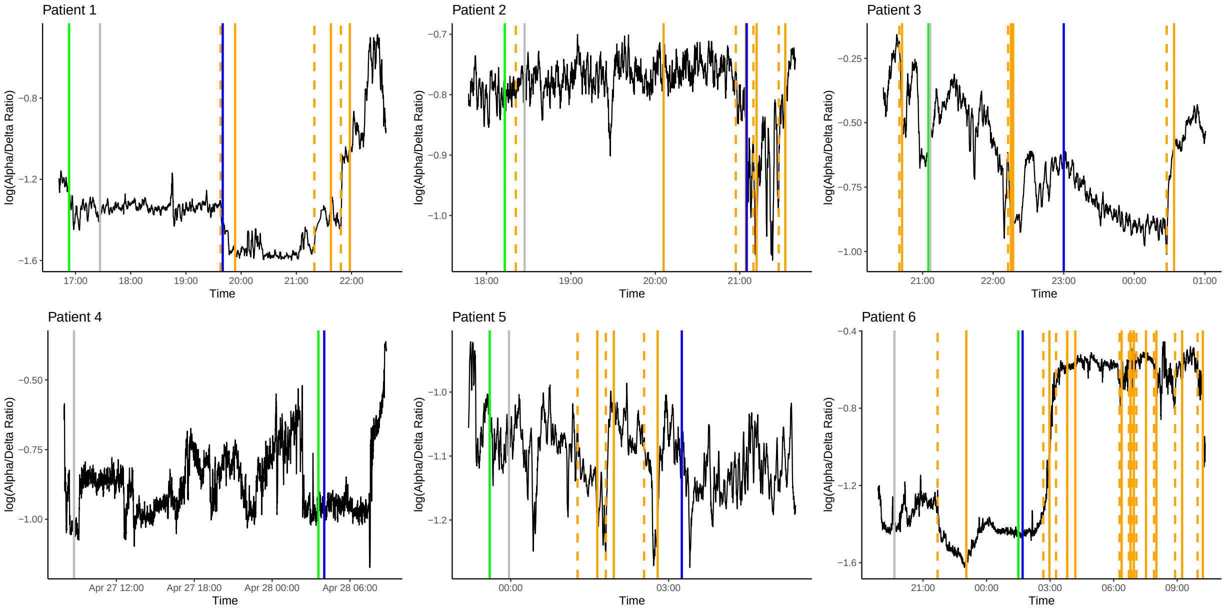

Figure V.4: Mann-Whitney CPM results on alpha/delta ratio for six patients with adverse events. The solid orange lines denote detection times, while the dashed orange lines denote the estimated detection times. The grey vertical lines represent the end of the baseline window. The green vertical lines denote the first EEG change noted in qEEG review, while the blue vertical lines denote clinical alarms.

may or may not be a prelude to the later catastrophic decline, but if so, having the change occur the training period may make it more difficult to respond to future changes. However, a shorter baseline captures less of the natural fluctuation seen in Figure V.3, making it more difficult to model the EEG process.

For five of the six patients in Figure V.4, multiple changes are detected, with changes soon after the clinical changes for patients 1, 2, and 6. For patients 2, 3, and 5, one or more changes is also detected in the period between the first EEG change and the clinical change. No detections are made for patient 4, likely because there is strong fluctuation in the pre-change data.

To assess the rate of detections in patients without adverse events, we perform the same analysis on the other 11 patients, with the results shown in Figure V.5. If no changes occur, and the ARMA fit is appropriate, we expect to see very few detected changes. Patients N1, N3, N7, and N8 show no changes, indicating that the fitted ARMA model provided a reasonable approximation for their behavior. Patients N2, N4, and N9 show a small number of changes; some of these may be the result of real changes unrelated to adverse events, some may be false alarms, and some may simply result from a poor model fit. For patients N5, N6, N10, and N11, many changes are detected, either because the baseline is too short to capture behavior, or the choice of model is unsuitable.

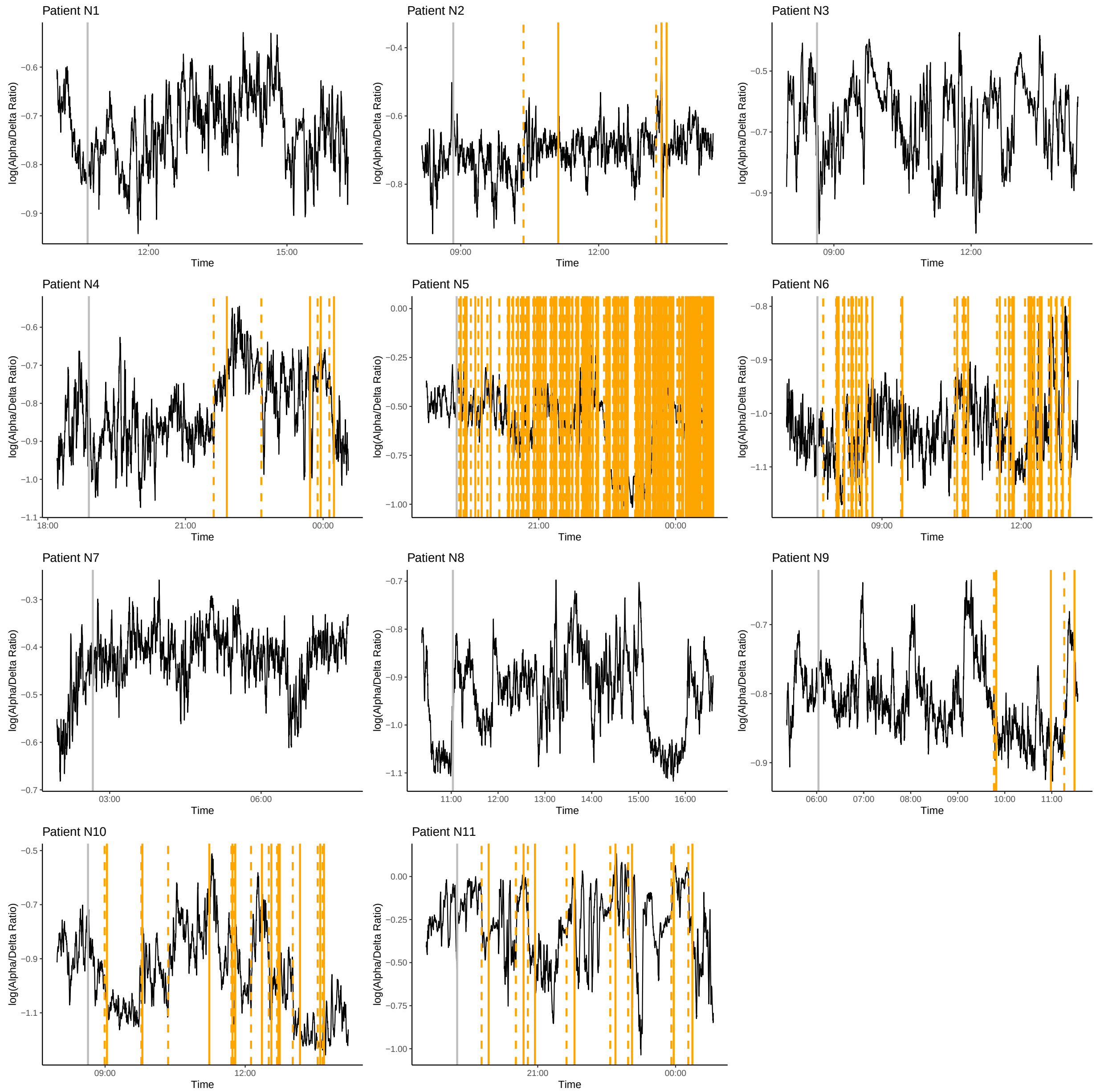
Patient N1
Patient N2
Patient N3
Patient N4
Patient N5
Patient N6
Patient N7
Patient N8
Patient N9
Patient N10
Patient N11

Figure V.5: Mann-Whitney CPM results on alpha/delta ratio for 11 patients without adverse events. The solid orange lines denote detection times, while the dashed orange lines denote the estimated detection times. The grey vertical lines represent the end of the baseline window.

Persistent changes with CUSUM detection. One possibility for attempting to reduce the number of detected changes in non-adverse patients is to fit other time series models, or to update the fitted model over time to account for non-stationarity. Alternatively, it may be possible to find other summaries of the EEG which are stable under normal brain function but show changes under catastrophic decline. However, results for other existing qEEG features are similar to Figure V.4 and Figure V.5, and many changes are still detected even when updating the fitted time series model.

Alternatively, we attempt to learn information about the nature of a detected change. When changes correspond to adverse events, we expect them to persist in brain behavior, whereas changes due to normal fluctuations seem more likely to be transitory. With this in mind, we construct the CUSUM statistic described above, which accumulates over time if changes persist. Figure V.6 shows the CUSUM statistic for the 11 non-adverse patients. For N5, N6, and N10, which had the most changes in Figure V.5, changes from baseline still appear to persist, but for the remaining 8 patients, changes appear more transitory.

Figure V.7 shows the CUSUM analysis on the 6 patients with adverse events. Changes do appear to accumulate for patients 1, 2, and 6 after the clinical change, but not for patients 3, 4, and 5. Examining the behavior of the CUSUM statistic over time may be helpful for distinguishing persistent and transitory changes, but is not a perfect solution.

V.4 Limitations and future work

The purpose of this chapter is to describe initial results from a case study with ICU patients, which will help inform further development of methods for detecting changes in the EEG data. Ultimately, our results demonstrate the challenge of detecting relevant changes in a system that regularly fluctuates, while maintaining some control on the number of detections when no adverse events occur. There are several limitations made apparent by our initial results:

1. Using just one or two qEEG features may be sufficient to visualize the effect of adverse events, but is likely insufficient to distinguish adverse events from normal, benign changes.
2. Capturing medium- to long-term behavior is important, but requires a baseline of sufficient length for model training, and as a reference for changepoint detection.
3. Defining a true changepoint, for reference with detection times, may also be challenging. At what point is a change predictive of a future adverse event?

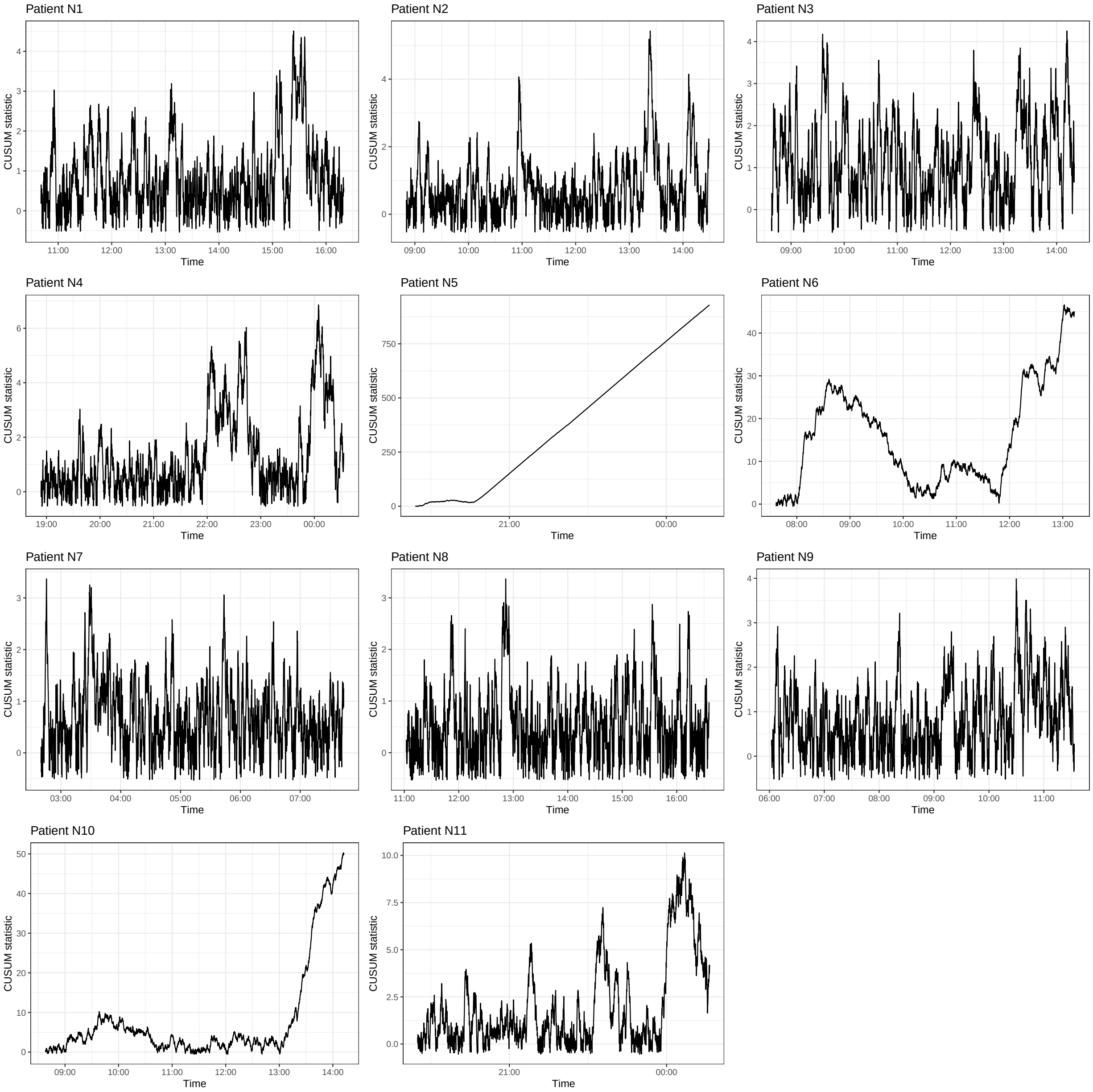

Figure V.6: CUSUM statistic for changepoint detection, for 11 normal patients without adverse events.

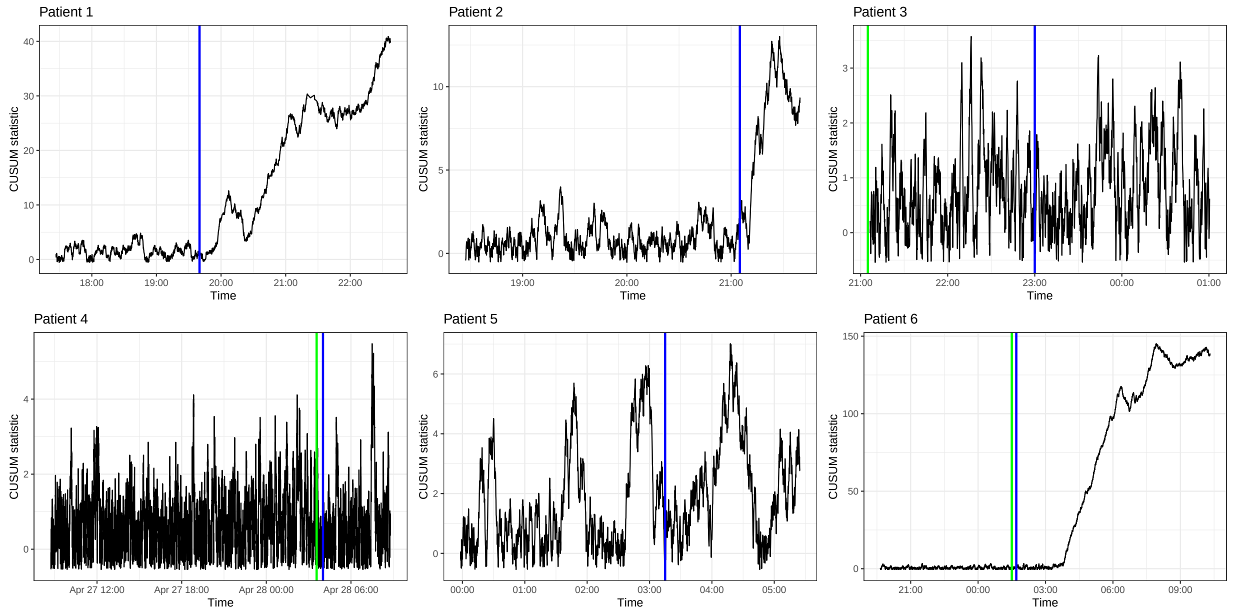

Figure V.7: CUSUM statistic for changepoint detection, for six patients with adverse events. The statistic begins after the baseline, which is approximately 1 hour long. The green vertical lines denote the first EEG change noted in qEEG review, while the blue vertical lines denote clinical alarms. When a green line is not shown, the first EEG change occurs during the baseline.

We plan to address these limitations in continued work. We are currently in the process of collecting further data, which will be recorded over a greater period of time, allowing us to better model pre-change brain behavior. We also plan to analyze a larger group of patients without adverse events. This will help with modeling normal EEG fluctuations, and may help characterize the normal relationship between qEEG features, allowing us to simultaneously use multiple features for changepoint detection. Finally, in the long term we hope to learn how to classify EEG changes, to distinguish between benign fluctuations and adverse events.

Bibliography

Ackerman, S., Dube, P., and Farchi, E. (2020). Sequential drift detection in deep learning classifiers. *arXiv preprint arXiv:2007.16109*. 49, 50, 53, 63

Altunay, S., Telatar, Z., and Erogul, O. (2010). Epileptic eeg detection using the linear prediction error energy. *Expert Systems with Applications*, 37(8):5661–5665. 75

Argyriou, A., Evgeniou, T., and Pontil, M. (2008). Convex multi-task feature learning. *Machine learning*, 73(3):243–272. 10

Argyriou, A., Micchelli, C. A., Pontil, M., and Ying, Y. (2007). A spectral regularization framework for multi-task structure learning. In *NIPS*, volume 1290, page 1296. Citeseer. 10

Axelrod, D. (1981). Cell-substrate contacts illuminated by total internal reflection fluorescence. *The Journal of Cell Biology*, 89(1):141–145. 6

Azizzadenesheli, K., Liu, A., Yang, F., and Anandkumar, A. (2019). Regularized learning for domain adaptation under label shifts. *arXiv preprint arXiv:1903.09734*. 49

Baron, M. I. (2000). Nonparametric adaptive change point estimation and on line detection. *Sequential Analysis*, 19(1-2):1–23. 55

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48. 17

Bekker, J. and Davis, J. (2020). Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4):719–760. 45

Bell, C., Gordon, L., and Pollak, M. (1994). An efficient nonparametric detection scheme and its application to surveillance of a bernoulli process with unknown baseline. *Lecture Notes-Monograph Series*, pages 7–27. 55, 56

Bickel, S., Brückner, M., and Scheffer, T. (2009). Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(9). 9, 55

Bohannon, K. P., Bittner, M. A., Lawrence, D. A., Axelrod, D., and Holz, R. W. (2017). Slow fusion pore expansion creates a unique reaction chamber for co-packaged cargo. *The Journal of general physiology*, 149(10):921–934. 6

Bowman, S. L., Soohoo, A. L., Shiwarski, D. J., Schulz, S., Pradhan, A. A., and Puthenveedu, M. A. (2015). Cell-autonomous regulation of mu-opioid receptor recycling by substance p. *Cell reports*, 10(11):1925–1936. 6

Brodsky, B. E., Darkhovsky, B. S., Kaplan, A. Y., and Shishkin, S. L. (1999). A nonparametric method for the segmentation of the eeg. *Computer methods and programs in biomedicine*, 60(2):93–106. 75

Brodsky, E. and Darkhovsky, B. S. (1993). *Nonparametric methods in change point problems*, volume 243. Springer Science & Business Media. 55

Brodsky, E. and Darkhovsky, B. S. (2000). *Non-parametric statistical diagnosis: problems and methods*, volume 509. Springer Science & Business Media. 55

Caicedo, J. C., Goodman, A., Karhohs, K. W., Cimini, B. A., Ackerman, J., Haghighi, M., Heng, C., Becker, T., Doan, M., McQuin, C., et al. (2019). Nucleus segmentation across imaging experiments: the 2018 data science bowl. *Nature methods*, 16(12):1247–1253. 5, 9

Chen, G., Lu, G., Shang, W., and Xie, Z. (2019). Automated change-point detection of eeg signals based on structural time-series analysis. *IEEE Access*, 7:180168–180180. 75

Chen, H. (2019). Sequential change-point detection based on nearest neighbors. *The Annals of Statistics*, 47(3):1381–1407. 55, 63

Chen, H. and Chu, L. (2019). *gStream: Graph-Based Sequential Change-Point Detection for Streaming Data*. R package version 0.2.0. 64

Cheng, J., Bambrick, H., Yakob, L., Devine, G., Frentiu, F. D., Thai, P. Q., Xu, Z., Hu, W., et al. (2020). Heatwaves and dengue outbreaks in hanoi, vietnam: New evidence on early warning. *PLoS neglected tropical diseases*, 14(1):e0007997. 68

Chicco, D. (2017). Ten quick tips for machine learning in computational biology. *BioData mining*, 10(1):1–17. 27

Christiansen, E. M., Yang, S. J., Ando, D. M., Javaherian, A., Skibinski, G., Lipnick, S., Mount, E., ONeil, A., Shah, K., Lee, A. K., et al. (2018). In silico labeling: predicting fluorescent labels in unlabeled images. *Cell*, 173(3):792–803. 5, 9

Chu, L. and Chen, H. (2018). Sequential change-point detection for high-dimensional and non-euclidean data. *arXiv preprint arXiv:1810.05973*. 55, 64

Cipolli, W. and Hanson, T. (2017). Computationally tractable approximate and smoothed polya trees. *Statistics and Computing*, 27(1):39–51. 62

Cipolli, W. and Hanson, T. (2019). Supervised learning via smoothed polya trees. *Advances in Data Analysis and Classification*, 13(4):877–904. 62

Cuong, H. Q., Hien, N. T., Duong, T. N., Phong, T. V., Cam, N. N., Farrar, J., Nam, V. S., Thai, K. T., and Horby, P. (2011). Quantifying the emergence of dengue in hanoi, vietnam: 1998–2009. *PLoS Negl Trop Dis*, 5(9):e1322. 68

DiCiccio, T. J., Efron, B., et al. (1996). Bootstrap confidence intervals. *Statistical science*, 11(3):189–228. 12

Dingle, A. A., Jones, R. D., Carroll, G. J., and Fright, W. R. (1993). A multistage system to detect epileptiform activity in the eeg. *IEEE Transactions on Biomedical Engineering*, 40(12):1260–1268. 75

Finnigan, S., Wong, A., and Read, S. (2016). Defining abnormal slow eeg activity in acute ischaemic stroke: Delta/alpha ratio as an optimal qeeg index. *Clinical Neurophysiology*, 127(2):1452–1459. 71, 73, 75

Gao, Z., Lu, G., Yan, P., Lyu, C., Li, X., Shang, W., Xie, Z., and Zhang, W. (2018). Automatic change detection for real-time monitoring of eeg signals. *Frontiers in physiology*, 9:325. 73, 75

Garg, A., Garg, J., Rao, Y., Upadhyay, G., and Sakhuja, S. (2011). Prevalence of dengue among clinically suspected febrile episodes at a teaching hospital in north india. *Journal of Infectious Diseases and Immunity*, 3(5):85–89. 48

Garg, S., Wu, Y., Balakrishnan, S., and Lipton, Z. C. (2020). A unified view of label shift estimation. *arXiv preprint arXiv:2003.07554*. 10

Giné, E. and Guillou, A. (2002). Rates of strong uniform consistency for multivariate kernel density estimators. In *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, volume 38, pages 907–921. Elsevier. 62

Goenka, A., Boro, A., and Yozawitz, E. (2018). Comparative sensitivity of quantitative eeg (qeeg) spectrograms for detecting seizure subtypes. *Seizure*, 55:70–75. 72

Gordon, L. and Pollak, M. (1994). An efficient sequential nonparametric scheme for detecting a change of distribution. *The Annals of Statistics*, pages 763–804. 55

Gordon, L. and Pollak, M. (1995). A robust surveillance scheme for stochastically ordered alternatives. *The Annals of Statistics*, pages 1350–1375. 55

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. (2009). Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5. 9, 53, 55, 59, 63

Grinspan, Z. M., Eldar, Y. C., Gopher, D., Gottlieb, A., Lammfromm, R., Mangat, H. S., Peleg, N., Pon, S., Rozenberg, I., Schiff, N. D., et al. (2016). Guiding principles for a pediatric neurology icu (neuropicu) bedside multimodal monitor: findings from an international working group. *Applied clinical informatics*, 7(2):380. 71

Han, W. and Atkinson, K. E. (2009). *Theoretical Numerical Analysis: A Functional Analysis Framework*. Springer. 109, 110, 111

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized additive models*, volume 43. CRC press. 21, 99

Hawkins, D. M., Qiu, P., and Kang, C. W. (2003). The changepoint model for statistical process control. *Journal of quality technology*, 35(4):355–366. 73

Hively, L., Gailey, P., and Protopopescu, V. (1999). Detecting dynamical change in nonlinear time series. *Physics Letters A*, 258(2-3):103–114. 75

Hsu, J. C., Hsieh, C.-L., and Lu, C. Y. (2017). Trend and geographic analysis of the prevalence of dengue in taiwan, 2010–2015. *International Journal of Infectious Diseases*, 54:43–49. 48, 68

Kanamori, T., Hido, S., and Sugiyama, M. (2009). A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(Jul):1391–1445. 53, 55, 59, 63

Kanamori, T., Suzuki, T., and Sugiyama, M. (2012). Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86(3):335–367. 63

Kannathal, N., Choo, M. L., Acharya, U. R., and Sadasivan, P. (2005). Entropies for detection of epilepsy in eeg. *Computer methods and programs in biomedicine*, 80(3):187–194. 71, 75

Kaplan, A. Y., Fingelkurts, A. A., Fingelkurts, A. A., Borisov, S. V., and Darkhovsky, B. S. (2005). Nonstationary nature of the brain activity as revealed by eeg/meg: methodological, practical and conceptual challenges. *Signal processing*, 85(11):2190–2212. 72, 75

Kass, R. E., Caffo, B. S., Davidian, M., Meng, X.-L., Yu, B., and Reid, N. (2016). Ten simple rules for effective statistical practice. 27

Kawahara, Y. and Sugiyama, M. (2009). Change-point detection in time-series data by direct density-ratio estimation. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 389–400. SIAM. 55

Klem, G. H., Lüders, H. O., Jasper, H., and Elger, C. (1999). The ten-twenty electrode system of the international federation. *Electroencephalogr Clin Neurophysiol*, 52(3):3–6. 74

Kou, Z. W., Mo, J. L., Wu, K. W., Qiu, M. H., Huang, Y. L., Tao, F., Lei, Y., Lv, L. L., and Sun, F. Y. (2019). Vascular endothelial growth factor increases the function of calcium-impermeable AMPA receptor GluA2 subunit in astrocytes via activation of protein kinase C signaling pathway. *Glia*, 67(7):1344–1358. 6

Kraus, O. Z., Grys, B. T., Ba, J., Chong, Y., Frey, B. J., Boone, C., and Andrews, B. J. (2017). Automated analysis of high-content microscopy data with deep learning. *Molecular systems biology*, 13(4):924. 9

Krivobokova, T., Kneib, T., and Claeskens, G. (2010). Simultaneous confidence bands for penalized spline estimators. *Journal of the American Statistical Association*, 105(490):852–863. 12, 99

Kuang, K., Cui, P., Athey, S., Xiong, R., and Li, B. (2018). Stable prediction across unknown environments. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1617–1626. 10, 13, 25

Lai, T. L. (1998). Information bounds and quick detection of parameter changes in stochastic systems. *IEEE Transactions on Information Theory*, 44(7):2917–2929. 52, 70

Lavielle, M. (2005). Using penalized contrasts for the change-point problem. *Signal processing*, 85(8):1501–1510. 75, 76

Lipton, Z. C., Wang, Y.-X., and Smola, A. (2018). Detecting and correcting for label shift with black box predictors. *arXiv preprint arXiv:1802.03916*. 10, 21, 23, 47, 48, 49, 53, 97, 98, 103

Liu, S., Yamada, M., Collier, N., and Sugiyama, M. (2013). Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83. 55

Logan, T., Bendor, J., Toupin, C., Thorn, K., and Edwards, R. H. (2017). α-Synuclein promotes dilation of the exocytotic fusion pore. *Nature Neuroscience*, 20(5):681–689. 6

Lorden, G. (1971). Procedures for reacting to a change in distribution. *The Annals of Mathematical Statistics*, 42(6):1897–1908. 51, 55, 64

Madrid Padilla, O. H., Athey, A., Reinhart, A., and Scott, J. G. (2019). Sequential nonparametric tests for a change in distribution: an application to detecting radiological anomalies. *Journal of the American Statistical Association*, 114(526):514–528. 53

Magliacane, S., van Ommen, T., Claassen, T., Bongers, S., Versteeg, P., and Mooij, J. M. (2017). Domain adaptation by using causal inference to predict invariant conditional distributions. *arXiv preprint arXiv:1707.06422*. 10, 25

Makin, T. R. and de Xivry, J.-J. O. (2019). Science forum: Ten common statistical mistakes to watch out for when writing or reviewing a manuscript. *Elife*, 8:e48175. 27

Makiyama, K. (2019). *densratio: Density Ratio Estimation*. R package version 0.2.1. 63

Maretto, R. V., Fonseca, L. M., Jacobs, N., Körting, T. S., Bendini, H. N., and Parente, L. L. (2020). Spatio-temporal deep learning approach to map deforestation in amazon rainforest. *IEEE Geoscience and Remote Sensing Letters*. 9

McDonald, D. (1990). A CUSUM procedure based on sequential ranks. *Naval Research Logistics*, 37(5):627–646. 55

Mohseni, H. R., Maghsoudi, A., and Shamsollahi, M. B. (2006). Seizure detection in eeg signals: A comparison of different approaches. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6724–6727. IEEE. 71, 75

Moustakides, G. V. (1986). Optimal stopping times for detecting changes in distributions. *The Annals of Statistics*, 14(4):1379–1387. 55

Moustakides, G. V., Polunchenko, A. S., and Tartakovsky, A. G. (2011). A numerical approach to performance analysis of quickest change-point detection procedures. *Statistica Sinica*, pages 571–596. 110

Newson, J. J. and Thiagarajan, T. C. (2019). Eeg frequency bands in psychiatric disorders: a review of resting state studies. *Frontiers in human neuroscience*, 12:521. 75

Nguyen, X., Wainwright, M. J., and Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861. 55, 58, 59

Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., and Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25):E5716–E5725. 5, 8, 9

Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1/2):100–115. 63

Palus, M., Komarek, V., Hrncir, Z., and Prochazka, T. (1999). Is nonlinearity relevant for detecting changes in eeg? 75

Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359. 9

Peters, J., Bühlmann, P., and Meinshausen, N. (2016). Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 947–1012. 10, 13, 25

Pippig, S., Andexinger, S., and Lohse, M. J. (1995). Sequestration and recycling of beta 2-adrenergic receptors permit receptor resensitization. *Molecular Pharmacology*, 47(4):666–676. 6

Polunchenko, A. S. and Tartakovsky, A. G. (2012). State-of-the-art in sequential change-point detection. *Methodology and computing in applied probability*, 14(3):649–684. 50

Preuss, P., Puchstein, R., and Dette, H. (2015). Detection of multiple structural breaks in multivariate time series. *Journal of the American Statistical Association*, 110(510):654–668. 76

Qu, H. and Gotman, J. (1997). A patient-specific algorithm for the detection of seizure onset in long-term eeg monitoring: possible use as a warning device. *IEEE transactions on biomedical engineering*, 44(2):115–122. 71, 75

Quiñonero-Candela, J., Sugiyama, M., Lawrence, N. D., and Schwaighofer, A. (2009). *Dataset shift in machine learning.* Mit Press. 9

Rabanser, S., Günnemann, S., and Lipton, Z. (2019). Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems 32.* 49, 53

Rammer, W. and Seidl, R. (2019). Harnessing deep learning in ecology: An example predicting bark beetle outbreaks. *Frontiers in plant science*, 10:1327. 9

Ramsay, J. and Silverman, B. (2005). *Functional data analysis.* Springer. 105

Rappoport, J. Z., Taha, B. W., Lemeer, S., Benmerah, A., and Simon, S. M. (2003). The AP-2 Complex Is Excluded from the Dynamic Population of Plasma Membrane-associated Clathrin. *Journal of Biological Chemistry*, 278(48):47357–47360. 6

Reynolds Jr, M. R. and Stoumbos, Z. G. (1999). A cusum chart for monitoring a proportion when inspecting continuously. *Journal of quality technology*, 31(1):87–108. 60

Rojas-Carulla, M., Schölkopf, B., Turner, R., and Peters, J. (2018). Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342. 10, 25

Ross, G. J. (2015). Parametric and nonparametric sequential change detection in R: The cpm package. *Journal of Statistical Software*, 66(3):1–20. 53, 63, 73

Ross, G. J. and Adams, N. M. (2012). Two nonparametric control charts for detecting arbitrary distribution changes. *Journal of Quality Technology*, 44(2):102–116. 53

Ross, G. J., Tasoulis, D. K., and Adams, N. M. (2011). Nonparametric monitoring of data streams for changes in location and scale. *Technometrics*, 53(4):379–389. 63, 73

Saab, M. and Gotman, J. (2005). A system to detect the onset of epileptic seizures in scalp eeg. *Clinical Neurophysiology*, 116(2):427–442. 71, 75

Saerens, M., Latinne, P., and Decaestecker, C. (2002). Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41. 10, 21, 23, 49, 53, 98

Sankaranarayanan, S., De Angelis, D., Rothman, J. E., and Ryan, T. a. (2000). The Use of pHluorins for Optical Measurements of Presynaptic Activity. *Biophysical Journal*, 79(4):2199–2208. 6

Schröder, A. L. and Ombao, H. (2019). Fresped: Frequency-specific change-point detection in epileptic seizure multi-channel eeg data. *Journal of the American Statistical Association*, 114(525):115–128. 71, 75, 76

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244. 9

Shoeb, A. H. and Guttag, J. V. (2010). Application of machine learning to epileptic seizure detection. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 975–982. 75, 76

Siegmund, D. and Venkatraman, E. (1995). Using the generalized likelihood ratio statistic for sequential detection of a change-point. *The Annals of Statistics*, pages 255–271. 52

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.21.2. 17

Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *The annals of statistics*, pages 1040–1053. 58

Storkey, A. (2009). When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, pages 3–28. 10, 49

Subbaswamy, A., Schulam, P., and Saria, S. (2019). Preventing failures due to dataset shift: Learning predictive models that transport. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3118–3127. PMLR. 9, 10, 25

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M. (2008). Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746. 9, 55, 63

Swanson, A., Kosmala, M., Lintott, C., Simpson, R., Smith, A., and Packer, C. (2015). Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna. *Scientific data*, 2(1):1–14. 9

Swisher, C. B., White, C. R., Mace, B. E., Dombrowski, K. E., Husain, A. M., Kolls, B. J., Radtke, R. R., Tran, T. T., and Sinha, S. R. (2015). Diagnostic accuracy of electrographic seizure detection by neurophysiologists and non-neurophysiologists in the adult icu using a panel of quantitative eeg trends. *Journal of Clinical Neurophysiology*, 32(4):324–330. 72

Tartakovsky, A., Nikiforov, I., and Basseville, M. (2014). *Sequential analysis: Hypothesis testing and changepoint detection*. Chapman and Hall/CRC. 52, 56, 73

Tartakovsky, A. G., Pollak, M., and Polunchenko, A. S. (2012a). Third-order asymptotic optimality of the generalized shiryaev–roberts changepoint detection procedures. *Theory of Probability & Its Applications*, 56(3):457–484. 55

Tartakovsky, A. G., Polunchenko, A. S., and Moustakides, G. V. (2009). Design and comparison of shiryaev–roberts-and cusum-type change-point detection procedures. In *Proceedings of the 2nd International Workshop in Sequential Methodologies*. 50, 109

Tartakovsky, A. G., Polunchenko, A. S., and Sokolov, G. (2012b). Efficient computer network anomaly detection by changepoint detection methods. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):4–11. 55, 61

Tartakovsky, A. G., Rozovskii, B. L., Blažek, R. B., and Kim, H. (2006a). Detection of intrusions in information systems by sequential change-point methods. *Statistical methodology*, 3(3):252–293. 55

Tartakovsky, A. G., Rozovskii, B. L., Blazek, R. B., and Kim, H. (2006b). A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. *IEEE transactions on signal processing*, 54(9):3372–3382. 55

Tuan, N. M., Nhan, H. T., Chau, N. V. V., Hung, N. T., Tuan, H. M., Van Tram, T., Le Da Ha, N., Loi, P., Quang, H. K., Kien, D. T. H., et al. (2015). Sensitivity and specificity of a novel classifier for the early diagnosis of dengue. *PLoS Negl Trop Dis*, 9(4):e0003638. 47, 48, 63, 66, 67, 68, 70

Unnikrishnan, J., Veeravalli, V. V., and Meyn, S. P. (2011). Minimax robust quickest change detection. *IEEE Transactions on Information Theory*, 57(3):1604–1614. 52, 56

Volkhonskiy, D., Burnaev, E., Nouretdinov, I., Gammerman, A., and Vovk, V. (2017). Inductive conformal martingales for change-point detection. In *Proceedings of the Sixth Workshop on Conformal and Probabilistic Prediction and Applications*, volume 60 of *Proceedings of Machine Learning Research*, pages 132–153. 73

Vovk, V., Nouretdinov, I., and Gammerman, A. (2003). Testing exchangeability on-line. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 768–775. 75

Wang, D., Miao, D., and Xie, C. (2011). Best basis-based wavelet packet entropy feature extraction and hierarchical eeg classification for epileptic detection. *Expert Systems with Applications*, 38(11):14314–14320. 75

Way, G. P., Kost-Alimova, M., Shibue, T., Harrington, W. F., Gill, S., Piccioni, F., Becker, T., Shafqat-Abbasi, H., Hahn, W. C., Carpenter, A. E., et al. (2021). Predicting cell health phenotypes using image-based morphology profiling. *Molecular Biology of the Cell*, 32(9):995–1005. 9

WHO (2020). Dengue and severe dengue. 47

Wiwanitkit, V. (2006). An observation on correlation between rainfall and the prevalence of clinical cases of dengue in thailand. *Journal of vector borne diseases*, 43(2):73. 48, 68

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)*, 73(1):3–36. 17

Wood, S. N. (2017). *Generalized additive models: an introduction with R*. CRC press. 12, 21, 99

Yu, S. S., Lefkowitz, R. J., and Hausdorff, W. P. (1993). Beta-adrenergic receptor sequestration. A potential mechanism of receptor resensitization. *Journal of Biological Chemistry*, 268(1):337–341. 6

Yu, Y. and Szepesvári, C. (2012). Analysis of kernel mean matching under covariate shift. *arXiv preprint arXiv:1206.4650*. 55

Yudowski, G. A., Puthenveedu, M. A., and von Zastrow, M. (2006). Distinct modes of regulated receptor insertion to the somatodendritic plasma membrane. *Nature neuroscience*, 9(5):622–627. 6

Appendix

Appendix A

Inference with classifier predictions

A.1 Label shift estimation

Let $\{(Z_i', Y_i', C_i')\}_{i=1}^m$ denote a set of labeled training data, and $\{(Z_j, C_j)\}_{j=1}^n$ a set of unlabeled test data with unobserved labels $Y_j \in \{0, 1\}$. Let $\mathcal{A}$ denote a classifier fit on the training data, with $\mathcal{A}(z) = \widehat{P}(Y_i' = 1|Z_i' = z)$. Under the label shift assumption, $P(Y = 1|Z = z, C = c) \neq P(Y' = 1|Z' = z, C' = c')$, and classifier predictions $\mathcal{A}(z)$ must be corrected to $\mathcal{A}_L(z, c)$ by equation (II.2).

This requires estimating the prevalence $P(Y = 1|C = c)$ for each $c \in \mathcal{C}$, which is challenging when labels are unobserved. Fortunately, the test prevalence $P(Y = 1|C = c)$ can be estimated under the label shift assumption. Here we summarize two different approaches, which in our experience are simple but generally reliable for label shift estimation.

A.1.1 Discretization method

The discretization approach is due to Lipton et al. (2018), and we summarize the method in Algorithm 1. Their approach relies on discretizing the predicted probabilities $\mathcal{A}(Z)$, and recognizing that under label shift $\mathcal{A}(Z) \perp\!\!\!\perp C|Y$.

Algorithm 1: Discretization method for label shift prevalence estimation (Lipton et al., 2018)

```
Data: Labeled training data $\{(Z'_i, Y'_i, C'_i)\}_{i=1}^m$
      Unlabeled test data $\{(Z_j, C_j)\}_{j=1}^n$
      Classifier $\mathcal{A}$ with $\mathcal{A}(z) = \widehat{P}(Y'_i = 1 | Z'_i = z)$
Input  : Discretization threshold $h \in (0,1)$
Output: Estimated prevalence $\widehat{P}(Y = 1 | C = c)$ for each $c \in \mathcal{C}$
init
    Calculate discretized predictions on training data: $\widehat{Y}'_i = \mathbb{1}\{\mathcal{A}(Z'_i) > h\}$ ;
    Calculate discretized predictions on test data: $\widehat{Y}_j = \mathbb{1}\{\mathcal{A}(Z_j) > h\}$ ;
for $c \in \mathcal{C}$ do
    $\boldsymbol{\pi}_{train} = \left[\frac{1}{m}\sum_{i=1}^m (1 - Y'_i),\ \frac{1}{m}\sum_{i=1}^m Y'_i\right]^T$ ;
    $\mathbf{M} \in \mathbb{R}^{2\times 2}$, with $\mathbf{M}_{ij} = \frac{1}{m}\sum_{k=1}^m \mathbb{1}\{\widehat{Y}'_k = i-1, Y'_k = j-1\}$, for $i, j \in \{1, 2\}$ ;
    $\widehat{\boldsymbol{\pi}}_{test} = \left[\frac{1}{\#\{i:C_i=c\}}\sum_{j=1}^n (1 - \widehat{Y}_j)\mathbb{1}\{C_j = c\},\ \frac{1}{\#\{i:C_i=c\}}\sum_{j=1}^n \widehat{Y}_j \mathbb{1}\{C_j = c\}\right]^T$ ;
    $\widehat{P}(Y = 1 | C = c) = (\mathbf{M}^{-1}\widehat{\boldsymbol{\pi}}_{test})[2] \cdot \boldsymbol{\pi}_{train}[2]$, where $\boldsymbol{v}[2]$ denotes the second element of vector $\boldsymbol{v}$ ;
end
return $\widehat{P}(Y = 1 | C = c)$ for each $c \in \mathcal{C}$
```

A.1.2 Fixed point method

The discretization method converts predicted probabilities in $(0, 1)$ to binary predictions by thresholding. This requires specifying a threshold, and the choice of threshold may impact the resulting prevalence estimates, particularly if the label shift assumption holds only approximately or there are few observations from one class. An alternative is to consider the label shift corrected probabilities $\mathcal{A}_L(z, c)$, calculated from $\mathcal{A}(z)$ via Bayes theorem (II.2).

Let $\pi_{test,c}$ be a putative value for $P(Y = 1|C = c)$, and $\mathcal{A}_L^{\pi_{test,c}}(z, c)$ the corrected probabilities using $\pi_{test,c}$ in (II.2). If $\mathcal{A}$ is a calibrated classifier on the training data, and $\pi_{test,c}$ is close to $P(Y = 1|C = c)$, then under the label shift assumption $\pi_{test,c} \approx \frac{1}{\#\{i:C_i=c\}} \sum_{j=1}^n \mathcal{A}_L^{\pi_{test,c}}(Z_j, C_j)\mathbb{1}\{C_j = c\}$. The fixed point method considers a range $[a, b] \subset (0, 1)$ of potential values for $P(Y = 1|C = c)$, and estimates prevalence by

$$\widehat{P}(Y = 1|C = c) = \underset{\pi_{test,c} \in [a,b]}{\arg\min} \left| \frac{1}{\#\{i : C_i = c\}} \sum_{j=1}^n \mathcal{A}_L^{\pi_{test,c}}(Z_j, C_j)\mathbb{1}\{C_j = c\} - \pi_{test,c} \right|. \tag{A.1}$$

(The restricted range $[a, b]$ is required to avoid trivial solutions). This is essentially the approached proposed by Saerens et al. (2002), just implemented as a search rather than through iterated EM updates.

A.2 Posterior sampling for logistic GAMs

The inference procedure described in Section II.3 requires the ability to sample sample a new classifier function $\mathcal{A}^*$ at each bootstrap step. As bootstrapping the training data and refitting the classifier at each step is potentially very computationally expensive, it may be necessary to use a classifier for which the variability is understood. In particular, we want to resample the full classification function at every step, not just characterize variability at a single point (this requirement is sufficient to construct global confidence bands for the probability function, not just pointwise confidence intervals). As an example, we show here how standard results for logistic GAMs can be incorporated into the bootstrap procedure described above.

Letting $Z_1, ..., Z_d$ denote the components of Z, the logistic GAM models $P(Y = 1|Z)$ by

$$\text{logit } P(Y = 1|Z) = f_1(Z_1) + \cdots + f_d(Z_d), \tag{A.2}$$

where $f_1, ..., f_d$ are smooth functions (Hastie and Tibshirani, 1990; Wood, 2017). The model is estimated with a spline fit. Let $\overline{\mathbf{Z}}$ be the design matrix for the spline fit (capturing the intercept and Z), let $\boldsymbol{\beta}$ be the spline coefficients, and λ the smoothing parameter. The spline fit is penalized by $\lambda\boldsymbol{\beta}^T\mathbf{S}\boldsymbol{\beta}$ where λ is the smoothing parameter and $\mathbf{S}$ is the spline smoothing matrix; for example, for smoothing splines $\lambda\boldsymbol{\beta}^T\mathbf{S}\boldsymbol{\beta}$ corresponds to a penalty on the integrated squared second derivative of the smooth. It turns out that this penalty term is equivalent to using the prior distribution $\boldsymbol{\beta}|\lambda \sim N(\mathbf{0}, (\lambda\mathbf{S})^-)$, where $(\lambda\mathbf{S})^-$ denotes the pseudo-inverse (Wood, 2017). The posterior distribution of $\boldsymbol{\beta}$, given the training data (X_i', Z_i', Y_i'), is approximately

$$\boldsymbol{\beta}|\{(Z_i', Y_i')\}_{i=1}^m, \lambda \sim N(\widehat{\boldsymbol{\beta}}, (\overline{\mathbf{Z}}^T\mathbf{W}\overline{\mathbf{Z}} + \lambda\mathbf{S})^{-1}), \tag{A.3}$$

where $\widehat{\boldsymbol{\beta}}$ are the estimated coefficients and $\mathbf{W}$ is the weights matrix from the last step of penalized iterative reweighted least squares estimation (Wood, 2017).

If we use a logistic GAM as our classifier, we can draw a new classifier function $\mathcal{A}^*$ in the bootstrap procedure discussed above by sampling from the posterior distribution of $\boldsymbol{\beta}$. While we expect draws from a posterior to give Bayesian credible intervals, posteriors for spline fits often have good frequentist properties (Krivobokova et al., 2010; Wood, 2017), and so it is reasonable to sample from the posterior in (A.3) to construct our frequentist bootstrap confidence intervals.

A.3 Bootstrap algorithms for inference

In Section II.3, we describe semiparametric bootstrap procedures for inference with prevalence $P(Y = y|C = c)$ and class-conditional feature means $\mathbb{E}[X|Y = y, C = c]$. Here we include the detailed algorithms for implementing each bootstrap procedure. Algorithm 2 describes the procedure from Section II.3.1 for constructing a confidence interval for prevalence; Algorithm 3 constructs confidence intervals for $\mathbb{E}[X|Y = y, C = c]$ using probability-weighted mixed effects models, as in Section II.3.2; and Algorithm 4 constructs confidence intervals for $\mathbb{E}[X|Y = y, C = c]$ using parametric mixture models assisted by label shift estimation, as in Section II.3.3.

Algorithm 2: Semiparametric bootstrap confidence intervals for prevalence, under label shift

```
Data: Labeled training data $\{(Z'_i, Y'_i, C'_i, K'_i)\}_{i=1}^m$
      Unlabeled test data $\{(Z_j, C_j, K_j)\}_{j=1}^n$
Input  : Number $B$ of bootstrap samples
         Level $1 - \alpha$ for confidence interval
Output: A confidence interval for $P(Y = 1|C = c)$
init
    Train classifier $\mathcal{A}$ with training data $\{(Z'_i, Y'_i)\}_{i=1}^m$;
    Calculate estimate $\widehat{P}(Y = 1|C = c)$ with label shift methods (A.1) ;
    Calculate label shift-corrected predictions $\mathcal{A}_L(Z_i, C_i)$ as in (II.2);
for $s = 1, ..., B$ do
    Sample $(Z_1'^*, Y_1'^*), ..., (Z_m'^*, Y_m'^*)$ by resampling rows $(Z'_i, Y'_i)$ with replacement;
    Train classifier $\mathcal{A}^*$ on bootstrap sample $(Z_1'^*, Y_1'^*), ..., (Z_m'^*, Y_m'^*)$;
    Sample $(Z_i^*, C_i^*)$ by resampling rows $(Z_i, C_i)$ with replacement;
    Using $(Z_i'^*, Y_i'^*)$, $\mathcal{A}^*$, and $Z_i^*$, calculate $\widehat{P}_s(Y^* = 1|C^* = c)$ with label shift methods (A.1);
end
return $1 - \alpha$ confidence interval from $\widehat{P}(Y = 1|C = c)$ and the $\widehat{P}_s(Y^* = 1|C^* = c)$ (e.g., a bootstrap percentile
  interval)
```

A.4 Mixed effects models with label-dependent random effects

As we discuss in Section II.3 and Section II.4, confidence intervals from the weighted mixed effects model (II.5) may lose coverage when random effects are label dependent. In Table II.2, we saw this decrease in coverage, and also that an additional variance calibration step can address the issue. Algorithm 5 details the full mixed effects procedure when random effects are label-dependent, including the variance calibration step. Note that variance calibration approximately doubles the time needed to construct a bootstrap confidence interval.

Algorithm 3: Semiparametric bootstrap confidence intervals with classifier predictions under label shift

```
Data: Labeled training data $\{(X_i', Z_i', Y_i', C_i', K_i')\}_{i=1}^m$
      Unlabeled test data $\{(X_j, Z_j, c_j, K_j)\}_{j=1}^n$
Input  : Number $B$ of bootstrap samples
         Level $1-\alpha$ for confidence interval
Output: A confidence interval for the parameters $\beta_{c,1}$ of the mixed effects model (II.5)
init
    Train classifier $\mathcal{A}$ with training data $\{(Z_i', Y_i')\}_{i=1}^m$;
    Calculate label shift-corrected predictions $\mathcal{A}_L(Z_i, C_i)$ as in (II.2);
    Fit the mixed effects model (II.5) for $y = 0$ and $y = 1$, using weights $w_{i,1} = \mathcal{A}_L(Z_i, c_i)$ and $w_{i,0} = 1 - w_{i,1}$. This
      gives parameter estimates $\widehat{\beta}_{c,y}$ and $\widehat{\omega}^2$, and observed random effects $\widehat{b}_k$;
    Define residuals $e_i = X_i - \widehat{b}_k$;
for $s = 1, ..., B$ do
    Sample $(Z_1'^*, Y_1'^*), ..., (Z_m'^*, Y_m'^*)$ by resampling rows $(Z_i', Y_i')$ with replacement;
    Train classifier $\mathcal{A}^*$ on bootstrap sample $(Z_1'^*, Y_1'^*), ..., (Z_m'^*, Y_m'^*)$;
    Sample $(Z_i^*, C_i^*, K_i^*, \mathcal{A}_L(Z_i^*, c_i^*), e_i^*)$ by resampling rows $(Z_i, C_i, K_i, \mathcal{A}_L(Z_i, C_i), e_i)$ with replacement;
    Sample $Y_i^* \overset{iid}{\sim} \text{Bernoulli}(\mathcal{A}_L(Z_i^*, C_i^*))$ for $i = 1, ..., n$;
    Sample $b_k^* \overset{iid}{\sim} N(0, \widehat{\omega}^2)$ for $k \in \mathcal{K}$;
    Generate $X_i^*$ by $X_i^* = e_i^* + b_k^*$ for $i = 1, ..., n$;
    Using $(Z_i'^*, Y_i'^*)$, $\mathcal{A}^*$, and $Z_i^*$, calculate $\widehat{P}(Y_i^* = 1|c_i^*)$ under the label shift assumption (A.1);
    $\mathcal{A}_L^*(Z_i^*, C_i^*) = \dfrac{\frac{\widehat{P}(Y_i^*=1|C_i^*)}{\widehat{P}(Y_i'^*=1)}\mathcal{A}^*(Z_i^*)}{\frac{\widehat{P}(Y_i^*=1|C_i^*)}{\widehat{P}(Y_i'^*=1)}\mathcal{A}^*(Z_i^*) + \frac{1-\widehat{P}(Y_i^*=1|C_i^*)}{1-\widehat{P}(Y_i'^*=1)}(1 - \mathcal{A}^*(Z_i^*))}$;
    Fit the mixed effects model (II.5) with observed data $(X_i^*, C_i^*, K_i^*)$ and weights $w_{i,1}^* = \mathcal{A}_L^*(Z_i^*, C_i^*)$, giving
      estimates $\widehat{\beta}_{c,1,s}^*$;
end
return $1-\alpha$ confidence interval from $\widehat{\beta}_{c,1}$ and the $\widehat{\beta}_{c,1,s}^*$ (e.g., a bootstrap percentile interval)
```

```
Algorithm 4: Semiparametric bootstrap confidence intervals with mixture models under label shift
Data: Labeled training data $\{(X_i', Z_i', Y_i', C_i', K_i')\}_{i=1}^m$
      Unlabeled test data $\{(X_j, Z_j, C_j, K_j)\}_{j=1}^n$
Input  : Number $B$ of bootstrap samples
         Level $1 - \alpha$ for confidence interval
Output: A confidence interval for the parameters $\beta_{c,1}$ of the mixture model (II.6)
init
    Train classifier $\mathcal{A}$ with training data $\{(Z_i', Y_i')\}_{i=1}^m$;
    Using $(Z_i', Y_i')$, $\mathcal{A}$, and $Z_i$, calculate $\widehat{P}(Y_i = 1|C_i)$ under the label shift assumption (A.1);
    Calculate label shift-corrected predictions $\mathcal{A}_L(Z_i, C_i)$ as in (II.2);
    Using $\widehat{P}(Y_i = 1|C_i)$ as mixing proportions, fit the mixed effects mixture model (II.6), giving parameter
      estimates $\widehat{\beta}_{c,y}$ and $\widehat{\omega}_y^2$ and observed random effects $\widehat{b}_{k,0}$, $y \in \{0, 1\}$;
    Define residuals $e_{i,0} = X_i - \widehat{b}_{k,1}$ and $e_{i,1} = X_i - \widehat{b}_{k,0}$;
for $s = 1, ..., B$ do
    Sample $(Z_1'^*, Y_1'^*), ..., (Z_m'^*, Y_m'^*)$ by resampling rows $(Z_i', Y_i')$ with replacement;
    Train classifier $\mathcal{A}^*$ on bootstrap sample $(Z_1'^*, Y_1'^*), ..., (Z_m'^*, Y_m'^*)$;
    Sample $(Z_i^*, C_i^*, K_i^*, \mathcal{A}_L(Z_i^*, C_i^*), e_{i,0}^*, e_{i,1}^*)$ by resampling rows $(Z_i, C_i, K_i, \mathcal{A}_L(Z_i, C_i), e_{i,0}, e_{i,1})$ with
      replacement;
    Sample $Y_i^* \overset{iid}{\sim} \text{Bernoulli}(\mathcal{A}_L(Z_i^*, c_i^*))$ for $i = 1, ..., n$;
    Sample $b_{k,0}^* \overset{iid}{\sim} N(0, \widehat{\omega}_0^2)$ and $b_{k,1}^* \overset{iid}{\sim} N(0, \widehat{\omega}_1^2)$ for $k \in \mathcal{K}$;
    Generate $X_i^*$ by $X_i^* = (e_{i,1}^* + b_{k,1}^*)Y_i^* + (e_{i,0}^* + b_{k,0}^*)(1 - Y_i^*)$ for $i = 1, ..., n$;
    Using $(Z_i'^*, Y_i'^*)$, $\mathcal{A}^*$, and $Z_i^*$, calculate $\widehat{P}(Y_i^* = 1|C_i^*)$ under the label shift assumption (A.1);
    Using $\widehat{P}(Y_i^* = 1|C_i^*)$ as mixing proportions, fit the mixed effects mixture model (II.6), giving parameter
      estimates $\widehat{\beta}_{c,1,s}^*$;
end
return $1 - \alpha$ confidence interval from $\widehat{\beta}_{c,1}$ and the $\widehat{\beta}_{c,1,s}^*$ (e.g., a bootstrap percentile interval)
```

Algorithm 5: Semiparametric bootstrap confidence intervals with classifier predictions under label shift, with variance calibration step

```
Data: Labeled training data $\{(X'_i, Z'_i, Y'_i, C'_i, K'_i)\}_{i=1}^m$
      Unlabeled test data $\{(X_j, Z_j, c_j, K_j)\}_{j=1}^n$
Input : Number $B$ of bootstrap samples
        Level $1-\alpha$ for confidence interval
Output: A confidence interval for the parameters $\beta_{c,1}$ of the mixed effects model (II.5)
init
    Train classifier $\mathcal{A}$ with training data $\{(Z'_i, Y'_i)\}_{i=1}^m$;
    Calculate label shift-corrected predictions $\mathcal{A}_L(Z_i, C_i)$ as in (II.2);
    Fit the mixed effects model (II.5) with $y = 1$ and weights $w_{i,1} = \mathcal{A}_L(Z_i, c_i)$, giving parameter estimates $\widehat{\beta}_{c,1}$ and
      $\widehat{\omega}_1^2$, and observed random effects $\widehat{b}_{k,1}$;
    Fit the mixed effects model (II.5) with $y = 0$ and weights $w_{i,0} = 1 - w_{i,1}$, giving parameter estimates $\widehat{\beta}_{c,0}$ and
      $\widehat{\omega}_0^2$, and observed random effects $\widehat{b}_{k,0}$;
    Define residuals $e_{i,0} = X_i - \widehat{b}_{k,1}$ and $e_{i,1} = X_i - \widehat{b}_{k,0}$;
for $s = 1, ..., B$ do
    Sample $(Z_1'^*, Y_1'^*), ..., (Z_m'^*, Y_m'^*)$ by resampling rows $(Z'_i, Y'_i)$ with replacement;
    Train classifier $\mathcal{A}^*$ on bootstrap sample $(Z_1'^*, Y_1'^*), ..., (Z_m'^*, Y_m'^*)$;
    Sample $(Z_i^*, C_i^*, K_i^*, \mathcal{A}_L(Z_i^*, c_i^*), e_{i,0}^*, e_{i,1}^*)$ by resampling rows $(Z_i, C_i, K_i, \mathcal{A}_L(Z_i, C_i), e_{i,0}, e_{i,1})$ with replacement;
    Sample $Y_i^* \overset{iid}{\sim} \text{Bernoulli}(\mathcal{A}_L(Z_i^*, C_i^*))$ for $i = 1, ..., n$;
    Sample $b_{k,0}^* \overset{iid}{\sim} N(0, \widehat{\omega}_0^2)$ and $b_{k,1}^* \overset{iid}{\sim} N(0, \widehat{\omega}_1^2)$ for $k \in \mathcal{K}$;
    Generate $X_i^*$ by $X_i^* = (e_{i,1}^* + b_{k,1}^*)Y_i^* + (e_{i,0}^* + b_{k,0}^*)(1 - Y_i^*)$ for $i = 1, ..., n$;
    Using $(Z_i'^*, Y_i'^*)$, $\mathcal{A}^*$, and $Z_i^*$, calculate $\widehat{P}(Y_i^* = 1|c_i^*)$ under the label shift assumption, e.g. using the method
      from Lipton et al. (2018);
    $\mathcal{A}_L^*(Z_i^*, C_i^*) = \frac{\frac{\widehat{P}(Y_i^*=1|C_i^*)}{\widehat{P}(Y_i'^*=1)}\mathcal{A}^*(Z_i^*)}{\frac{\widehat{P}(Y_i^*=1|C_i^*)}{\widehat{P}(Y_i'^*=1)}\mathcal{A}^*(Z_i^*) + \frac{1-\widehat{P}(Y_i^*=1|C_i^*)}{1-\widehat{P}(Y_i'^*=1)}(1 - \mathcal{A}^*(Z_i^*))}$;
    Fit the mixed effects model (II.5) with observed data $(X_i^*, C_i^*, K_i^*)$ and weights $w_{i,1}^* = \mathcal{A}_L^*(Z_i^*, C_i^*)$ and
      $w_{i,0}^* = 1 - w_{i,1}^*$, giving estimates $\widehat{\omega}_{1,s}^{*2}$ and $\widehat{\omega}_{0,s}^{*2}$;
    Calculate the true sample variance $v_{y,s}^2 = \frac{1}{|\mathcal{K}|-1} \sum_k b_{k,y}^{*2}$ ;
end
Regress $v_{y,s}^2$ on $\widehat{\omega}_{y,s}^{*2}$, producing an estimating function $\widehat{f}_y$ with $\widehat{v}_{y,s}^2 = \widehat{f}_y(\widehat{\omega}_{y,s}^{*2})$, for $y \in \{0, 1\}$ ;
Calculate the adjusted variances: $\widehat{\omega}_{y,\text{adj}}^2 = \widehat{f}_y(\widehat{\omega}_y^2)$, for $y \in \{0, 1\}$ ;
for $s = 1, ..., B$ do
    Sample $(Z_1'^*, Y_1'^*), ..., (Z_m'^*, Y_m'^*)$ by resampling rows $(Z'_i, Y'_i)$ with replacement;
    Train classifier $\mathcal{A}^*$ on bootstrap sample $(Z_1'^*, Y_1'^*), ..., (Z_m'^*, Y_m'^*)$;
    Sample $(Z_i^*, C_i^*, K_i^*, \mathcal{A}_L(Z_i^*, c_i^*), e_{i,0}^*, e_{i,1}^*)$ by resampling rows $(Z_i, C_i, K_i, \mathcal{A}_L(Z_i, C_i), e_{i,0}, e_{i,1})$ with replacement;
    Sample $Y_i^* \overset{iid}{\sim} \text{Bernoulli}(\mathcal{A}_L(Z_i^*, C_i^*))$ for $i = 1, ..., n$;
    Sample $b_{k,0}^* \overset{iid}{\sim} N(0, \widehat{\omega}_{0,\text{adj}}^2)$ and $b_{k,1}^* \overset{iid}{\sim} N(0, \widehat{\omega}_{1,\text{adj}}^2)$ for $k \in \mathcal{K}$;
    Generate $X_i^*$ by $X_i^* = (e_{i,1}^* + b_{k,1}^*)Y_i^* + (e_{i,0}^* + b_{k,0}^*)(1 - Y_i^*)$ for $i = 1, ..., n$;
    Using $(Z_i'^*, Y_i'^*)$, $\mathcal{A}^*$, and $Z_i^*$, calculate $\widehat{P}(Y_i^* = 1|c_i^*)$ under the label shift assumption, e.g. using the method
      from Lipton et al. (2018);
    $\mathcal{A}_L^*(Z_i^*, C_i^*) = \frac{\frac{\widehat{P}(Y_i^*=1|C_i^*)}{\widehat{P}(Y_i'^*=1)}\mathcal{A}^*(Z_i^*)}{\frac{\widehat{P}(Y_i^*=1|C_i^*)}{\widehat{P}(Y_i'^*=1)}\mathcal{A}^*(Z_i^*) + \frac{1-\widehat{P}(Y_i^*=1|C_i^*)}{1-\widehat{P}(Y_i'^*=1)}(1 - \mathcal{A}^*(Z_i^*))}$;
    Fit the mixed effects model (II.5) with observed data $(X_i^*, C_i^*, K_i^*)$ and weights $w_{i,1}^* = \mathcal{A}_L^*(Z_i^*, C_i^*)$, giving
      estimates $\widehat{\beta}_{c,1,s}^*$;
end
return $1-\alpha$ confidence interval from $\widehat{\beta}_{c,1}$ and the $\widehat{\beta}_{c,1,s}^*$ (e.g., a bootstrap percentile interval)
```

Assumptions	Normal?	Training Data	Test Data
(A1), (A2), (A3)	yes	$Y' \sim Bernoulli(0.2)$ $Z'\|Y' = 0 \sim N(0,1)$ $Z'\|Y' = 1 \sim N(3,1)$	$Y \sim Bernoulli(0.4)$ $Z\|Y = 0 \sim N(0,1)$ $Z\|Y = 1 \sim N(3,1)$ $b_k \sim N(0,0.5)$ $X_i = Z_i + b_{k_i} + N(0,0.2)$
	no	$Y' \sim Bernoulli(0.2)$ $Z'\|Y' = 0 \sim SN(0,2,3)$ $Z'\|Y' = 1 \sim 8 - SN(3,2,3)$	$Y \sim Bernoulli(0.4)$ $Z\|Y = 0 \sim SN(0,2,3)$ $Z\|Y = 1 \sim 8 - SN(3,2,3)$ $b_k \sim N(0,0.5)$ $X_i = Z_i + b_{k_i} + N(0,0.2)$
(A2), (A3)	yes	$Y' \sim Bernoulli(0.2)$ $Z'\|Y' = 0 \sim N(-0.5,1)$ $Z'\|Y' = 1 \sim N(3,1)$	$Y \sim Bernoulli(0.4)$ $Z\|Y = 0 \sim N(0,1)$ $Z\|Y = 1 \sim N(3,1)$ $b_k \sim N(0,0.5)$ $X_i = Z_i + b_{k_i} + N(0,0.2)$
	no	$Y' \sim Bernoulli(0.2)$ $Z'\|Y' = 0 \sim SN(-0.5,2,3)$ $Z'\|Y' = 1 \sim 8 - SN(3,2,3)$	$Y \sim Bernoulli(0.4)$ $Z\|Y = 0 \sim SN(0,2,3)$ $Z\|Y = 1 \sim 8 - SN(3,2,3)$ $b_k \sim N(0,0.5)$ $X_i = Z_i + b_{k_i} + N(0,0.2)$
(A1), (A2)	yes	$Y' \sim Bernoulli(0.2)$ $Z'\|Y' = 0 \sim N(0,1)$ $Z'\|Y' = 1 \sim N(3,1)$	$Y \sim Bernoulli(0.4)$ $Z\|Y = 0 \sim N(0,1)$ $Z\|Y = 1 \sim N(3,1)$ $b_k \sim N(0,0.5)$ $X_i = Z_i + b_{k_i} + N(0,0.2) + \mathbb{1}(Y_i = 1)$
	no	$Y' \sim Bernoulli(0.2)$ $Z'\|Y' = 0 \sim SN(0,2,3)$ $Z'\|Y' = 1 \sim 8 - SN(3,2,3)$	$Y \sim Bernoulli(0.4)$ $Z\|Y = 0 \sim SN(0,2,3)$ $Z\|Y = 1 \sim 8 - SN(3,2,3)$ $b_k \sim N(0,0.5)$ $X_i = Z_i + b_{k_i} + N(0,0.2) + \mathbb{1}(Y_i = 1)$

Table A.1: Simulation settings for assessing performance of bootstrap inference procedures.

A.5 Simulation settings

In Section II.4, we conduct simulations to assess the impact of assumptions on the coverage of our bootstrap confidence intervals. We describe three different scenarios, with different combinations of (A1), (A2), and (A3) holding true. We also assess the impact of deviations from parametric assumptions on the mixture model approach. In Table A.1 we provide the simulation settings, describing how training and test data was simulated in each scenario. Note that $N(\mu, \sigma^2)$ denotes a Gaussian distribution with mean μ and variance σ^2, and $SN(\xi, \omega, \alpha)$ denotes a skewed normal distribution with location ξ, scale ω, and shape α.

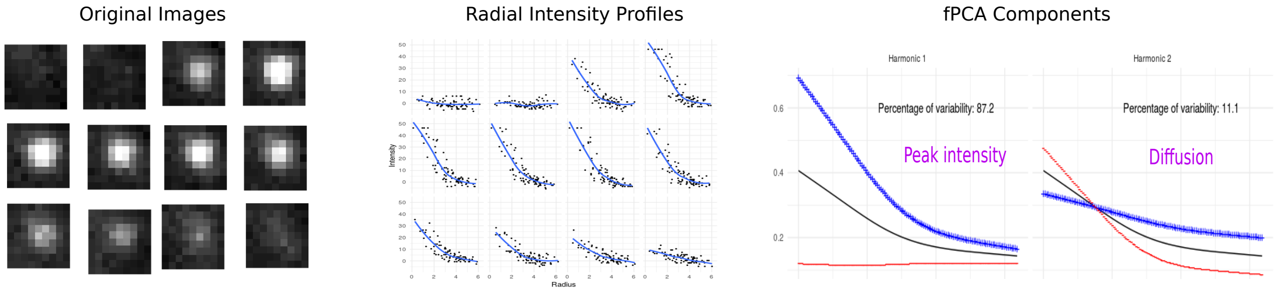

Figure A.1: Functional PCA on radial intensity profiles. Left: the original frames for a puff from TIRF microscopy. Center: the corresponding radial intensity profiles for the frames on the left, with smooth curves showing the overall shape in each frame. Right: the first two fPCA component functions for the radial intensity profiles, plotted as differences from the mean function (the black curve). A positive score for the component function is plotted with blue +'s, while a negative score for the component function is plotted with red −'s.

A.6 Functional PCA features

After automatic event detection and particle tracking, each detected event on the cell surface is represented by a series of greyscale 9 × 9 pixel frames, with the intensity of fluorescence recorded for each pixel. As shown in Figure II.1, there are often clear differences in the individual frames for puffs and nonpuffs, and more importantly in the evolution of frames over time.

To capture the behavior of puffs over time, we first consider the two-dimensional intensity function within each frame. Noticing that puffs tend to have symmetric intensity functions, we can reduce the intensity function to a one-dimensional function of distance from the center of the frame (Figure A.1). Each event then becomes a collection of radial intensity functions. As demonstrated in Figure A.1, the radial intensity functions tend to have similar shapes, with only a few modes of variation. This suggests that functional PCA (fPCA) (Ramsay and Silverman, 2005) could provide effective dimension reduction of the radial profiles.

Before performing fPCA, each detected event was scaled to have the same peak intensity. Figure A.1 shows the first two principal component functions, which together make up about 98% of the variability in radial intensity profiles. As we might expect, peak intensity in a frame is the main component of variation, captured by the first principal component. The second principal component, account for about 11% of the variability in radial intensity profiles, captures diffusivity of fluorescence in the frame. Each event is then represented as a bivariate time series of scores for the first two principal components. As suggested by Figure II.1, time series of component scores for puffs are expected to have a characteristic pattern, whereas scores for nonpuffs are expected to be much noisier. To visualize fPCA score time series, we consider each event as a path through two-dimensional principal component score space. Figure A.2 shows these paths for a puff and a nonpuff, illustrating the differences we expect to see in these time series.

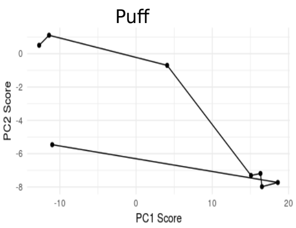

Figure A.2: Differences between fPCA score paths for puffs and nonpuffs.

Figure A.2 suggests that featurizing the fPCA score paths could be useful for classifying puffs and nonpuffs. We construct several features:

- *ConvexArea* and *ConvexPerimeter*: the area and perimeter of a convex hull around the score path
- *Randomness*: a measure of randomness in the time series of first principal component scores
- *Smoothness*: the average distance between points in the observed score path and the kernel-smoothed score path

A.7 TIRF microscopy mixture model results

In Section II.5, we summarize the results of inference on puff *Smoothness*, using the mixed effects approach and the mixture model approach. The full results are provided here. Figure A.3 shows the distribution of *Smoothness* within each test cell, for puffs and nonpuffs. The distributions in Figure A.3 suggest that a two-component Gaussian mixture is reasonable, and so we fit the hierarchical mixture described in (II.6). However, because the parametric model does not hold exactly, and the proportion of puffs is small, the estimated puff distributions are poor (Figure A.3). With the mixing proportions specified, Figure A.4 shows the resulting fit, which is much improved over Figure A.3. The estimated means in each cell are given by Table A.2; the slight bias in puff means arises because the distribution of *Smoothness* is only approximately Gaussian.

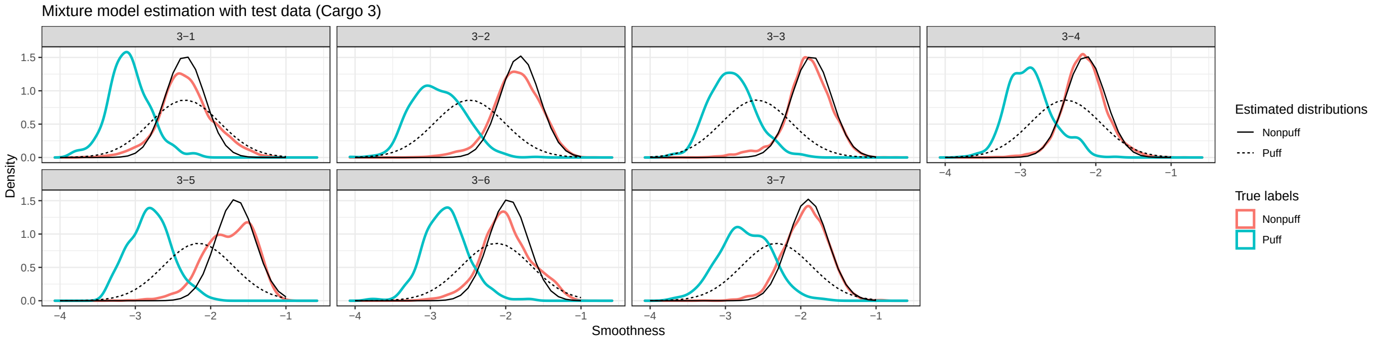

Figure A.3: The distribution of *Smoothness* in each cell, for puffs and nonpuffs. Estimates of the true densities are shown from kernel density estimation with the true labels, while the black curves show the fitted normal distributions from a hierarchical Gaussian mixture model. For the mixture model, the mixing proportion for puffs and nonpuffs is estimated as a parameter of the model.

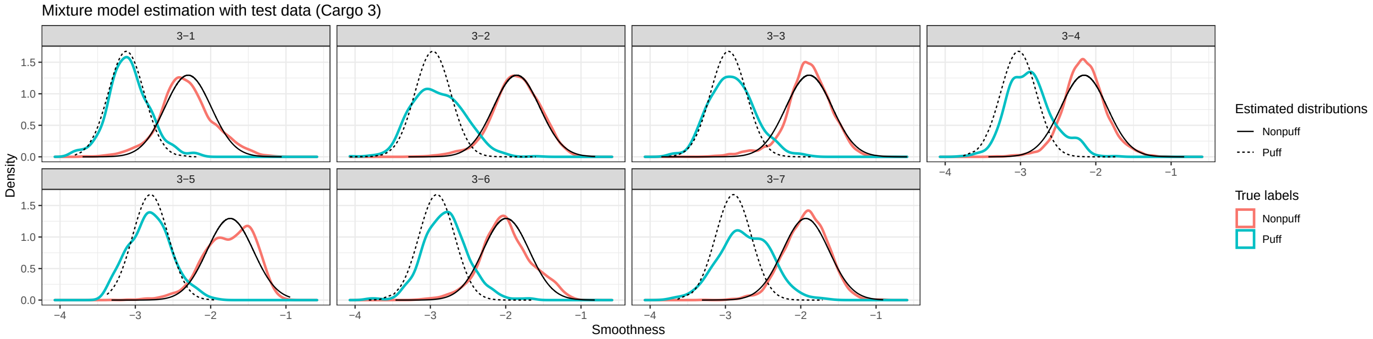

Figure A.4: The distribution of *Smoothness* in each cell, for puffs and nonpuffs. Estimates of the true densities are shown from kernel density estimation with the true labels, while the black curves show the fitted normal distributions from a hierarchical Gaussian mixture model. For the mixture model, the mixing proportion in each cell is estimated beforehand with the label shift correction, to improve identifiability.

Cell	True means		Mixture model estimates	
	Puff	Nonpuff	Puff	Nonpuff
3-1	-3.08	-2.32	-3.12	-2.30
3-2	-2.90	-1.86	-2.97	-1.86
3-3	-2.91	-1.90	-2.96	-1.89
3-4	-2.87	-2.15	-3.03	-2.16
3-5	-2.80	-1.75	-2.80	-1.74
3-6	-2.79	-1.99	-2.91	-1.99
3-7	-2.73	-1.93	-2.89	-1.93

Table A.2: Mean of *Smoothness* in each Condition 3 cell. The true means are estimated with the true labels, while the mixture model estimates result from a mixed effects Gaussian mixture model (II.6), where the mixing proportion in each cell is estimated with the label shift correction. There is slight bias in the mixture model estimates of puff means shown here; this occurs because the distribution of *Smoothness* for puffs is only approximately Gaussian, and is slightly right-skewed (Figure A.4).

Appendix B

Sequential changepoint detection for label shift in classification

B.1 Proof of Theorem 1

Proof. Following Tartakovsky et al. (2009), we can express the desired expectations as solutions to Fredholm integral equations of the second kind. In particular, let $v_1(x) = \mathbb{E}_i(T^x(A))$, and $v_2(x) = \mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_m)$. Then,

$$v_1(x) = 1 + \int\limits_0^A v_1(y)k(x,y)dy \tag{B.1}$$

$$v_2(x) = 1 + \int\limits_0^A v_2(y)\widetilde{k}(x,y)dy, \tag{B.2}$$

where $k(x,y) = \frac{\partial}{\partial y}\mathbb{P}_i\left(\lambda(X) \leq \frac{y}{\Psi(x)}\right)$ (Tartakovsky et al., 2009), and $\widetilde{k}(x,y) = \frac{\partial}{\partial y}\mathbb{P}_i\left(\widehat{\lambda}_m(X) \leq \frac{y}{\Psi(x)}|\widehat{\lambda}_m\right)$. Using this representation, we show that $||v_1 - v_2||_\infty$ is small.

First, let K and $\widetilde{K}$ be the integral operators defined by the kernels $k(x,y)$ and $\widetilde{k}(x,y)$. Then under assumption (B3), we have $K, \widetilde{K} : C([0,A]) \to C([0,A])$ are compact (Han and Atkinson, 2009). Furthermore, $v_1 = (I - K)\mathbf{1}$ and $v_2 = (I - \widetilde{K})\mathbf{1}$, where $\mathbf{1}(x) \equiv 1$.

Next we show that $Kv = v$ and $\widetilde{K}v = v$ have only the trivial solution $v = 0$. Considering $K, \widetilde{K}$ on $L^2(0, A)$, we have $K, \widetilde{K} : L^2(0, A) \to L^2(0, A)$ because $k, \widetilde{k}$ are Hilbert-Schmidt kernel functions under (B3). Then consider the adjoint operators K^* and $\widetilde{K}^*$. Moustakides et al. (2011) demonstrate that the maximal eigenvalue of K^* is strictly less than 1 when $\lambda(X)$ is continuous, and therefore the same holds for K. So, $Kv = v$ has only the trivial solution $v = 0$, with the same result for $\widetilde{K}$.

Then by the Fredholm alternative theorem, $(I - K)^{-1}, (I - \widetilde{K})^{-1} : C([0, A]) \to C([0, A])$ are bijective and bounded (Han and Atkinson, 2009). Furthermore,

$$||K - \widetilde{K}||_\infty \le \sup_{x \in [0,A]} \int_0^A |k(x, y) - \widetilde{k}(x, y)| dy \le \int_0^\infty |f^i_\lambda(s) - f^i_{\widehat{\lambda}_m}(s)| ds = 2TV(f^i_\lambda, f^i_{\widehat{\lambda}_m}). \tag{B.3}$$

By (B5), $TV(f^i_\lambda, f^i_{\widehat{\lambda}_m}) \xrightarrow{p} 0$, and so by Theorem 2.3.5 in Han and Atkinson (2009), we have that

$$|\mathbb{E}_i(\widetilde{T}^x(A)|\widehat{\lambda}_m) - \mathbb{E}_i(T^x(A))| \le ||v_2 - v_1||_\infty \le ||\widetilde{K}^{-1}||_\infty ||(K - \widetilde{K})v_1||_\infty \le O_P(TV(f^i_\lambda, f^i_{\widehat{\lambda}_m})). \tag{B.4}$$

Finally, we can improve the bound in the case when f^i_λ is bounded. Note that

$$\begin{aligned} \left| \int_0^A (k(x, y) - \widetilde{k}(x, y)) v_1(y) dy \right| \le & \left| \int_0^\infty f_{\Psi(x)\lambda}(y) \overline{v}_1(y) dy - \int_0^\infty f_{\Psi(x)\widehat{\lambda}_m}(y) \overline{v}_1(y) dy \right| \\ & + v_1(A) \left| \int_A^\infty (f_{\Psi(x)\lambda}(y) - f_{\Psi(x)\widehat{\lambda}_m}(y)) dy \right|, \end{aligned} \tag{B.5}$$

where $\overline{v}_1(y) = v_1(y)$ for $y \le A$, and $\overline{v}_1(y) = v_1(A)$ for $y \ge A$. If f^i_λ is bounded, then v_1 is Lipschitz, and

$$\left| \int_0^\infty f_{\Psi(x)\lambda}(y) \overline{v}_1(y) dy - \int_0^\infty f_{\Psi(x)\widehat{\lambda}_m}(y) \overline{v}_1(y) dy \right| \le C \sup_{||h||_L \le 1} \left| \int_0^\infty f_{\Psi(x)\lambda}(y) h(y) dy - \int_0^\infty f_{\Psi(x)\widehat{\lambda}_m}(y) h(y) dy \right| \tag{B.6}$$

$$\le C\Psi(A)\mathbb{E}_i[|\widehat{\lambda}_m(X) - \lambda(X)|], \tag{B.7}$$

by Kantorovich-Rubinstein duality. Since $\left| \int_A^\infty (f_{\Psi(x)\lambda}(y) - f_{\Psi(x)\widehat{\lambda}_m}(y)) dy \right| \le ||F^i_\lambda - F^i_{\widehat{\lambda}_m}||_\infty$, this concludes the proof. □

B.2 Proof of Corollary 1

Proof. The proof of Corollary 1 is similar to the proof of Theorem 1, but in a finite-dimensional space. Since π_0 and π_∞ are known, and $\log(\pi_0/\pi_\infty)$ and $\log((1-\pi_0)/(1-\pi_\infty))$ are both rational, then R_t^x and $\widetilde{R}_t^x$ are Markov chains on the same finite state space, where x and the states are linear combinations of $\log(\pi_0/\pi_\infty)$ and $\log((1-\pi_0)/(1-\pi_\infty))$. Let v_1 denote the vector of expected stopping times when starting the optimal detection procedure in each state, and v_2 the corresponding vector for the estimated detection procedure. Then, $(I-K)v_1 = \mathbf{1}$ and $(I-\widetilde{K})v_2 = \mathbf{1}$, where $\mathbf{1}$ is the vector of all 1's, and K and $\widetilde{K}$ are transition probability matrices for the Markov chain.

In particular, all elements of K are either 0, $\mathbb{P}(Y_i = 0)$, or $\mathbb{P}(Y_i = 1)$. The corresponding elements of $\widetilde{K}$ are

$$\widetilde{K}_{ij} = \begin{cases} 0 & K_{ij} = 0 \\ \mathbb{P}(\mathcal{A}(X) = 0|\mathcal{A}) & K_{ij} = \mathbb{P}(Y_i = 0) \\ \mathbb{P}(\mathcal{A}(X) = 1|\mathcal{A}) & K_{ij} = \mathbb{P}(Y_i = 1). \end{cases} \tag{B.8}$$

Therefore, $|\widetilde{K}_{ij} - K_{ij}| \leq O_P(\mathbb{P}_i(\mathcal{A}(X) = 0|Y = 1, \mathcal{A}) + \mathbb{P}_i(\mathcal{A}(X) = 1|Y = 0, \mathcal{A}))$, and as in Theorem 1 the proof follows again by applying Theorem 2.3.5 from Han and Atkinson (2009). □

B.3 Proof of Theorem 2

Proof. For ease of notation, we drop the subscript $i \in \{0, \infty\}$ and the dependence on the classifier $\mathcal{A}$ from the expectations; our goal is to show $|\mathbb{E}(\widetilde{T}_w(A)) - \mathbb{E}(T_w(A))| \xrightarrow{p} 0$. Let $U_t(A) = \min\{T_w(A), t\}$ and $\widetilde{U}_t(A) = \min\{\widetilde{T}_w(A), t\}$. Then for any $t_0 > 0$,

$$\begin{aligned} |\mathbb{E}(\widetilde{T}_w(A)) - \mathbb{E}(T_w(A))| \leq |\mathbb{E}(\widetilde{T}_w(A)) - \mathbb{E}(\widetilde{U}_{t_0}(A))| + |\mathbb{E}(\widetilde{U}_{t_0}(A)) - \mathbb{E}(U_{t_0}(A))| + \\ |\mathbb{E}(U_{t_0}(A)) - \mathbb{E}(T_w(A))|. \end{aligned} \tag{B.9}$$

For $\varepsilon > 0$, let $\mathcal{C}_{t_0,\varepsilon} = \left\{X_1, ..., X_{t_0} : \sup\limits_{t \leq t_0} |\widetilde{R}_{t,w} - R_{t,w}| < \varepsilon | \mathcal{A}\right\}$. Then

$$|\mathbb{E}(\widetilde{U}_{t_0}(A)) - \mathbb{E}(U_{t_0}(A))| \leq |\mathbb{E}(\widetilde{U}_{t_0}(A)) - \mathbb{E}(\widetilde{U}_{t_0}(A)|\mathcal{C}_{t_0,\varepsilon})| + |\mathbb{E}(\widetilde{U}_{t_0}(A)|\mathcal{C}_{t_0,\varepsilon}) - \mathbb{E}(U_{t_0}(A))|, \tag{B.10}$$

and $|\mathbb{E}(\widetilde{U}_{t_0}(A)) - \mathbb{E}(\widetilde{U}_{t_0}(A)|\mathcal{C}_{t_0,\varepsilon})| \leq 2t_0(1-\mathbb{P}(\mathcal{C}_{t_0,\varepsilon}))$. Also, if $X_1, ..., X_{t_0} \in \mathcal{C}_{t_0,\varepsilon}$ then $U_{t_0}(A-\varepsilon) \leq \widetilde{U}_{t_0}(A) \leq U_{t_0}(A+\varepsilon)$, and so

$$\begin{aligned}|\mathbb{E}(\widetilde{U}_{t_0}(A)|\mathcal{C}_{t_0,\varepsilon}) - \mathbb{E}(U_{t_0}(A))| \leq |\mathbb{E}(U_{t_0}(A+\varepsilon)) - \mathbb{E}(T_w(A+\varepsilon))| + |\mathbb{E}(T_w(A+\varepsilon)) - \mathbb{E}(T_w(A-\varepsilon))| + \\ |\mathbb{E}(T_w(A-\varepsilon)) - \mathbb{E}(U_{t_0}(A-\varepsilon))|.\end{aligned} \tag{B.11}$$

Now let $\eta > 0$. By continuity of $\mathbb{E}(T_w(A))$, there exists $\varepsilon > 0$ such that $|\mathbb{E}(T_w(A+\varepsilon)) - \mathbb{E}(T_w(A-\varepsilon))| < \eta/6$. Next, let $c_1 > 0$ and $0 < \delta_1 < \min\{\frac{a}{\pi_\infty}, \frac{1-b}{1-\pi_\infty}\}$. For $\pi_0 \in \Pi_0$, define $\lambda_{\pi_0}^{\min}$ by

$$\lambda_{\pi_0}^{\min}(x) = \begin{cases} \lambda_{\pi_0}(x) - \delta_1 & x \in \mathcal{S}_{c_1} \\ \min\{\frac{a}{\pi_\infty}, \frac{1-b}{1-\pi_\infty}\} & x \notin \mathcal{S}_{c_1}, \end{cases} \tag{B.12}$$

and define $T_w^{\min}$ by replacing λ_{π_0} with $\lambda_{\pi_0}^{\min}$ in (IV.8), and where we choose δ_1 and c_1 sufficiently small that $\mathbb{E}(T_w^{\min})$ is finite. Therefore, if $\sup\limits_{\substack{x\in\mathcal{S}_{c_1}\\ \pi_0\in\Pi_0}} |\widehat{\lambda}_{\pi_0,\mathcal{A},m}(x) - \lambda_{\pi_0}(x)| < \delta_1$, then $|\mathbb{E}(\widetilde{T}_w(A)) - \mathbb{E}(\widetilde{U}_{t_0}(A))| \leq |\mathbb{E}(T_w^{\min}(A)) - \mathbb{E}(U_{t_0}^{\min}(A))|$.

Now choose t_0 sufficiently large that

$$\begin{aligned} |\mathbb{E}(T_w^{\min}(A)) - \mathbb{E}(U_{t_0}^{\min}(A))| &< \eta/6 \\ |\mathbb{E}(U_{t_0}(A+\varepsilon)) - \mathbb{E}(T_w(A+\varepsilon))| &< \eta/6 \\ |\mathbb{E}(U_{t_0}(A-\varepsilon)) - \mathbb{E}(T_w(A-\varepsilon))| &< \eta/6 \\ |\mathbb{E}(U_{t_0}(A)) - \mathbb{E}(T_w(A))| &< \eta/6. \end{aligned} \tag{B.13}$$

Finally, we just have to control $2t_0(1-\mathbb{P}(\mathcal{C}_{t_0,\varepsilon}))$. If $\sup\limits_{\substack{x\in\mathcal{S}_c\\ \pi_0\in\Pi_0}} |\widehat{\lambda}_{\pi_0,\mathcal{A},m}(x)-\lambda_{\pi_0}(x)| < \delta$ and $X_1,...,X_{t_0}\in\mathcal{S}_c$, then

$$\begin{aligned}
\sup_{t\le t_0}|\widetilde{R}_{t,w}-R_{t,w}| &= \sup_{t\le t_0}\left|\max_{t-m_\alpha\le k\le t}\int\limits_{\Pi_0}\prod_{i=k}^{t}\widehat{\lambda}_{\pi_0,\mathcal{A},m}(X_i)w(\pi_0)d\pi_0 - \max_{t-m_\alpha\le k\le t}\int\limits_{\Pi_0}\prod_{i=k}^{t}\lambda_{\pi_0}(X_i)w(\pi_0)d\pi_0\right| \\
&\le \sup_{t\le t_0}\max_{t-m_\alpha\le k\le t}\left|\int\limits_{\Pi_0}\prod_{i=k}^{t}\widehat{\lambda}_{\pi_0,\mathcal{A},m}(X_i)w(\pi_0)d\pi_0 - \int\limits_{\Pi_0}\prod_{i=k}^{t}\lambda_{\pi_0}(X_i)w(\pi_0)d\pi_0\right| \\
&\le \sup_{t\le t_0}\int\limits_{\Pi_0}\left(\max_{t-m_\alpha\le k\le t}\left|\prod_{i=k}^{t}\widehat{\lambda}_{\pi_0,\mathcal{A},m}(X_i)-\prod_{i=k}^{t}\lambda_{\pi_0}(X_i)\right|\right)w(\pi_0)d\pi_0 \\
&\le \max\left\{(M+\delta)^{m_\alpha}-M^{m_\alpha},\ M^{m_\alpha}-(M-\delta)^{m_\alpha}\right\}.
\end{aligned} \tag{B.14}$$

where $M=\max\left\{\frac{1}{\pi_\infty}\ ,\ \frac{1}{1-\pi_\infty}\right\}$. Choose $\delta_2<\delta_1$ such that the right hand side is at most ε, and $c_2<c_1$ such that $2t_0(1-\mathbb{P}(X_i\in\mathcal{S}_{c_2})^{t_0})<\eta/6$. Then, $\sup\limits_{\substack{x\in\mathcal{S}_{c_2}\\ \pi_0\in\Pi_0}}|\widehat{\lambda}_{\pi_0,\mathcal{A},m}(x)-\lambda_{\pi_0}(x)|<\delta_2$ implies that $|\mathbb{E}(\widetilde{T}_w(A))-\mathbb{E}(T_w(A))|<\eta$, and so

$$\mathbb{P}\left(|\mathbb{E}(\widetilde{T}_w(A))-\mathbb{E}(T_w(A))|<\eta\right)\ge\mathbb{P}\left(\sup_{\substack{x\in\mathcal{S}_{c_2}\\ \pi_0\in\Pi_0}}|\widehat{\lambda}_{\pi_0,\mathcal{A},m}(x)-\lambda_{\pi_0}(x)|<\delta_2\right)\to 1. \tag{B.15}$$

As this works for all $\eta>0$, then we conclude that $|\mathbb{E}(\widetilde{T}_w(A))-\mathbb{E}(T_w(A))|\xrightarrow{p}0$ as desired. □

B.4 Details for Example 1

Under $\mathbb{P}_\infty$, $X\sim N(0,1)$ and under $\mathbb{P}_0$, $X\sim N(\mu,1)$. Therefore,

$$\lambda(x)=\frac{\exp\{-0.5(x-\mu)^2\}}{\exp\{-0.5x^2\}}=\exp\left\{\mu x-\frac{\mu^2}{2}\right\}. \tag{B.16}$$

Let $\widehat{\mu}$ be an estimate of μ, then $\widehat{\lambda}(x)=\exp\left\{\widehat{\mu}x-\frac{\widehat{\mu}^2}{2}\right\}$.

Next, we need f^i_λ and $f^i_{\widehat{\lambda}}$. Since $P(\widehat{\lambda}(X)\le s)=P(X\le\log(s)/\mu+\mu/2)$, then $f^i_\lambda(s)=f^i_X(\log(s)/\mu+\mu/2)/(\mu s)$. Likewise, $f^i_{\widehat{\lambda}}(s)=f^i_X(\log(s)/\widehat{\mu}+\widehat{\mu}/2)/(\widehat{\mu}s)$. Since the pre- and post-change distributions are normal, then

$$f^{\infty}_{\lambda}(s) \propto \frac{1}{\mu s} \exp\left\{-\frac{1}{2}\left(\frac{\log s}{\mu} + \frac{\mu}{2}\right)^2\right\} \qquad f^{\infty}_{\widehat{\lambda}}(s) \propto \frac{1}{\widehat{\mu} s} \exp\left\{-\frac{1}{2}\left(\frac{\log s}{\widehat{\mu}} + \frac{\widehat{\mu}}{2}\right)^2\right\} \tag{B.17}$$

$$f^{0}_{\lambda}(s) \propto \frac{1}{\mu s} \exp\left\{-\frac{1}{2}\left(\frac{\log s}{\mu} - \frac{\mu}{2}\right)^2\right\} \qquad f^{\infty}_{\widehat{\lambda}}(s) \propto \frac{1}{\widehat{\mu} s} \exp\left\{-\frac{1}{2}\left(\frac{\log s}{\widehat{\mu}} + \frac{\widehat{\mu}}{2} - \mu\right)^2\right\}. \tag{B.18}$$

Clearly, (B2) is satisfied. To show that (B3) is satisfied, we prove that f^i_{λ} and $f^i_{\widehat{\lambda}}$ are Lipschitz. We have

$$\frac{d}{ds} f^{\infty}_{\lambda}(s) = \frac{1}{(\mu s)^2} \exp\left\{-\frac{1}{2}\left(\frac{\log s}{\mu} + \frac{\mu}{2}\right)^2\right\}\left(-\frac{\log s}{\mu} - \frac{3\mu}{2}\right) \tag{B.19}$$

$$\frac{d}{ds} f^{0}_{\lambda}(s) = \frac{1}{(\mu s)^2} \exp\left\{-\frac{1}{2}\left(\frac{\log s}{\mu} - \frac{\mu}{2}\right)^2\right\}\left(-\frac{\log s}{\mu} - \frac{\mu}{2}\right). \tag{B.20}$$

Since both derivatives are bounded, f^{∞}_{λ} and f^{0}_{λ} are Lipschitz. Similarly, $f^{\infty}_{\widehat{\lambda}}$ and $f^{0}_{\widehat{\lambda}}$ are Lipschitz.

Finally, to show that (B5) is bounded, and provide the upper bound in Theorem 1, we need to show that $TV(f^i_{\widehat{\lambda}}, f^i_{\lambda}) \xrightarrow{p} 0$. We have

$$|f^{\infty}_{\widehat{\lambda}}(s) - f^{\infty}_{\lambda}(s)| \leq \left|\frac{1}{\widehat{\mu}} - \frac{1}{\mu}\right| \left|\frac{1}{s} \exp\left\{-\frac{1}{2}\left(\frac{\log s}{\widehat{\mu}} + \frac{\widehat{\mu}}{2}\right)\right\}\right| \tag{B.21}$$

$$+ \frac{1}{\mu}\left|\frac{1}{s} \exp\left\{-\frac{1}{2}\left(\frac{\log s}{\widehat{\mu}} + \frac{\widehat{\mu}}{2}\right)^2\right\} - \frac{1}{s} \exp\left\{-\frac{1}{2}\left(\frac{\log s}{\mu} + \frac{\mu}{2}\right)^2\right\}\right|. \tag{B.22}$$

First,

$$\int\limits_0^\infty \left|\frac{1}{\widehat{\mu}} - \frac{1}{\mu}\right| \left|\frac{1}{s} \exp\left\{-\frac{1}{2}\left(\frac{\log s}{\widehat{\mu}} + \frac{\widehat{\mu}}{2}\right)\right\}\right| ds = O_P\left(\left|\frac{1}{\widehat{\mu}} - \frac{1}{\mu}\right|\right) = O_P(|\widehat{\mu} - \mu|). \tag{B.23}$$

Next,

$$\left|\frac{1}{s} \exp\left\{-\frac{1}{2}\left(\frac{\log s}{\widehat{\mu}} + \frac{\widehat{\mu}}{2}\right)^2\right\} - \frac{1}{s} \exp\left\{-\frac{1}{2}\left(\frac{\log s}{\mu} + \frac{\mu}{2}\right)^2\right\}\right| = |\widehat{\mu} - \mu| \left|\frac{1}{\mu^3 s} e^{-0.125(\mu^2 + 2\log s)^2/\mu^2} (\log^2(s) - 0.25\mu^4)\right|$$
$$+ O_P(|\widehat{\mu} - \mu|^2), \tag{B.24}$$

and

$$\int\limits_0^\infty |\widehat{\mu} - \mu| \left|\frac{1}{\mu^3 s} e^{-0.125(\mu^2 + 2\log s)^2/\mu^2} (\log^2(s) - 0.25\mu^4)\right| ds = O_P(|\widehat{\mu} - \mu|). \tag{B.25}$$

Therefore,

$$TV(f^i_{\widehat{\lambda}}, f^i_{\lambda}) = \int\limits_0^\infty |f^\infty_{\widehat{\lambda}}(s) - f^\infty_{\lambda}(s)| ds \leq O_P(|\widehat{\mu} - \mu|). \tag{B.26}$$

B.5 Details for Example 2

Under both pre- and post-change distributions, $X|Y = y \sim N(\boldsymbol{\mu_y}, \boldsymbol{\Sigma})$. Let $p(x) = \mathbb{P}_\infty(Y = 1|X = x)$, and let $\mathcal{A}$ be the LDA classifier with predicted probabilities $\mathcal{A}(x) = \widehat{\mathbb{P}}_\infty(Y = 1|X = x)$, given by (IV.12). Then, $\lambda(X)$ and $\widehat{\lambda}(X)$ are linear transformations of $p(X)$ and $\mathcal{A}(X)$ respectively, by (IV.2) and (IV.5). Therefore, to check the assumptions it is sufficient to consider $f_{p|Y=y}$ and $f_{\mathcal{A}|Y=y}$, the conditional densities of $p(X)|Y = y$ and $\mathcal{A}(X)|Y = y$.

First, we have

$$\begin{aligned} f_{p|Y=y}(s) &= \frac{1}{s - s^2}\phi\left(\frac{\log\left(\frac{s(1-\pi_\infty)}{(1-s)\pi_\infty}\right) + \frac{1}{2}\left(\boldsymbol{\mu_1}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_1} - \boldsymbol{\mu_0}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_0}\right) - \boldsymbol{\mu_y}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu_1} - \boldsymbol{\mu_0})}{\sqrt{(\boldsymbol{\mu_1} - \boldsymbol{\mu_0})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu_1} - \boldsymbol{\mu_0})}}\right) \\ f_{\mathcal{A}|Y=y}(s) &= \frac{1}{s - s^2}\phi\left(\frac{\log\left(\frac{s(1-\pi_\infty)}{(1-s)\pi_\infty}\right) + \frac{1}{2}\left(\widehat{\boldsymbol{\mu_1}}^T\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu_1}} - \widehat{\boldsymbol{\mu_0}}^T\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu_0}}\right) - \boldsymbol{\mu_y}^T\widehat{\boldsymbol{\Sigma}}^{-1}(\widehat{\boldsymbol{\mu_1}} - \widehat{\boldsymbol{\mu_0}})}{\sqrt{(\widehat{\boldsymbol{\mu_1}} - \widehat{\boldsymbol{\mu_0}})^T\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\Sigma}\widehat{\boldsymbol{\Sigma}}^{-1}(\widehat{\boldsymbol{\mu_1}} - \widehat{\boldsymbol{\mu_0}})}}\right), \end{aligned} \tag{B.27}$$

where ϕ is the standard normal density. For simplicity, we write this as

$$\begin{aligned} f_{p|Y=y}(s) &= \frac{1}{s - s^2}\phi\left(\frac{\log\left(\frac{s(1-\pi_\infty)}{(1-s)\pi_\infty}\right) + a}{b}\right) \\ f_{\mathcal{A}|Y=y}(s) &= \frac{1}{s - s^2}\phi\left(\frac{\log\left(\frac{s(1-\pi_\infty)}{(1-s)\pi_\infty}\right) + \widehat{a}}{\widehat{b}}\right). \end{aligned} \tag{B.28}$$

Now, we show that these densities are Lipschitz. For $p(X)|Y = y$, we have

$$\frac{d}{ds}f_{p|Y=y} = \frac{\left(-a + b^2(2s-1) - \log\left(\frac{s(1-\pi_\infty)}{(1-s)\pi_\infty}\right)\right)}{b^2(1-s^2)s^2}, \tag{B.29}$$

which is bounded. Similarly, $\frac{d}{ds}f_{\mathcal{A}|Y=y}$ is bounded. Therefore, $f_{p|Y=y}$ and $f_{\mathcal{A}|Y=y}$ are Lipschitz, so f^i_λ and $f^i_{\widehat{\lambda}}$ are all Lipschitz, which satisfies (B3).

Next, we want to show convergence in total variation distance. Note that $|f^i_\lambda - f^i_{\widehat{\lambda}}| = O_P(|f_{p|Y=1} - f_{\mathcal{A}|Y=1}|) + O_P(|f_{p|Y=0} - f_{\mathcal{A}|Y=0}|)$, so it suffices to show that $TV(f_{p|Y=y}, f_{\mathcal{A}|Y=y})$ converges. Using $a, \widehat{a}, b, \widehat{b}$ from above, we get

$$TV(f_{p|Y=y}, f_{\mathcal{A}|Y=y}) = \int |f_{\mathcal{A}|Y=y}(s) - f_{p|Y=y}(s)|ds = O_P(|\widehat{a} - a|) + O_P(|\widehat{b} - b|). \tag{B.30}$$

The right hand side converges by strong consistency of $\widehat{\boldsymbol{\mu_y}}$, $y \in \{0, 1\}$, and $\widehat{\boldsymbol{\Sigma}}^{-1}$, satisfying (B5).