When Training and Test Sets are Different: Characterising Learning Transfer

Amos J Storkey
Institute of Adaptive and Neural Computation
School of Informatics, University of Edinburgh
a.storkey@ed.ac.uk

March 8, 2013

1 Overview

In this chapter, a number of common forms of dataset shift are introduced, and each is related to a particular form of causal probabilistic model. Examples are given for the different types of shift, and some corresponding modelling approaches. By characterising dataset shift in this way, there is potential for the development of models which capture the specific types of variations, combine different modes of variation, or do model selection to assess whether dataset shift is an issue in particular circumstances. As an example of how such models can be developed, an illustration is provided for one approach to adapting Gaussian process methods for a particular type of dataset shift called Mixture Component Shift.

After the issue of dataset shift is introduced, the distinction between conditional and unconditional models is elaborated in Section 3. This difference is important in the context of dataset shift, as it will be argued in Section 5 that dataset shift makes no difference for causally conditional models. This form of dataset has been called covariate shift. In Section 6, another simple form of dataset shift is introduced: prior probability shift. This is followed by Section 7 on sample selection bias, Section 8 on imbalanced data and Section 9 on domain shift. Finally three different types of source component shift are given in Section 10. One example of modifying Gaussian process models to apply to one form of source component shift is given in Section 11. A brief discussion on the issue of determining whether shift occurs (Section 12) and on the relationship to Transfer Learning (Section 13) concludes the chapter.

2 Introduction

A camera company develops some expert pattern recognition software for their cameras but now wants to sell it for use on other cameras. Does it need to worry about the differences?

The country Albodora has done a study that shows the introduction of a particular measure has aided in curbing underage drinking. Bodalecia's politicians are impressed by the results and what to utilize Albodora's approach in their own country. Will it work?

A consultancy provides network intrusion detection software, developed using machine learning techniques on data from 4 years ago. Will the software still work as well now as it did when it was first released? If not, do the company need to do a whole further analysis, or are there some simple changes that can be made to bring the software up to scratch?

In the real world, the conditions in which we use the systems we develop will differ from the conditions in which they were developed. Typically environments are non-stationary, and sometimes the difficulties of matching the development scenario to the use are too great or too costly.

In contrast, textbook predictive machine learning methods work by ignoring these differences. They presume either that the test domain and training domain match, or that it makes no difference if they do not match. In this book we will be asking about what happens when we allow for the possibility of *dataset shift*. What happens if we are explicit in recognizing that in reality things might change from the idealized training scheme we have set up?

The scenario can be described a little more systematically. Given some data, and some modelling framework, a model can be learnt. This model can be used for making predictions $P(\mathbf{y}|\mathbf{x})$ for some targets $\mathbf{y}$ given some new $\mathbf{x}$. However, if there is a possibility that something may have changed between training and test situations, it is important to ask if a different predictive model should be used. To do this, it is critical to develop an understanding of the appropriateness of particular models in the circumstance of such changes. Knowledge of how best to model the potential changes will enable better representation of the result of these changes. There is also the question of what needs to be done do to implement the resulting process. Does the learning method itself need to be changed, or is there just post-hoc processing that can be done to the learnt model to account for the change?

The problem of dataset shift is closely related to another area of study known by various terms such as *transfer learning* or *inductive transfer*. Transfer Learning deals with the general problem of how to transfer information from a variety of previous different environments to help with learning, inference and prediction in a new environment. Dataset shift is more specific: it deals with the business of relating information in (usually) two closely related environments to help with the prediction in one given the data in the other(s).

Faced with the problem of dataset shift, we need to know what we can do. If it is possible to characterise the types of changes that occur from training to test situation, this will help in knowing what techniques are appropriate. In this chapter some of the most typical types of dataset shift will be characterised.

The aim, here, is to provide an illustrative introduction to dataset shift. There is no attempt to provide an exhaustive, or even systematic literature review: indeed the literature is far too extensive for that. Rather, the hope is that by taking a particular view on the problem of dataset shift, it will help to provide an organisational structure which will enable the large body of work in all these areas to be systematically related and analysed, and will help establish new developments in the field as a whole.

Gaussian process models will be used as illustrations in parts of this chapter. It would be foolish to reproduce an introduction to this area when there are already very comprehensible alternatives. Those who are unfamiliar with Gaussian processes, and want to follow the various illustrations, are referred to [19]. Gaussian processes are a useful predictive modelling tool with some desirable properties. They are directly applicable to regression problems, and can be used for classification via

logistic transformations. Only the regression case will be discussed here.

3 Conditional and Generative Models

This chapter will describe methods for dataset shift using probabilistic models. A probabilistic model relates the variables of interest by defining a joint probability distribution for the values those variables take. This distribution determines which values of the variables are more or less probable, and hence how particular variables are related: it may be that the probability that one variable takes a certain value is very dependent on the state of another. A good model is a probability distribution that describes the understanding and the occurrence of those variables well. Very informally, a model that assigns low probability to things that are not observed and relationships that are forbidden or unlikely and high probability to observed and likely items is favoured over a model that does not.

In the realm of probabilistic predictive models it is useful to make a distinction between conditional and generative models. The term generative model will be used to refer to a probabilistic model (effectively a joint probability distribution) over all the variables of interest (including any parameters). Given a generative model we can generate artificial data from the model by sampling from the required joint distribution, hence the name. A generative model can be specified using a number of conditional distributions. Suppose the data takes the form of covariate $\mathbf{x}$ and target $\mathbf{y}$ pairs. Then by way of example, $P(\mathbf{y}, \mathbf{x})$ can be written as $P(\mathbf{x}|\mathbf{y})P(\mathbf{y})$, and may also be written in terms of other hidden *latent* variables which are not observable. For example we could believe the distribution $P(\mathbf{y}, \mathbf{x})$ depends on some other factor $\mathbf{r}$ and we would write

$$P(\mathbf{y}, \mathbf{x}) = \int d\mathbf{r} P(\mathbf{y}, \mathbf{x}|\mathbf{r})P(\mathbf{r}) \tag{1}$$

where the integral is a *marginalisation* over the $\mathbf{r}$, which simply means that as $\mathbf{r}$ is never known it needs to be integrated over in order to obtain the distribution for the observable quantities $\mathbf{y}$ and $\mathbf{x}$. Necessarily distributions must also be given for any latent variables.

Conditional models are not so ambitious. In a conditional model the distribution of some smaller set of variables is given for each possible known value of the other variables. In many useful situations (such as regression) the value of certain variables (the covariates) is always known, and so there is no need to model them. Building a conditional model for variables $\mathbf{y}$ given other variables $\mathbf{x}$ implicitly factorises the joint probability distribution over $\mathbf{x}$ and $\mathbf{y}$, as well as parameters (or latent variables) Θ_x and Θ_y, as $P(\mathbf{y}|\mathbf{x}, \Theta_y)P(\mathbf{x}|\Theta_x)P(\Theta_y)P(\Theta_x)$. If the values of $\mathbf{x}$ are always given, it does not matter how good the model $P(\mathbf{x})$ is: it is never used in any prediction scenario. Rather, the quality of the conditional model $P(\mathbf{y}|\mathbf{x})$ is all that counts, and so conditional models only concern themselves with this term. By ignoring the need to model the distribution of $\mathbf{x}$ well, it is possible to choose more flexible model parameterisations than with generative models. Generative models are required to tractably model both the distributions over $\mathbf{y}$ and $\mathbf{x}$ accurately. Another advantage of conditional modelling is that the fit of the predictive model $P(\mathbf{y}|\mathbf{x})$ is never compromised in favour of a better fit of the unused model $P(\mathbf{x})$ as they are decoupled.

If the generative model actually accurately specifies a known generative process for the data, then the choice of modelling structure may fit the real constraints much better than a conditional model and hence result in a more accurate parameterisation. In these situations generative models may fare better than conditional

ones. The general informal consensus is that in most typical predictive modelling scenarios standard conditional models tend to result in lower errors than standard generative models. However this is no hard rule and is certainly not rigorous.

It is easy for this terminology to get confusing. In the context of this chapter we will use the term conditional model for any model that factorises the joint distribution (having marginalised for any parameters) as $P(\mathbf{y}|\mathbf{x})P(\mathbf{x})$, and unconditional model for any other form of factorisation. The term generative model will be used to refer to any joint model (either of conditional or unconditional form) which is used to represent the whole data in terms of some useful factorisation, possibly including latent variables. In most cases the factorised form will represent a (simplified) causal generative process. We may use the term causal graphical model in these situations to emphasise that the structure is more than just a representation of some particular useful factorisation, but is presumed to be a factorisation that respects the way the data came about.

It is possible to analyse data using a model structure that is not a causal model but still has the correct relationships between variables for a static environment. One consequence of this is that it is perfectly reasonable to use a conditional form of model for domains that are not causally conditional: many forms of model can be statistically equivalent. If the $P(\mathbf{x})$ does not change then it does not matter. Hence conditional models can perform well in many situations where there is no dataset shift regardless of the underlying beliefs about the generation process for the data. However in the context of dataset shift, there is presumed to be an interventional change to some (possibly latent) variable. If the true causal model is not a conditional model, then this change will implicitly cause a change to the relationship $P(\mathbf{y}|\mathbf{x})$. Hence the learnt form of the conditional model will no longer be valid. Recognition of this is vital: just because a conditional model performs well in the context of no dataset shift does not imply its validity or capability in the context of dataset shift.

4 Real-Life Reasons for Dataset Shift

Whether using unconditional or conditional models, there is a presumption that the distributions they specify are static; i.e. they do not change between the time we learn them and the time we use them. If this is not true, and the distributions change in some way, then we need to model for that change, or at least the possibility of that change. To postulate such a model requires an examination of the reasons why such a shift may occur.

Though there are no doubt an infinite set of potential reasons for these changes, there are a number of ways of collectively characterising many forms of shift into qualitatively different groups. The following will be discussed in this chapter:

Simple Covariate Shift is when only the distributions of covariates $\mathbf{x}$ change and everything else is the same.

Prior Probability Shift is when only the distribution over $\mathbf{y}$ changes and everything else stays the same.

Sample Selection Bias is when the distributions differ as a result of an unknown sample rejection process.

Imbalanced Data is a form of deliberate dataset shift for computational or modelling convenience.

Domain Shift involves changes in measurement.

Source Component Shift involves changes in strength of contributing components.

Each of these relates to a different form of model. Unsurprisingly, each form suggests a particular approach for dealing with the change. As each model is examined in the proceeding sections, the particular nature of the shift will be explained, some of the literature surrounding that type of dataset shift will be mentioned, and a graphical illustration of the overall model will be given. The graphical descriptions will take a common form: they will illustrate the probabilistic graphical (causal) model for the generative model. Where the distributions of a variable may change between train and test scenarios, the corresponding network node is darkened. Each figure will also illustrate data undergoing the particular form of shift by providing samples for the training (light) and test (dark) situations. These diagrams should quickly illustrate the type of change that is occurring. In the descriptions, a subscript tr will denote a quantity related to the training scenario, and a subscript te will denote a quantity relating to the test scenario. Hence $P_{\text{tr}}(\mathbf{y})$ and $P_{\text{te}}(\mathbf{y})$ are the probability of $\mathbf{y}$ in training and test situations respectively.

5 Simple Covariate Shift

The most basic form of dataset shift occurs when the data is generated according to a model $P(\mathbf{y}|\mathbf{x})P(\mathbf{x})$ and where the distribution $P(\mathbf{x})$ changes between training and test scenarios. As only the covariate distribution changes, this has been called covariate shift [20]. See Figure 1 for an illustration of the form of causal model for covariate shift.

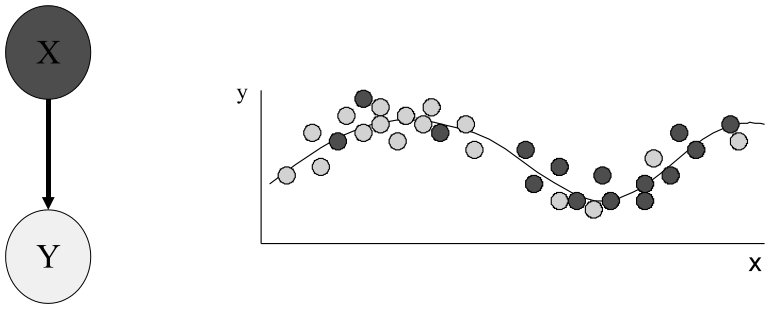

Figure 1: Simple Covariate Shift. Here the causal model indicated the targets $\mathbf{y}$ are directly dependent on the covariates $\mathbf{x}$. In other words the predictive function and noise model stays the same, it is just the typical locations $\mathbf{x}$ of the points at which the function needs to be evaluated that change. In this figure and throughout the causal model is given on the left with the node that varies between training and test made darker. To the right is some example data, with the training data in shaded light and the test data shaded dark.

A typical example of covariate shift occurs in assessing the risk of future events given current scenarios. Suppose the problem was to assess the risk of lung cancer in 5 years ($\mathbf{y}$) given recent past smoking habits ($\mathbf{x}$). In these situations we can be sure that the occurrence or otherwise of *future* lung cancer is not a causal factor of *current* habits. So in this case a conditional relationship of the form $P(\mathbf{y}|\mathbf{x})$ is a reasonable causal model to consider[1]. Suppose now that changing circumstances

[1]Of course there are always possible confounding factors, but for the sake of this illus-

(e.g. a public smoking ban) affect the distribution over habits $\mathbf{x}$. How do we account for that in our prediction of risk for a new person with habits $\mathbf{x}^*$?

It will perhaps come as little surprise that the fact that the covariate distribution changes should have no effect on the model $P(\mathbf{y}|\mathbf{x}^*)$. Intuitively this makes sense. The smoking habits of some person completely independent of me should not affect my risk of lung cancer if I make no change at all. From a modelling point of view we can see this from our earlier observation in the static case this is simply a conditional model: it gives the same prediction for given $\mathbf{x}$, $P(\mathbf{y}|\mathbf{x})$ regardless of the distribution $P(\mathbf{x})$. Hence in the case of dataset shift, it still does not matter what $P(\mathbf{x})$ is, or how it changes. The prediction will be the same.

This may seem a little laboured, but the point is important to make in the light of various pieces of recent work that suggest there are benefits in doing something different if covariate shift occurs. The claim is that if the class of models that is being considered for $P(\mathbf{y}|\mathbf{x})$ does not contain the true conditional model, then improvements can be gained by taking into account the nature of the covariate shift. In the next section we examine this, and see that this work effectively makes a change of global model class for $P(\mathbf{y}|\mathbf{x})$ between the training and test cases. This is valuable as it makes it clear that if the desire is (asymptotic) risk minimisation for a constant modelling cost, then there may be gains to be made by taking into account the test distribution. Following this discussion we show that Gaussian processes are nonparametric models that truly are conditional models, in that they satisfy Kolmogorov Consistency. This same characteristic does not follow for probabilistic formulations of Support Vector Classifiers.

5.1 Is there really no modelling implication?

There are a number of recent papers that have suggested that something different does need to be done in the context of covariate shift. For example in [20], the author proposes an importance reweighting of data points in their contribution to the estimator error: points lying in regions of high test density are more highly weighted that those in low density regions. This was extended in [26], with the inclusion of a generalisation error estimation method for this process of adapting for covariate shift. In [24, 25], the importance re-weighting is made adaptable on the basis of cross-validation error.

The papers make it clear that there *is* some benefit to be obtained by doing something different in the case of covariate shift. The argument here is that these papers indicate a computational benefit rather than a fundamental modelling benefit. These papers effectively compare different global model classes for the two cases: case one, where covariate shift is compensated for, and case two where covariate shift is not compensated for. This is not immediately obvious because the apparent model class is the same. It is just that in compensating for covariate shift the model class is utilised locally (the model does not need to account for training data that is seen but is outside the support of the test data distribution), whereas when not compensating the model class is used globally.

As an example, consider using a linear model to fit nonlinear data (Figure 2a). When not compensating for covariate shift, we obtain the fit given by the dashed line. When compensating for covariate shift, we get the fit given by the solid line. In the latter case, there is no attempted explanation for much of the observed

tration we choose to ignore that for now. It is also possible the samples are not drawn independently and identically distributed due to population effects (e.g. passive smoking) but that too is ignored here.

training data, which is fit very poorly by the model. Rather the model class is being used locally. As a contrast consider the case of a local linear model (Figure 2b). Training the local linear model explains the training data well, and the test data well. However only one of the local linear components is really used when doing prediction. Hence the effort spent computing the linear components for regions outside of the support of the test data was wasted.

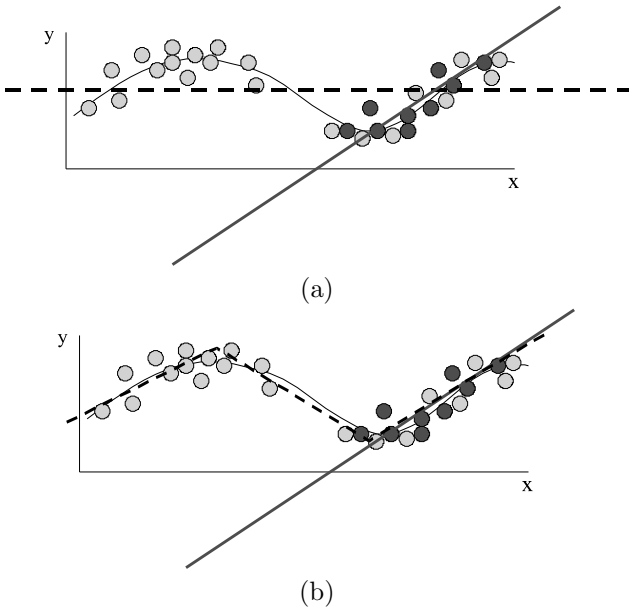

Figure 2: Covariate shift for mis-specified models: (a) The linear model is a poor fit to the global data (dashed line). However by focussing on the local region associated with the test data distribution the fit (full line) is much better as a local linear model is more appropriate. (b) The global fit for a local linear model is more reasonable, but involves the computation of many parameters that are never used in the prediction.

There are a number of important contributions that stem from the recent study of covariate shift. It clarifies that there are potential computational advantages of adjusting for covariate shift due to the fact that it may be possible to use a simpler model class but only focus on a local region relevant to the test scenario, rather than worrying about the global fit of the model. There is no need to compute parameters for a more complicated global model, or for a multitude of local fits that are never used. Furthermore it also makes use of an issue in semi-supervised learning: the nature of the clusters given by the test distribution might be an indicator of a data region that can be modelled well by a simple model form.

There is a another contention that is certainly worth considering here. Some might argue that there are situations where there can be strong a priori knowledge about the model form for the test data, but very little knowledge about the model form for the training data, as that may, for example, be contaminated with a number of other data sources about which little is known. In this circumstance it seems that it is vital to spend the effort modelling the known form of model for the test region, ignoring the others. This is probably a very sound policy. Even so, there is still the possibility that even the test region is contaminated by these other sources. If it is

possible to untangle the different sources this could serve to improve things further. This is discussed more in the context of source component shift.

5.2 Gaussian Processes and Conditional Modelling

Suppose instead of using a linear model, a Gaussian process is used. How can we see that this really is a conditional model where the distribution of the covariates has no effect on the predictions? This follows from the fact that no matter what other covariate samples we see, the prediction for our current data remains the same; that is, Gaussian processes satisfy Kolmogorov consistency:

$$P(\{y_i\}|\{\mathbf{x}_i\},\{\mathbf{x}^k,y^k\}) = \int dy^* P(\{y_i\},y^*|\{\mathbf{x}_i\},\mathbf{x}^*,\{\mathbf{x}^k,y^k\}) \quad (2)$$

$$= P(\{y_i\}|\{\mathbf{x}_i\},\mathbf{x}^*,\{\mathbf{x}^k,y^k\}) \quad (3)$$

where (2) results from the definition of a Gaussian process, and (3) from basic probability theory (marginalisation). In this equation the y_i are the test targets, $\mathbf{x}_i$ the test covariates, $\mathbf{x}^k$ and y^k, the training data, and $\mathbf{x}^*, y^*$ a potential extra training point. However we never know the target y^* and so it is marginalised over. The result is that introducing the new covariate point $\mathbf{x}^*$ has had no predictive effect.

Using Gaussian processes in the usual way involves training on all the data points: the estimated conditional model $P(\mathbf{y}|\mathbf{x})$ has made use of all the available information. If one of the data points was downweighted (or removed altogether) the effect would simply be greater uncertainty about the model, resulting in a broader posterior distribution over functions.

It may be considered easier to specify a model class for a local region than a model class for the data as a whole. Practically this may be the case. However by specifying that a particular model may be appropriate for any potential local region, we are effectively specifying a model form for each different region of space. This amounts to specifying a global model anyway, and indeed one derivation of the Gaussian process can be obtained from infinite local radial basis function models [7].

Are all standard nonparametric models also conditional models? In fact some common models are not: the Support Vector Machine (SVM) classifier does not take this form. In [21, 22], it is shown that in order for the support vector machine to be defined as a probabilistic model, a global compensation factor needs to be made due to the fact that the SVM classifier does not include a normalisation term in its optimisation. One immediate consequence of this compensation is that the probabilistic formulation of the SVM does not satisfy Kolmogorov consistency. Hence the SVM is dependent on the density of the covariates in its prediction.

This can be shown, purely by way of example, for the linear SVM regression case. Generalisations are straightforward. We present an outline argument here, following the notation in [19]. The linear support vector classifier maximises

$$\exp\left(-\sum_{i=1}^{N}(1-y_i(\mathbf{w}^T.\mathbf{x}_i)_+\right)\exp\left(-\frac{1}{2C}|\mathbf{w}|^2\right). \quad (4)$$

where C is some constant, y_i are the training targets, $\mathbf{x}_i$ are the covariates (augmented with an addition unit attribute) and $\mathbf{w}$ the linear parameters. The $(.)_+$ notation is used to denote the function $(x)_+ = x$ iff $x > 0$ and is zero otherwise.

Equation (4) can be rewritten as

$$\left[\prod_{i=1}^{N} \frac{1}{Z_i(\mathbf{w})} \exp(-(1 - y_i(\mathbf{w}^T\mathbf{x}_i)_+)\right] Z(\mathbf{w}) \exp\left(-\frac{1}{2C}|\mathbf{w}|^2\right). \tag{5}$$

where $Z_N = \prod_{i=1}^{N} Z_i(\mathbf{w})$, and $Z_i(\mathbf{w}) = \sum_{y_i=\pm 1} \exp(-(1 - y_i(\mathbf{w}^T\mathbf{x}_i)_+)$ is a normalisation constant, so now

$$\frac{1}{Z_i(\mathbf{w})} \exp(-(1 - y_i(\mathbf{w}^T\mathbf{x}_i)_+) \stackrel{\text{def}}{=} P(y_i|\mathbf{w}) \tag{6}$$

can be interpreted as a probability. Hence the support vector objective can be written

$$\left[\prod_{i=1}^{N} P(y_i|\mathbf{w})\right] Z_N(\mathbf{w}) \exp\left(-\frac{1}{2C}|\mathbf{w}|^2\right). \tag{7}$$

Consider the cases $N = N^*$ and $N = N^* + 1$. Starting with the latter, marginalization over y_{N^*+1} is now straightforward as it only occurs as a probability. So the marginal objective is now

$$\left[\prod_{i=1}^{N^*} P(y_i|\mathbf{w})\right] Z_{N^*+1}(\mathbf{w}) \exp\left(-\frac{1}{2C}|\mathbf{w}|^2\right). \tag{8}$$

Marginalising out over w_{N^*+1} then gives

$$\left[\prod_{i=1}^{N^*} P(y_i|\mathbf{w})\right] Z_{N^*+1}(\mathbf{w}) \exp\left(-\frac{1}{2C}|\mathbf{w}|^2\right). \tag{9}$$

However $Z_{N^*+1}(\mathbf{w}) \neq Z_{N^*}(\mathbf{w})$ due to the extra product term. Specifically the dependence on $\mathbf{w}$ is different, so the objective (9) does not match the objective (7) for $N = N^*$. Hence the support vector objective for the case of an unknown value of target at a given point is different from the objective function without considering that point. The standard probabilistic interpretation of the support vector classifier does not satisfy Kolmogorov consistency, and seeing a covariate at a point will affect the objective function even if there is no knowledge of the target at that point. Hence the SVM classifier is in some way dependent on the covariate density, as it is dependent purely on the observation of covariates themselves.

6 Prior Probability Shift

Prior probability shift is a common issue in simple generative models. A popular example stems from the availability of naive Bayes models for the filtering of spam email. In cases of Prior Probability Shift, an assumption is made that a causal model of the form $P(\mathbf{x}|\mathbf{y})P(\mathbf{y})$ is valid (see Figure 3) and Bayes rule is used to inferentially obtain $P(\mathbf{y}|\mathbf{x})$. Naive Bayes is one model that makes this assumption. The difficulty occurs if the distribution $P(\mathbf{y})$ changes between training and test situations. As $\mathbf{y}$ is what we are trying to predict it is unsurprising that this form of dataset shift will affect the prediction.

For a known shift in $P(\mathbf{y})$, prior probability shift is easy to correct for. As it is presumed that $P(\mathbf{x}|\mathbf{y})$ does not change, this model can be learnt directly from the training data. However the learnt $P_{\text{tr}}(\mathbf{y})$ is no longer valid, and needs to be replaced by the known prior distribution in the test scenario $P_{\text{te}}(\mathbf{y})$.

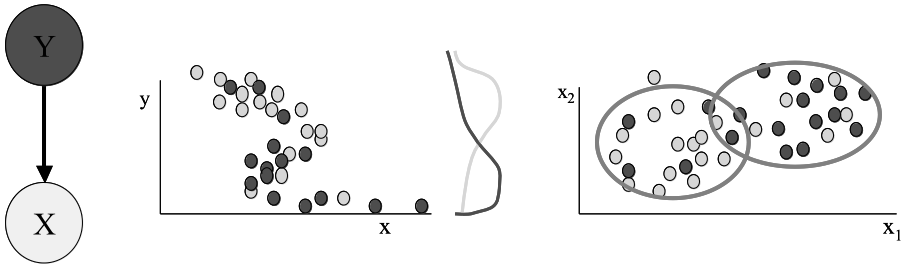

Figure 3: Prior Probability Shift. Here the causal model indicated the covariates $\mathbf{x}$ are directly dependent on the predictors $\mathbf{y}$. The distribution over $\mathbf{y}$ can change, and this effects the predictions in both the continuous case (left) and the class conditional case (right).

If, however, the distribution $P_{\text{te}}(\mathbf{y})$ is not known for the test scenario, then the situation is a little more complicated. Making a prediction

$$P(\mathbf{y}|\mathbf{x}) = \frac{P(\mathbf{x}|\mathbf{y})P(\mathbf{y})}{P(\mathbf{x})} \tag{10}$$

is not possible without knowledge of $P(\mathbf{y})$. But given the model $P(\mathbf{x}|\mathbf{y})$ and the covariates for the test data, certain distributions over $\mathbf{y}$ are more or less likely. Consider the spam filter example again. If in the test data, the vast majority of the emails contain spammy words, rather than hammy words, we would rate $P(\text{spam}) = 0$ as an unlikely model compared with other models such as $P(\text{spam}) = 0.7$. In saying this we are implicitly using some a priori model of what distributions $P(\text{spam})$ are acceptable to us, and then using the data to refine this model.

Restated, to account for prior probability shift where the precise shift is unknown a prior distribution over valid $P(\mathbf{y})$ can be specified, and the posterior distribution over $P(\mathbf{y})$ computed from the test covariate data. Then the predicted target is given by the sum of the predictions obtained for each $P(\mathbf{y})$ weighted by the posterior probability of $P(\mathbf{y})$.

Suppose $P(\mathbf{y})$ is parameterised by θ, and a prior distribution for $P(\mathbf{y})$ is defined through a prior on the parameters $P(\theta)$. Also assume that the model $P_{\text{tr}}(\mathbf{x}|\mathbf{y})$ has been learnt from the training data. Then the prediction taking into account the parameter uncertainty and the observed test data is

$$P(\mathbf{y}_1|\{\mathbf{x}_i\}) = \int d\theta P(\mathbf{y}_1|\mathbf{x}_1, \theta) P_{\text{te}}(\theta|\{\mathbf{x}_i\}) \tag{11}$$

$$= \int d\theta \frac{P_{\text{tr}}(\mathbf{x}_1|\mathbf{y}_1)P(\mathbf{y}_1|\theta)}{P_{\text{tr}}(\mathbf{x}_1|\theta)} P_{\text{te}}(\theta|\{\mathbf{x}_i\}) \tag{12}$$

where

$$P_{\text{te}}(\theta|\{\mathbf{x}_i\}) \propto \prod_i \sum_{\mathbf{y}_i} P_{\text{tr}}(\mathbf{x}_i|\mathbf{y}_i)P(\mathbf{y}_i|\theta)P(\theta) \tag{13}$$

and where i counts over the test data, i.e. these computations are done for the targets $\mathbf{y}_i$ for test points $\mathbf{x}_i$. The ease with which this can be done depends on how many integrals or sums are tractable, and whether the posterior over θ can be represented compactly.

7 Sample Selection Bias

Sample selection bias is a statistical issue of critical importance in numerous analyses. One particular area where selection bias must be considered is survey design. Sample selection bias occurs when the training data points $\{\mathbf{x}_i\}$ (the sample) do not accurately represent the distribution of the test scenario (the population) due to a selection process for each item i that is (usually implicity) dependent on the target variable $\mathbf{y}_i$.

In doing surveys, the desire is to estimate population statistics by surveying a small sample of the population. However, it is easy to set up a survey that means that certain groups of people are less likely to be included in the survey than others because, either they refuse to be involved, or they were never in a position to ask to be involved. A typical street survey, for example, is potentially biased against people with poor mobility who may be more likely to be using other transport methods than walking. A survey in a train station is more likely to catch people engaging in leisure travel than busy commuters with optimized journeys who may refuse to do the survey for lack of time.

Sample selection bias is certainly not restricted to surveys. Other examples include estimating the average speed of drivers by measuring the speeds of cars passing a stationary point on a motorway; more fast drivers will pass the point than slow drivers, simply on account of their speed. In any scenario relying on measurement from sensors, sensor failure may well be more likely in environmental situations that would cause extreme measurements. Also the process of data cleaning can itself introduce selection bias. For example, in obtaining handwritten characters, completely unintelligible characters may be discarded. But it may be that certain characters are more likely to be written unclearly.

Sample selection bias is also the cause of the well known phenomenon called "regression to the mean." Suppose that a particular quantity of importance (e.g. number of cases of illness X) is subject to random variations. However that circumstance could also be affected by various causal factors. Suppose two that, across the country, the rate of illness X is measured, and is found to be excessive in particular locations Y. As a result of that, various measures are introduced to try to curb the number of illnesses in these regions. The rate of illnesses are measured again and, lo and behold, things have improved and regions Y no longer have such bad rates of illnesses. As a result of that change it is tempting for the uninitiated to conclude that the measures were effective. However as the regions Y were chosen on the basis of a statistic that is subject to random fluctuations, and the regions were chosen because this statistic took an extreme value, even if the measures had no effect at all the illness rates would be expected to reduce at the next measurement precisely because of the random variations. This is sample selection bias because the sample taken to assess improvement was precisely the sample that was most likely to improve anyway. The issue of reporting bias is also a selection bias issue. "Interesting" positive results are more likely to be reported than "boring" negative ones.

The graphical model for sample selection bias is illustrated in Figure 4. Consider two models: P_{tr} denotes the model for the training set, and P_{te} the model for the test set. For each datum $(\mathbf{x}, \mathbf{y})$ in the training set:

$$P_{\text{tr}}(\mathbf{y}, \mathbf{x}) = P(\mathbf{y}, \mathbf{x}|v = 1) = P(v = 1|\mathbf{y}, \mathbf{x})P(\mathbf{y}|\mathbf{x})P(\mathbf{x}) \tag{14}$$

and for each datum in the test set:

$$P_{\text{te}}(\mathbf{y}, \mathbf{x}) = P(\mathbf{y}, \mathbf{x}) = P(\mathbf{y}|\mathbf{x})P(\mathbf{x}). \tag{15}$$

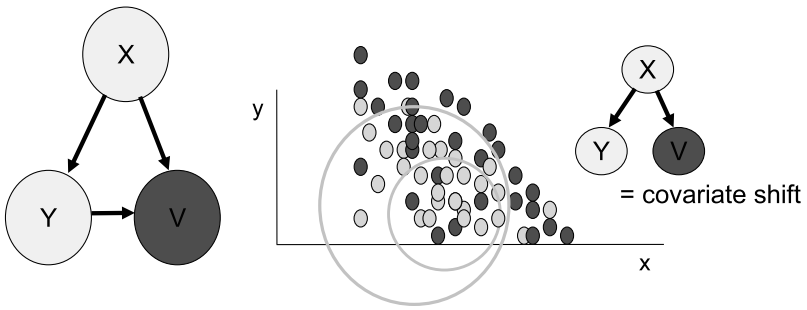

Figure 4: Sample Selection Bias. The actual observed training data is different from the test data because some of the data is more likely to be excluded from the sample. Here v denotes the selection variable, and an example selection function is given by the equiprobable contours. The dependence on $\mathbf{y}$ is crucial as without it there is no bias and this becomes a case of simple covariate shift.

Here v is a binary selection variable that decides whether a datum would be included in the training sample process ($v = 1$) or rejected from the training sample ($v = 0$).

In much of the sample selection literature this model has been simplified by assuming

$$P(\mathbf{y}|\mathbf{x}) = P(\boldsymbol{\epsilon} = \mathbf{y} - \mathbf{f}(\mathbf{x})) \text{ and} \tag{16}$$

$$P(v = 1|\mathbf{y}, \mathbf{x}) = P(\nu > g(\mathbf{x})|\mathbf{y} - f(\mathbf{x})) = P(\nu > g(\mathbf{x})|\boldsymbol{\epsilon}) \tag{17}$$

for some densities $P(\boldsymbol{\epsilon})$ and $P(\nu|\boldsymbol{\epsilon})$, function g and map $\mathbf{f}$. The issue is to model $\mathbf{f}$, which is the dependence of the targets $\mathbf{y}$ on covariates $\mathbf{x}$, while also modelling for g, which produces the bias. In words the model says there is a (multivariate) regression function for $\mathbf{y}$ given covariates $\mathbf{x}$, where the noise is independent of $\mathbf{x}$. Likewise equation (17) describes a classification function for the selection variable v in terms of $\mathbf{x}$, but where the distribution is dependent on the deviation of $\mathbf{y}$ from its predictive mean. Note that in some of the literature, there is an explicit assumption that v depends on some features in addition to $\mathbf{x}$ that control the selection. Here this is simplified by including these features in $\mathbf{x}$ and adjusting the dependence encoded by $\mathbf{f}$ accordingly.

Study of sample selection bias has a long history. [8] proposed the first solution to the selection bias problem which involved presuming $\mathbf{y} = y$ is scalar (hence also $\boldsymbol{\epsilon} = \epsilon$ and $\mathbf{f} = f$), f and g are linear, and the joint density $P(\epsilon, \nu) = P(\epsilon)P(\nu|\epsilon)$ is Gaussian. Given this the likelihood of the parameters can be written down for a given complete dataset (a dataset including the rejected samples). However in computing the maximum likelihood solution for the regression parameters, it turns out the rejected samples are not needed. Note that in the case that ϵ and μ are independent, and $P(\epsilon, \nu) = P(\epsilon)P(\mu)$, there is no predictive bias, and this is then a case of simple covariate shift.

Since the seminal paper by Heckman, many other related approaches have been proposed. These include those that relax the Gaussianity assumption for μ and σ, most commonly by mapping the Gaussian variables through a known nonlinearity before using them [16] and using semi-parametric methods directly on $P(\epsilon|\nu)$ [9]. More recent methods include [31], where the author focuses on the case where $P(v|\mathbf{x}, \mathbf{y}) = P(v|\mathbf{y})$, [6] which looks at maximum entropy density estimation under selection bias and [11] which focuses on using additional unbiased covariate data to help estimate the bias. More detailed analysis of the historical work on selection bias is available in [29] and a characterisation of types of selection bias is given in [10].

8 Imbalanced Data

It is quite possible to have a multiclass machine learning problem where one or more classes is very rare compared with others. This is called the problem of *imbalanced data*. Indeed the prediction of rare events (e.g. loan defaulting) often provide the most challenging problems. This imbalanced data problem is a common cause of dataset shift *by design*.

If the prediction of rare events is the primary issue, to use a balanced dataset may involve using a computationally infeasible amount of data just in order to get enough rare cases to be able to characterize the class accurately. For this reason it is common to "balance" the training dataset by throwing away data from the common classes so that there is an equal amount of data corresponding to each of the classes under consideration. Note that here, the presumption is not that the model would not be right for the imbalance data, rather that is is computationally infeasible to use the imbalanced data. However the data corresponding to the common class is discarded, simply because typically that is less valuable: the common class may already be easy to characterise fairly well as it has large amounts of data already.

The result of discarding data, though, is that the distribution in the training scenario no longer matches the imbalanced test scenario. However it is this imbalanced scenario that the model will be used for. Hence some adjustment needs to be made to account for the deliberate bias that is introduced. The graphical model for imbalanced data is shown in Figure 5 along with a two class example.

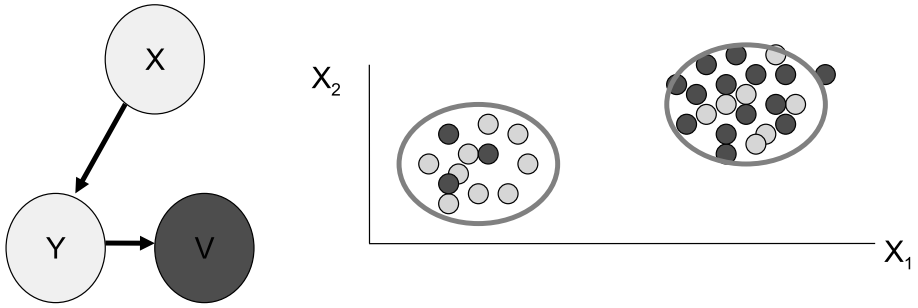

Figure 5: Imbalanced Data: imbalanced data is sample selection bias with a designed known bias that is dependent on only the class label. Data from more common classes is more likely to be rejected in the training set in order to balance out the number of cases of each class.

In the conditional modelling case, dataset shift due to re-balancing imbalanced data is just the sample selection bias problem with a known selection bias (as the selection bias was by design not by constraint or accident). In other words, we have selected proportionally more of one class of data than another precisely for no reason other than the class of the data. Variations on this theme can also be seen in certain types of stratified random surveys where particular strata are oversampled because they are expected to have a disproportionate effect on the statistics of interest, and so need a larger sample to increase the accuracy with which their effect is measured.

In a target-conditioned model (of the form $P(\mathbf{x}|\mathbf{y})P(\mathbf{y})$), dataset shift due to imbalanced data is just prior probability shift with a known shift. This is very simple to adjust for as only $P(\mathbf{y})$ needs to be changed. This simplicity can mean that some people choose generative models over conditional models for imbalanced data problems. Because the imbalance is decoupled from the modelling it is transparent that the imbalance itself will not affect the learnt model.

In a classification problem, the output of a conditional model is typically viewed as a probability distribution over class membership. The difficulty is that these probability distributions were obtained on training data that was biased in favour of rare classes compared to the test distribution. Hence the output predictions need to be weighted by the reciprocal of the known bias and renormalised in order to get the correct predictive probabilities. In theory these renormalised probabilities should be used in the likelihood and hence in any error function being optimised.

In practice it is not uncommon for the required reweighting to be ignored, either through naivety, or due to the fact that the performance of the resulting classifier appears to be better. This is enlightening as it illustrates the importance of not simply focusing on the probabilistic model without also considering the decision theoretic implications. By incorporating a utility or loss function a number of things can become apparent. First, predictive performance on the rare classes is often more important than that on common classes. For example in emergency prediction, we prefer to sacrifice a number of false positives for the benefit of another true positive. By ignoring the reweighting, the practitioner is saying that the bias introduced by the balancing matches the relative importance of false positives and true positives. Furthermore introduction of a suitable loss function can reduce the problem where a classifier puts all the modelling effort into improving the many probabilities that are already nearly certain at the sacrifice of the small number of cases associated with the rarer classes. Most classifiers share a number of parameters between predictors of the rare and common classes. It is easy for the optimisation of those parameters to be swamped by the process of improving the probability of the prediction of the common classes at the expense of any accuracy on the rare classes. However the difference between a probability of 0.99 and 0.9 may not make any difference to what we do with the classifier and so actually makes no difference to the real results obtained by using the classifier, if predictive probabilities are actually going to be ignored in practice.

Once again the literature on imbalanced data is significant, and there is little chance of doing the field great justice in this small space. In [5] the authors give an overview of the content of a number of workshops in this area, and the papers referenced provide an interesting overview of the field. One paper [13] from the AAAI workshops looks at a number of different strategies for Learning from Imbalanced Datasets. SMOTE [4] is a more recent approach that has received some attention. In [1] the authors look at the issue of imbalanced data specifically in the context of support vector machines, and an earlier paper [30] also focusses on support vector machines and considers the issue of data imbalance while discussing the balance between sensitivity and specificity. In the context of linear program boosting, the paper [17] considers the implications of imbalanced data, and tests this on a text classification problem. As costs and probabilities are intimately linked, the paper [32] discusses how to jointly deal with these unknowns. The fact that adjusting class probabilities does make a practical difference can be found in [15]. Further useful analysis of the general problem can be found in [13].

9 Domain Shift

In life, the meaning of numbers can change. Inflation reduces the value of money. Lighting changes can effect the appearance of a particular colour or the meaning of a position can change dependent on the current frame of reference. Furthermore there is often the possibility of changes in measurement units. All of these can cause dataset shift. We call this particular form of dataset shift *domain shift*. This term is borrowed from linguistics, where it refers to changes in the domain of discourse. The

same entity can be referred to in different ways in different domains of discourse: for example, in one context metres might be an obvious unit of measurement, and in another inches may be more appropriate.

Domain shift is characterized by the fact that the measurement system, or method of description, can change. One way to understand this is to postulate some underlying unchanging latent representation of the covariate space. We denote a latent variable in this space by $\mathbf{x}_0$. Such a variable could, for example, be a value in yen indexed adjusted to a fixed date. The predictor variable $\mathbf{y}$ is dependent on this latent $\mathbf{x}_0$. The difficulty is that we never observe $\mathbf{x}_0$. We only observe some map $\mathbf{x} = \mathbf{f}(\mathbf{x}_0)$ into the observable space. And that map can change between training and test scenarios.

Modelling for domain shift involves estimating the map between representations using the distributional information. A good example of this is estimating gamma correction for photographs. Gamma correction is a specific parametric nonlinear map of pixel intensities. Given two unregistered photographs of a similar scene from different cameras, the appearance may be different due to the camera gamma calibration or due to postprocessing. By optimising the parameter to best match the pixel distributions we can obtain a gamma correction such that the two photographs are using the same representation. A more common scenario is that a single camera moves from a well lit to a badly lit region. In this context, gamma correction is correction for changes due to lighting - an estimate of the gamma correction needed to match some idealized pixel distribution can be computed. Another form of explicit density shift include estimating doppler shift from diffuse sources.

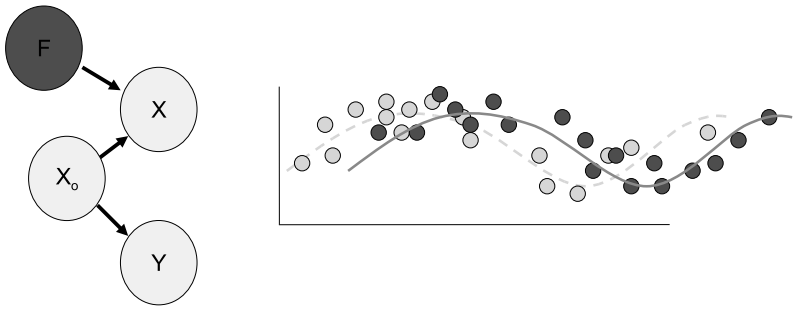

Figure 6: Domain Shift: The observed covariates $\mathbf{x}$ are transformed from some idealised covariates $\mathbf{x}_0$ via some transformation F, which is allowed to vary between datasets. The target distribution $P(\mathbf{y}|\mathbf{x}_0)$ is unchanged between test an training datasets, but of course the distribution $P(\mathbf{y}|\mathbf{x}_0)$ does change if F changes.

10 Source Component Shift

Source Component Shift may be the most common form of dataset shift. In the most general sense it simply states that the observed data is made up from data from a number of different sources, each with their own characteristics, and the proportions of those sources can vary between training and test scenarios.

Source component shift is ubiquitous: a particular product is produced in a number of factories, but the proportions sourced from each factory varies dependent on a retailer's supply chain; voting expectations vary depending on type of work, and different places in a country have different distributions of jobs; a major furniture

store wants to analyse advertising effectiveness amongst a number of concurrent advertising streams, but the effectiveness of each is likely to vary with demographic proportions; the nature of network traffic on a university's computer system varies with time of year due to the fact that different student groups are present or absent at different times.

It would seem likely that most of the prediction problems that are the subject of study or analysis involve at least one of:

- Samples that could come from one of a number of sub-populations, between which the quantity to be predicted may vary.
- Samples chosen are subject to factors that are not fully controlled for, and that could change in different scenarios.
- Targets are aggregate values averaged over a potentially varying population.

Each of these provides a different potential form of source component shift. The three cases correspond to *mixture component shift*, *factor component shift* and *mixing component shift* respectively. These three cases will be elaborated further.

The causal graphical model for source component shift is illustrated in Figure 7. In all cases of source component shift there is some changing environment that jointly affects the values of the samples that are drawn. This may sound indistinguishable from sample selection bias, and indeed these two forms of dataset shift are closely related. However with source component shift the causal model states that the change is a change in the *causes*. In sample selection bias, the change is a change in the *measurement process*. This distinction is subtle but important from a modelling point of view. At this stage it is worth considering the three different cases of source component shift.

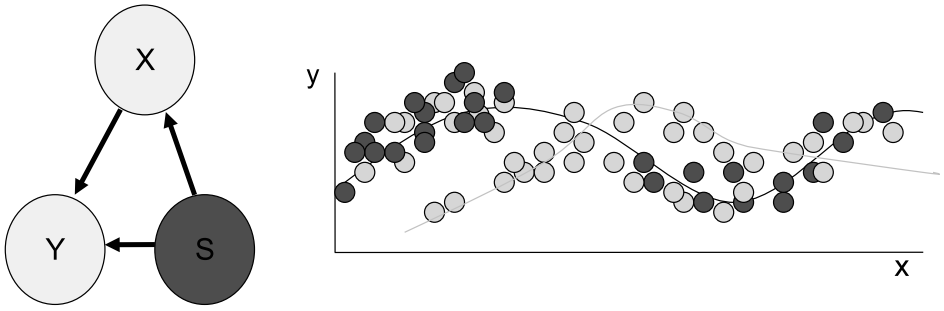

Figure 7: Source component shift. A number of different sources of data are represented in the dataset, each with its own characteristics. Here S denotes the source proportions and these can vary between test and training scenarios. In mixture component shift, these sources are mixed together in the observed data, resulting in two or more confounded components.

Mixture Component Shift In mixture component shift, the data consists directly of samples of $(\mathbf{x}, \mathbf{y})$ values that come from a number of different sources. However for each datum the actual source (which we denote by s) is unknown. Unsurprisingly these different sources occur in different proportions $P(s)$, and are also likely to be responsible for different ranges of values for $(\mathbf{x}, \mathbf{y})$: the distribution $P(\mathbf{y}, \mathbf{x}|s)$ is conditionally dependent on s. Typically, it is presumed that the effects of the sources $P(\mathbf{y}, \mathbf{x}|s)$ are the

same in all situations, but that the *proportions* of the different sources vary between training and test scenarios. This distinction is a natural extension to prior probability shift, where now the shift in prior probabilities is in a latent space rather than in the space of the target attributes.

Factor Component Shift Here the data is dependent on a number of factors that influence the probability, where each factor is decomposable into a form and a strength. For concreteness sake, a common form of factor model decomposes $P(\mathbf{x}, \mathbf{y})$ as

$$P(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp\left(\sum_k \alpha_k \Phi_k(\mathbf{x}, \mathbf{y})\right) \tag{18}$$

for form exponents $\Phi_k(\mathbf{x}, \mathbf{y})$ and strength exponents α_k. Factor component shift occurs when the form of the factors remains the same, but the strength of the factors changes between training and test scenario.

Mixing Component Shift In mixing component shift, the scenario is the same as mixture component shift, but where the measurement is an aggregate: consider sampling whole functions independently from many IID mixture component shift models. Then under a mixing component shift model, the observation at $\mathbf{x}$ is now an average of the observations at $\mathbf{x}$ for each of those samples. The probability of obtaining an $\mathbf{x}$ is as before. Presuming the applicability of a central limit theorem, the model can then be written as

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp\left((\mathbf{y} - \boldsymbol{\mu}(\mathbf{x}))\boldsymbol{\Sigma}^{-1}(\mathbf{x})(\mathbf{y} - \boldsymbol{\mu}(\mathbf{x}))\right)\right) \tag{19}$$

where the mean $\boldsymbol{\mu}(\mathbf{x}) = \sum_s P(s|\mathbf{x})\boldsymbol{\mu}_s$ and the covariance $\boldsymbol{\Sigma} = \sum_s P(s|\mathbf{x})\boldsymbol{\Sigma}_s$ are given by combining the means $\boldsymbol{\mu}_s$ and covariances $\boldsymbol{\Sigma}_s$ of the different components s, weighted by their probability of contribution at point $\mathbf{x}$ (usually called the responsibility).

Although all three of these are examples of source component shift, the treatment each requires is slightly different. The real issue is being able to distinguish the different sources and their likely contributions in the test setting. The ease or otherwise with which this can be done will depend to a significant extent on the situation, and on how much prior knowledge about the form of the sources there is. It is noteworthy that, at least in mixture component shift, the easier it is to distinguish the sources, the less relevant it is to model the shift: sources that do not overlap in $\mathbf{x}$ space are easier to distinguish, but also mean that there is no mixing at any given location to confound the prediction.

It is possible to reinterpret sample selection bias in terms of source component shift if we view the different rejection rates as relating to different sources of data. By setting

$$P_{\text{te}}(s) \propto \int d\mathbf{x} d\mathbf{y} P(\mathbf{x}, \mathbf{y}|P(v = 1|\mathbf{x}, \mathbf{y}) = s) \tag{20}$$

$$P(\mathbf{x}, \mathbf{y}|s) \propto P(\mathbf{x}, \mathbf{y}|P(v = 1|\mathbf{x}, \mathbf{y}) = s) \tag{21}$$

$$P_{\text{tr}}(s) \propto s \int d\mathbf{x} d\mathbf{y} P(\mathbf{x}, \mathbf{y}|P(v = 1|\mathbf{x}, \mathbf{y}) = s) \tag{22}$$

we can convert a sample selection bias model into a source component shift model. In words, the source s is used to represent how likely the rejection would be, and hence each source generates regions of $\mathbf{x}, \mathbf{y}$ space that have equiprobable selection

probabilities under the sample selection bias problem. At least from this particular map between the domains, the relationship is not very natural, and hence from a generative point of view the general source component shift and general sample selection bias scenarios are best considered to be different from one another.

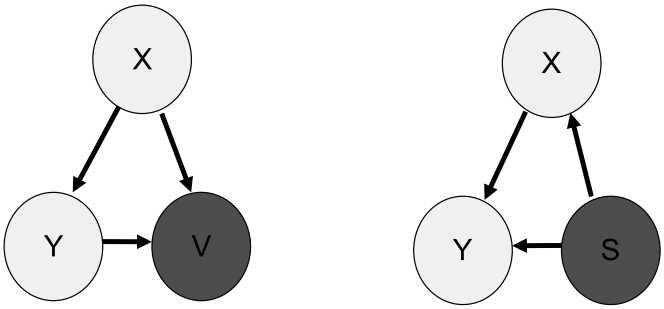

Figure 8: Sample selection bias (left) and source component shift (right) are related. The sources are equated to regions of $(\mathbf{x}, \mathbf{y})$ space with equiprobable sample rejection probabilities under the sample selection bias model. Then the proportions for these sources vary between training and test situations. Here $\mathbf{x}$ and $\mathbf{y}$ are the covariates and targets respectively, s denotes the different sources, and v denotes the sample selection variable.

11 Gaussian Process Methods for Dataset Shift

Gaussian processes have proven their capabilities for nonlinear regression and classification problems. But how can they be used in the context of dataset shift? In this section, we consider how Gaussian process methods can be adapted for mixture component shift.

11.1 Mixture Component Shift Model

In mixture component shift, there are a number of possible components to the model. We will describe here a two source problem, where the covariate distribution for each source is described as a mixture model (a mixture of Gaussians will be used). The model takes the following form

- The distribution of the training data and test data are denoted P_{tr} and P_{te} respectively, and are unknown in general.
- Source 1 consists of M_1 mixture distributions for the covariates, where mixture t is denoted $P_{1t}(\mathbf{x})$. Each of the components is associated[2] with regression model $P_1(\mathbf{y}|\mathbf{x})$.
- Source 2 consists of M_2 mixture distributions for the covariates, where mixture t is denoted $P_{2t}(\mathbf{x})$. Each of the components is associated with the regression model $P_2(\mathbf{y}|\mathbf{x})$.

[2]If a component i is associated with a regression model j, this means that any datum $\mathbf{x}$ generated from the mixture component i, will also have a corresponding $\mathbf{y}$ generated from the associated regression model $P_j(\mathbf{y}|\mathbf{x})$.

- The training and test data distributions take the following form:

$$P_{\text{tr}}(\mathbf{x}) = \sum_{t=1}^{M_1} \beta_1 \gamma_{1t}^D P_{1t}(\mathbf{x}) + \sum_{t=1}^{M_2} \beta_2 \gamma_{2t}^D P_{2t}(\mathbf{x}) \text{ and } P_{\text{te}}(\mathbf{x}) = \sum_{t=1}^{M1} \gamma_{1t}^T P_{1t}(\mathbf{x}) \tag{23}$$

Here β_1 and β_2 are parameters for the proportions of the two sources in the training data, γ_{1t}^D are the relative proportions of each mixture from source 1 in the training data, and γ_{2t}^D are the relative proportions of each mixture from source 2 in the training data. Finally γ_{1t}^T are the proportions of each mixture from source 1 in the test data. Once again, D and T denote the training and test datasets respectively. Note that source 2 does not occur in the test dataset. All these parameters are presumed unknown. In general we will assume the mixtures are Gaussian, when the form $N(\mathbf{x}; \mathbf{m}, \mathbf{K})$ will be used to denote the Gaussian distribution function of $\mathbf{x}$, with mean $\mathbf{m}$ and covariance $\mathbf{K}$.

For Gaussian process models for $P_1(\mathbf{y}|\mathbf{x})$ and $P_2(\mathbf{y}|\mathbf{x})$, with mixture parameters collected as $\mathbf{\Omega}$, and the mixing proportions collected as $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ we have the full probabilistic model

$$P(\{\mathbf{y}^\mu, \mathbf{x}^\mu | \mu \in D\}, \{\mathbf{x}^\nu | \nu \in T\} | \boldsymbol{\beta}, \mathbf{\Omega}) = \sum_{\{s^\mu\}, \{t^\mu\}} \prod_{\mu \in D} P(s^\mu|\boldsymbol{\beta}) P(t^\mu|\boldsymbol{\gamma}, s^\mu) P_{s^\mu t^\mu}(\mathbf{x}^\mu|\Omega_{t^\mu}) P_{s^\mu}(\mathbf{y}^\mu|\mathbf{x}^\mu) \prod_{\nu \in T} P(t^\nu|\boldsymbol{\gamma}) P_{1t^\nu}(\mathbf{x}^\nu|\mathbf{\Omega}) \tag{24}$$

where s^μ denotes the source, and t^μ denotes the mixture component. In words this model says

- For each item in the training set:
  - Decide which source generated this datum.
  - Decide which of the mixtures associated with this source generated the covariates.
  - Sample the covariates from the relevant mixture.
  - Sample the target from the Gaussian process (conditioned on the covariates) associated with this source.
- For each item in the test set:
  - Decide which of the mixtures from source 1 generated the covariates (source 2 is not represented in the test data).
  - Generate the covariates from that mixture.

11.2 Learning and Inference

The primary computational issue in learning and inference in this model is the difficulty of summing over all the allocations of data points to mixture components. For Gaussian processes, this computation is harder than in most parametric models as we cannot expect to be able to do standard Expectation Maximisation. Expectation Maximisation algorithms involve iterative computation of responsibilities $P(s^\mu)$ for each data point μ and then a maximum likelihood parameter estimation for the parameters given the responsibilities. However as Gaussian processes are nonparametric, the distribution is not independent of the allocation. Hence whether one point is allocated to one mixture or not will immediately effect the distribution over all other mixtures.

Here, a variational approximation is proposed, which enables a variational Expectation Maximisation procedure to be used. The approximation takes the form of an intermediate approximating Gaussian process for each mixture component and factorised responsibilities.

For simplicity, we will assume that the target is a scalar value: we are interested in regression. The issues of generalisation to multidimensional targets are the same as in standard Gaussian process models. Furthermore for ease of notations the targets for all of the N data points y^μ are collected into a vector $\mathbf{y} = (y^1, y^2, \dots y^N)^T$. The same is done for all other relevant scalar quantities such as the indicators $\mathbf{s}$ etc. The quantities $\mathbf{f}_1$ and $\mathbf{f}_2$ denote the collections of values of each noise-free Gaussian process at all the points $\{\mathbf{x}^\mu\}$, and noise σ^2.

The Gaussian process mixture can be written as

$$P(\mathbf{y}|\{\mathbf{x}^\mu\}) = \sum_{\mathbf{s}} \int d\mathbf{f}_1 \; d\mathbf{f}_2 P(\mathbf{f}_1, \mathbf{f}_2, \mathbf{s}, \mathbf{y}|\{\mathbf{x}^\mu\}) \tag{25}$$

where

$$P(\mathbf{f}_1, \mathbf{f}_2, \mathbf{s}, \mathbf{y}|\{\mathbf{x}^\mu\}) = P(\mathbf{f}_1|\{\mathbf{x}^\mu\}) P(\mathbf{f}_2|\{\mathbf{x}^\mu\}) \times \prod_\mu \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \left[s^\mu (y^\mu - f^\mu)^2 + (1 - s^\mu)(y^\mu - f^\mu)^2 \right] \right). \tag{26}$$

Note $P(\mathbf{f}_1|\{\mathbf{x}^\mu\})$ and $P(\mathbf{f}_2|\{\mathbf{x}^\mu\})$ are simply the prior Gaussian process regressors for the two sources.

By using a variational approximation of the form $Q(\mathbf{f}_1)Q(\mathbf{f}_2)\prod_\mu Q(s^\mu)$ and iteratively reducing the KL divergence $KL(Q||P)$ we obtain the following approximation procedure for an iterative solution of the covariate shift model. Here α_{st}^μ is used to denote the responsibility of mixture t of source s for point μ in the training set. The term $\alpha_s^\mu = \sum_t \alpha_{st}^\mu$ is the responsibility of source s for the point μ.

- Perform a standard Gaussian mixture model Expectation Maximisation to initialise the responsibilities α_s^μ for each of the two sources.
- Iterate:
  - Compute the pseudo-variances σ^2/α_s^μ for each point and each source.
  - Build the covariance C_1 for source 1 from the covariance of the Gaussian process, and an additive pseudo-noise given by a matrix with the pseudo-variances for source 1 down the diagonal.
  - Do the same for source 2 to obtain C_2.
  - Compute the mean predictions $(f_1^*)^\mu$ and $(f_2^*)^\mu$ at points $\{\mathbf{x}^\mu\}$ for Gaussian processes with training covariances C_1, and C_2, and prediction covariances given by the original covariance functions.
  - Compute the parameter updates for the Gaussian processes using the usual hyper-parameter optimisations, and the updates for the various mixture components using:

$$\mathbf{m}_{st} = \frac{\sum_{\mu\in(D,T)} \alpha_{st}^\mu \mathbf{x}^\mu}{\sum_{\mu\in(D,T)} \alpha_{st}^\mu} \;, \; \mathbf{K}_{st} = \frac{\sum_{\mu\in(D,T)} \alpha_{st}^\mu (\mathbf{x}^\mu - \mathbf{m}_{st})(\mathbf{x}^\mu - \mathbf{m}_{st})^T}{\sum_{\mu\in(D,T)} \alpha_{st}^\mu} \tag{27}$$

– Compute the new responsibilities for each mixture, each source and each data point using:

$$\alpha_{st}^{\mu} = \frac{\beta_s \gamma_{st}^D P_{st}(\mathbf{x}^\mu|\mathbf{\Omega}) P(y^\mu|(f_s^*)^\mu, \sigma^2)}{\sum_{s,t} \beta_s \gamma_{st}^D P_{st}(\mathbf{x}^\mu|\mathbf{\Omega}) P(y^\mu|(f_s^*)^\mu, \sigma^2)} \text{ and } \alpha_{1t}^{\nu} = \frac{\gamma_{1t}^T P_{1t}(\mathbf{x}^\mu|\mathbf{\Omega})}{\sum_t \gamma_{1t}^T P_{1t}(\mathbf{x}^\mu|\mathbf{\Omega})} \tag{28}$$

$$\beta_s = \frac{1}{|D|} \sum_{\mu \in D,t} \alpha_{st}^{\mu} \ , \ \gamma_{st}^D = \frac{1}{|D|} \sum_{\mu \in D} \frac{\alpha_{st}^{\mu}}{\beta_s} \ , \ \gamma_{1t}^T = \frac{1}{|T|} \sum_{\nu \in T} \alpha_{1t}^{\nu} \tag{29}$$

where

$$P(y^\mu|(f_s^*)^\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y^\mu - (f_s^*)^\mu)^2\right). \tag{30}$$

- Predict the result on the test data using the Gaussian process prediction with the covariance between data points given by C_1 and covariance between test and data given by the usual covariance function.

See [28] for more details of this approach to Gaussian process mixtures. Intuitively, this process increases the noise term on data that are poorly explained by one of the mixtures. A datum with an increased noise term will have less influence on the overall Gaussian process regressor that is learnt. This model is related to a mixture of experts model [12, 14], but where there is a coupling of the regression function between different mixtures. and the covariate density itself is also modelled. A similar model was developed in [27], but only for linear regressors, and single Gaussian components per regressor. This model has the usual deficits associated with mixture models, including local minima issues, and the difficulties in deciding on a suitable number of mixtures. The infinite mixture of Gaussian process experts [18] is another mixture of experts model, but one that uses Gaussian processes and does not suffer from model size selection issues. However it too does not have the distribution in covariate space (although this could be added to the model without major difficulties). The main issues of adapting this for use here are that of having to resort to Markov Chain Monte-Carlo methods rather than variational methods, and incorporating the match to the test dataset. These are surmountable issues. In the current context, Bayesian Information Criterion methods can be used [23] for selection of the number of mixtures, but it may not always work well as it is both approximate and a heuristic for latent variable problems. One other consequence of the model selection issue is that that this implementation of the model may well perform more poorly than a straight Gaussian process in cases of no dataset shift. This issue is discussed more generally in the next section.

12 Shift or No Shift?

One big issue in all types of dataset shift is determining whether there is, in fact, any type of shift at all. It is possible that using a modelling method which can account for covariate shift may produce worse results than a standard model on data for which no shift occurs. This is first because introducing the possibility of shift allows for a large scope of possible representations that waters down the more concrete (but rigid) assumptions that presuming no shift makes. Second, the various methods used in modelling covariate shift may have their own deficiencies (such as local minima) that mean that they do not properly include the no-shift case: for a maximum likelihood solution may prefer to improve the likelihood by utilising the freedom of the dataset shift model to overfit, even if presuming no shift would generalise better.

At this point, there are some real practicalities that should outweigh theoretical niceties. It may be interesting to consider how to determine whether covariate shift occurs on the basis of the training covariates, training targets and test covariates alone. It may also be useful in making a choice about a limited number of models to consider. However in many realistic scenarios (the main exceptions being single future prediction cases[3]), a practitioner would be negligent not to check a model in the actual environment it is being developed for before rolling out the use of the model. There must come a stage at which some test targets are obtained, and at which some assessment is done on the basis of those. Furthermore even a few test targets provide a large amount of information regarding dataset shift, in the same way that semi-supervised learning can provide major benefits over unsupervised learning. It would also seem peculiar if a no-shift model was not one of the small basket of models considered at this stage, unless a particular form of dataset shift was guaranteed a priori. The major improvements available from a semi-supervised approach in the test domain should never be neglected: targets in the test domain are very valuable information.

13 Dataset Shift and Transfer Learning

Dataset shift and transfer learning are very related. Transfer Learning considers the issue of how information can be taken from a number of only partially related training scenarios and used to provide better prediction in one of those scenarios than would be obtained from that scenario alone. Hence dataset shift consists of the case where there are only two scenarios, and one of those scenarios has no training targets. Multi-task learning is also related. In multi-task learning the response for a given input on a variety of tasks is obtained, and information between tasks is used to aid prediction. Multi-task learning can be thought of a special case of transfer learning where there is some commonality in training covariates between tasks, and where the covariates have the same meaning across scenarios (hence domain shift is precluded).

There is recent work on utilising Gaussian processes for multi-task learning [3]. Unlike the methods developed here, this approach relies on having target data for all scenarios to help in relating them. Many approaches to document analysis (e.g. Latent Dirichlet Allocation [2] and many related techniques) are in fact methods for mixture component shift, applied to unsupervised problems in more general multi-dataset scenarios. The major advantage of having multiple datasets is that it is possible to characterise the differences between the datasets.

14 Conclusions

Modelling Dataset Shift is a challenging problem, but one with significant real world consequence. The failures that arise from ignoring the possibility of dataset shift (e.g. sample selection bias) have been known for a long time. Furthermore models that work well in static scenarios (such as the use of a conditional model) can fail in situations of shift. By characterising the different forms of dataset shift, we can begin to get a handle on the ways the data can be expected to change. Though sample selection bias and imbalanced data have been studied for many decades as subjects in their own right, some common forms of shift, such as source component shift and domain shift may also be worthy of further explicit study. Hopefully, by

[3]As an example, a pollster predicting election results has no recourse to the voting patterns of the population as a whole until it is too late.

relating the different types of shift, more general methods will become available that can cope with a number of different forms of shift at the same time. Such methods may help automate the process of prediction even in the case of changing environments. The aim is to develop methods that are robust to, and automatically accommodate for dataset shift.

One big question that should be considered is whether it is important to study dataset shift in its own right, or whether there is more to be gained by the general study of methods for learning transfer that could be directly applied to dataset shift. Though the basket of approaches in the two fields may well be similar, there are methods that will require either some test targets, or multiple training domains to work, both of which may be unavailable in a standard dataset shift problem. One thing is certain though, study of dataset shift and transfer learning cannot be done in isolation of one another, and in a world of data abundance, it may well be worth asking whether a scenario with a single training dataset as well as a single unlabelled test dataset is really the best way of expressing a given problem.

References

[1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *Machine Learning: ECML 2004*, pages 39–50, 2004.

[2] D. M. Blei, A. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] E. V. Bonilla, F. A. Agakov, and C. K. I. Williams. Kernel multi-task learning using task-specific features. In *Proceedings of AISTATS 2007*, 2007.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[5] N. V. Chawla, N. Japkowich, and A. Kolcz. Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explorations*, 6:1–6, 2004.

[6] M. Dudík, R. E. Schapire, and S. J. Phillips. Correcting sample selection bias in maximum entropy density estimation. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, 2006.

[7] M. N. Gibbs and D. J. C. MacKay. Efficient implementation of Gaussian processes. Unpublished Manuscript, 1997.

[8] J. J. Heckman. Shadow prices, market wages, and labor supply. *Econometrica*, 42(4):679–694, 1974.

[9] J. J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47:153–161, 1979.

[10] J. J. Heckman. Varieties of selection bias. *The American Economic Review*, 80(2):313–318, may 1990.

[11] J. Huang, A. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19*, pages 601–608, 2007.

[12] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

[13] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6:429–449, 2002.

[14] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

[15] P. Latinne, M. Saerens, and C. Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities may significantly improve classifiction accuracy: Evidence from a multi-class problem in remote sensing. In *Proc. 18th International Conf. on Machine Learning*, pages 298–305. Morgan Kaufmann, San Francisco, CA, 2001.

[16] L. Lee. Some approaches to the correction of selectivity bias. *Review of Economic Studies*, 49:355–372, 1982.

[17] J. Leskovec and J. Shawe-Taylor. Linear programming boosting for uneven datasets. In *ICML 2003*, 2003.

[18] C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems 14*, 2002.

[19] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, Massachusetts, 2006.

[20] H. Shimodeira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244, 2000.

[21] P. Sollich. Probabilistic interpretations and Bayesian methods for support vector machines. In *ICANN99*, pages 91–96, 1999.

[22] P. Sollich. Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Machine Learning*, 46:21–52, 2002.

[23] A. J. Storkey and M. Sugiyama. Mixture regression for covariate shift. In *Advances in Neural Information Processing Systems 19*, 2007.

[24] M. Sugiyama, B. Blankertz, M. Krauledat, G. Dornhege, and K.-R. Müller. Importance weighted cross-validation for covariate shift. In K. Franke, K.-R. Müller, B. Nickolay, and R. Schäfer, editors, *DAGM 2006*, pages 354–363. Springer LNCS 4174, 2006.

[25] M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, May 2007.

[26] M. Sugiyama and K.-R. Müller. Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, 23(4):249–279, 2005.

[27] H. G. Sung. *Gaussian Mixture Regression and Classification*. PhD thesis, Rice University, 2004.

[28] V. Tresp. Mixtures of Gaussian processes. In *Advances in Neural Information Processing Systems 13*, 2001.

[29] F. Vella. Estimating models with sample selection bias: A survey. *The Journal of Human Resources*, 33:127–169, 1998.

[30] K. Veropoulos, N. Cristianini, and C. Campbell. Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on Artificial Intelligence, (IJCAI99), Stockholm, Sweden, 1999*, 1999.

[31] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In R. Greiner and D. Schuurmans, editors, *Proc. of the 21st Int. Conf. on Machine Learning (ICML)*, pages 114–122, 2004.

[32] B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, 2001.