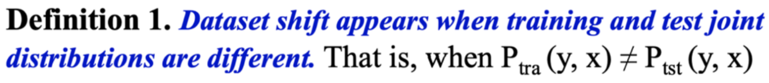
1. Understanding Dataset Shift

-Introduction

- Dataset shift is a challenging situation where the joint distribution of inputs and outputs differs between the training and test stages.
- The key theme of this article can be summarized in a single sentence: **Dataset shift is when the training and test distributions are different.**
- The problem of dataset shift can stem from the way input features are utilized, the way training and test sets are selected, data sparsity, shifts in the data distribution due to non-stationary environments, and also from changes in the activation patterns within layers of deep neural networks.
- Definition



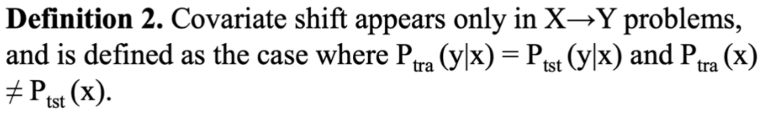
Definition 1. ***Dataset shift appears when training and test joint distributions are different.*** That is, when $P_{tra}(y, x) \neq P_{tst}(y, x)$

-why important?

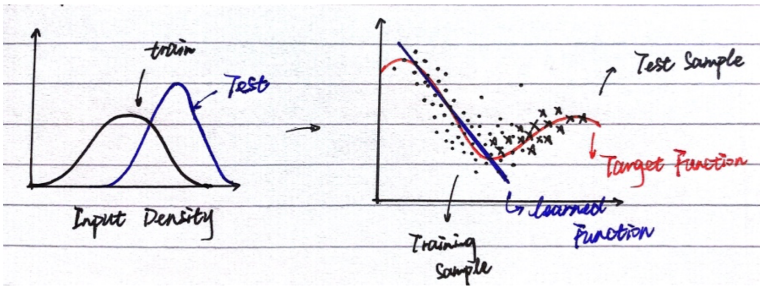
- It is application-dependent and thus relies largely on the skill of the data scientist to examine and resolve. For example, how does one determine when the dataset has shifted sufficiently to pose a problem to our algorithms? If only certain features begin to diverge, how do we determine the trade-off between the loss of accuracy by removing features and the loss of accuracy by a misrepresented data distribution?

1.1 Covariate shift

- Covariate shift is the change in the distribution of the covariates specifically, that is, the independent variables. This is normally due to changes in state of latent variables, which could be temporal (even changes to the stationarity of a temporal process), or spatial, or less obvious.
- Covariate shift is the scholarly term for when **the distribution of the data** (i.e. the input features) changes.
- Definition



Definition 2. Covariate shift appears only in X→Y problems, and is defined as the case where $P_{tra}(y|x) = P_{tst}(y|x)$ and $P_{tra}(x) \neq P_{tst}(x)$.



- Examples

1. Face recognition algorithms that are trained predominantly on younger faces, yet

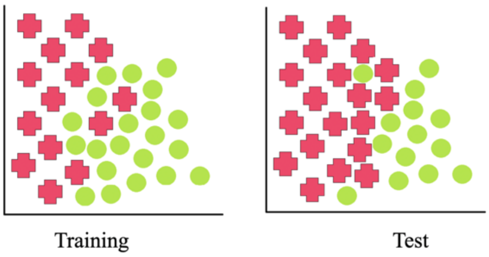
the dataset has a much larger proportion of older faces in it.

2. Predicting life expectancy but having very few samples in the training set of individuals that smoke, and many more samples of this in the training set.

- In this case, there is no change in the underlying relationship between the input and output (the regression line is still the same), yet part of that relationship is data-sparse, omitted, or misrepresented such that the test set and training set do not reflect the same distribution.
- Covariance shift can cause a lot of problems when performing cross-validation. Cross-validation is almost unbiased without covariate shift but it is heavily biased under covariate shift.

1.2 Prior probability shift

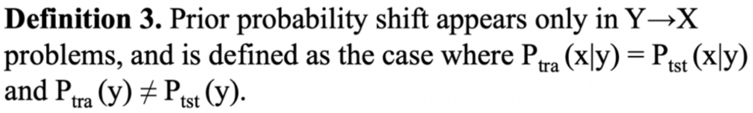
- While covariate shift focuses on changes in **the feature (x) distribution**, prior probability shift focuses on changes in **the distribution of the class variable y**.



Training

Test

- This type of shifting may seem slightly more confusing but is it essentially the reverse of covariate shift. An intuitive way to think about it might be to consider an unbalanced dataset.
- Definition



Definition 3. Prior probability shift appears only in Y→X problems, and is defined as the case where $P_{tra}(x|y) = P_{tst}(x|y)$ and $P_{tra}(y) \neq P_{tst}(y)$.

- Example

If the training set has equal prior probabilities on the number of spam emails that you receive (i.e. the probability of an email being spam is 0.5), then we would expect 50% of the training set to contain spam emails and 50% to contain non-Spam.

If, in reality, only 90% of our emails are spam (perhaps not unlikely), then our prior probability of the class variables has changed.

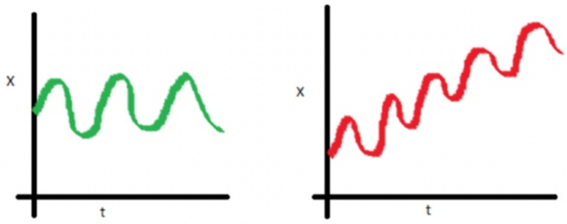
This idea has relations to data sparsity and biased feature selection that are factors in causing covariance shift, but instead of influencing our input distribution, they instead influence our output distribution.

This problem only occurs in Y -> X problems and is commonly associated with naive Bayes (hence the spam example, since naive Bayes is commonly used to filter spam emails).

1.3 Concept shift

- Concept shift is different from covariate and prior probability shift in that it is not related to the data distribution or the class distribution but instead is related to the relationship between the two variables.
- An intuitive way to think about this idea is by looking at time series analysis. In time series analysis, it is common to examine whether the time series is stationary before performing any analysis, as stationary time series are much easier to analyze than non-stationary time series.

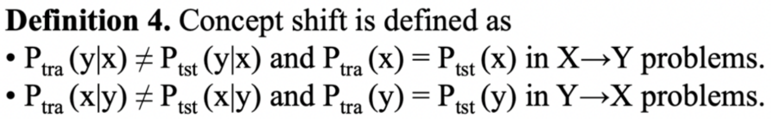
  This is easier because the relationship between the input and output is not consistently changing! There are ways of detrending a time series to make it stationary, but this does not always work (such as in the case of stock indices that generally contain little autocorrelation or secular variation).



Stationary series

Non-Stationary series

- Definition



Definition 4. Concept shift is defined as
- $P_{tra}(y|x) \neq P_{tst}(y|x)$ and $P_{tra}(x) = P_{tst}(x)$ in X→Y problems.
- $P_{tra}(x|y) \neq P_{tst}(x|y)$ and $P_{tra}(y) = P_{tst}(y)$ in Y→X problems.

- Example

To give a more concrete example, let's say we examined the profits of companies before the 2008 financial crisis and made an algorithm to predict the profit based on factors such as the industry, number of employees, information about products, and so on. If our algorithm is trained on data from 2000–2007, but are not using it to predict the same information after the financial crisis, it is likely to perform poorly.

So what changed? Clearly, the overall relationship between the inputs and outputs changed due to the new socio-economic environment, and, if these are not reflected in our variables (such as having a dummy variable for the date that the financial crisis occurred and training data before and after this date) then our model is going to suffer the consequences of concept shift.

In our specific case, we would expect to see profits change markedly in the years after the financial crisis (this is an example of **an interrupted time series**).

1.4 Internal covariate shift (an important subtype of covariate shift)

- Researchers found that due to the variation in the distribution of activations from the output of a given hidden layer, which are used as the input to a subsequent layer, the network layers can suffer from covariate shift which can impede the training of deep neural networks.

- The situation without batch normalization, network activations are exposed to varying data input distributions that propagate through the network and distort the learned distributions.

  This idea is the stimulus of **batch normalization**, proposed by Christian Szegedy and Sergey Ioffe in their 2015 paper *"Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift"*.

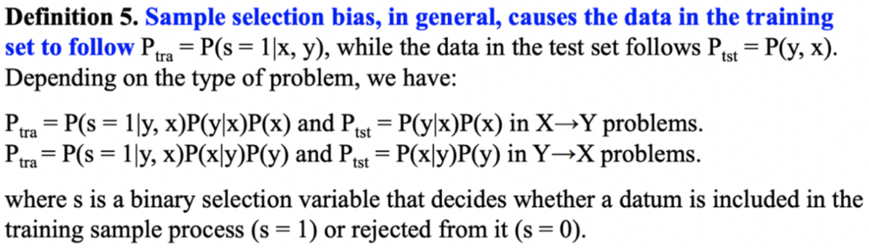
  The authors propose that internal covariate shift in the hidden layers slows down training and requires lower learning rates and careful parameter initialization. They resolve this by normalizing the inputs to hidden layers by adding a batch normalization layer.

2. Major Causes of Dataset Shift

- The two most common causes of dataset shift are **(1) sample selection bias** and **(2) non-stationary environments.**

2.1 Sample selection bias

- Sample selection bias is not a flaw with any algorithm or handling of the data. It is purely a systematic flaw in the process of data collection or labeling which causes nonuniform selection of training examples from a population, which causes biases to form during training.
- Definition



Definition 5. **Sample selection bias, in general, causes the data in the training set to follow** $P_{tra} = P(s = 1|x, y)$, while the data in the test set follows $P_{tst} = P(y, x)$. Depending on the type of problem, we have:

$P_{tra} = P(s = 1|y, x)P(y|x)P(x)$ and $P_{tst} = P(y|x)P(x)$ in X→Y problems.
$P_{tra} = P(s = 1|y, x)P(x|y)P(y)$ and $P_{tst} = P(x|y)P(y)$ in Y→X problems.

where s is a binary selection variable that decides whether a datum is included in the training sample process (s = 1) or rejected from it (s = 0).

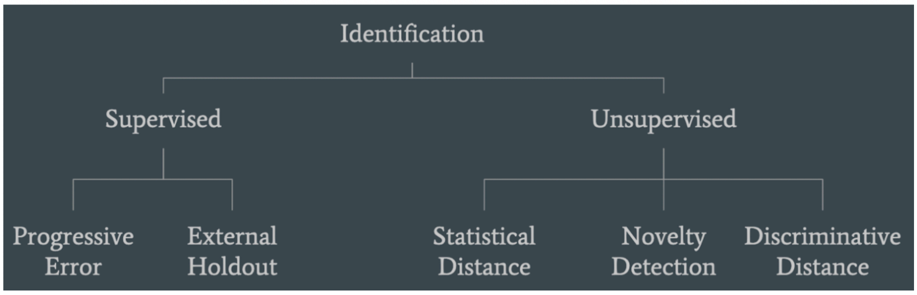
2.2 Non-stationary environments

- When the training environment is different from the test one, whether it is due to a temporal or a spatial change.

3. Identifying Dataset Shift

- There are several methods that can be used to determine whether shifting is present in a dataset and its severity.
- Unsupervised methods are perhaps the most useful ways of identifying dataset shift, as they do not require post-hoc analysis to be done, the latency of which cannot be afforded in some production systems. Supervised methods exist which essentially look

at growing errors as the model runs and the performance on an external holdout (validation set).



3.1 Statistical Distance

- The statistical distance method is useful for detecting if your model predictions change over time. This is done by creating and using histograms. By making histograms, you are not only able to detect whether your model predictions change over time, but also check if your most important features change over time. Simply put, you form histograms of your training data, keep track of them over time, and compare them to see any changes. This method is used commonly by financial institutions on credit-scoring models.
- There are several metrics which can be used to monitor the change in model predictions over time. These include the **Population Stability Index (PSI)**, **Kolmogorov-Smirnov statistic**, **Kullback-Lebler divergence** (or other **f-divergences**), and **histogram intersection**.
- The major disadvantage of this method is that is not great for high-dimensional or sparse features. However, it can be very useful and in my opinion should be the first thing to try when dealing with this issue.

3.2 Novelty Detection

- A method that is more amenable to fairly complex domains such as computer vision, is novelty detection. The idea is to create a model for modeling source distribution. Given a new data point, you try to test what is the likelihood that this data point is drawn from the source distribution. For this method, you can use various techniques such as a one-class support vector machine, available in most common libraries.
- If you are in a regime of homogenous but very complex interactions (e.g. visual, audio, or remote sensing), then this is a method you should look into, because in that case, the statistical distance (histogram method) won’t be as effective a method.
- The major disadvantage of this method is that it cannot tell you explicitly what has changed, only that there has been a change.

3.3 Discriminative Distance

- The discriminative distance method is less common, nonetheless, it can be effective. The intuition is that you want to train a classifier to detect whether or not an example is from the source or target domain. You can use the training error as proxy of the distance between those two distributions. The higher the error, the closer they are (i.e. the classifier cannot discriminate between the source and target domain).
- Discriminative distance is widely applicable and high dimensional. Though it takes time and can be very complicated, this method is a useful technique if you are doing domain

adaptation (and for some deep learning methods, this may be the only feasible technique that exists).

- This method is good for high-dimensional and sparse data, and is widely applicable. However, it can only be done offline and is more complicated to implement than the previous methods.

4. Handling Dataset Shift

- How do you correct dataset shift? If possible, you should always retrain. Of course, in some situations, it may not be possible, for example, if there are latency problems with retraining. In such cases, there are several techniques for correcting dataset shift.

4.1 Feature Removal

- By utilizing the statistical distance methods discussed above which are used to identify covariate shift, we can use these as measures of the extent of the shifting. We can set a boundary on what is deemed an acceptable level of shift, and analyzing individual features or through an ablation study, we can determine which features are most responsible for the shifting and remove these from the dataset.
- As you may expect, there is a trade-off between removing features that contribute to the covariate shift and having additional features and tolerating some covariate shift. This trade-off is something that the data scientist would need to assess on a case-by-case basis.
- A feature that differs a lot during training and test, but does not give you a lot of predictive power, should always be dropped. As an example, PSI is used in risk management and an arbitrary value of 0.25 is used as the limit, above which this is deemed as a major shift.

4.2 Importance Reweighting

- The main idea with importance reweighting is that you want to upweight training instances that are very similar to your test instances. Essentially, you try to change your training data set such that it looks like it was drawn from the test data set. The only thing required for this method is unlabeled examples for the test domain. This may result in data leakage from the test set.
- To make it clear how this works, we basically reweight each of the training examples by the relative probability of the training and test set. We can do this by density estimation, through kernel methods such as kernel mean matching, or through discriminative reweighting.

4.3 Adversarial Search

- The adversarial search method uses an adversarial model where the learning algorithm attempts to construct a predictor that is robust to the deletion of features at test time.
- The problem is formulated as finding the optimal minimax strategy with respect to an adversary which deletes features and shows that the optimal strategy may be found by either solving a quadratic program or using efficient bundle methods for optimization.