UniTs - University of Trieste

Distribution Shift Report

Authors:

**Andrea Spinelli, Giacomo Amerio,
Giovanni Lucarelli, Tommaso Piscitelli**

January 19, 2025

github.com/giovanni-lucarelli/distribution-shift

Abstract

This is the report of our project about distribution shift ...

Contents

1 Introduction

1.1 Dataset Shift

In the field of machine learning and predictive modeling, it is often assumed that data distributions remain static, meaning they do not change between the training and deployment phases of the model.

However, in practice, this assumption is rarely satisfied: data distributions can undergo significant changes between the training and testing scenarios.

This phenomenon is known as ”dataset shift” and is closely related to another field of study, referred to by various terms such as ”transfer learning” or ”inductive transfer”.

Transfer learning addresses the problem of how information can be drawn from a number of only partially related training scenarios and used to provide better predictions in one of those scenarios compared to using only that specific scenario.

Therefore, dataset shift represents a more specific case: it deals with relating information in, typically, two closely related environments to improve prediction in one given the dataset in the other.

Given this issue, it is crucial to develop an understanding of the suitability of particular models under such changing conditions, and it is necessary to consider whether a different predictive model should be employed.

Among the various forms of dataset shift, covariate shift, studied and described by Shimodaira in 2000, is one of the most extensively researched forms.

It encompasses situations where the distribution of the covariates, $P(X)$ changes, while the conditional relationship $P(Y \mid X)$, representing the relationship between the covariates X and the target Y, remains unchanged.

In this case, the typical values of the covariates observed during testing differ from those observed during training.

1.1.1 Most common causes of dataset shift

The two most common and studied causes of dataset shift are:

1. Sample selection bias
2. Non-stationary environments

Sample selection bias occurs when there is a discrepancy in the data distribution due to the training data being obtained through a biased method, and therefore not reliably representing the real environment in which the classifier will be used (the test set).

It is not a flaw of an algorithm or data management but a systematic defect in the process of collecting or labeling data, which causes a non-uniform selection of training examples from a population, leading to the formation of bias during training.

Dataset shift resulting from sample selection bias is particularly relevant when dealing with imbalanced classification problems, as in highly imbalanced domains, the minority class is especially sensitive to single classification errors due to its typically low number of samples.

In real-world applications, it is often the case that data is not stationary (in time or space). One of the most relevant non-stationary scenarios involves adversarial classification problems, such as spam filtering and network intrusion detection.

1.2 Covariate Shift

As previously mentioned, covariate shift is a specific type of dataset shift often encountered in machine learning.

It occurs when the distribution of input data changes between the training environment and the operational environment, but there is no change in the underlying relationship between the input and output.

Mathematically, this case can be defined as follows:

Definition: *Covariate shift* occurs only in problems of the type $X \to Y$ and is defined as the case where:

$$P_{\text{tra}}(Y \mid X) = P_{\text{tst}}(Y \mid X) \quad \text{and} \quad P_{\text{tra}}(X) \neq P_{\text{tst}}(X)$$

where P_{tra} and P_{tst} represent the probability distributions in the training data and test data, respectively.

Covariate shift is a phenomenon that can affect a wide range of machine learning models, regardless of the task they are designed to perform.

It is commonly encountered in scenarios where models classify data or predict trends based on input features.

This issue is particularly relevant in diverse machine learning applications, including but not limited to:

1. Image categorization and facial recognition systems
2. Speech recognition and translation software
3. Diagnostic and screening tools in healthcare

For example consider a model designed to distinguish between cats and dogs. Our training data might consist of images like those shown in Figure 1.1, present in a specific dataset.

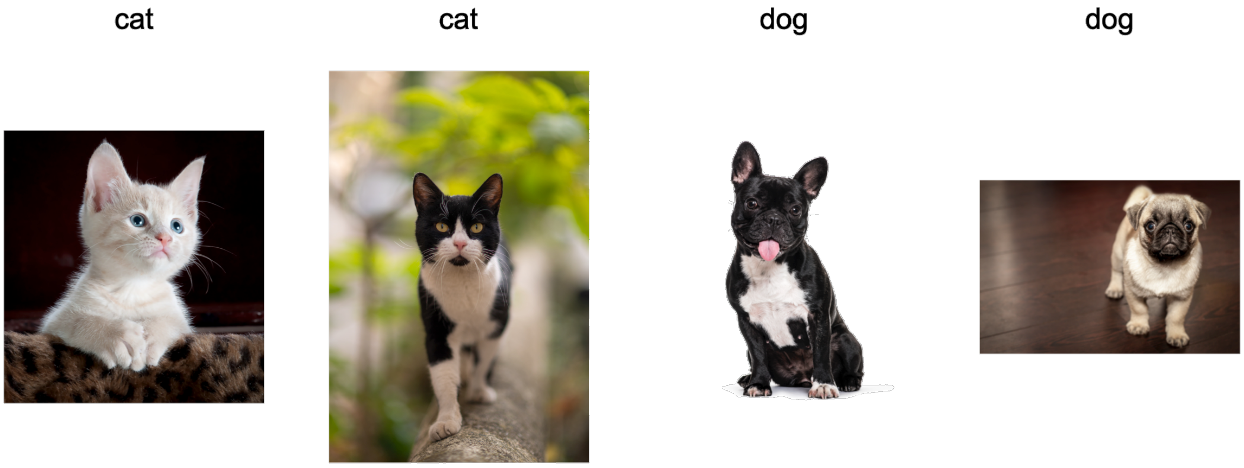

Figure 1.1. Training data for distinguishing cats and dogs.

At test time, we are asked to classify the images in Figure 1.2. Once deployed, the model will not accurately distinguish between cats and dogs because the feature distribution will differ.

Model may achieve a high degree of accuracy on a labeled training dataset, identifying and classifying the object in an image.

However, when deployed with real-time data, changes in the input distribution can significantly impact the model's accuracy.

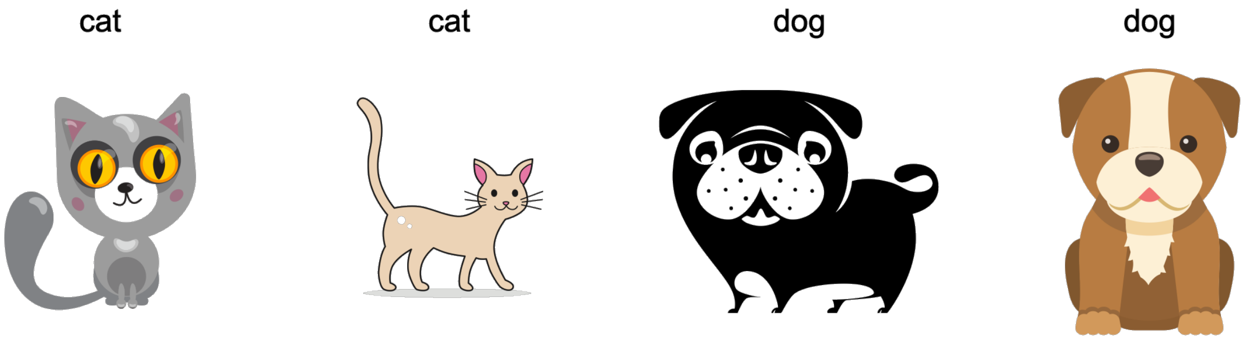

Figure 1.2. Test data for distinguishing cats and dogs.

The same can be accured in the case of facial recognition. The training data might not include subjects from specific ethnicities or age groups.

When the model is deployed in a real-world environment, subjects that do not align with the training data may exhibit an unrecognizable feature distribution.

Another cause of covariate shift could be variations in environmental conditions, such as lighting.

For instance, an image categorization model trained under specific lighting conditions may perform poorly when deployed in an operational setting with different lighting.

So covariate shift, also known as covariate drift, is a very common issue in machine learning. Supervised learning models are often trained with labeled data.

A data scientist typically prepares and labels the training data, identifying and analyzing outliers to maintain a high level of data quality.

However, the same level of oversight cannot be guaranteed in an operational environment since professionals will not have direct control over the input data once the model is deployed.

This means that the availability and quantity of training data can be limited, and consequently, the distribution of input data in this subset of training data is unlikely to exactly mirror the characteristics of data in a real-world environment.

Figure 1.3 shows an example of different distribution between training data and test data, creating a division between the two datasets.

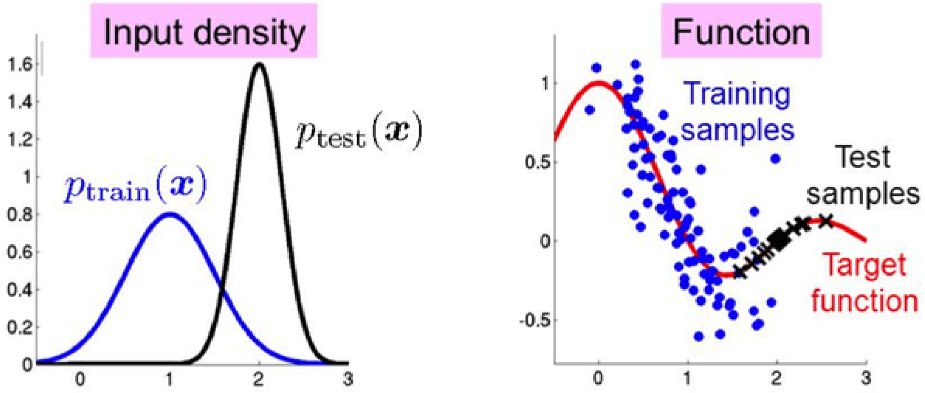

Figure 1.3. Example of covariate shift.

This will have a negative impact on the accuracy of the model, as the algorithms will have been trained to map input data to output data and may fail to recognize the features of inputs from a different distribution, as shown in Figure 1.4.

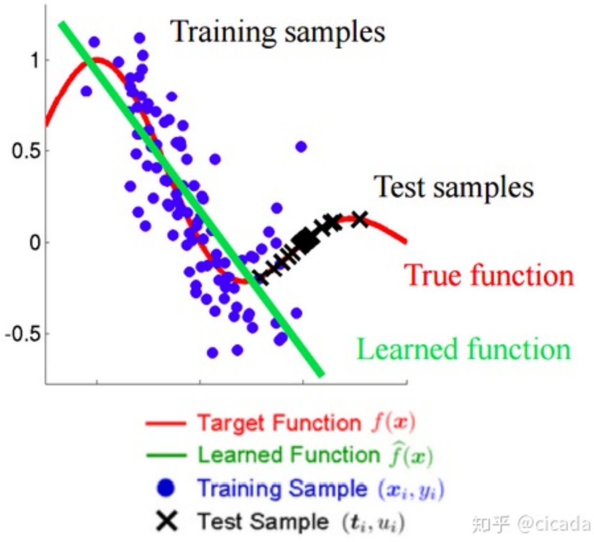

Figure 1.4. Example of inaccurate model.

This means that the model may become less accurate or completely ineffective. This issue represents a critical aspect in machine learning, as a highly performant model on training data may not remain accurate once deployed.

The goal is to determine the extent to which the shift affects the model, take measures to address the issues, and improve the model's accuracy.

Addressing this issue allows models to be readjusted to improve their accuracy. Covariate shift can provide insights into the degree of generalization of the model, which refers to the model's ability to apply learned features from training data to new data.

Low levels of generalization can result from overfitting, where the model is overly aligned with the training data, making it ineffective when encountering new data with a different distribution.

2 Data Generation

To evaluate the performance of the proposed models under varying degrees of covariate distribution shift, we simulated a training dataset and multiple test datasets, each exhibiting a distinct degree of shift. This setup enables a comprehensive assessment of the models' generalization capabilities.

The training dataset consists of three features, denoted as X_1, X_2, and X_3, and a binary target variable Y. The features were generated from a multivariate normal distribution $\mathscr{N}(\mu, \mathbf{\Sigma})$, where each element of the mean vector μi was sampled from a uniform distribution $\mathscr{U}_{[0,1]}$, and the elements of the covariance matrix $[\mathbf{\Sigma}]_{i,j}$ were sampled from $\mathscr{U}_{[-1,1]}$. To ensure the validity of the covariance matrix, after the generation, it was then transformed into a symmetric positive semi-definite matrix. The target variable Y is a binary variable with values in $\{0, 1\}$. It was generated by first constructing a second-order polynomial model with all possible interaction terms and random coefficients drawn from $\mathscr{U}_{[-1,1]}$. The output of this polynomial was then transformed using the standard logistic function to produce probabilities, which were thresholded to determine the binary values of Y.

A fully shifted test dataset was generated from a new multivariate normal distribution $\mathscr{N}(\mu_{0.05}, \mathbf{\Sigma}_s)$. Here, $\mu_{0.05}$ represents a mean vector centered at the 5th percentile of the original μ, and $\mathbf{\Sigma}_s$ is a covariance matrix distinct from $\mathbf{\Sigma}$. This shift was deliberately designed to focus on a region of the sample space with limited representation in the training data, thereby introducing a significant covariate distribution shift. The target variable Y for this dataset was generated in the same manner as in the training dataset.

To assess model performance across a continuum of distribution shifts, we created a series of datasets using statistical mixtures of the training dataset and the fully shifted dataset. Specifically, we defined ten datasets, $\mathscr{D}_p$, where $p \in \{0.0, 0.1, \ldots, 1.0\}$ represents the mixing probability. The dataset $\mathscr{D}_{0.0}$ corresponds to data generated from the same distribution as the training dataset (but with new samples), allowing for an evaluation of the models on unseen, unshifted data. Conversely, $\mathscr{D}_{1.0}$ corresponds to the fully shifted dataset. Intermediate values of p represent datasets with increasing proportions of shifted data, enabling a systematic investigation of model robustness under varying degrees of covariate distribution shift.

3
Performance Degradation

3.1 Evaluation Metric: AUC

We defined the Area under the Curve (AUC) as the metric to evaluate the performance of the models. The AUC is a measure of the performance of a classification model. It is calculated as the area under the ROC curve. The ROC curve is a graphical representation of the true positive rate against the false positive rate. The AUC ranges from 0 to 1, where 0 indicates a model with no predictive power and 1 indicates a model with perfect predictive power.

The AUC evaluates how well the model ranks positive instances against negative ones, this means that whenever a positive and negative example are chosen at random, the model will rank the positive example higher than the negative one. This characteristic allows it to remain effective even when there are shifts in the underlying data distribution.

3.2 Models

Simple covariate shift can lead to serious degradations in model performance. In this chapter, we will analyze how different statistical learning models' performances are affected by covariate shift.

We evaluated a diverse set of models, ranging from simple linear classifiers to more sophisticated ensemble methods, to comprehensively assess their robustness under covariate shift conditions:

- **Logistic Regression**: A linear classifier that serves as a baseline model;
- **Decision Tree**: A simple non-linear model that creates a tree-like structure of decision rules;
- **Generalized Additive Model (GAM)**: A flexible statistical model that combines the interpretability of linear models with the ability to capture non-linear relationships;
- **Gradient Boosted Trees**: An ensemble learning method that creates sequential decision trees to iteratively correct prediction errors. Each tree focuses on the residuals from previous predictions, with optimized step sizes to minimize the overall loss function using gradient descent;
- **eXtreme Gradient Boosting (XGBoost)**: A highly optimized implementation of gradient boosting machines known for its efficiency and performance.

Each model was selected for its unique characteristics and widespread use in practical applications, providing a comprehensive view of how different learning approaches handle distribution shifts.

3.3 Results

We firstly evaluated the performance of each vanilla model on the aforementioned statistical mixtures of original and shifted test data.

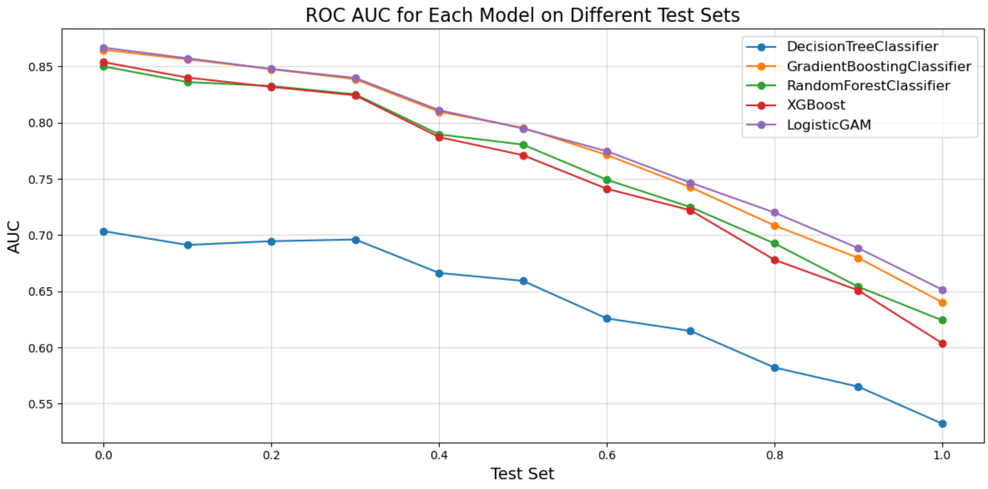

Figure 3.1. Performance of vanilla models on datasets with varying degrees of covariate shift. The AUC scores of the models are plotted against the mixing probability p, which represents the proportion of shifted data in the test set.

As we can see from the plot, at low levels of mixed data, the models can still perform realatively well. However, as the proportion of mixed data increases, the performance of the models degrades significantly. The decision tree model is the most sensitive to covariate shift, while the GAM model is the most robust.

Given these preliminary results, we proceeded with hyperparameter optimization to potentially enhance model performance.

Fine Tuning

The best choice for hyperparameters in this models might be different depending on the dataset. In order to find the hyperparameters that best fit the data, we performed a hyperparameter tuning using the `GridSearchCv` function from the `scikit-learn` library, wich performs a cross-validation (k=5) to select the best hyperparameters for each model.

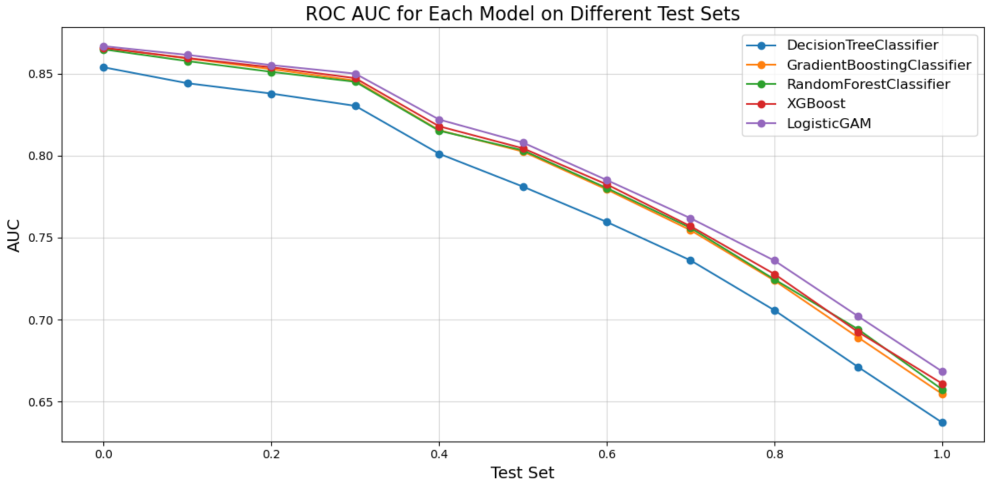

Figure 3.2. Performance of fine tuned models on datasets with varying degrees of covariate shift.

As shown in Figure 3.2, the fine-tuned models exhibit improved performance compared to the vanilla models. The decision tree model, which was the most sensitive to covariate shift, now performs almost on par with the other models. The GAM model remains the most robust, with consistently high AUC scores across all mixing probabilities.

Overfitting

In order to overfit the models, we created a custom script to perform grid search withot cross-validation. We favored the hyperparameters that achieved the highest score on the training set, while ignoring the necessary precaution to avoid overfitting, i.e. we overshot the maximum depth reachable by the trees.

Since the logistic GAM is a method which automatically selects the functions that best fit the data, we did not perform any overfitting on it. The results for the other models are shown in Figure 3.3.

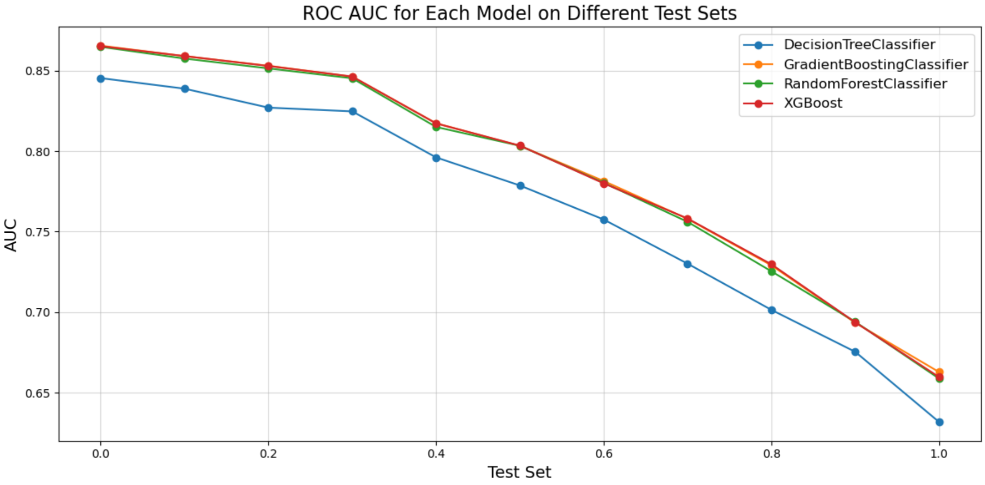

Figure 3.3. Performance of overfitted models on datasets with varying degrees of covariate shift.

As shown in Figure 3.3, we can clearly see that as we go through increasing percentage of mixed data points, the performance of the models are not comparable with the ones that were fine tuned. As the plot forthere points out, the overfitting lead to an overall levelling of the models' performances, making it difficult to spot any significant difference between them.

4
Performance Enhancement

One of the most uded approach to mitigate covariate shift conseqences is *Reweighting*, which consists in quantify the degree of distribution shift and then apply a correction to the model [1]. Another approach is *Data Augmentation*, which consists in generating new data points from the original ones, in order to make the model more robust to the distribution shift [2].

In this chapter we propose a robust training method which, in a preliminary analysis, seams to outperform the other models in terms of robustness to covariate shift.

This method is based on the idea of **Data Augmentation**. Instead of using training data as it is, we create new data applying the follofing transformation to the original data:

Algorithm 4.1 Custom Data Augmentation

```
Input: $Data, N$
Output: $Data_{\text{aug}}$

$Size \leftarrow len(Data)$
$Data_{\text{new}} \leftarrow Data$
$Data_{\text{tr}} \leftarrow$ random subset of $N\%$ of $Data$
for $x_i$ in $Data_{\text{tr}}$ do
    $x'_i \leftarrow \begin{cases} X_i + \varepsilon & \text{with probability } 0.5 \\ X_i - \varepsilon & \text{with probability } 0.5 \end{cases}$
    $y'_i \leftarrow y_i$
end for
$Data_{\text{aug}} \leftarrow Data_{\text{new}} \cup Data_{\text{tr}}$
$Data_{\text{aug}} \leftarrow Downsample(Data_{\text{aug}}, Size)$
return $Data_{\text{aug}}$
```

We firstly evaluated this method on the same classification task as the other models, but, since we needed a better way to theoretically understand the inner processes of the training, we decided to apply it to a simple 1-dimensional regression problem.

Bibliography

[1] Haoran Zhang et al. *”Why did the Model Fail?”: Attributing Model Performance Changes to Distribution Shifts*. 2023. arXiv: `2210.10769` `[cs.LG]`. URL: `https://arxiv.org/abs/2210.10769`.

[2] Mengnan Zhao et al. *Adversarial Training: A Survey*. 2024. arXiv: `2410.15042` `[cs.LG]`. URL: `https://arxiv.org/abs/2410.15042`.