Covariate Shift in Machine Learning

In real world problems, it may happen that train and test datasets have not the quality of a machine learning model. This phenomenon is called **dataset shift**.

In this post, you will learn what is and how data scientist deal with **covariate shift**, which is one of the dataset shift types.

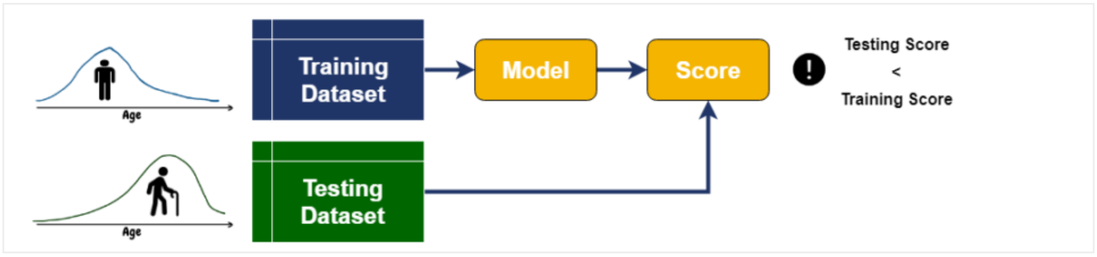
What is Dataset Shift?

Imagine that you are working as a data scientist in an insurance company. Your goal is to predict the price of an insurance policy based on the historical data including variables such as age, employment type, income, etc.

You received a training and testing datasets, then performed feature engineering, feature selection, modelling and compared training and testing scores. It turns out that whichever model you use, your **testing score is always much lower than the training score**. What could be the reason behind it?

You took a closer look into training and testing datasets and it turned out that the average age of policyholders in your testing dataset is higher than in the training dataset!

In other words, the **age distribution in the training set is different than in the testing set**.



Example of a dataset shift. The distribution of age in the training dataset is different than in the testing dataset.

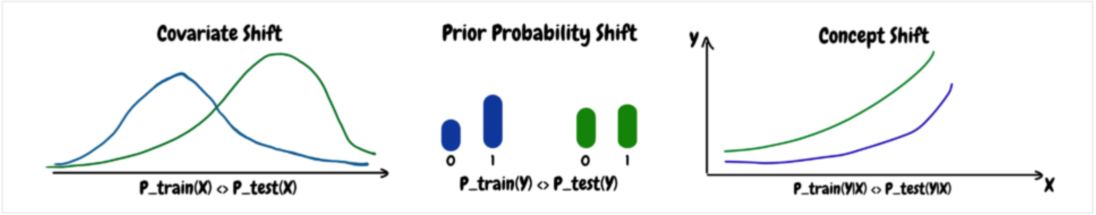
Due to the fact that age is a very important variable in price modelling you price for older people, as it was fitted on younger policyholders.

This is exactly what is called **dataset shift** (also called dataset drift).

Types of Dataset Shift

We can divide the dataset shift into three types:

- **Covariate Shift –** Shift in the independent variables.
  In short words, this is exactly like the above example with policyholders ages. This is a topic of this post.
- **Prior Probability Shift –** Shift in the target variable.
  In short words, it happens when the distribution of target variable in training and testing dataset is different. For instance, unbalanced classes between training and testing datasets in classification problems.
- **Concept Shift –** Shift in the relationship between the independent and the target variable.
  In short words, it happens when training data is collected at different point of time comparing to testing data (also called non-stationarity in time series analysis).

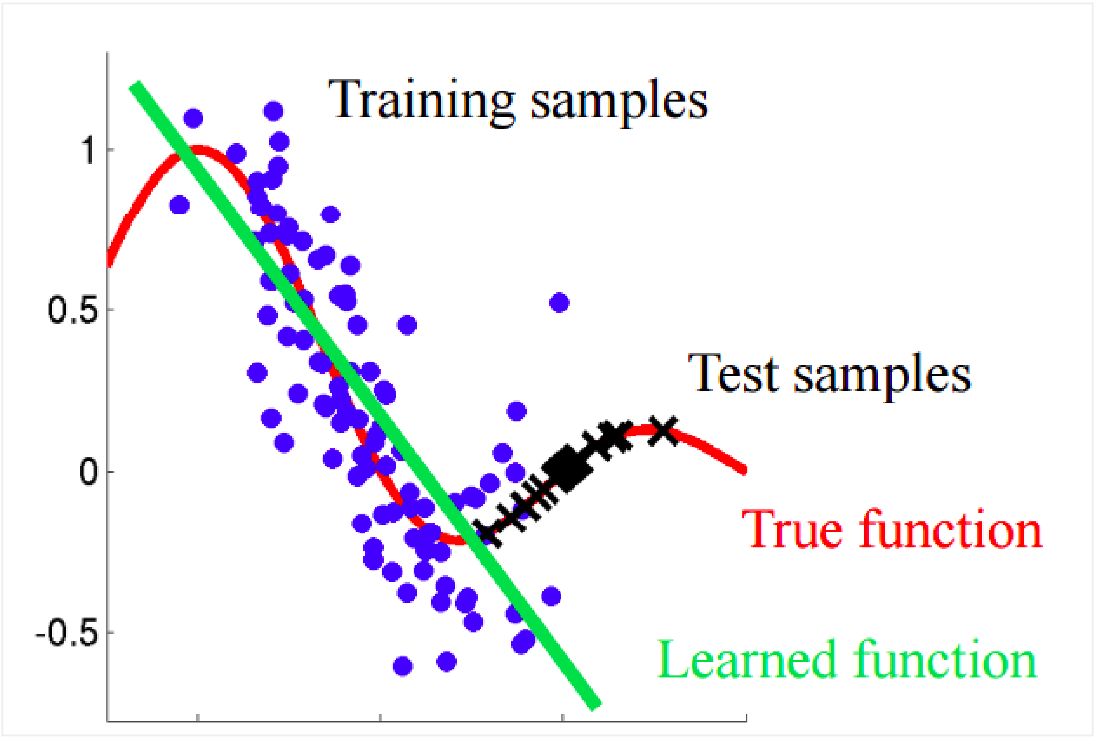


Three types of dataset shift. (1) Different X distributions in covariate shift. (2) Different Y distributions in prior probability shift. (3) Different relationship between Y and X in concept shift.

Covariate Shift

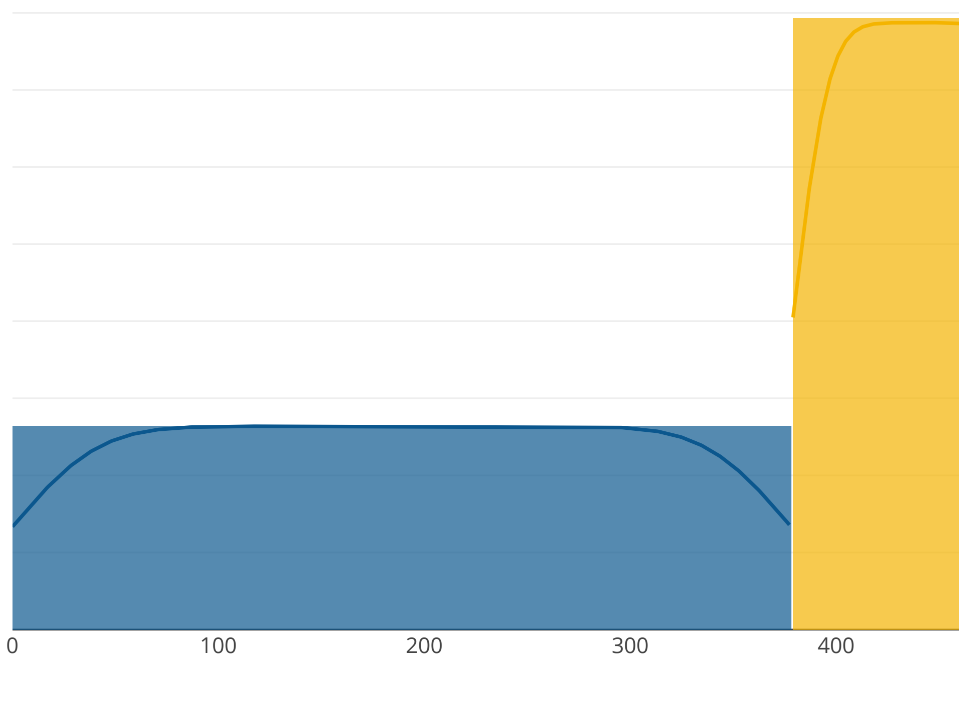
Definition

Covariate shift occurs when the distribution of input variables is different between training and testing dataset. Mathematically, covariate shift occurs if $P_{train}(X) \neq P_{test}(X)$ and $P_{train}(Y|X) = P_{test}(Y|X)$, where X is a feature.



Visualization of covariate shift. Image source

Let's analyze an example. Sometimes, a dataset includes ID variable which is an index of observations. The distributions of ID variable in training and testing datasets are clearly non-overlapping. For instance, the ID in training dataset can be from 0 to 378 and in test dataset is from 379.



EDIT CHART

If we merge the datasets and reshuffle, we can still tell which observation belongs to train and test datasets by examing ID variable. It means that ID is a drifting variable. Of course, it just the simplest example we can imagine, normally we have more complicated cases in which it is not so straightforward to spot a difference between distributions.

How to deal with drifting variables? The simplest strategy is to delete the variable **manually**, but it only works for features like ID for which it is obvious that it's drifting.

Can we **automatically identify** the drifting variables?

Solution

Here is the simple algorithm:

training or testing dataset,

2. Merge training and testing datasets,
3. Create a classification model taking **examined feature** as an explanatory variable and **origin feature** as a target on a part of merged dataset (~75%),
4. Predict on the rest of merged dataset (~25%) and calculate **ROC-AUC**.
5. If ROC-AUC is greater than a threshold (let's say 0.8) we identify the examined variable as drifting.

As we identified the drifting variables, we can drop them and safely carry on next steps (feature engineering, modelling, etc).

If we do not want to simply drop a drifting variable, there exist alternative approaches:

- Importance weighted cross validation
- Integrated optimization problem. Discriminative learning
- Kernel mean matching

Example

The code snippet below is using Boston Housing Pricing dataset with ID variable added. We expect that the ROC-AUC score for ID variable will be over 0.8, so that we identify it as drifting variable. Let's see if our algorithm works.

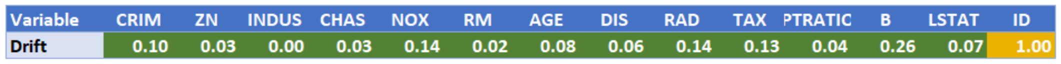
Feel free to use full code hosted on GitHub.

```
# Import objects
import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import KFold, cross_val_predict, train_test_split
from sklearn.datasets import load_boston

# Load data
dataset = load_boston()
df = pd.DataFrame(dataset.data, columns=dataset.feature_names)

# Randomly split into train and test datasets
```

```
# Add ID variable (we expect that the variable will be identified as drifting)
```

```python
df_train['ID'] = np.arange(len(df_train))
df_test['ID'] = np.arange(len(df_train),len(df))

# Loop over each column in X and calculate ROC-AUC
drifts = []
for col in df_train.columns:
    # Select column
    X_train = df_train[[col]]
    X_test = df_test[[col]]

    # Add origin feature
    X_train["target"] = 0
    X_test["target"] = 1

    # Merge datasets
    X_tmp = pd.concat((X_train, X_test),
                      ignore_index=True).drop(['target'], axis=1)
    y_tmp= pd.concat((X_train.target, X_test.target),
                   ignore_index=True)

    X_train_tmp, X_test_tmp, \
    y_train_tmp, y_test_tmp = train_test_split(X_tmp,
                                               y_tmp,
                                               test_size=0.25,
                                               random_state=1)

    # Use Random Forest classifier
    rf = RandomForestClassifier(n_estimators=50,
                                n_jobs=-1,
                                max_features=1.,
                                min_samples_leaf=5,
                                max_depth=5,
                                random_state=1)

    # Fit
    rf.fit(X_train_tmp, y_train_tmp)

    # Predict
    y_pred_tmp = rf.predict_proba(X_test_tmp)[:, 1]

    # Calculate ROC-AUC
    score = roc_auc_score(y_test_tmp, y_pred_tmp)

    drifts.append((max(np.mean(score), 1 - np.mean(score)) - 0.5) * 2)

print(drifts)
```

The drifts produced:



Variable	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIC	B	LSTAT	ID
Drift	0.10	0.03	0.00	0.03	0.14	0.02	0.08	0.06	0.14	0.13	0.04	0.26	0.07	1.00

ID variable has drift of 1.0 and it is identified as a drifting variable.

The ID ROC-AUC is 1.0, so it is clearly a drifting variable. You can use exactly the same algorithm for your real datasets to identify more sophisticated drifting variables!

Moreover, more robust version can be found in automated machine learning packaged called MLBox

Summary

You just learned what are the threats of drifting variables in real machine learning tasks and how to tackle the problem.

I encourage you to explore other types of dataset shift (concept and prior probability shifts) as well as other approaches to approach covariate shift:

- Importance weighted cross validation
- Integrated optimization problem. Discriminative learning
- Kernel mean matching

Dawid Kopczyk

July 29, 2019

Machine Learning

Previous post

Leave a Reply

Your email address will not be published.

Name

Email

Website

Notify me of follow-up comments by email.

Notify me of new posts by email.

Post Comment

SEARCH FORM

Search form

RECENT POSTS

- » Covariate Shift in Machine Learning
- » Variational Quantum Circuits
- » From Decision Trees to Gradient Boosting

BLOG LINKS

Privacy Policy