Imperial College
London

DEEP LEARNING (70010)

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Correction Tutorial 4 - Covariate Shift

1 Introduction

Covariate shift is a type of causal (x causes y) distribution shift, which assumes that the distribution of input can change over time, while the labeling function $P(y|x)$ stays the same. It is called "covariate shift" because the problem comes from the shift of the features (covariates) distribution.

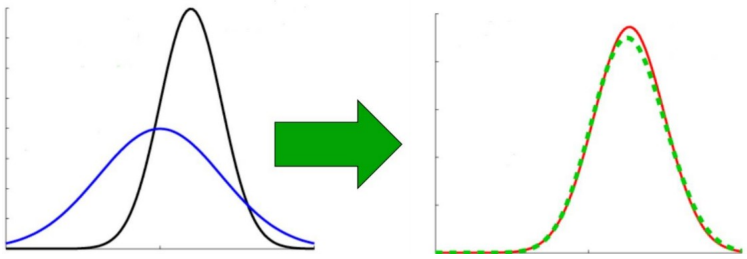

Figure 1: Effect of covariate shift correction on data distribution

1.1 Examples

Let us consider the following training set (figure 2) and test set (figure 3).

Figure 2: Training set

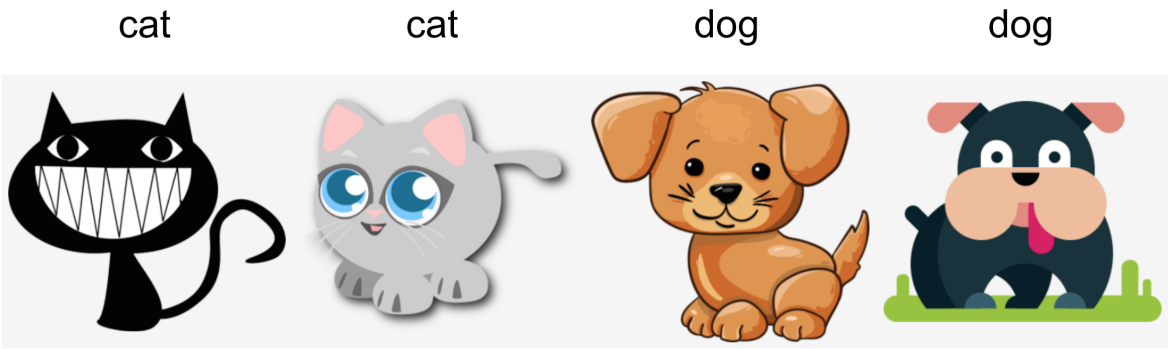

Figure 3: Test set

While the labels are the same, our samples have very different features. This is an extreme case of covariate shift, but there are many subtles cases. By default, you can assume that any deep learning model suffers from covariate shift. The real questions are (1) how much does it affect your use case and (2) how to make that model robust to your use case.

Here is a list of real examples where scientist have overlooked the importance of the data distribution while building a model:

- English speech recognition trained on native English speakers and applied to any English speaker.
- Classifying images as either cats or dogs and omitting certain species from the training set that are seen in the test set.
- Prostate cancer detection trained on old men with cancer and healthy university students, applied to potentially sick old men.

1. For the last case, can you explain why this data distribution is very problematic (compare test time and real world application) ?

A1: At training and test time, the results will be very good as the network will learn to differentiate young and old people. In a real world setting, the network will see only old men with potential illness, and tell them they have cancer, only because they are old.

2 Mathematical expression

We want to estimate $P(y|x)$ and we have access to $(\boldsymbol{x}_i, y)$. Covariate shift arises because the *training distribution* comes from $q(\boldsymbol{x})$ (source distribution) while the *test distribution* comes from $p(\boldsymbol{x})$ (target distribution).

We can write:

$$p(x, y) = p(x)q(y|x) \tag{1}$$

The training risk is written as:

$$\underset{w}{\text{minimize}} \int \int q(x)q(y|x)l(f(x, w), y)\, dy\, dx \tag{2}$$

$$\text{or, } \underset{w}{\text{minimize}} \frac{1}{m} \sum_{i=1}^{m} l(f(x_i, w), y_i) \tag{3}$$

where l is a loss function, x the training samples, y the corresponding labels and m the number of samples.

The test risk is different, and written as:

$$\underset{w}{\text{minimize}} \int \int \underline{p(x)}q(y|x)l(f(x, w), y)\, dy\, dx \tag{4}$$

2.1 Weighting problem

Now that you have the mathematical notations ready, find a way to correct the covariate shift by weighting the distributions q and p. (*Hint: look at eq. 1*)

$$\int p(x)f(x)dx = \int q(x)\alpha(x)f(x)dx \tag{5}$$

2. What is $\alpha(x)$?

3. Rewrite eq. 3 with α plugged in.

A2: $\alpha(x) = \frac{p(x)}{q(x)}$
A3: $\underset{w}{\text{minimize}} \frac{1}{m} \sum_{i=1}^{m} \alpha(x) l(f(x_i, w), y_i)$

2.2 Estimating α

How do we estimate α ? Multiple approaches exist, like operator-theoretic approaches using minimum norm or maximum entropy principles. To use such methods, you would need samples from both distributions p and q. Note that only the features are needed ($\boldsymbol{x} \sim p(\boldsymbol{x})$) and not the labels ($y \sim p(y)$), therefore these methods are usable in real world settings.

Here, we are going to estimate α with a very effective approach called logistic regression (softmax regression for binary classification). We train a binary classifier to differentiate between our training set q and test data p. If it cannot distinguish between the 2 distributions, it means that the instances are representative of the two distributions. On the other hand, instances that can be discriminated should be overweighted or underweighted accordingly.

We assume that we have an equal number of instances from both distributions $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$. We define z as the labels, with $z = 1$ for $p(\boldsymbol{x})$ and $z = -1$ for $q(\boldsymbol{x})$. We can therefore write:

$$P(z = 1|\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{p(\boldsymbol{x}) + q(\boldsymbol{x})} \iff \frac{P(z = 1|\boldsymbol{x})}{P(z = -1|\boldsymbol{x})} = \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \tag{6}$$

4. By using the logistic regression approach $P(z = 1|\boldsymbol{x}) = \frac{1}{1+\exp(-h(\boldsymbol{x}))}$, where h is a parameterized function, compute α.

A4:

$$\alpha(\boldsymbol{x}) = \frac{\frac{1}{1+\exp(-h(\boldsymbol{x}))}}{\frac{-h(\boldsymbol{x})}{1+\exp(-h(\boldsymbol{x}))}} = \exp(h(\boldsymbol{x}))$$

2.3 Covariate shift correction algorithm

You are now given a training set $\{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$ and an unlabeled test set $\{\boldsymbol{u}_1, ..., \boldsymbol{u}_m)\}$. We assume that $\boldsymbol{x}_i$ are sampled from the source distribution $q(\boldsymbol{x})$ and that $\boldsymbol{u}_j$ are sampled from the target distribution $p(\boldsymbol{x})$, where $1 \leq i \leq n$ and $1 \leq j \leq m$.

5. Design a prototypical algorithm (no code, plain English and math) to correct covariate shift.

6. In which case does this method breaks? (*hint: check for extreme cases*)

A5:

1. Generate a binary-classification training set $(x_n, -1), (u_m, 1)$.
2. Train a binary classifier using logistic regression to get function .
3. Weigh training data using $\alpha = \exp((\boldsymbol{x}))$ or better $\alpha = \min(\exp((\boldsymbol{x})), c)$ for some constant c.
4. Use weights α for training on $(x_1, y_1), \ldots, (x_n, y_n)$

A6: If a sample has 0 chance of appearing in the training set but does appear in the test set, it will have infinite weight.