What to do when your training and testing data come from different distributions?

When data shift happens, it gives rise to a problem of domain adaptation, which typically focuses on adapting a model from a single source domain to a target domain. One of the major problems that we might be facing is: Data mismatch.

Figure out whether the problem was due to data mismatch!

Let’s divide the dataset into train, bridge, validation, and test set (when we don’t have a separate test set). Here the bridge set gets a subset of data from the train set and that it is not included while training the model. Let’s assume the validation and the test set comprise of the target distribution dataset. After training the model using the train set, we shall predict the values in the bridge set and also in the validation set.

For example: the error is as follows:

Train set: 2% error

Bridge set: 3% error

Validation set: 10% error

The 8% error difference between the train and validation set could mean that we are facing a data mismatch problem as the validation set involves data representative of the target distribution. It means that when the model faces unseen data that do not have the same distribution as the data in the train set, it doesn’t perform well, and the predictions are far off.

How to deal with the problem?

To reduce the data mismatch error, you would need to somehow incorporate the characteristics of the dev/test datasets — the target distribution — into the train set.

How?

1. Collect more data representative of the test set distribution and add them to the train set.
2. Error analysis: Analyzing the errors on the dev set and how they are different from the errors on the train set could give us ideas to address the data mismatch problem.

For example, if you find many of the errors on the dev set occur when the background of the animal’s image is rocky, you can mitigate such errors by adding animal images with rocky background to the train set.

1. Artificial data synthesis: Another way to incorporate the characteristics of the dev/test sets into the train set is to synthesize data with similar characteristics.

For example, we mentioned before that the images in our dev/test sets are mostly blurry in contrast to the clear images from the web that make most of our train set. You can artificially add blurriness to the images of the train set to be more similar to the dev/test sets. But the problem of this approach is: we could end up overfitting our classifier to the artificial characteristics we made.

In our example, the blurriness we artificially made by some mathematical function might only be a small sub-set of the blurriness that exists in the images of the target distribution. In other words, the blurriness in the target distribution could be due to many reasons. For example, fog, low resolution camera, subject movement could all be causes. But our synthesized blurriness may not represent all of these causes. Making sure subtle changes of the test dataset are captured well might help to mitigate the problem.