Covariate Shift - Consequences and good practice

Covariate shift, re-weight training data, active sampling

Joyce Wang | Software Engineer
Sep 2017
Motivation

What is going on here?

Validation Accuracy = 0.96

Query Accuracy = 0.67
Outline

● What is covariate shift?
   ○ why would it occur?
   ○ what consequence would it have?

● How to detect covariate shift?
   ○ visualization method
   ○ quantitative method

● Strategies to handle covariate shift
   ○ training data reweighting
   ○ active learning
Covariate Shift

When the distribution on training and test/query sets do not match, we are facing *covariate shift*, or *sample selection bias*.

**Against** fundamental assumption:

Both the training and query data should be drawn from the same population / distribution.
Distribution Mismatch

Training data and query data are drawn from almost the **same** population

Training data and query data are drawn from completely **different** population
Covariate Shift - Commonplace

Lack of randomness

Inadequate samples

Biased sampling rules
Covariate Shift - Consequence

- Overfitting on training examples
- Unreliable predictions

Example: binary classification
Detect Covariate Shift
Detect Covariate Shift

- Visualization
- Membership modelling
- Uncertainty quantification
Visualize Training and Query Data

What if I have high-dimensional data?

- Per dimension visualization
- Dimensionality reduction (PCA, t-SNE)

We need more robust methods.
We apply a model to predict the probability of a new point being a member of training set.

For example, one-class SVM could classify new data as similar or different to the training set.
Uncertainty Quantification

1. Fit a *probabilistic* model to training set

2. Every prediction has uncertainty (confidence interval) associated with it

3. Determine covariate shift with uncertainty of predictions
Uncertainty Quantification

low uncertainty $\rightarrow$ similar to training dataset

high uncertainty $\rightarrow$ not similar to training dataset
Handle Covariate Shift
Handle Covariate Shift

- Training Sample Reweighting
  - Make the distribution of training data look like the distribution of query data.

- Active Sampling
  - Help model gain understanding about query data and learn effectively.
Sample Reweighting

- Build a classifier to classify training and query sets
  - e.g. logistic regression

Color training points by the probability of being in query set

| Low | Median | High |
Sample Reweighting

- Reweight every training point in learning process.

<table>
<thead>
<tr>
<th>Training samples</th>
<th>Probability of being in query set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9872</td>
</tr>
<tr>
<td>2</td>
<td>0.8754</td>
</tr>
<tr>
<td>3</td>
<td>0.7913</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>n-1</td>
<td>0.2877</td>
</tr>
<tr>
<td>n</td>
<td>0.1867</td>
</tr>
</tbody>
</table>
Overlap is essential to apply sample re-weighting.
Active Learning

- Train a **probabilistic** model.
- Predict query set with trained model.

Find the query point with *that is expected to most improve the model*

- Get the *target value* for that most useful point.
- Put the point into training set.
Active Learning - Demo
Active Learning - Demo
Active Learning - Demo
Active Learning - Demo
Active Learning - Demo
Active Learning - Demo
Active Learning - Demo
## Comparison of Strategies for Handling Covariate Shift

<table>
<thead>
<tr>
<th></th>
<th>Sample Reweighting</th>
<th>Active Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td>● achievable if you cannot get more samples</td>
<td>● no need for overlap</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● gain more understanding about query data</td>
</tr>
<tr>
<td><strong>Disadvantages</strong></td>
<td>● need overlap between training and query sets</td>
<td>● not achievable if you cannot get more samples</td>
</tr>
<tr>
<td></td>
<td>● less understanding on data</td>
<td></td>
</tr>
</tbody>
</table>
Thank you

twitter @joycexinyuewang
email joyce.wang@data61.csiro.au
Reference

- **Density Ratio Estimation in Machine Learning**

- **Correcting Sample Selection Bias by Unlabeled Data**
Uncertainty Quantification

probability of positive label
Sample Reweighting

- Reweight every training point in minimizing loss function.

\[ L(\theta) = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} J(y_i, \hat{y}_i(\theta)) \]

where
- \( L(\theta) \) is the loss function we aim to minimize with respect to \( \theta \)
- \( J(y_i, \hat{y}_i) \) is the cost associated with a single sample
- \( y_i \) is the actual target value for training sample \( i \)
- \( \hat{y}_i \) is the predicted target value of training sample \( i \)

<table>
<thead>
<tr>
<th>Training samples</th>
<th>Probability of being in query set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9872</td>
</tr>
<tr>
<td>2</td>
<td>0.8754</td>
</tr>
<tr>
<td>3</td>
<td>0.7913</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>n-1</td>
<td>0.2877</td>
</tr>
<tr>
<td>n</td>
<td>0.1867</td>
</tr>
</tbody>
</table>
Acquisition Function

- Reduce the maximum uncertainty
  \[
  \arg\max_x \left| \text{Var}\{f(x)\} \right| \quad \text{where } f(x) = \hat{y}
  \]

- Reduce the maximum upper confidence bound
  \[
  \arg\max_x \left| f(x) + \kappa \sigma(x) \right| \quad \text{where } f(x) = \hat{y}
  \]

- Reduce the total uncertainty
  \[
  \arg\max_x \int_x \text{Var}\{f^*(x)\} \quad \text{where } f^*(x) = \hat{y} \quad \text{after including new sample}
  \]

- Utility function if policy is known
## Detect Covariate Shift - Comparison

<table>
<thead>
<tr>
<th></th>
<th>Visualization</th>
<th>Membership Modelling</th>
<th>Uncertainty Quantification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantage</strong></td>
<td>quick</td>
<td>informative</td>
<td>informative</td>
</tr>
<tr>
<td></td>
<td></td>
<td>quantitative</td>
<td>quantitative</td>
</tr>
<tr>
<td><strong>Disadvantage</strong></td>
<td>subjective</td>
<td>sensitive to tuning</td>
<td>difficult to work</td>
</tr>
<tr>
<td></td>
<td>open to</td>
<td>parameters</td>
<td>with large-size data</td>
</tr>
<tr>
<td></td>
<td>interpretation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Sample Reweighting

- Apply trained classifier to obtain the probability of each training point being inside query set

Use cross-validation to avoid over-fitting.

<table>
<thead>
<tr>
<th>Training samples</th>
<th>Probability of being in query set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9872</td>
</tr>
<tr>
<td>2</td>
<td>0.8754</td>
</tr>
<tr>
<td>3</td>
<td>0.7913</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>n-1</td>
<td>0.2877</td>
</tr>
<tr>
<td>n</td>
<td>0.1867</td>
</tr>
</tbody>
</table>
Hold-out / Development set

Apply model to predict the y value

Glossary

Training data

Split

90% Training set

10% Hold-out / Development set

Test data

used to

Validate model (optional)

Query data
Sample Reweighting

- Reweight every training point in learning process.