Efficient Sampling-Based Kernel Mean Matching

Swarup Chandra, Ahsanul Haque, Latifur Khan and Charu Aggarwal*
University of Texas at Dallas
*IBM Research
Data Classification

Training Data

Model

Test Data
Data Classification

- Training Data
- Similar Data Distribution
- Test Data

Model
**Data Classification**

- **Training Data**
- **Test Data**
- **Domain 1**
- **Domain 2**

**Domain Adaptation**

**Model**
Data Classification

Domain 1

Training Data

Test Data

Biased Data

Sample Selection Bias

Model
Data Classification

• Biased Training Data
  • Limited labeled data
  • High cost
• Covariate shift assumption
  • Between training (tr) and test (te) distribution:
    \[ p_{tr}(y|x) = p_{te}(y|x) \]
  • Equal conditional distribution:
    \[ p_{tr}(x) = p_{te}(x) \]
  • Unequal marginal distribution:
• Use training labels to classify test data:
  • Solution: make
  • Compute instance weight
    \[ \beta(x) = \frac{p_{te}(x)}{p_{tr}(x)} \]
Kernel Mean Matching

• Minimize mean distance between weighted training data distribution and test data distribution

\[ \left\| E_{x \sim p_{tr}(x)}[\beta(x)\phi(x)] - E_{x \sim p_{te}(x)}[\phi(x)] \right\| \]

• Maximum Mean Discrepancy

\[ \hat{\beta} \approx \text{minimize} \frac{1}{2} \beta^T K \beta - \kappa^T \beta \]

subject to \( \beta(x^{(i)}) \in [0, B], \forall i \in \{1 \ldots n_{tr}\} \)

\[ \left| \sum_{i=1}^{n_{tr}} \beta(x^{(i)}) - n_{tr} \right| \leq n_{tr} \epsilon \]

\[ K^{(ij)} = h(x^{(i)}_{tr}, x^{(j)}_{tr}) \quad \kappa^{(i)} = \frac{n_{tr}}{n_{te}} \sum_{j=1}^{n_{te}} h(x^{(i)}_{tr}, x^{(j)}_{te}) \]

Requires complete training and test data to be in the memory
Kernel Mean Matching

- Time Complexity: $O(n_{tr}^3 + n_{tr}^2d + n_{tr}n_{te}d)$

- Related Work: Ensemble Kernel Mean Matching

Sampling-Based Approach

• Very Fast Kernel Mean Matching (VFKMM)
  • $m/n$ bootstrap sampling
  • Sample training data with replacement
  • Minimum number of samples such that each training instance is associated with at least one sample

\[
\frac{\ln \eta}{m \ln \left(1 - \frac{1}{n_{tr}}\right)}
\]
Sampling-Based Approach

• Extended Very Fast Kernel Mean Matching (EVFKM)
  • Sample training data with replacement
  • Split test data into k parts (sampling without replacement)
Empirical Evaluation

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Features</th>
<th>Total Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ForestCover</td>
<td>54</td>
<td>50,000</td>
</tr>
<tr>
<td>KDD</td>
<td>34</td>
<td>50,000</td>
</tr>
<tr>
<td>Syn002</td>
<td>70</td>
<td>50,000</td>
</tr>
<tr>
<td>MNIST</td>
<td>780</td>
<td>50,000</td>
</tr>
</tbody>
</table>

- Available on UCI data repository
- Training data bias induction:

\[ p(\xi = 1|x^{(i)}) = \exp \frac{-||x^{(i)} - x||}{\sigma} \]

Competing Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cenKMM</td>
<td>Original Method</td>
</tr>
<tr>
<td>ensKMM</td>
<td>Related Work* (split test data)</td>
</tr>
<tr>
<td>ensTrKMM</td>
<td>Baseline Method (split training data)</td>
</tr>
<tr>
<td>VFKMM</td>
<td>Proposed Method (sample training data)</td>
</tr>
<tr>
<td>EVFKMM</td>
<td>Extended VFKMM (also split test data)</td>
</tr>
</tbody>
</table>

SR: Sampling with replacement
SWR: Sampling without replacement

Results

Execution time with different sample size

- (a) ForestCover
- (b) KDD
- (c) Syn002
- (d) MNIST

$k = \text{number of test data split}$
$m = \text{training sample size}$

For uniformity: $k \left( \propto \frac{1}{m} \right)$
Results

NMSE with different sample size

\[
NMSE = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{\hat{\beta}(x^{(i)})}{\sum_{j=1}^{n} \hat{\beta}(x^{(j)})} - \frac{\beta(x^{(i)})}{\sum_{j=1}^{n} \beta(x^{(j)})} \right)
\]
Results

NMSE with different number of samples

(a) ForestCover  (b) KDD  (c) Syn002  (d) MNIST

Graphs showing NMSE for different datasets and models.
Results

Execution time with different training dataset size

(a) ForestCover  (b) KDD  (c) Syn002  (d) MNIST
Results

NMSE with different training dataset size

(a) ForestCover  (b) KDD  (c) Syn002  (d) MNIST
Conclusion

• Scalable sampling-based method for Kernel Mean Matching
  • Use M/N bootstrap sampling to generate training data
  • Combine training data instance weights
• Fully scalable KMM
  • Sampling over training dataset
  • Splitting of test dataset.
• Empirical results show large improvements in execution time with similar error.