Synchronization-based Classification on Distributed Concept-drifting Data Streams
Introduction

Classification

- Classification is a type machine learning task which infers a function from labeled training data.

Distributed and parallel Classification

- The abundance of data and the need to process larger amounts of data have triggered machine learning development.
- Classic classification algorithms are modified into scaled-up versions which require for distributed machine learning.

Streaming Classification

- Another development of machine learning is in processing continuous supply of data.
- The training needs to be performed again from the beginning with the new arrived data and it is costly and time-consuming, dealing with concept-drift.
Recent works can be summarized in two basic models

Central Learning Model and Distributed Learning Model are two basic models established to deal with distributed data streams.

Fig. 2 (a) Central Learning Model suffers inexpensive data storage and communication (b) Distributed Learning Model suffers the presence of concept drift and lack of modeling the dynamic dependence among streams.
Motivation

Focusing on Challenges on Distributed Data Streams Classification

- How to handle concept-drift of local streams
- How to learn and model the dynamic dependence or association among data streams over time?
- How to combine all information for prediction

How to utilize the similarity and learn the association of these large-scale data with distributed data streams?
Modeling the Association Among Data Streams

Since different data streams often has association in real world data driven-applications, so we established a new learning model.

Fig.2 Principle of Combined data for prediction
Overview of this Framework

Fig. 3 Framework Overview
Learn Local patterns by dynamic Prototype-based Learning

Maintain a small set of important prototypes for each data streams

a) error-driven learning approach
b) synchronization-inspired constrained clustering
c) PCA and statistics

Fig. 4 An illustration of the P-Tree structure.
Error-driven Representativeness Learning

How to dynamically select the short-term and/or long-term representative examples?

**Basic idea:** Leverage the prediction performance of test examples to infer the representativeness of examples by lazy learning: nearest neighbor classifier.

\[ \text{Rep}(y) = \text{Rep}(y) + \text{Sign}(x_p, x_l) \]

where \( \text{Sign}(x, y) \) is the sign function, and 1 if \( x \) equals \( y \), -1 otherwise.

- **High representativeness** —— Keep
- **Low representativeness** —— Delete
- **Unchanged representativeness?** —— Summarization (Sync. Algorithm)
Summarization: Constrained Clustering by Synchronization

\[ x_i(t + \Delta t) = x_i(t) + \frac{1}{|N_\varepsilon(x(t))|} \cdot \sum_{y \in N_\varepsilon(x(t)), eq(lx, ly)} \sin(y_i(t) - x_i(t)) \]

(a) Constrained clustering by synchronization

(b) Prototype-based data representation

Cannot Link
PCA and statistics

- **Principle Component Analysis (PCA):** Analyze the change of each class data distribution by principle component of two sets of examples.

- **Statistical Analysis:** Compute a suitable statistic, which is sensitive to data class distribution changes between the two sets of examples.

![Fig.5 PCA-based concept drift analysis](image1)

![Fig.6 Statistical Analysis](image2)
weight vector

Fig. 8 Data Structure for Weight Vector
How to Learn information from other data streams

a) Maintain a weight vector for each other data streams by using dynamic error-driven learning

b) Learn the relevant data streams which are really useful for prediction of test data

**Fig. 7** Process of Learning information from other streams.
A decay function for the weight

For each contribution (correct prediction), it will be decreased over time with a decay function. (i.e., the old correct prediction is less important than the recent correct prediction using other data streams)

$$W_k^{(i)}(t) = \sum_{k=1}^{n} W_k^{(i)}(x_k, t) = \sum_{k=1}^{n} \alpha^{-\lambda t} \cdot W_0^{(i)}(x_k)$$
Fig. 10 Ensemble Learning Process: using Weighted Majority
- **Ensemble Learning Framework**:

\[
\arg \max_c P_c(x) = \left( \sum_{i=1}^{k-1} W_k^{(i)}(t) \text{pre}_c(x) + W^* \text{pre}_{c_r}(x) \right)
\]
The Process of Learning Independence from Other Data Streams

- Dealing Static data with outliers

Data with Outliers

Helping Learning Data with Outliers
Dealing data with dynamic nature

Arriving Data with out enough information

Adding information for Arriving Data
Experiments & Results
Synthetic Data

The hyperplane in 2-dimensional space was used to simulate different time-changing concepts by altering its orientation and position in a smooth or sudden manner.
- Dealing data with parting plane

- Dealing data without parting plane
Supplement Data to Help Predict

- Dealing data with parting plane

- Dealing data without parting plane
## Prediction performance

- **Data with Parting Plane**

<table>
<thead>
<tr>
<th>Central Accuracy</th>
<th>96.62%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helpful for Prediction</td>
<td>628</td>
</tr>
<tr>
<td>Harmful for Prediction</td>
<td>633</td>
</tr>
<tr>
<td>Right Prediction but not helpful</td>
<td>51362</td>
</tr>
<tr>
<td>Helpful but not use</td>
<td>0</td>
</tr>
<tr>
<td>Harmful but not use</td>
<td>242</td>
</tr>
</tbody>
</table>

- **Data without Parting Plane**

<table>
<thead>
<tr>
<th>Central Accuracy</th>
<th>98.31%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helpful for Prediction</td>
<td>229</td>
</tr>
<tr>
<td>Harmful for Prediction</td>
<td>22</td>
</tr>
<tr>
<td>Right Prediction but not helpful</td>
<td>49485</td>
</tr>
<tr>
<td>Helpful but not use</td>
<td>0</td>
</tr>
<tr>
<td>Harmful but not use</td>
<td>4</td>
</tr>
</tbody>
</table>
Data Set

1. Electricity:
   - Contains 45,312 instances,
   - which was collected from the Australian New South Wales Electricity Market for every five minutes.

2. Forest Covertype:
   - Containing 581,012 instances
   - describes seven forest cover types on a 3030 meter grid with 54 different geographic measurements.

3. Sensor:
   - Containing 2,219,803 instances
   - stream contains information (temperature, humidity, light, and sensor voltage) collected from 54 sensors
4. Power supply:

- Contains 29,928 instances, 2 attributes, and 24 classes
- An Italy electricity company which records the power from two sources: power supply from main grid and power transformed from other grids

5. Kddcup99:

- Contains 494,021 instances, 41 attributes, and 23 classes
- was collected from the KDD CUP challenge in 1999, and the task is to build predictive models capable of distinguishing between intrusions and normal connections
## Performance of data stream classification algorithm on real-world data sets

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Electricity</th>
<th>Forest Covertype</th>
<th>Sensor</th>
<th>Power Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Accuracy</td>
<td>NO.</td>
<td>Accuracy</td>
<td>Accuracy</td>
<td>Accuracy</td>
</tr>
<tr>
<td>0</td>
<td>69.31%</td>
<td>88.57%</td>
<td>75.63%</td>
<td>86.58%</td>
</tr>
<tr>
<td>1</td>
<td>68.65%</td>
<td>88.68%</td>
<td>74.63%</td>
<td>87.38%</td>
</tr>
<tr>
<td>2</td>
<td>65.99%</td>
<td>88.51%</td>
<td>75.67%</td>
<td>87.33%</td>
</tr>
<tr>
<td>3</td>
<td>69.07%</td>
<td>88.58%</td>
<td>75.63%</td>
<td>86.63%</td>
</tr>
<tr>
<td>4</td>
<td>69.78%</td>
<td>88.64%</td>
<td>57.9%</td>
<td>80.65</td>
</tr>
<tr>
<td>Global Accuracy</td>
<td>71.13%</td>
<td>89.50%</td>
<td>73.91%</td>
<td>85.61%</td>
</tr>
<tr>
<td></td>
<td>Central Accuracy</td>
<td>Helpful for Prediction</td>
<td>Harmful for Prediction</td>
<td>Right Prediction but not helpful</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------</td>
<td>------------------------</td>
<td>------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td><strong>CovtypeNorm</strong></td>
<td>89.5%</td>
<td>10442</td>
<td>5204</td>
<td>461045</td>
</tr>
<tr>
<td><strong>ElecNormNew</strong></td>
<td>71.13%</td>
<td>2179</td>
<td>1529</td>
<td>25041</td>
</tr>
<tr>
<td><strong>Power Supply</strong></td>
<td>85.61%</td>
<td>47</td>
<td>51</td>
<td>706</td>
</tr>
</tbody>
</table>
Sensitivity w.r.t. number of data streams (e.g. Covertype)

We test data with changing the number of streams.

Fig. 9 Accuracy w.r.t. number of data streams
Sensitivity w.r.t. number of neighbors (e.g. Covertype)

We test data with changing the number of k.
When k=1,3,5,10,

Fig.9 Accuracy w.r.t. neighbors
Sensitivity w.r.t. with different factor $\lambda$

We test our Algorithm with different $\lambda$. The result shows that it is stable with the $\lambda$ value 0.01, 0.1, 0.5, 1.

Let Covertype data as an example, the central accuracy is all around 90.5.
Conclusion

✓ Our method successfully deal with the concept-drift.

✓ This study provides a distributed classifying model which can learn the relevance of different streams.

✓ The final prediction can combine the information from local classifier.
Thanks for your attention!

Q & A