Incremental
Generalized Eigenvalue Classification
on Data Streams

Mario R. Guarracino, Salvatore Cuciniello, Davide Feminiano

High Performance and Networking Institute
National Research Council, Italy

Overview

✓ What is supervised learning?
✓ Classification on data streams
✓ GEPSVM: Generalized Eigenvalue Problem Support Vector Machine
✓ Problems
✓ Incremental classification
✓ SI-ReGEC: Stream Incremental Regularized Eigenvalue Classifier
✓ Numerical results
✓ Conclusions
What is supervised learning?

- Supervised learning refers to the capability of a system to learn from examples.
- The trained system is able to provide an answer for each new question.
- Supervised means the desired output for the training set is provided by an external teacher.
- Binary classification is among the most successful methods for supervised learning.

Classification on data streams

- Many applications:
  - Computer network traffic (spam, intrusion detection,...)
  - Bank transactions (fraud, credit cards,...)
  - Web search (link rating)
  - Video/Audio sensors (video surveillance, face identification,...)
Binary classification problem can be formulated as a generalized eigenvalue problem (GEPSVM).

Find $x'w_1 = \gamma_1$ the closest to A and the farthest from B.

Let $[w_1, \gamma_1]$ and $[w_m, \gamma_m]$ be eigenvectors associated to min and max eigenvalues of $Gx = \lambda Hx$.

$a \in A \iff$ closer to $x'w_1 = \gamma_1 = 0$ than to $x'w_m = \gamma_m = 0$.

$b \in B \iff$ closer to $x'w_m = \gamma_m = 0$ than to $x'w_1 = \gamma_1 = 0$. 
Problems

- Standard classification methods rely on the persistence of a complete training set.
- Data not well represented by a persistent collection of items.
- Data may be accessed but not completely loaded in main memory or stored.

Incremental classification

A new approach consists in finding a small and robust subset of the training set while accessing data available in the window.

When the window is full, all points within are processed by the classifier.
It is possible to incrementally train an algorithm using one point at time, analyzing its contribution of information.

Improvements:

✓ A smaller set of points reduces the probability of overfitting the problem

✓ It is computationally more efficient in predicting new points

✓ As new points become available in the window, their influence is evaluated with respect to the existing classifier
SI-ReGEC: Stream Incremental Regularized Eigenvalue Classifier

At each step, data in window are processed with the incremental learning classifier...

And hyperplanes are built

Step by step new points are processed

...and SI-ReGEC updates hyperplanes configuration
SI-ReGEC: Stream Incremental Regularized Eigenvalue Classifier

But not all points are considered...

Some of them are discarded if their information contribution is useless

New unknown incoming points are classified by their distance from the hyperplanes
An incremental learning technique based on GEPSVM that determines classification models based on a very small sample of data from the stream.

SI-ReGEC: Stream Incremental Regularized Eigenvalue Classifier

1: \( C = \text{load}(\text{stream}, \text{wsize}) \)
2: \( C_0 = \text{kmean}(C, ksn) \)
3: \( \Gamma_0 = C \setminus C_0 \)
4: \( \{R_0, M_0\} = \text{Classify}(C, C_0) \)
5: \( \text{repeat} \)
6: \( k = 1 \)
7: \( \text{while } |\Gamma_k| > 0 \text{ do} \)
8: \( x_k = \arg \max_{x \in \Gamma_k} \{\text{dist}(x, P_{\text{train}}(x))\} \)
9: \( \{R_k, M_k\} = \text{Classify}(C, \{C_{k-1} \cup \{x_k\}\}) \)
10: \( \text{if } R_k > R_{k-1} \text{ then} \)
11: \( \Gamma_k = \Gamma_{k-1} \setminus \{x_k\} \)
12: \( k = k + 1 \)
13: \( \text{end if} \)
14: \( \text{end while} \)
15: \( C_k = C_0 \cup C_k \)
16: \( \Gamma_0 = \text{load}(\text{stream}, \text{wsize}) \)
17: \( \text{until } |\Gamma_0| = 0 \)

Numerical Results

Large-noisy-crossed-norm Data set

200,000 points with 20 features equal divided in 2 classes

100,000 train points 100,000 test points

Each class is drawn from a multivariate normal distribution
Numerical Results

<table>
<thead>
<tr>
<th>Parameter</th>
<th>B</th>
<th>ED</th>
<th>FP</th>
<th>EM</th>
<th>EM+E</th>
<th>SI-ReGEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error (%)</td>
<td>3.2</td>
<td>9.1</td>
<td>3.2</td>
<td>4.5</td>
<td>6.7</td>
<td><strong>2.88</strong></td>
</tr>
<tr>
<td>subset train</td>
<td>8321</td>
<td>4172</td>
<td>8452</td>
<td>1455</td>
<td>5308</td>
<td>413</td>
</tr>
<tr>
<td>window size</td>
<td>100000</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
</tbody>
</table>

SI-ReGEC has the lowest error and uses the smallest incremental set

B: Batch SVM  
ED: Error-driven KNN  
FP: Fixed partition SVM  
EM: Exceeding-margin SVM  
EM+E: Fixed margin + errors

Larger windows lead to smaller train subset and execution time increases with window growth

<table>
<thead>
<tr>
<th>Wsize</th>
<th>Acc.</th>
<th>Growth Rate</th>
<th>Avg. Time</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>96.13%</td>
<td>15.75</td>
<td>4.25s</td>
<td>8.5e-4s</td>
</tr>
<tr>
<td>1000</td>
<td>96.92%</td>
<td>4.31</td>
<td>15.1s</td>
<td>1.5e-3s</td>
</tr>
<tr>
<td>2000</td>
<td>96.56%</td>
<td>2.63</td>
<td>61.7s</td>
<td>3.1e-3s</td>
</tr>
<tr>
<td>4000</td>
<td>97.45%</td>
<td>1.81</td>
<td>232.49s</td>
<td>7.0e-3s</td>
</tr>
</tbody>
</table>

Data is processed at 123.5 Gb/day on standard hardware
Conclusions

✓ SI-ReGEC:
  1. achieves a classification accuracy well comparable with other methods
  2. produces smaller incremental training sets

Future Work:
Investigate how to dynamically adapt window size to stream rate and nonstationary data streams

Thanks!