Representative Subsets for Big Data Learning using $k$-NN Graphs

Raghvendra Mall, Vilen Jumutc
Rocco Langone, Johan A.K. Suykens

KU Leuven, ESAT-STADIUS

IEEE BigData 2014
Introduction

- **Choices for predictive models for Big Data Learning are limited.**
- One way is to build fast, scalable learning algorithms supporting parallelization.
- **The other direction is Sampling.**

Advantages & Disadvantages

- Build predictive model on a small representative subset of the data.
- Predictive model has out-of-sample extension property (Alzate & Suykens, IEEE PAMI, 2010), (Mall & Suykens, PAKDD, 2013 & IEEE TNNLS, in press).
- Sampling is used efficiently in large scale kernel based methods, prototype learning, manifold learning methods etc.
- **Probabilistic sampling techniques are non-deterministic leading to large variations in performance.**
Contributions

- Propose a framework that overcomes issues of probabilistic sampling.
- Select representative subsets that retain natural cluster structure in data.
- Compare and evaluate resulting subset with other subset selection techniques for big data learning tasks including classification & clustering.

Proposed Framework

- $k$-NN = $k$-Nearest Neighbor
- LSSVM = Least Squares Support Vector Machines (Suykens et al., 2002).
- SD-LSSVM = Subsampled-Dual LSSVM (Mall & Suykens, IEEE TNNLS, in press).
## Distributed $k$-NN Graph Generation Framework

### Initial Setup
- Convert Big Data into unweighted and undirected $k$-NN graph.
- Generating elements of a kernel matrix ($\Omega$) in parallel.
- Similarity between points (nodes) is used as weight of edges in $k$-NN graph.
- Only top $k$ neighbors of a node are selected in the graph.

### RBF kernel matrix

$$\Omega = \begin{pmatrix} K(x_1, x_1) & \ldots & K(x_1, x_N) \\ \vdots & \ddots & \vdots \\ K(x_N, x_1) & \ldots & K(x_N, x_N) \end{pmatrix},$$  

(1)

- Computation of the kernel matrix is based on the widely used radial-basis function (RBF), $K(x, y) = \exp(-\frac{||x-y||^2}{2\sigma^2})$.
- Pre-compute $\sigma$ using Silverman’s Rule of Thumb [6] as $\sigma = \hat{\sigma} N^{-1/(d+4)}$. 
Computational Tricks

- A batch cluster-based approach is used for creating kernel matrix.
- For each node a batch subset $\mathcal{D}_p \subset \mathcal{D}$ (big data) is loaded along with the corresponding matrix slice $X_p$, $p = 1, \ldots, N$.
- Pre-computed means $\mu(X_p)$ and variances $\text{Var}(X_p)$ are used to estimate the final $\hat{\sigma}$ by averaging.
- Map-Reduce setting (Agarwal et al, JMLR, 2014) explained in Figures 1-4 is used to calculate kernel matrix slices $\Omega^{(p)}$ related to $\mathcal{D}_p$.
- Sort in ascending order the columns of $\Omega^{(p)}$.
- Pick the indices corresponding of the top $k$ values.
- The implementation was done using Julia programming language (http://julialang.org/).
First loading in parallel for the $p$-th node a subset $\mathcal{D}_p \subset \mathcal{D}$ of the big data and the corresponding matrix slice $X_p$.

**Figure:** (1) Loading the data slices
Calculating slices $\Omega^{(p)}$ of the kernel matrix and sorting the columns to map the slices into indices.

**Figure:** (2) Mapping of the slices
The reduce operation is to pick the top $k$ indices and form the $k$-NN subgraphs.

**Figure:** (3) Reducing to $k$-NN subgraphs
Finally, merge all $k$-NN subgraphs into an aggregated $k$-NN graph.

**Figure:** (4) Creating the $k$-NN graph
FURS selects several nodes with high degree centrality from different dense regions in the $k$-NN graph capturing the inherent community structure.

1. Given: A $k$-NN graph $G(V,E)$.
2. Result: A subset comprising representative nodes whose cardinality is $M$.
3. Calculate weighted degree of each node by aggregating weights on its edges.
4. Calculate median degree ($m$) of the network & reject nodes with degree < $m$.
5. Sort nodes in descending order based on their degree.
6. Select and add node with highest degree to subset $S$.
7. Deactivate its neighborhood.
8. Remove selected node keeping the topography intact.
9. If all nodes deactivated.
   - Re-activate all nodes.
   - Sort nodes in descending order based on their degree.
10. Repeat Step 5 – 9 till we have selected $M$ nodes.
Figure: Generalization results using different subset selection techniques for the synthetic 4 Gaussians (4G) dataset using a subset of size 285 out of 28,500 points.
## Classification Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>( N )</th>
<th>SR</th>
<th>SRE</th>
<th>FURS(_k=10)</th>
<th>FURS(_k=100)</th>
<th>FURS(_k=500)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4G</td>
<td>28,500</td>
<td>0.395±0.235</td>
<td>0.898±0.375</td>
<td>0.252±0.023</td>
<td>0.331±0.063</td>
<td>0.414±0.070</td>
</tr>
<tr>
<td>5G</td>
<td>81,000</td>
<td>0.264±0.225</td>
<td>0.298±0.142</td>
<td>0.358±0.296</td>
<td>0.082±0.016</td>
<td>0.086±0.009</td>
</tr>
<tr>
<td>Magic</td>
<td>19,020</td>
<td>25.32±3.913</td>
<td>28.44±4.446</td>
<td>33.07±2.076</td>
<td>31.05±4.143</td>
<td>36.13±3.062</td>
</tr>
<tr>
<td>Shuttle</td>
<td>58,000</td>
<td>2.437±1.104</td>
<td>2.330±0.958</td>
<td>4.223±0.998</td>
<td>1.482±0.600</td>
<td>1.980±0.715</td>
</tr>
<tr>
<td>Skin</td>
<td>245,057</td>
<td>0.578±0.387</td>
<td>0.254±0.078</td>
<td>3.277±1.133</td>
<td>0.494±0.078</td>
<td>0.772±0.080</td>
</tr>
</tbody>
</table>

**Table:** Averaged generalization errors along with their standard deviations for SD-LSSVM (Mall & Suykens, IEEE TNNLS, in press) model for different subset selection techniques.
Synthetic Clustering Experiment

Figure: Comparison of clustering performance of KSC model for different subset selection techniques on 4G dataset using a subset of size 285 out of 28,500 points.
### Table: Comparison of FURS with other subset selection techniques w.r.t. quality metrics Davies-Bouldin (DB) and silhouette (SIL).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Points</th>
<th>M</th>
<th>Random</th>
<th>Rènyi entropy</th>
<th>FURS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>DB</td>
<td>SIL</td>
<td></td>
</tr>
<tr>
<td>4G</td>
<td>28,500</td>
<td>285</td>
<td>0.804 ± 0.167</td>
<td>0.667 ± 0.059</td>
<td>2.909 ± 2.820</td>
</tr>
<tr>
<td>5G</td>
<td>81,000</td>
<td>810</td>
<td>2.011 ± 0.998</td>
<td>0.649 ± 0.049</td>
<td>0.885 ± 0.276</td>
</tr>
<tr>
<td>Batch</td>
<td>13,910</td>
<td>139</td>
<td>4.287 ± 0.764</td>
<td>0.503 ± 0.066</td>
<td>3.969 ± 0.783</td>
</tr>
<tr>
<td>House</td>
<td>34,112</td>
<td>342</td>
<td>0.612 ± 0.154</td>
<td>0.679 ± 0.073</td>
<td>0.507 ± 0.028</td>
</tr>
<tr>
<td>Mopsi Finland</td>
<td>13,467</td>
<td>135</td>
<td>0.897 ± 0.935</td>
<td>0.824 ± 0.085</td>
<td><strong>0.526</strong> ± <strong>0.223</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KDDCupBio</td>
<td>145,751</td>
<td>1458</td>
<td><strong>2.932</strong> ± <strong>1.205</strong></td>
<td></td>
<td>4.233 ± 1.82</td>
</tr>
<tr>
<td>Power</td>
<td>2,049,280</td>
<td>2,050</td>
<td>2.558 ± 1.374</td>
<td></td>
<td>2.0619 ± 0.760</td>
</tr>
</tbody>
</table>
Conclusion

- Proposed a method to obtain representative subset of the big data.
- Converted the big data into a $k$-NN graph using a distributed framework.
- Selected representative subset using FURS algorithm.
- Illustrated effectiveness of the selected representative subset for big data classification and clustering tasks.
A. Agarwal, O. Chapelle, M. Dudk, and J. Langford.
A reliable effective terascale linear learning system.

C. Alzate and J.A.K. Suykens.
Multiway spectral clustering with out-of-sample extensions through weighted kernel pca.

R. Mall, R. Langone, and J.A.K. Suykens.
FURS: Fast and unique representative subset selection retaining large scale community structure.

R. Mall and J.A.K. Suykens.
Very sparse LS-SVM Reductions for large scale data.
IEEE Transactions of Neural Networks and Learning Systems, in press.

R. Mall and Johan A.K. Suykens.
Sparse variations to fixed-size least squares support vector machines for large scale data.

B.W. Silverman.
Density Estimation for Statistics and Data Analysis.

Least Squares Support Vector Machines.

T. White.
Hadoop: The Definitive Guide.