Schedule

Morning Session:
08:30 - 09:15 Welcome and Presentation of Results (Organizers)
09:15 - 10:00 Ronan Collobert - Large Scale Learning Which Is Actually Useful
10:00 - 10:15 Coffee Break
10:15 - 10:35 Jochen Garcke - AV SVM
10:35 - 11:05 Hsiang Fy Yu - liblinear
11:05 - 11:35 Yossi Richter - Parallel Decision Tree

Afternoon Session
14:00 - 14:30 Han-Shen Huang et.al - Triple Jump Linear SVM
14:30 - 15:00 Marc Boulle - Averaging of Selective Naive Bayes Classifiers
15:00 - 15:45 Chih-Jen Lin - Training Support Vector Machines: Status and Challenges
15:45 - 16:00 Coffee Break
16:00 - 16:30 Olivier Chapelle, Sathiya Keerthi - SDM L1/2 and Newton SVM (presented by Chih-Jen Lin)
16:30 - 17:00 Antoine Bordes - SGD-QN, LaRank
17:00 - 18:00 Discussion and Summary
Large Scale Learning - Challenge
(Learning with Millions of Examples and Dimensions)

Sören Sonnenburg, Vojtech Franc,
Elad Yom-Tov and Michele Sebag

Fraunhofer FIRST.IDA, Berlin

July 9, 2008
Outline

1. Motivation
2. Evaluation Criteria
3. Statistics
4. Preliminary Results
Large Scale Problems

What makes a Problem Large Scale?

- Large number of data points
- Extremely high dimensionality
- High effort algorithms $O(N^3)$
- Large memory requirements

⇒ Anything that reaches current computers limits: computational, memory, transfer costs

One may define a large scale problem to be a problem which to solve reaches current computers limits be it computational, memory or transfer costs wise. For machine learning this translates to high effort algorithms (e.g. $O(N^3)$), large number of data points or high dimensionality.
Our Motivation

**Current SVM solvers**

- Joachims 2005, $\text{SVM}^{\text{perf}}$ is *much* faster than $\text{SVM}^{\text{light}}$
- Own experiments: $\text{SVM}^{\text{light}}$ is *much* faster than $\text{SVM}^{\text{perf}}$
- Shalev-Shwartz et.al. 2007, Pegasos is much faster than $\text{SVM}^{\text{light,perf}}$
- Own experiments: Pegasos is much slower than $\text{SVM}^{\text{light,perf}}$
- Teo et.al. 2007, $\text{SVM}^{\text{perf}}$ is a special case of BMRM
- Own experiments: BMRM is much faster than $\text{SVM}^{\text{perf}}$
- new $\text{SVM}^{\text{perf}2.1}$ similar in speed to BMRM
- Bottou 2007, SGD done right outperforms competitors

There is no reliable way to tell which method is faster!
We need a fair comparison!

Large Scale Learning Challenge

Main Goal
- Evaluation under exact same fair conditions to answer: **Which learning method is most accurate given limited resources?**
- Evaluation based on training time, test error (or objective value, etc. specific to method)

Additional Goals
- Which method gives the overall best classification performance?
- Which classifier is the most training time efficient while achieving a good test error?
- Approximation vs. Exact Algorithms?
- What should one tune? Data representation? Feature selection? Core algorithm?
Two tracks:
- Wild Competition / Parallel
- Method Specific:
  - Linear SVM
  - RBF SVM

Setup:
- Method are trained on diverse labeled datasets (size $10^2,3,4,5,6,7,\ldots$); unlabeled validation set and test set

Evaluation
- Record training time, validation and test output for 10 intermediate points
- Timing “calibrated” using program measuring floating point, integer, memory speed; At the end re-evaluation on a single machine.
- Live feedback for validation set
- Feedback for test set after end of competition
- Competitors are required to submit a detailed explanation of the used methods.
Setup and Evaluation Criteria

Setup Evaluation Criteria

- Time vs. Test Error or Objective Value
- Dataset Size vs. Time ($O(n^s)$)
- Dataset Size vs. Test Error

We will compute **Performance Figures** and **Scalar Measures**.

⇒ **Compute Scalar Evaluation Scores for Final Evaluation**
Scalar Measures: Test Error, Time for error within 5%, Area under curve
Dataset Size vs. Time

Scalar Measure - Slope in Log-Log Plot $O(n^s)$
Scalar Measures: Dataset size for error within 5%, Area under curve
Adjusted Goals and Evaluation for SVMs

Goals for SVMs

- What is the relation between objective value vs. test error?
- What is the relation between stopping conditions and test error?
- Which algorithm is good on what kind of data set ((un)balanced, high or low dimensional, range of $C$, etc.)
Adjusted Evaluation for SVMs

Setup and Evaluation Criteria for SVMs

- Linear SVM with sparse data representation
- RBF Kernel SVM with dense data representation
- Run SVM for given fixed values of C and kernel width
- Record objective value while training
- Additional stopping criterion: target objective value
- Figures: Time vs. C, Time vs. Objective, Time vs. Test Error and Objective
- Scalars: Total time for model selection (all Cs and kernel widths), Time to reach target objective
Time Line

- February - Start of Competition
- June - End of Open Competition
- Ongoing - We perform re-evaluation on a single CPU Linux machine with 32G of memory
- 9 July 2008 - Evaluation in an ICML’2008 workshop
- 15 October 2008 - Planned Large Scale JMLR Track (under review) for best performing methods
## Statistics

- Number of subscribed people: 49
- Number of submissions: 44 (in total 459 entries)
- Most submissions are linear methods, variants of L1/L2-SVM solvers
  - SGD-Qn (**Antoine Bordes**)
  - Newton (**Olivier Chapelle**)
  - Dual coord. descent (**Hsiang Fy Yu**)
  - Tripple Jump SVM (**Han-Shen Huang and Chun-Nan Hsu**)
  - Interior point (Kistian Woodsend)
- Some non-linear
  - Parallel decision tree (**Yossi Richter**)
  - Averaging of selective naive Bayes (**Marc Boulle**)
  - Averaging of RBF-kernel SVM (**Jochen Garcke**)
Statistics

- Most active: Christian Woodsend (Interior Point SVM: linear, parallel, wild track) and Antoine Bordes (SGD-QN, Larank), Olivier Chapelle and Sathiya Keerthi (SDM-L1/L2, Newton, . . .)
- Most interest in Wild track (263 entries)
- Most interest in Alpha dataset (72 submissions)
- Least interest Parallel and SVM tracks
- Programming Languages: C/C++, Java, Matlab
### Overview Datasets

#### Datasets

<table>
<thead>
<tr>
<th></th>
<th>Number of examples</th>
<th></th>
<th>Dims</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>training</td>
<td>testing</td>
<td>valid.</td>
<td></td>
</tr>
<tr>
<td>alpha</td>
<td>5000000</td>
<td>300000</td>
<td>100000</td>
<td>500</td>
</tr>
<tr>
<td>beta</td>
<td>5000000</td>
<td>300000</td>
<td>100000</td>
<td>500</td>
</tr>
<tr>
<td>gamma</td>
<td>5000000</td>
<td>300000</td>
<td>100000</td>
<td>500</td>
</tr>
<tr>
<td>delta</td>
<td>5000000</td>
<td>300000</td>
<td>100000</td>
<td>500</td>
</tr>
<tr>
<td>epsilon</td>
<td>5000000</td>
<td>300000</td>
<td>100000</td>
<td>2000</td>
</tr>
<tr>
<td>zeta</td>
<td>5000000</td>
<td>300000</td>
<td>100000</td>
<td>2000</td>
</tr>
<tr>
<td>fd</td>
<td>5469800</td>
<td>1623120</td>
<td>532400</td>
<td>900</td>
</tr>
<tr>
<td>ocr</td>
<td>3500000</td>
<td>1500000</td>
<td>670000</td>
<td>1156</td>
</tr>
<tr>
<td>dna</td>
<td>500000000</td>
<td>3627840</td>
<td>1000000</td>
<td>800</td>
</tr>
<tr>
<td>webspam</td>
<td>3500000</td>
<td>98387</td>
<td>500000</td>
<td>16777216</td>
</tr>
<tr>
<td></td>
<td>optimal model</td>
<td>classifier</td>
<td>aoPRC</td>
<td>aoPRC</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>------------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>alpha</td>
<td>quadratic</td>
<td>0.009774</td>
<td>0.0854</td>
<td></td>
</tr>
<tr>
<td>beta</td>
<td>quadratic</td>
<td>0.304563</td>
<td>0.4577</td>
<td></td>
</tr>
<tr>
<td>gamma</td>
<td>semi-quad</td>
<td>0.009418</td>
<td>0.0116</td>
<td></td>
</tr>
<tr>
<td>delta</td>
<td>semi-quad</td>
<td>0.071005</td>
<td>0.0801</td>
<td></td>
</tr>
<tr>
<td>epsilon</td>
<td>linear</td>
<td>0.033645</td>
<td>0.0341</td>
<td></td>
</tr>
<tr>
<td>zeta</td>
<td>linear</td>
<td>0.011361</td>
<td>0.0116</td>
<td></td>
</tr>
<tr>
<td>dna</td>
<td>WD-SVM</td>
<td>&lt; 0.4558</td>
<td></td>
<td>0.8030</td>
</tr>
<tr>
<td>ocr</td>
<td>n.n.</td>
<td>n.n.</td>
<td></td>
<td>0.1579</td>
</tr>
<tr>
<td>fd</td>
<td>n.n.</td>
<td>n.n.</td>
<td></td>
<td>0.2314</td>
</tr>
<tr>
<td>webspam</td>
<td>n.n.</td>
<td>n.n.</td>
<td></td>
<td>0.0004</td>
</tr>
</tbody>
</table>
## Preliminary Results - Wild Track

<table>
<thead>
<tr>
<th>Rank</th>
<th>Score</th>
<th>Submitter</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.40</td>
<td>antoine - Antoine Bordes</td>
<td>SgdQn</td>
</tr>
<tr>
<td>2</td>
<td>2.50</td>
<td>chap - Olivier Chapelle</td>
<td>Newton SVM</td>
</tr>
<tr>
<td>3</td>
<td>2.70</td>
<td>yahoo - Olivier Keerthi</td>
<td>SDM SVM L1</td>
</tr>
<tr>
<td>3</td>
<td>2.70</td>
<td>yahoo - Olivier Keerthi</td>
<td>SDM SVM L2</td>
</tr>
<tr>
<td>5</td>
<td>5.50</td>
<td>ker2 - Porter Chang CTJ</td>
<td>L SVM01</td>
</tr>
<tr>
<td>6</td>
<td>5.90</td>
<td>antoine - Antoine Bordes</td>
<td>LaRankConverged</td>
</tr>
<tr>
<td>7</td>
<td>6.20</td>
<td>rofu - Rofu yu</td>
<td>liblinear</td>
</tr>
<tr>
<td>8</td>
<td>7.00</td>
<td>kristian - Kristian Woodsend</td>
<td>IPM SVM 2</td>
</tr>
</tbody>
</table>

(still running: LR - Gavin Cawley, Averaging of Selective Naive Bayes Classifiers - Marc Boulle, AV SVM single - Jochen Garcke)
Preliminary Results - Alpha: Time vs. Error
Preliminary Results - Alpha: Size vs. Error
Preliminary Results - Beta: time vs. error

- Newton SVM
- AV SVM single
- AV SVM
- Averaging of Selective Naïve
- Averaging of Selective Naïve
- SDM SVM L1
- SDM SVM L2
- WRRR
- SgdOn
- LR
- Interior Point SVM
- CTJ LSVMO1
- Coordinate descent dual l1 li...
- libLinear
- ORRR
- IPM SVM 2
- LaRankConverged
- ocas
- CTJ LSVMO2
- ORRR Ensemble
Preliminary Results - Beta: Size vs. Error
Preliminary Results - Gamma: Time vs. Error
Preliminary Results - Gamma: Size vs. Error
Preliminary Results - Delta: Time vs. Error

The graph shows the aoPRC (Area Under the Precision-Recall Curve) for different algorithms as the dataset size increases. The x-axis represents the dataset size, while the y-axis shows the aoPRC values. Various algorithms are represented by different colors and markers.

- SgdOn
- LR
- Averaging of Selective Naive
- Averaging of Selective Naive
- Interior Point SVM
- AV SVM single
- SDM SVM L2
- IPM SVM L2
- AV SVM
- Newton SVM
- ORRR
- CTJ LSVM02
- CTJ LSVM01
- SDM SVM L1
- ORRR Ensemble
- Coordinate descent dual l1 l1
- liblinear
- ocsap
- LaRankConverged

The graph indicates how each algorithm performs relative to others as the size of the dataset increases.
Preliminary Results - Epsilon: Size vs. Error
Preliminary Results - Zeta: Size vs. Error
Preliminary Results - DNA: Time vs. Error

The graph shows the relationship between CPU time (in seconds) and area under the precision-recall curve (aoPRC) for different learning algorithms. The algorithms compared include SSDM SVM L2, SGD, SDM SVM L1, Newton SVM, liblinear, and LaRank. The graph indicates that as CPU time increases, the aoPRC generally decreases, except for the LaRank algorithm, which shows a different trend.
Preliminary Results - Webspam: Time vs. Error
Preliminary Results - Webspam: Size vs. Error
Preliminary Results - FD: Time vs. Error
Preliminary Results - OCR: Time vs. Error

![Graph showing the relationship between CPU time (s) and aoPRC for various algorithms.](image)
Preliminary Results - OCR: Size vs. Error

The graph shows the aoPRC (Average Precision at Recall) against dataset size for different methods. The x-axis represents the dataset size, while the y-axis shows the aoPRC. The methods include SDM SVM L1, SgdQn, SDM SVM L2, Newton SVM, LaRankConverged, and lliblinear. The graph indicates a trend where the aoPRC decreases as the dataset size increases for all methods.
<table>
<thead>
<tr>
<th>Rank</th>
<th>Score</th>
<th>Submitter</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.80</td>
<td>rofu - Rofu yu</td>
<td>liblinear</td>
</tr>
<tr>
<td>2</td>
<td>2.30</td>
<td>rofu - Rofu yu</td>
<td>CDD l1 linear svm</td>
</tr>
<tr>
<td>3</td>
<td>2.90</td>
<td>antoine - Antoine Bordes</td>
<td>LaRankConverged</td>
</tr>
<tr>
<td>4</td>
<td>4.00</td>
<td>yahoo - Olivier Keerthi</td>
<td>SDM SVM L1</td>
</tr>
<tr>
<td>5</td>
<td>5.30</td>
<td>ker2 - Porter Chang</td>
<td>CTJ LSVM02</td>
</tr>
<tr>
<td>6</td>
<td>5.40</td>
<td>kristian - Kristian Woodsend</td>
<td>IPM SVM 2</td>
</tr>
<tr>
<td>7</td>
<td>6.20</td>
<td>kristian - Kristian Woodsend</td>
<td>Interior Point SVM</td>
</tr>
<tr>
<td>8</td>
<td>6.30</td>
<td>fravoj - Vojtech Franc</td>
<td>svm-ocas</td>
</tr>
<tr>
<td>9</td>
<td>7.50</td>
<td>aiiaSinica - Han-Shen Huang</td>
<td>lsvm04</td>
</tr>
<tr>
<td>10</td>
<td>7.80</td>
<td>aiiaSinica - Han-Shen Huang</td>
<td>linearSVM03</td>
</tr>
</tbody>
</table>

(re-evaluation pending)
Which learning method is most accurate given limited resources?

- The method which uses the right tradeoff between model complexity and accuracy.
- Low model error overweights the optimization accuracy. ⇒ one should tune the model first then derive efficient algorithms.
- No big difference in speed nor scaling of linear SVM optimizers.
- Use as much data as possible to improve performance (for complex/correct model).
- Participants used given svmlight data representation, no domain specific knowledge ⇒ a lot of room for improvement.
Wild track

- SGD-QN - Antoine Bordes
- Newton SVM - Olivier Chapelle
- SDM SVM L1/L2 - Olivier Chapelle, Sathiya Keerthi

SVM track

- Liblinear - Hsiang Fy Yu

Additional best student awards

- Interior point SVM - Kristian Woodsend
- Triple Jump SVM - Han-Shen Huang and Chun-Nan Hsu
Planned Large Scale JMLR Track (tentative - under review)

- Submission: 15 October 2008
- Decision: 15 December 2008
- Final versions: 15 February 2009

Open to challenge participants and others

- for fair comparison among methods new contributors need to follow challenge protocol, i.e. to use the challenge data sets and performance figures in their evaluation.

Another Challenge? Volunteers?