Large Scale Learning - Competition

(Learning with Millions of Examples and Dimensions)

Sören Sonnenburg and Vojtech Franc

Fraunhofer FIRST.IDA, Berlin

December 21, 2007

FIRST

Fraunhofer Institut
Rechnerarchitektur
und Softwaretechnik

Large Scale Problems

What makes a Problem Large Scale?

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

$\Rightarrow$ **Anything that reaches current computers limits: computational, memory, transfer costs**

Applications

- Bioinformatics (Splice Sites, Gene Boundaries,. . .)
- IT-Security (Network traffic)
- Text-Classification (Spam vs. Non-Spam)

Our Motivation

Current SVM solvers

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

There is no reliable way to tell which method is faster!

Reasons

Evaluation was done using different criteria!

- Different Parameters $C, \varepsilon, \lambda, \ldots$
- Meaning of parameters different
- Evaluation based on test error, objective value, ...
- Programming Errors, Inefficient Code
- Other accidental mistakes.

We need a fair comparison!

Proposal for a 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?

Competition

- **Two tracks:**
  - Method Specific: SVMs, **Others?** ⇒ **Help us organizing!**
  - Wild Competition
- **Setup:**
  - Method are trained on diverse labeled datasets (size $10^{2,3,4,5,6,7,\cdots}$); unlabeled validation set and test set
  - 40M examples - human splice dataset (strings of length 398)
  - 100-500K websites web-spam data (16M dims)
  - 100K examples - image classification dense 10K dimensions
  - **More?** ⇒ **Please share the dataset!**
- **Evaluation**
  - Record training time, validation and test output for ≥ 10 intermediate points
  - Timing "calibrated" using program measuring floating point, integer, memory speed
  - Live feedback for validation set
  - Feedback for test set after end of competition

Time Line

- January/February - Announce Competition
- Beginning of June - End of Open Competition
- We perform re-evaluation on a single CPU Linux machine with 32G of memory
- 9 July 2008 - Evaluation in an ICML'2008 workshop

Proceedings in LNCS Springer for best performing methods

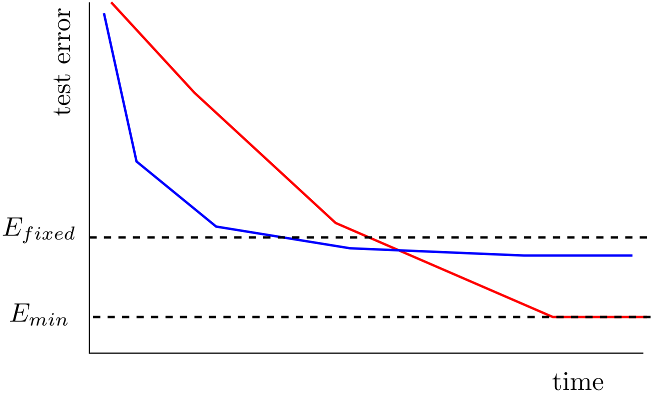
Setup and Evaluation Criteria

Setup Evaluation Criteria

- Training time vs. Test Error or Objective Value
- Dataset Size vs. Training time ($\mathcal{O}(n^s)$)
- Dataset Size vs. Test Error or Objective Value

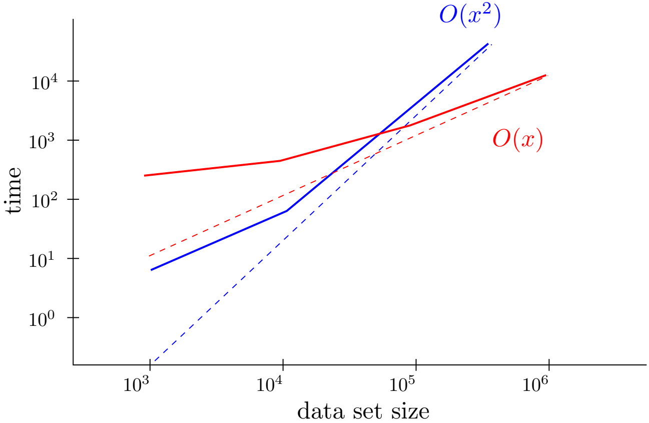
$\Rightarrow$ **Compute Scalar Evaluation Scores for Final Evaluation**

Evaluation: Training Time vs. Test Error



Scalar Measures: Test Error and Time for fixed Test Error

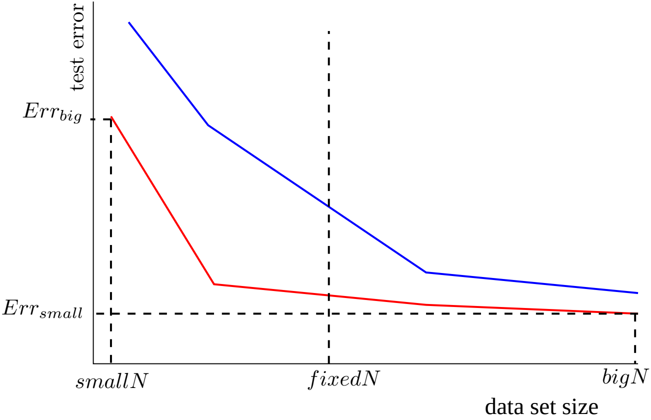
Dataset Size vs. Training Time



Scalar Measure - Slope in Log-Log Plot $\mathcal{O}(n^s)$

Fraunhofer Institut Rechnerarchitektur und Softwaretechnik

Dataset Size vs. Test Error



Scalar Measure - Test Error for fixed number of Examples and "Gain":= $\frac{Err_{small}}{Err_{big}} \cdot \frac{smallN}{bigN}$

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.)

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 to train for all Cs, Time to reach target objective

Items that need Discussion

- Evaluation Criteria Scores
- Which other datasets?
- Which other methods specific tracks?
- Data distribution? P2P torrent network?
- Should we include other constraints (low memory, time deadlines)?
- Anyone willing to manage other tracks (parallel, boosting, neural nets,. . .)?
- Any other comments, suggestions?