An Empirical Evaluation of Supervised Learning in High Dimensions

Rich Caruana, Nikos Karampatziakis and Ainur Yessenalina

Department of Computer Science Cornell University

Cornell University

1 Previous Empirical Comparisons

- STATLOG (1995)
  - Did not have boosting, SVMs and other recent methods.
- Caruana and Niculescu-Mizil (2006)
  - Included newer methods, very thorough.
  - On average, boosted trees were the best, followed by random forests.
- Neither study considered problems of high dimensionality.
- *Are those conclusions valid in high dimensions?*
- *Teaser: Yes, up to some dimensionality. But in higher dimensions things are different in a semi-obvious way. . .*

2 Methodology

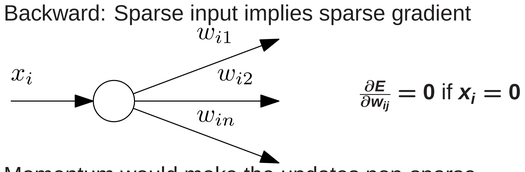
- 11 datasets, ranging from 700 to 700K dimensions, mainly from biology, text and link prediction domains.
- 10 state of the art learning algorithms
- 100's of parameter settings
- 3 metrics: accuracy, squared loss, area under the ROC
- Why not use more than these three?

3 Small difficulties

- Coping with squared loss → calibration (Platt & Isotonic).
- Coping with different baselines → standardization
- Interpretation: a standardized score of 1.02 indicates 2% improvement over typical method.

4 Implementation Tricks

- Most high dimensional data is sparse.
- Specialized implementations for handling sparse data.
- Neural Nets
  - Forward: Matrix times sparse vector multiplication
  - Backward: Sparse input implies sparse gradient



$\frac{\partial E}{\partial w_{ij}} = 0$ if $x_i = 0$

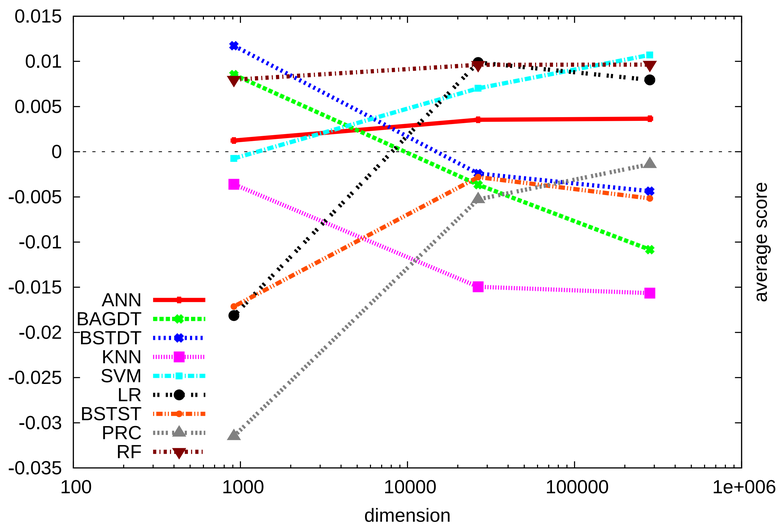
  - Momentum would make the updates non-sparse
- Decision Trees: Indexing by feature
- Kernel SVMs: Specialized large scale SVM solver LaSVM
- Still, experiments took 5-6 weeks in 40 cpus.

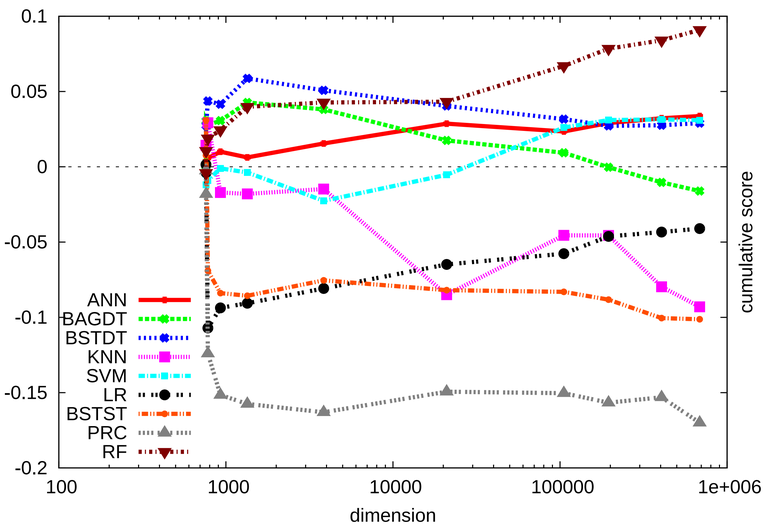
5 Average Over All Three Metrics

DIM	761	761	780	927	1344	3448	21K	105K	195K	405K	685K	—
AVG	Stu	Cal	Dig	Tis	Cry	Kdd	R-S	Cite	Dse	Spam	Imdb	Mean
RF	0.994	1.021	1.009	1.007	1.019	1.005	1.001	1.032	**1.013**	**1.006**	**1.007**	1.010
ANN	1.006	0.997	1.005	1.005	0.996	1.016	1.015	0.993	1.006	1.004	1.002	1.004
BST	0.998	1.040	**1.018**	0.998	**1.021**	0.987	0.988	0.988	0.995	1.000	1.001	1.003
SVM	0.992	0.990	1.003	1.010	0.997	0.968	**1.020**	1.041	1.006	1.000	1.000	1.002
BGT	1.001	**1.043**	0.997	1.003	1.015	0.992	0.977	0.989	0.989	0.989	0.994	0.999
LR	1.002	0.993	0.886	**1.016**	1.003	1.017	1.018	1.009	1.013	1.003	1.002	0.997
KNN	**1.022**	1.000	1.017	0.946	0.999	1.006	0.920	**1.052**	1.000	0.962	0.986	0.992
BSS	1.012	1.033	0.890	0.982	0.998	**1.017**	0.993	0.999	0.994	0.986	0.999	0.991
PRC	0.996	0.978	0.883	0.967	0.993	0.991	1.016	0.999	0.993	1.004	0.983	0.982
NB	0.961	0.927	0.799	0.922	0.958	0.995	1.000	1.000	0.987	0.943	0.950	0.949

- Random Forests on really high dimensions.
- Random Forests vs. Boosted Trees.
- Consistency of ANNs.
- Diversity of best models.
- Not apparent from this table: calibration with Isotonic Regression is almost always better than Platt's method or no calibration.

6 Trends





7 Conclusions

- Our results confirm the findings of previous studies in low dimensions.
- But as dimensionality increases, boosted trees fall behind random forests.
- Non-linear methods can do well in high dimensions.
  - But they need appropriate regularization. (ANNs, Kernel SVMs, Random Forests)
- Calibration never hurts and almost always helps even for methods such as logistic regression and neural nets.

8 Acknowledgments

- This work began as a group project in a graduate machine learning course at Cornell.
- We thank everyone who participated in the course and especially the following students: Sergei Fotin, Michael Friedman, Myle Ott, Raghu Ramanujan, Alec Berntson, Eric Breck, and Art Munson.

Random forest and other tree software:
http://www.cs.cornell.edu/∼nk/fest

Email: {caruana,nk,ainur,}@cs.cornell.edu

WWW: http://www.cs.cornell.edu