An Empirical Evaluation of Supervised Learning in High Dimensions

Rich Caruana    Nikos Karampatziakis    Ainur Yessenalina

Department of Computer Science, Cornell University

July 8, 2008
Previous Empirical Comparisons

- STATLOG (1995)
  - Did not have boosting, SVMs and other recent methods.
- Caruana and Niculescu-Mizil (2006)
  - Included newer methods.
  - Evaluated on 11 datasets and 8 metrics.
  - On average, boosted trees were the best.
Previous Empirical Comparisons

- **STATLOG (1995)**
  - Did not have boosting, SVMs and other recent methods.
- **Caruana and Niculescu-Mizil (2006)**
  - Included newer methods.
  - Evaluated on 11 datasets and 8 metrics.
  - On average, boosted trees were the best.

- Neither study considered problems of high dimensionality.
Previous Empirical Comparisons

- **STATLOG (1995)**
  - Did not have boosting, SVMs and other recent methods.
- **Caruana and Niculescu-Mizil (2006)**
  - Included newer methods.
  - Evaluated on 11 datasets and 8 metrics.
  - On average, boosted trees were the best.

Neither study considered problems of high dimensionality.

Are the conclusions of previous studies valid in high dimensions?
Previous Empirical Comparisons

- STATLOG (1995)
  - Did not have boosting, SVMs and other recent methods.

- Caruana and Niculescu-Mizil (2006)
  - Included newer methods.
  - Evaluated on 11 datasets and 8 metrics.
  - On average, boosted trees were the best.

Neither study considered problems of high dimensionality.

Are the conclusions of previous studies valid in high dimensions?

Teaser: Previous conclusions are valid up to some dimensionality. But in higher dimensions things are different in a semi-obvious way...
Motivation

- High dimensional learning tasks increasingly more common
  - Biological data
  - Text: bag-of-words data
  - Images
  - Link analysis

- Recent advances in effective techniques to handle them
  - SVMs
  - $L_1$ regularization
Outline

- Methodology
- Challenges
- Results
- Conclusions
## Datasets

<table>
<thead>
<tr>
<th>Problem</th>
<th>( \approx ) Attr</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sturn</td>
<td>760</td>
<td>Ornithology dataset</td>
</tr>
<tr>
<td>Calam</td>
<td>760</td>
<td>Ornithology dataset</td>
</tr>
<tr>
<td>Digits</td>
<td>780</td>
<td>Image recognition, MNIST, (&lt; 5 ) versus ( \geq 5 )</td>
</tr>
<tr>
<td>Tis</td>
<td>930</td>
<td>Protein translation problem</td>
</tr>
<tr>
<td>Cryst</td>
<td>1300</td>
<td>Protein crystallography diffraction</td>
</tr>
<tr>
<td>KDD98</td>
<td>4K</td>
<td>Predict if person will donate money</td>
</tr>
<tr>
<td>R-S</td>
<td>21K</td>
<td>Text classification</td>
</tr>
<tr>
<td>Dse</td>
<td>200K</td>
<td>Sentiment analysis</td>
</tr>
<tr>
<td>Spam</td>
<td>400K</td>
<td>Text classification</td>
</tr>
<tr>
<td>Cite</td>
<td>100K</td>
<td>Link prediction</td>
</tr>
<tr>
<td>Imdb</td>
<td>685K</td>
<td>Link prediction</td>
</tr>
</tbody>
</table>

- Use original train/validation/test if available.
- Otherwise split 40%/10%/50% in train/validation/test
Learning Algorithms

- Artificial Neural Nets (ANN*)
  Fully connected two layer nets, trained with SGD, early stopping
Learning Algorithms

- Artificial Neural Nets (ANN*)
- Support Vector Machines (SVM)
  Linear and kernel poly degree 2 & 3, RBF (SVM\textsuperscript{light}, LaSVM)
Learning Algorithms

- Artificial Neural Nets (ANN*)
- Support Vector Machines (SVM)
- Logistic Regression (LR)

Regularized with either $L_1$ or $L_2$ norm (BBR package)
Learning Algorithms

- Artificial Neural Nets (ANN*)
- Support Vector Machines (SVM)
- Logistic Regression (LR)
- Naive Bayes (NB*)

Continuous variables are modeled as coming from a Gaussian
Learning Algorithms

- Artificial Neural Nets (ANN*)
- Support Vector Machines (SVM)
- Logistic Regression (LR)
- Naive Bayes (NB*)
- Distance Weighted $k$NN (KNN*)

Locally weighted averaging with tuned euclidean distance
Learning Algorithms

- Artificial Neural Nets (ANN*)
- Support Vector Machines (SVM)
- Logistic Regression (LR)
- Naive Bayes (NB*)
- Distance Weighted $k$NN (KNN*)
- Bagged Decision Trees (BAGDT*)

Average of 100 trees trained on bootstrap samples
Learning Algorithms

- Artificial Neural Nets (ANN*)
- Support Vector Machines (SVM)
- Logistic Regression (LR)
- Naive Bayes (NB*)
- Distance Weighted \(k\)NN (KNN*)
- Bagged Decision Trees (BAGDT*)
- Random Forests (RF*)

Like \(5 \times \text{BAGDT}\) but each split considers \(\alpha \sqrt{d}\) random features
Learning Algorithms

- Artificial Neural Nets (ANN*)
- Support Vector Machines (SVM)
- Logistic Regression (LR)
- Naive Bayes (NB*)
- Distance Weighted $k$-NN (KNN*)
- Bagged Decision Trees (BAGDT*)
- Random Forests (RF*)
- Boosted Decision Trees (BSTDT*)

Adaboost with up to 1024 trees
Learning Algorithms

- Artificial Neural Nets (ANN*)
- Support Vector Machines (SVM)
- Logistic Regression (LR)
- Naive Bayes (NB*)
- Distance Weighted $k$NN (KNN*)
- Bagged Decision Trees (BAGDT*)
- Random Forests (RF*)
- Boosted Decision Trees (BSTDT*)
- Boosted Stumps (BSTST*)
- Adaboost with up to $2^{14}$ stumps
Learning Algorithms

- Artificial Neural Nets (ANN*)
- Support Vector Machines (SVM)
- Logistic Regression (LR)
- Naive Bayes (NB*)
- Distance Weighted $k$NN (KNN*)
- Bagged Decision Trees (BAGDT*)
- Random Forests (RF*)
- Boosted Decision Trees (BSTDT*)
- Boosted Stumps (BSTST*)
- Voted Perceptrons (PRC*)

Average of many linear perceptrons
Performance Metrics

- We used:
  - Area under ROC (AUC) — Ordering Metric
  - Accuracy (ACC) — Threshold Metric
  - Root mean squared error (RMS) — Probability Metric
- Why not use more than these three?
Performance Metrics

- We used:
  - Area under ROC (AUC) — Ordering Metric
  - Accuracy (ACC) — Threshold Metric
  - Root mean squared error (RMS) — Probability Metric

- Why not use more than these three?
- Performance metrics are correlated.
Output of ANN, Logistic Regression etc. can be interpreted as
\[ p(y = 1|x). \]

SVMs, Boosting etc. do not predict good probabilities.

These methods will do very poorly on squared loss.

Calibrate predictions of all models to make comparison fair.

- Platt’s method: Fits a sigmoid
  \[
  p(y = 1|x) = \frac{1}{1 + e^{\alpha h(x) + \beta}}
  \]

- Isotonic Regression: Fits a monotonic non-decreasing function. We learn a stepwise-constant function via the PAV algorithm. Optimal w.r.t. squared loss.

For more information see (Niculescu-Mizil & Caruana 2005).
Small difficulty

- For accuracy and AUC larger values indicate better performance. For squared error smaller is better.
Small difficulty

- For accuracy and AUC larger values indicate better performance. For squared error smaller is better.
- This is easily fixed if we use $1 - \text{squared error}$. 
Small difficulty

- For accuracy and AUC larger values indicate better performance. For squared error smaller is better.
- This is easily fixed if we use $1 - \text{squared error}$.
- For AUC baseline is 0.5, for accuracy and squared error baseline depends on problem.
- We would like to average across different problems and metrics.
**Standardization**

- *Typical* performance = median performance over all methods.
- One solution: Standardize performance scores by dividing by typical performance for that problem and metric.
- Values above (below) 1 indicate better (worse) than typical performance.
- Interpretation: a standardized score of 1.02 indicates 2% improvement over typical method.
Summary of Methodology

For every method and dataset

- Train models with different parameter settings
- Calibrate them using the validation set
  - For every performance metric
    - Pick model+calibration method with best performance on validation set
    - Report standardized performance on the test set
Scale of the Study

10 learning methods
×
100’s of parameter settings per method
=
1,000 expensive models trained per problem
×
11 Boolean classification test problems
=
11,000 models
×
3 performance metrics
=
33,000 model performance evaluations
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

\[
\frac{\partial E}{\partial w_{ij}} = 0 \text{ if } x_i = 0
\]

Momentum would make the updates non-sparse

Decision Trees: Indexing by feature

Kernel SVMs: Specialized large scale SVM solver LaSVM
Caveats

- Experiments took 5-6 weeks in 40 cpus.
- 5-fold cross-validation would be nice but too expensive.
  - Bootstrap analysis similar to the previous study.
- Binary classification only.
- Cannot try every flavor of every algorithm.
- 11 datasets so far.
### Average Over All Three Metrics

<table>
<thead>
<tr>
<th>DIM</th>
<th>761</th>
<th>761</th>
<th>780</th>
<th>927</th>
<th>1344</th>
<th>3448</th>
<th>21K</th>
<th>105K</th>
<th>195K</th>
<th>405K</th>
<th>685K</th>
<th>—</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>0.994</td>
<td>1.021</td>
<td>1.009</td>
<td>1.007</td>
<td>1.019</td>
<td>1.005</td>
<td>1.001</td>
<td>1.032</td>
<td>1.013</td>
<td>1.006</td>
<td>1.007</td>
<td>1.010</td>
</tr>
<tr>
<td>ANN</td>
<td>1.006</td>
<td>0.997</td>
<td>1.005</td>
<td>0.996</td>
<td>1.016</td>
<td>1.015</td>
<td>0.993</td>
<td>1.006</td>
<td>1.004</td>
<td>1.002</td>
<td>1.004</td>
<td>1.003</td>
</tr>
<tr>
<td>BST</td>
<td>0.998</td>
<td>1.040</td>
<td>0.998</td>
<td>1.021</td>
<td>0.987</td>
<td>0.988</td>
<td>0.988</td>
<td>0.995</td>
<td>1.000</td>
<td>1.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>0.992</td>
<td>0.990</td>
<td>1.003</td>
<td>1.010</td>
<td>0.997</td>
<td>0.968</td>
<td>1.020</td>
<td>1.041</td>
<td>1.006</td>
<td>1.000</td>
<td>1.000</td>
<td>1.002</td>
</tr>
<tr>
<td>BGT</td>
<td>1.001</td>
<td>1.043</td>
<td>0.997</td>
<td>1.003</td>
<td>1.015</td>
<td>0.992</td>
<td>0.977</td>
<td>0.989</td>
<td>0.989</td>
<td>0.994</td>
<td></td>
<td>0.999</td>
</tr>
<tr>
<td>LR</td>
<td>1.002</td>
<td>0.993</td>
<td>0.886</td>
<td>1.016</td>
<td>1.003</td>
<td>1.017</td>
<td>1.018</td>
<td>1.009</td>
<td>1.013</td>
<td>1.003</td>
<td>1.002</td>
<td>0.997</td>
</tr>
<tr>
<td>KNN</td>
<td>1.022</td>
<td>1.000</td>
<td>1.017</td>
<td>0.946</td>
<td>0.999</td>
<td>1.006</td>
<td>0.920</td>
<td>1.052</td>
<td>1.000</td>
<td>0.962</td>
<td>0.986</td>
<td>0.992</td>
</tr>
<tr>
<td>BSS</td>
<td>1.012</td>
<td>1.033</td>
<td>0.890</td>
<td>0.982</td>
<td>0.998</td>
<td>1.017</td>
<td>0.993</td>
<td>0.999</td>
<td>0.994</td>
<td>0.986</td>
<td>0.999</td>
<td>0.991</td>
</tr>
<tr>
<td>PRC</td>
<td>0.996</td>
<td>0.978</td>
<td>0.883</td>
<td>0.967</td>
<td>0.993</td>
<td>0.991</td>
<td>1.016</td>
<td>0.999</td>
<td>0.993</td>
<td>1.004</td>
<td>0.983</td>
<td>0.982</td>
</tr>
<tr>
<td>NB</td>
<td>0.961</td>
<td>0.927</td>
<td>0.799</td>
<td>0.922</td>
<td>0.958</td>
<td>0.995</td>
<td>1.000</td>
<td>1.000</td>
<td>0.987</td>
<td>0.943</td>
<td>0.950</td>
<td>0.949</td>
</tr>
</tbody>
</table>
## Average Over All Three Metrics

<table>
<thead>
<tr>
<th>DIM</th>
<th>761</th>
<th>761</th>
<th>780</th>
<th>927</th>
<th>1344</th>
<th>3448</th>
<th>21K</th>
<th>105K</th>
<th>195K</th>
<th>405K</th>
<th>685K</th>
<th>—</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.010</td>
</tr>
<tr>
<td>ANN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.004</td>
</tr>
<tr>
<td>BST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.003</td>
</tr>
<tr>
<td>SVM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.002</td>
</tr>
<tr>
<td>BGT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.999</td>
</tr>
<tr>
<td>LR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.997</td>
</tr>
<tr>
<td>KNN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.992</td>
</tr>
<tr>
<td>BSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.991</td>
</tr>
<tr>
<td>PRC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.982</td>
</tr>
<tr>
<td>NB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.949</td>
</tr>
</tbody>
</table>
## Average Over All Three Metrics

<table>
<thead>
<tr>
<th>DIM</th>
<th>761</th>
<th>761</th>
<th>780</th>
<th>927</th>
<th>1344</th>
<th>3448</th>
<th>21K</th>
<th>105K</th>
<th>195K</th>
<th>405K</th>
<th>685K</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>1.013</td>
<td>1.006</td>
<td>1.007</td>
<td>1.010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANN</td>
<td>1.006</td>
<td>1.004</td>
<td>1.002</td>
<td>1.004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BST</td>
<td>0.995</td>
<td>1.000</td>
<td>1.001</td>
<td>1.003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>1.006</td>
<td>1.000</td>
<td>1.000</td>
<td>1.002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGT</td>
<td>0.989</td>
<td>0.989</td>
<td>0.994</td>
<td>0.999</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>1.013</td>
<td>1.003</td>
<td>1.002</td>
<td>0.997</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KNN</td>
<td>1.000</td>
<td>0.962</td>
<td>0.986</td>
<td>0.992</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSS</td>
<td>0.994</td>
<td>0.986</td>
<td>0.999</td>
<td>0.991</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRC</td>
<td>0.993</td>
<td>1.004</td>
<td>0.983</td>
<td>0.982</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>0.987</td>
<td>0.943</td>
<td>0.950</td>
<td>0.949</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIM</td>
<td>761</td>
<td>761</td>
<td>780</td>
<td>927</td>
<td>1344</td>
<td>3448</td>
<td>21K</td>
<td>105K</td>
<td>195K</td>
<td>405K</td>
<td>685K</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>AVG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>0.994</td>
<td>1.021</td>
<td>1.009</td>
<td>1.007</td>
<td>1.019</td>
<td>1.005</td>
<td>1.001</td>
<td>1.032</td>
<td>1.013</td>
<td>1.006</td>
<td>1.007</td>
</tr>
<tr>
<td>ANN</td>
<td>0.998</td>
<td>1.040</td>
<td>1.018</td>
<td>0.998</td>
<td>1.021</td>
<td>0.987</td>
<td>0.988</td>
<td>0.988</td>
<td>0.995</td>
<td>1.000</td>
<td>1.001</td>
</tr>
<tr>
<td>BST</td>
<td>0.999</td>
<td>1.040</td>
<td>1.018</td>
<td>0.998</td>
<td>1.021</td>
<td>0.987</td>
<td>0.988</td>
<td>0.988</td>
<td>0.995</td>
<td>1.000</td>
<td>1.001</td>
</tr>
<tr>
<td>SVM</td>
<td>0.999</td>
<td>1.040</td>
<td>1.018</td>
<td>0.998</td>
<td>1.021</td>
<td>0.987</td>
<td>0.988</td>
<td>0.988</td>
<td>0.995</td>
<td>1.000</td>
<td>1.001</td>
</tr>
<tr>
<td>BGT</td>
<td>0.999</td>
<td>1.040</td>
<td>1.018</td>
<td>0.998</td>
<td>1.021</td>
<td>0.987</td>
<td>0.988</td>
<td>0.988</td>
<td>0.995</td>
<td>1.000</td>
<td>1.001</td>
</tr>
<tr>
<td>LR</td>
<td>0.999</td>
<td>1.040</td>
<td>1.018</td>
<td>0.998</td>
<td>1.021</td>
<td>0.987</td>
<td>0.988</td>
<td>0.988</td>
<td>0.995</td>
<td>1.000</td>
<td>1.001</td>
</tr>
<tr>
<td>KNN</td>
<td>0.999</td>
<td>1.040</td>
<td>1.018</td>
<td>0.998</td>
<td>1.021</td>
<td>0.987</td>
<td>0.988</td>
<td>0.988</td>
<td>0.995</td>
<td>1.000</td>
<td>1.001</td>
</tr>
<tr>
<td>BSS</td>
<td>0.999</td>
<td>1.040</td>
<td>1.018</td>
<td>0.998</td>
<td>1.021</td>
<td>0.987</td>
<td>0.988</td>
<td>0.988</td>
<td>0.995</td>
<td>1.000</td>
<td>1.001</td>
</tr>
<tr>
<td>PRC</td>
<td>0.999</td>
<td>1.040</td>
<td>1.018</td>
<td>0.998</td>
<td>1.021</td>
<td>0.987</td>
<td>0.988</td>
<td>0.988</td>
<td>0.995</td>
<td>1.000</td>
<td>1.001</td>
</tr>
<tr>
<td>NB</td>
<td>0.999</td>
<td>1.040</td>
<td>1.018</td>
<td>0.998</td>
<td>1.021</td>
<td>0.987</td>
<td>0.988</td>
<td>0.988</td>
<td>0.995</td>
<td>1.000</td>
<td>1.001</td>
</tr>
</tbody>
</table>

Mean values for each column:
- RF: 1.010
- ANN: 1.004
- BST: 1.003
- SVM: 1.002
- BGT: 0.999
- LR: 0.997
- KNN: 0.992
- BSS: 0.991
- PRC: 0.982
- NB: 0.949
**Average Over All Three Metrics**

<table>
<thead>
<tr>
<th>DIM</th>
<th>761</th>
<th>761</th>
<th>780</th>
<th>927</th>
<th>1344</th>
<th>3448</th>
<th>21K</th>
<th>105K</th>
<th>195K</th>
<th>405K</th>
<th>685K</th>
<th>—</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AVG</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>1.013</td>
<td>1.007</td>
<td>1.007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANN</td>
<td>1.004</td>
<td>1.003</td>
<td>1.002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BST</td>
<td>1.002</td>
<td>0.999</td>
<td>0.997</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>1.043</td>
<td>1.016</td>
<td>1.016</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGT</td>
<td>1.022</td>
<td>1.017</td>
<td>1.052</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>1.017</td>
<td>1.017</td>
<td>1.017</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KNN</td>
<td>1.018</td>
<td>1.021</td>
<td>1.020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSS</td>
<td>1.017</td>
<td>1.017</td>
<td>1.017</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRC</td>
<td>1.017</td>
<td>1.017</td>
<td>1.017</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>1.017</td>
<td>1.017</td>
<td>1.017</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

R. Caruana, N. Karampatziakis, A. Yessenalina

Learning in High Dimensions
## Average Over All Three Metrics

<table>
<thead>
<tr>
<th>AVG</th>
<th>DIM</th>
<th>761</th>
<th>761</th>
<th>780</th>
<th>927</th>
<th>1344</th>
<th>3448</th>
<th>21K</th>
<th>105K</th>
<th>195K</th>
<th>405K</th>
<th>685K</th>
<th>—</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIM</td>
<td>761</td>
<td>761</td>
<td>780</td>
<td>927</td>
<td>1344</td>
<td>3448</td>
<td>21K</td>
<td>105K</td>
<td>195K</td>
<td>405K</td>
<td>685K</td>
<td>—</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Stu</th>
<th>Cal</th>
<th>Dig</th>
<th>Tis</th>
<th>Cry</th>
<th>Kdd</th>
<th>R-S</th>
<th>Cite</th>
<th>Dse</th>
<th>Spam</th>
<th>IMDB</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>0.994</td>
<td>1.021</td>
<td>1.009</td>
<td>1.007</td>
<td>1.019</td>
<td>1.005</td>
<td>1.001</td>
<td>1.032</td>
<td>1.013</td>
<td>1.006</td>
<td>1.007</td>
<td>1.010</td>
<td></td>
</tr>
<tr>
<td>ANN</td>
<td>1.006</td>
<td>0.997</td>
<td>1.005</td>
<td>1.005</td>
<td>0.996</td>
<td>1.016</td>
<td>1.015</td>
<td>0.993</td>
<td>1.006</td>
<td>1.004</td>
<td>1.002</td>
<td>1.004</td>
<td></td>
</tr>
<tr>
<td>BST</td>
<td>0.998</td>
<td>1.040</td>
<td><strong>1.018</strong></td>
<td>0.998</td>
<td><strong>1.021</strong></td>
<td>0.987</td>
<td>0.988</td>
<td>0.988</td>
<td>0.995</td>
<td>1.000</td>
<td>1.001</td>
<td>1.003</td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>0.992</td>
<td>1.090</td>
<td>1.003</td>
<td>1.010</td>
<td>0.997</td>
<td>0.968</td>
<td><strong>1.020</strong></td>
<td>1.041</td>
<td>1.006</td>
<td>1.000</td>
<td>1.000</td>
<td>1.002</td>
<td></td>
</tr>
<tr>
<td>BGT</td>
<td>1.001</td>
<td><strong>1.043</strong></td>
<td>0.997</td>
<td>1.003</td>
<td>1.015</td>
<td>0.992</td>
<td>0.977</td>
<td>0.989</td>
<td>0.989</td>
<td>0.989</td>
<td>0.994</td>
<td>0.999</td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>1.002</td>
<td>0.993</td>
<td>0.886</td>
<td><strong>1.016</strong></td>
<td>1.003</td>
<td>1.017</td>
<td>1.018</td>
<td>1.009</td>
<td>1.013</td>
<td>1.003</td>
<td>1.002</td>
<td>0.997</td>
<td></td>
</tr>
<tr>
<td>KNN</td>
<td><strong>1.022</strong></td>
<td>1.000</td>
<td>1.017</td>
<td>0.946</td>
<td>0.999</td>
<td>1.006</td>
<td>0.920</td>
<td><strong>1.052</strong></td>
<td>1.000</td>
<td>0.962</td>
<td>0.986</td>
<td>0.992</td>
<td></td>
</tr>
<tr>
<td>BSS</td>
<td>1.012</td>
<td>1.033</td>
<td>0.890</td>
<td>0.982</td>
<td>0.998</td>
<td><strong>1.017</strong></td>
<td>0.993</td>
<td>0.999</td>
<td>0.994</td>
<td>0.986</td>
<td>0.999</td>
<td>0.991</td>
<td></td>
</tr>
<tr>
<td>PRC</td>
<td>0.996</td>
<td>0.978</td>
<td>0.883</td>
<td>0.967</td>
<td>0.993</td>
<td>0.991</td>
<td>1.016</td>
<td>0.999</td>
<td>0.993</td>
<td>1.004</td>
<td>0.983</td>
<td>0.982</td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>0.961</td>
<td>0.927</td>
<td>0.799</td>
<td>0.922</td>
<td>0.958</td>
<td>0.995</td>
<td>1.000</td>
<td>1.000</td>
<td>0.987</td>
<td>0.943</td>
<td>0.950</td>
<td>0.949</td>
<td></td>
</tr>
</tbody>
</table>

- Not apparent from this table: calibration with Isotonic Regression is almost always better than Platt’s method or no calibration.
Trends - Moving Average

-0.035
-0.030
-0.025
-0.020
-0.015
-0.010
-0.005
0.000
0.005
0.010
0.015

100
1000
10000
100000
1e+006

average score
dimension

example
Trends - Moving Average

-0.035
-0.03
-0.025
-0.02
-0.015
-0.01
-0.005
0
0.005
0.01
0.015

100
1000
10000
1e+006

average score

dimension

ANN
BAGDT
BSTDT
KNN
SVM
LR
BSTST
PRC
RF

R. Caruana, N. Karampatziakis, A. Yessenalina
Learning in High Dimensions
Trends - Moving Average

Average score vs. dimension for various models:
- ANN
- BAGDT
- BSTDT
- KNN
- SVM
- LR
- BSTST
- PRC
- RF

R. Caruana, N. Karampatziakis, A. Yessenalina
Learning in High Dimensions
Trends - Moving Average

R. Caruana, N. Karampatziakis, A. Yessenalina

Learning in High Dimensions
Trends - Cumulative Performance

![Graph showing cumulative performance trends](graph.png)
Introduction Methodology Challenges

Results

Trends - Cumulative Performance

R. Caruana, N. Karampatziakis, A. Yessenalina
Learning in High Dimensions
Trends - Cumulative Performance

R. Caruana, N. Karampatziakis, A. Yessenalina
Learning in High Dimensions
Trends - Cumulative Performance

-0.2
-0.15
-0.1
-0.05
0
0.05
0.1
0.05
0.1
0.5
1
1e+006
100
1000
10000
100000
1e+006

cumulative score

dimension

ANN
BAGDT
BSTDT
KNN
SVM
LR
BSTST
PRC
RF

R. Caruana, N. Karampatziakis, A. Yessenalina
Learning in High Dimensions
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.
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

Questions?