An Empirical Evaluation of Supervised Learning in High Dimensions

Rich Caruana Nikos Karampatziakis Ainur Yessenalina

Department of Computer Science, Cornell University

July 3, 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

Problem	$\approx$ Attr	Domain
Sturn	760	Ornithology dataset
Calam	760	Ornithology dataset
Digits	780	Image recognition, MNIST, < 5 versus ≥ 5
Tis	930	Protein translation problem
Cryst	1300	Protein crystallography diffraction
KDD98	4K	Predict if person will donate money
R-S	21K	Text classification
Dse	200K	Sentiment analysis
Spam	400K	Text classification
Cite	100K	Link prediction
Imdb	685K	Link prediction

- 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^{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 kNN (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 kNN (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 kNN (KNN*)
- Bagged Decision Trees (BAGDT*)
- Random Forests (RF*)

Like 5×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 kNN (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 kNN (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 kNN (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.

Calibration

- 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 $1-$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 $1-$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

$\times$

100's of parameter settings per method

$=$

1,000 expensive models trained per problem

$\times$

11 Boolean classification test problems

$=$

11,000 models

$\times$

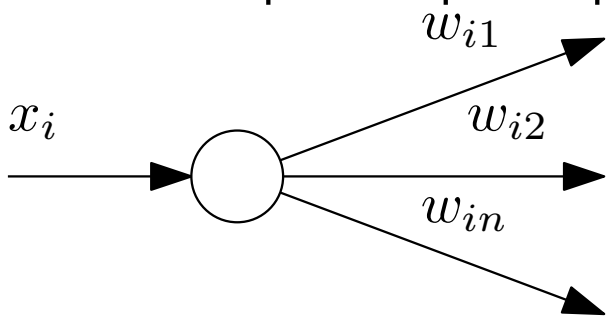
3 performance metrics

$=$

33,000 model performance evaluations

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



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

  - 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

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

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												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

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									**1.013**	**1.006**	**1.007**	1.010
ANN									1.006	1.004	1.002	1.004
BST									0.995	1.000	1.001	1.003
SVM									1.006	1.000	1.000	1.002
BGT									0.989	0.989	0.994	0.999
LR									1.013	1.003	1.002	0.997
KNN									1.000	0.962	0.986	0.992
BSS									0.994	0.986	0.999	0.991
PRC									0.993	1.004	0.983	0.982
NB									0.987	0.943	0.950	0.949

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												1.004
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												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

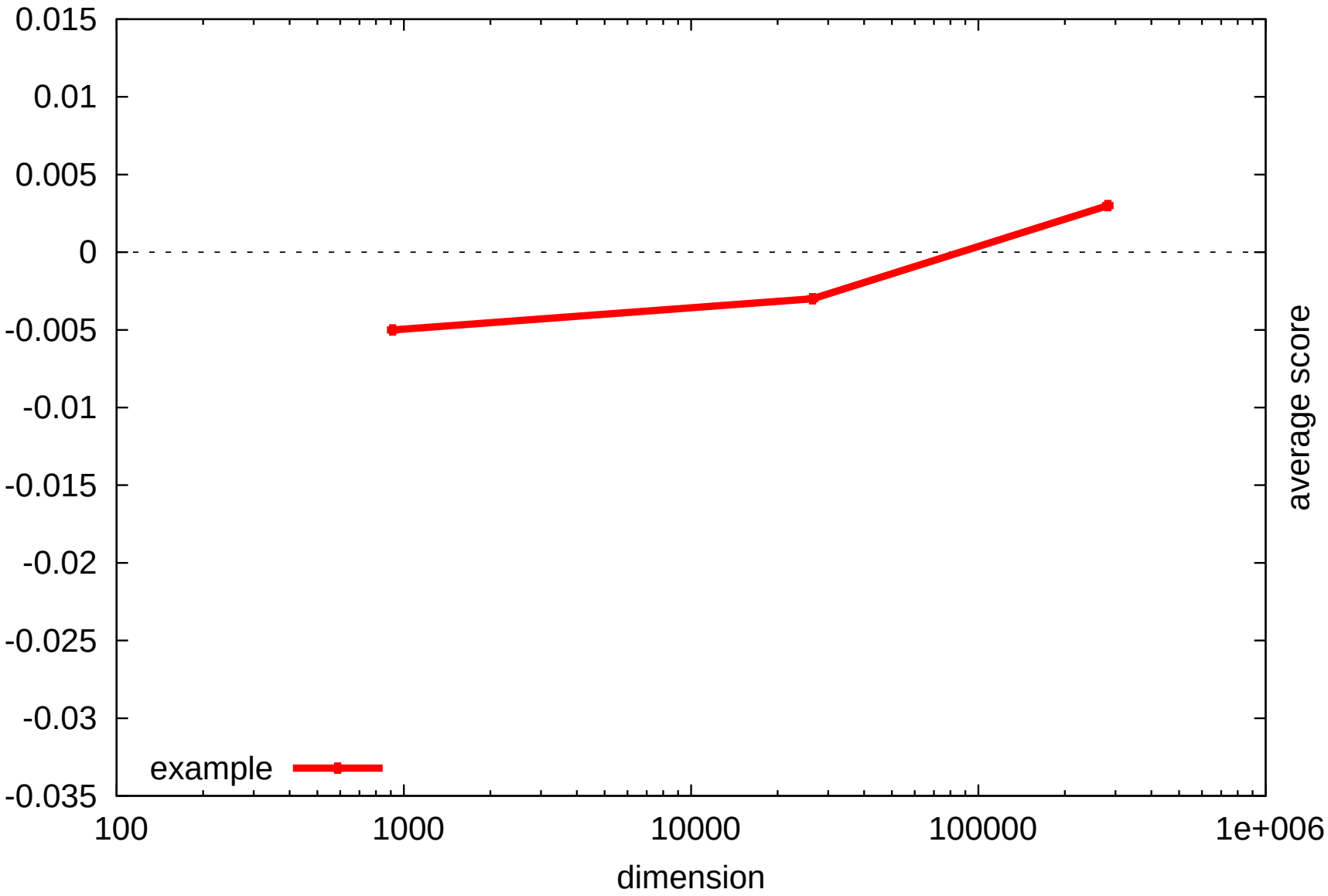
DIM	761	761	780	927	1344	3448	21K	105K	195K	405K	685K	—
AVG	Stu	Cal	Dig	Tis	Cry	Kdd	R-S	Cite	Dse	Spam	Imdb	Mean
RF									**1.013**	**1.006**	**1.007**	1.010
ANN	1.006	0.997	1.005	1.005	0.996	1.016	1.015	0.993	1.006	1.004	1.002	1.004
BST			**1.018**		**1.021**							1.003
SVM							**1.020**					1.002
BGT		**1.043**										0.999
LR				**1.016**								0.997
KNN	**1.022**							**1.052**				0.992
BSS						**1.017**						0.991
PRC												0.982
NB												0.949

Average Over All Three Metrics

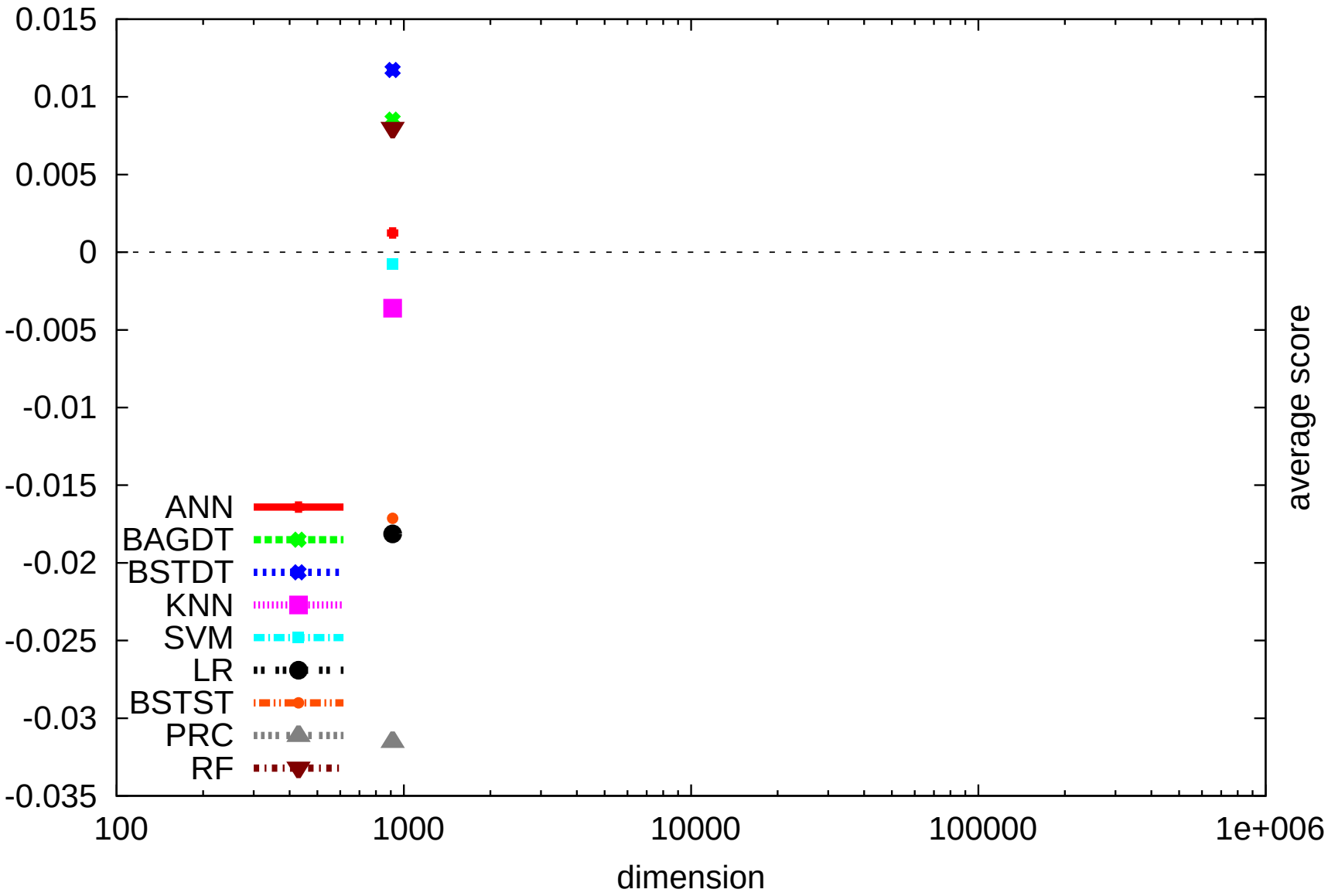
DIM	761	761	780	927	1344	3448	21K	105K	195K	405K	685K	—
AVG	STU	CAL	DIG	TIS	CRY	KDD	R-S	CITE	DSE	SPAM	IMDB	MEAN
RF	0.994	1.021	1.009	1.007	1.019	1.005	1.001	1.032	**1.013**	**1.006**	**1.007**	1.010
ANN	1.006	0.997	1.005	1.005	0.996	1.016	1.015	0.993	1.006	1.004	1.002	1.004
BST	0.998	1.040	**1.018**	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

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

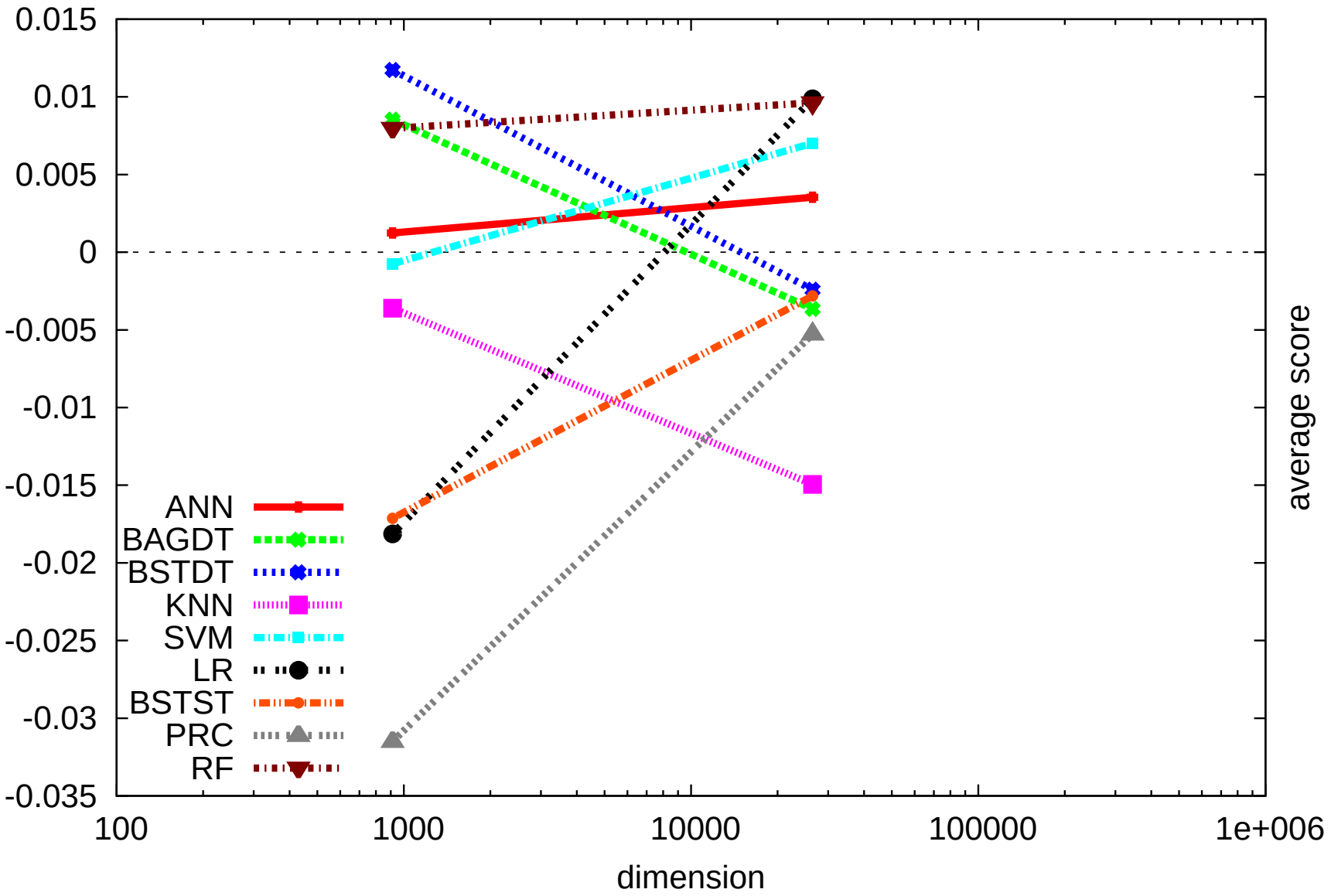
Trends - Moving Average



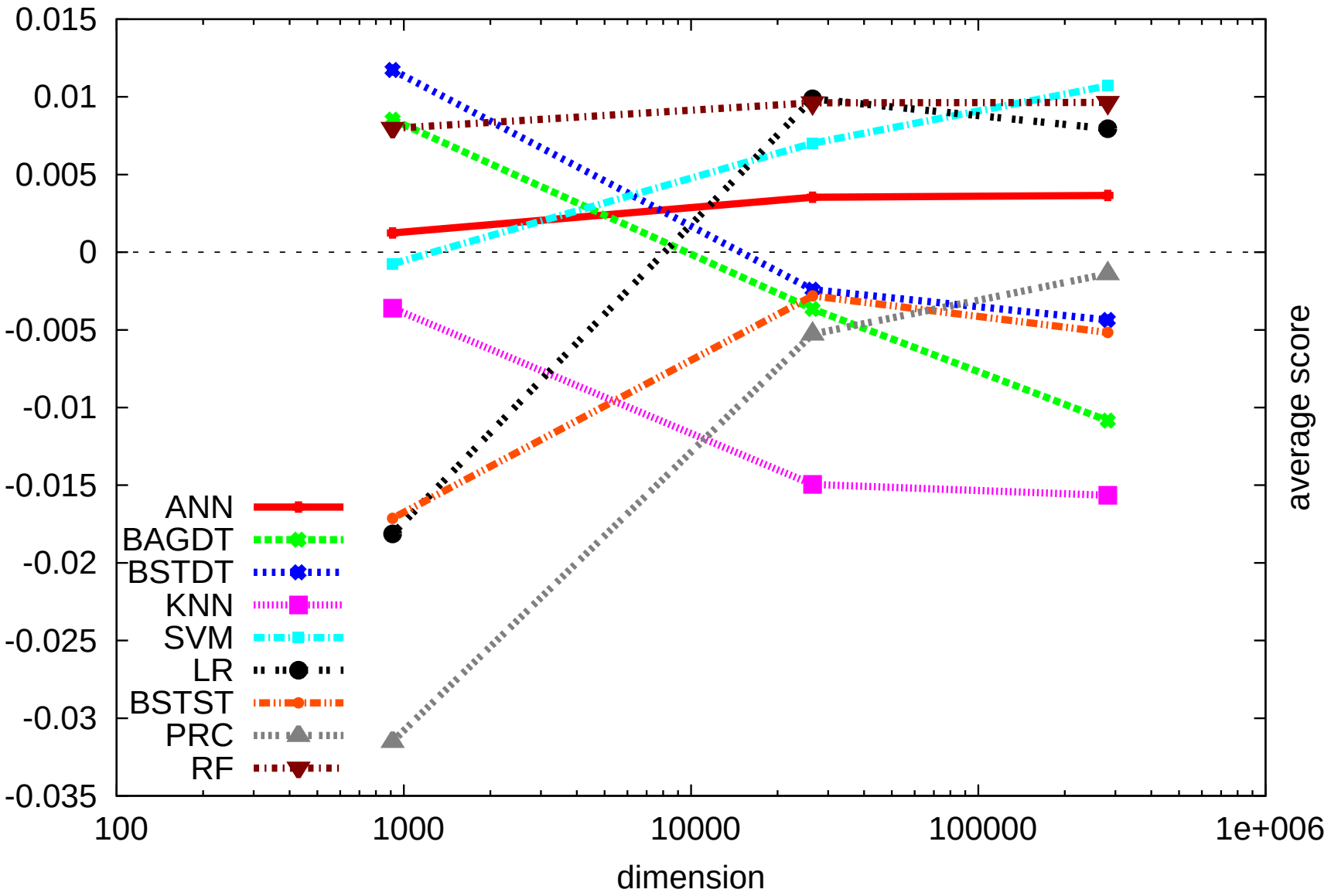
Trends - Moving Average



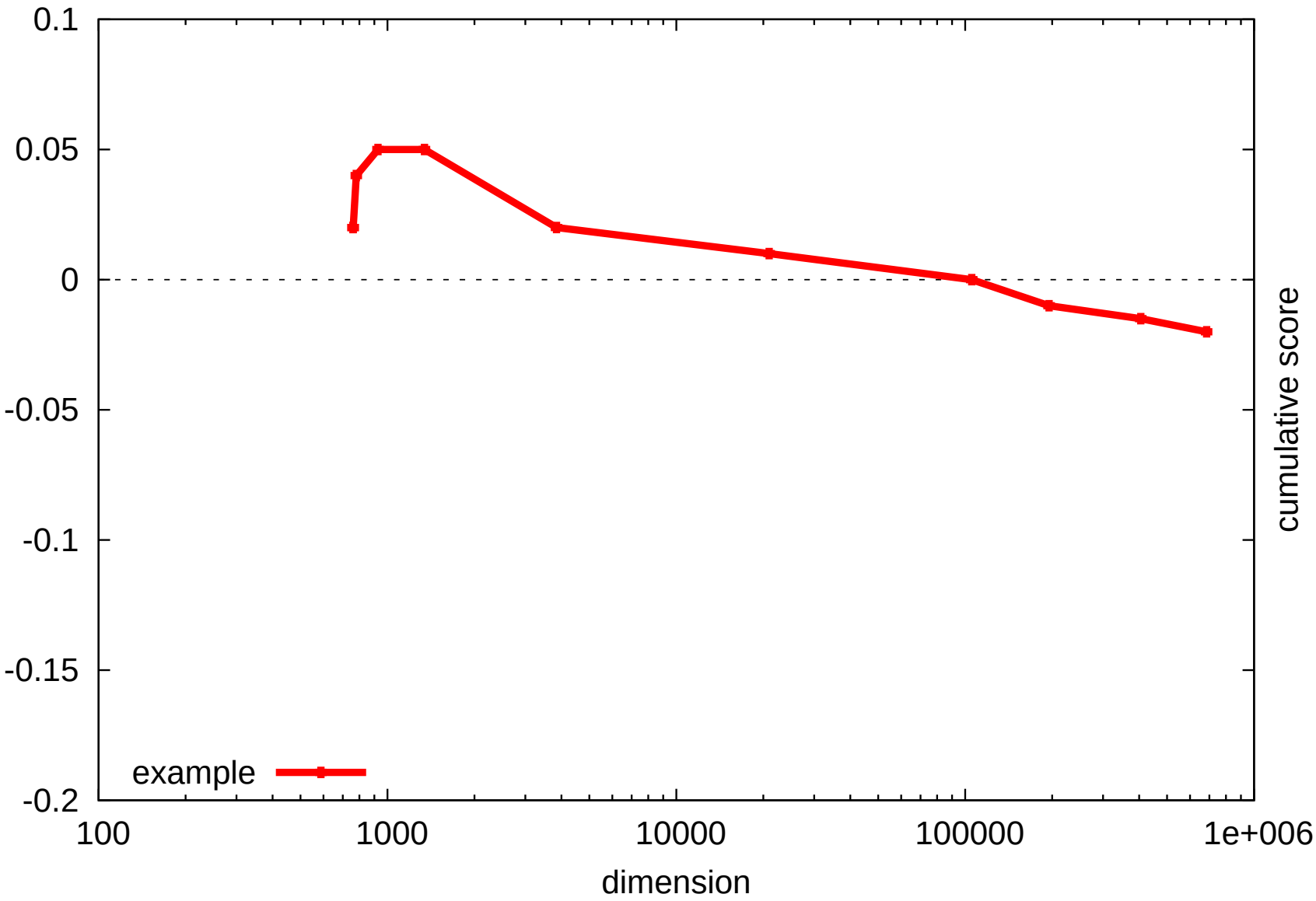
Trends - Moving Average



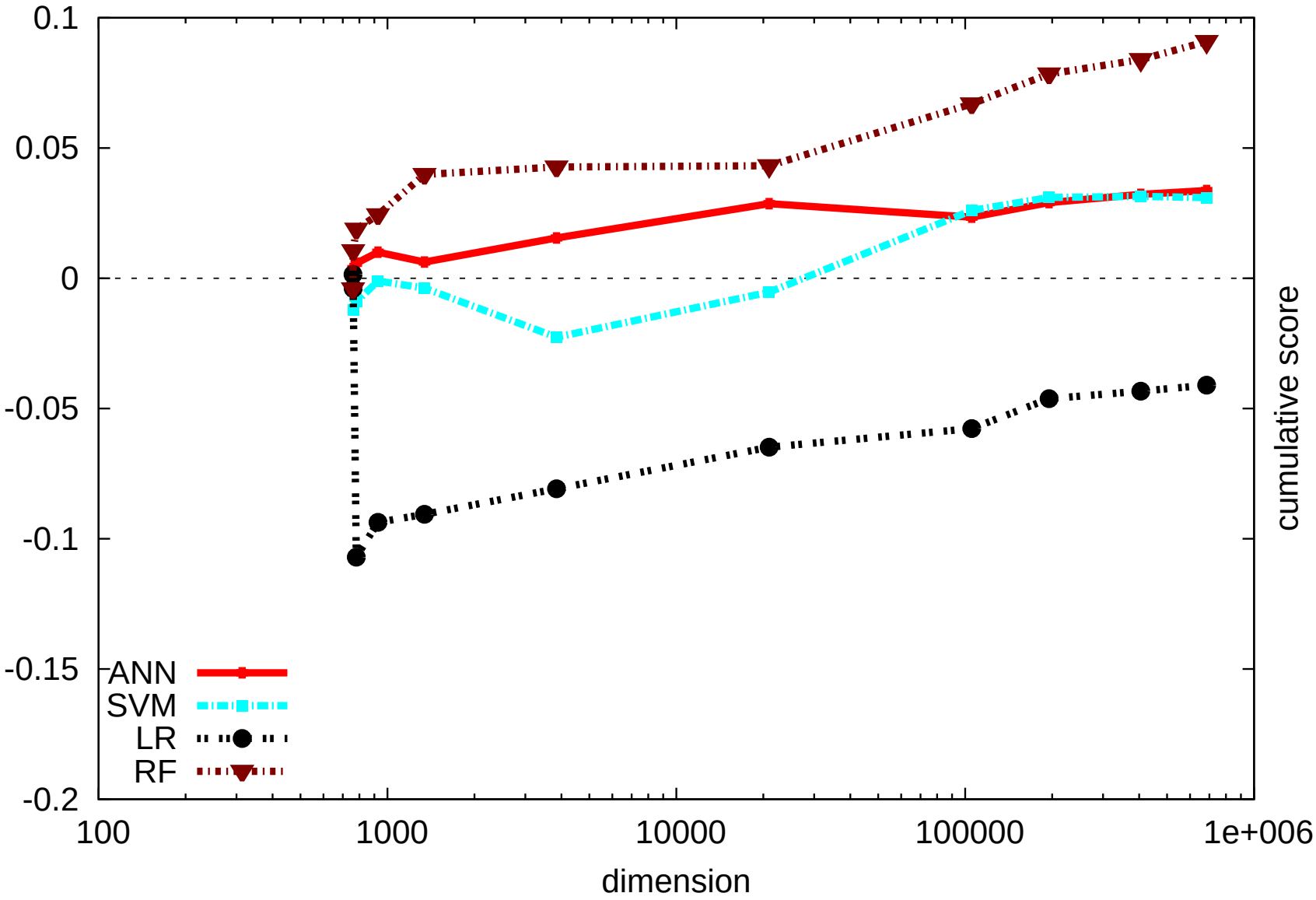
Trends - Moving Average



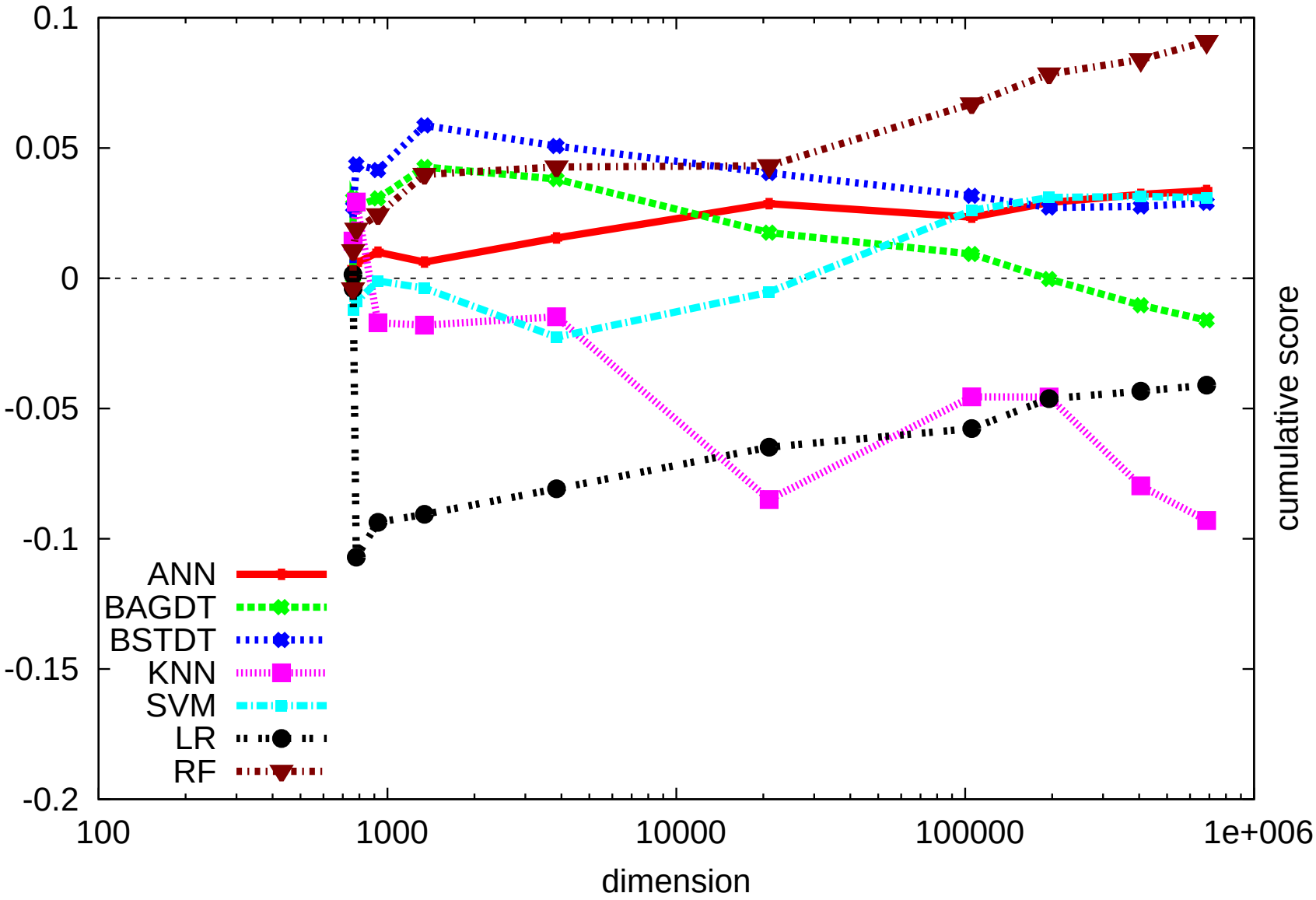
Trends - Cumulative Performance



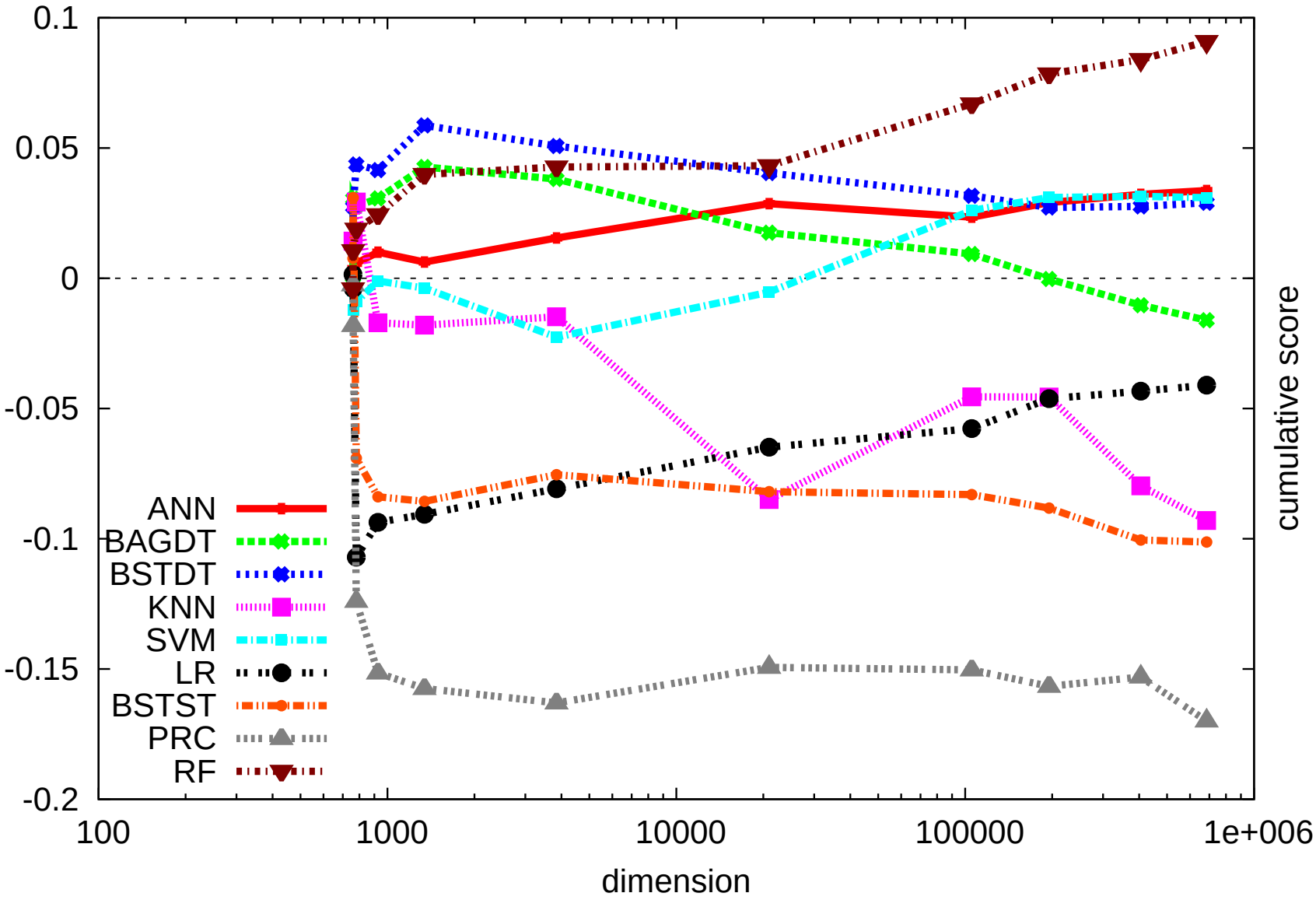
Trends - Cumulative Performance



Trends - Cumulative Performance



Trends - Cumulative Performance



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?