Active Learning for Logistic Regression

Andrew I. Schein
The University of Pennsylvania
Department of Computer and Information Science
Philadelphia, PA 19104-6389 USA
ais@cis.upenn.edu

April 21, 2005

What is Active Learning?

A scenario where learning agents interact with their environment.

...instead of passively receiving inputs.

We will focus on pool-based active learning for classification:

Observations $\mathbf{x}_n$ are given without their corresponding class labels y_n.

Our goal: Sequentially pick $\mathbf{x}_n$ to label to train the best classifier.

Example:

We have a pool of 2000 documents with no topic label.

Which documents do we label to build the best topic classifier?

Why Focus on Active Learning for Logistic Regression?

- Active learning in other settings already well studied.

- Logistic regression popular in a variety of applications:
  - Natural language processing
  - Biological sequence modeling
  - Economics
  - Social sciences

- Generalizations exist for more complex modeling problems:
  - The maximum entropy classifier
  - The conditional random field model

The Setting: Classification with Noise

$$\text{Training Set } \mathcal{D} \quad = \quad \{\mathbf{x}_n, y_n\}_1^N . \tag{1}$$

We assume the classification setting with noise...

There exists a function $t(\mathbf{x}, c)$ such that:

$$\mathrm{P}(Y = c|\mathbf{x}_n) = t(\mathbf{x}_n, c) \tag{2}$$

This is the “true model” which we estimate using $\pi(c, \mathbf{x}_n, \hat{\mathbf{w}}; \mathcal{D})$.

Logistic Regression

A Maximum Entropy Method for Class Probability Estimation

- Binary

$$\pi(c = 1, \mathbf{x}_n, \hat{\mathbf{w}}; \mathcal{D}) = \frac{1}{1 + \exp(-\mathbf{x}_n \cdot \mathbf{w})} \tag{3}$$

- Multiple Classes

$$\pi(c, \mathbf{x}_n, \hat{\mathbf{w}}; \mathcal{D}) = \frac{\exp(\mathbf{x}_n \cdot \mathbf{w}_c)}{\sum_{c'} \exp(\mathbf{x}_n \cdot \mathbf{w}_{c'})} \tag{4}$$

$\mathbf{x}_n$ is a vector of predictors for observation n
$\mathbf{w}_y$ is a vector of weights indexed by class y

Thesis

Discover best practices for active learning with logistic regression by:

- Examining active learning heuristics in logistic regression context.
- Developing loss function methods for logistic regression.
- Identifying when methods work and don’t work.
- Supporting conclusions with extensive empirical evaluation.
  - Most thorough evaluation of active learning for logistic regression
  - Most thorough evaluation of a loss function strategy.

Talk Outline

1. Derive Loss Function Methods for Logistic Regression
2. Motivate and Explain the Heuristics Evaluated
3. Describe Evaluation Strategy
4. Evaluation Results
5. Analysis of Results
6. Conclusions

How to use a Loss Function in Active Learning

Loss ideally is measured on a test set, but the pool is a surrogate.

Let $\phi(\mathcal{D})$ be a loss computed over pool. It depends on training set $\mathcal{D}$.

Our goal is to pick $\arg\min_{\mathcal{D}} \phi(\mathcal{D})$.

We can pick examples by maximizing expected benefit:

$$\begin{aligned} E_{y_n}\left[\phi(\mathcal{D} \cup (\mathbf{x}_n, y_n))\right] &= \hat{\mathsf{P}}(y_n = 0|\mathbf{x}_n)\phi(\mathcal{D} \cup (\mathbf{x}_n, 0)) \\ &+ \hat{\mathsf{P}}(y_n = 1|\mathbf{x}_n)\phi(\mathcal{D} \cup (\mathbf{x}_n, 1)). \end{aligned}$$

$\hat{\mathsf{P}}$ is the current model.

All we need now is a loss function and a way to compute it over the pool.

Analysis of Squared Loss

Define squared loss as follows:

$$\sum_{nc} \mathsf{E}[(y_{nc} - \pi(c, \mathbf{x}_n; \mathcal{D}))^2 | \mathbf{x}_n, \mathcal{D}] = \sum_{nc} \mathsf{E}[(y_{nc} - \mathsf{E}[c|\mathbf{x}_n])^2 | \mathbf{x}_n, \mathcal{D}] \quad \text{“noise”}$$
$$+ \sum_{nc} (\pi(c, \mathbf{x}_n; \mathcal{D}) - \mathsf{E}[c|\mathbf{x}_n])^2$$

y_{nc} is an indicator function.

E is expectation w.r.t. actual distribution P$(y, \mathbf{x})$.

The first term, “noise,” is independent of the training set.

The second term captures error due to using training set $\mathcal{D}$.

Next: Take an expectation over training sets of fixed size: $\mathsf{E}_{\mathcal{D}}$

Mean Squared Error

Taking the expectation of the training set dependent term we get:

$$\text{MSE} \doteq \sum_{nc} \mathsf{E}_{\mathcal{D}}[(\pi(c, \mathbf{x}_n; \mathcal{D}) - \mathsf{E}[c|\mathbf{x}_n])^2]. \tag{5}$$

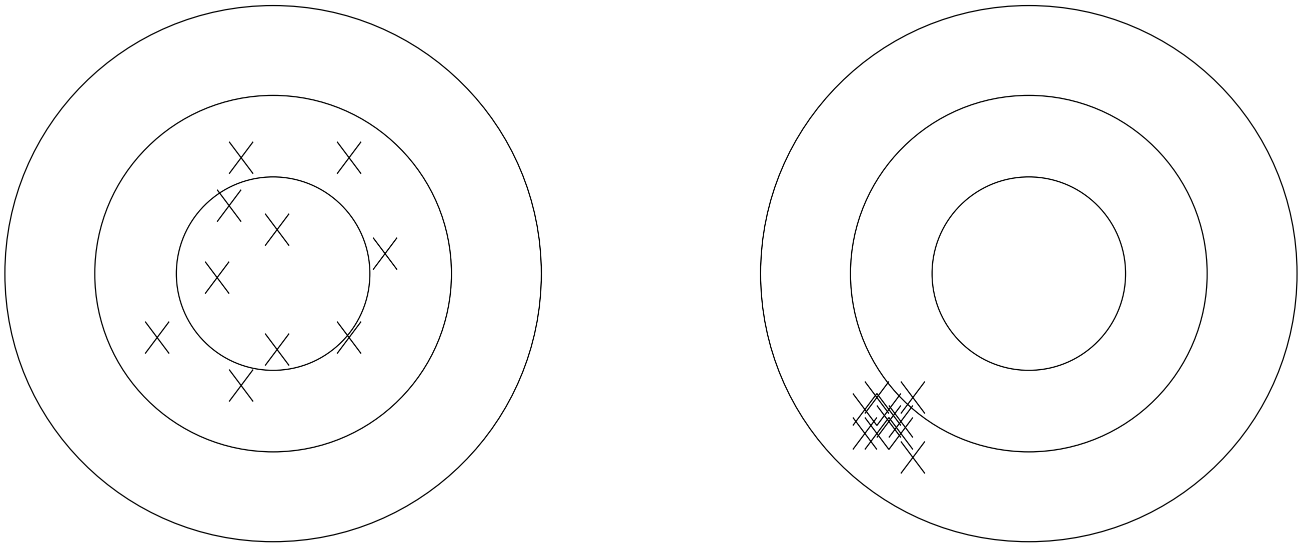
This is the mean squared error (MSE).

Computed over a test set or pool as a surrogate (sum over n).

MSE decomposes as follows:

$$\begin{aligned} \text{MSE} &= \sum_{nc} (\mathsf{E}_{\mathcal{D}}[\pi(c, \mathbf{x}_n; \mathcal{D})] - \mathsf{E}[c|\mathbf{x}_n])^2 \quad \textit{“squared bias”} \\ &+ \sum_{nc} \mathsf{E}_{\mathcal{D}}[(\pi(c, \mathbf{x}_n; \mathcal{D}) - \mathsf{E}_{\mathcal{D}}[\pi(c, \mathbf{x}_n; \mathcal{D})])^2]. \quad \textit{“variance”} \end{aligned} \tag{6}$$

A Graphical Presentation of Bias and Variance



Low Bias, High Variance High Bias, Low Variance

A Criterion For Picking a Training Set

$$\begin{aligned} \text{MSE} &= \sum_{nc} (\mathsf{E}_{\mathcal{D}}[\pi(c, \mathbf{x}_n; \mathcal{D})] - \mathsf{E}[c|\mathbf{x}_n])^2 \quad \textit{“squared bias”} \\ &+ \sum_{nc} \mathsf{E}_{\mathcal{D}}[(\pi(c, \mathbf{x}_n; \mathcal{D}) - \mathsf{E}_{\mathcal{D}}[\pi(c, \mathbf{x}_n; \mathcal{D})])^2]. \quad \textit{“variance”} \end{aligned} \tag{7}$$

MSE is difficult to compute since $\mathsf{E}[y_{nc}|\mathbf{x}_{nc}]$ is unknown.

Bias estimation requires a nonparametric method (such as bootstrap).

Variance estimation can take advantage of model structure.

A Variance Reduction Approach

Step 1... take a Taylor expansion:

$$\begin{aligned} \pi(c, \mathbf{x}_n, \hat{\mathbf{w}}; \mathcal{D}_s) &= \pi(c, \mathbf{x}_n, \overline{\mathbf{w}}; \mathcal{D}_s) \\ &+ \mathbf{g}_n(c)(\hat{\mathbf{w}} - \overline{\mathbf{w}}) + O(\frac{1}{s}), \end{aligned} \tag{8}$$

$\overline{\mathbf{w}}$ is the expected parameter estimate for fixed training set size s.

$\mathbf{g}_n(c) = \frac{\partial}{\partial \mathbf{w}} \pi(c, \mathbf{x}_n, \hat{\mathbf{w}}; \mathcal{D})$.

Asymptotics follow from efficiency of ML estimate $\hat{\mathbf{w}}$.

A Variance Reduction Approach

Step 2: compute variance of Taylor expansion:

$$\sum_{nc} \text{Var}[\pi(c, \mathbf{x}_n; \mathcal{D}_s)] \simeq \sum_{nc} \text{Var}[\mathbf{g}_n(c)(\hat{\mathbf{w}} - \overline{\mathbf{w}})] \quad (9)$$

$$\simeq \text{tr}\left\{AF^{-1}\right\} \quad (10)$$

A encodes the pool predictor distribution: $\sum_{nc} g_n(c) g_n(c)'$.

The second equation follows from asymptotic normality of $(\hat{\mathbf{w}} - \overline{\mathbf{w}})$.

F is the Fisher information matrix.

$O(K^3 D^3)$ to compute naively for most of our evaluation settings.

K = number of categories
D = number of predictors

What Other Loss Functions Can We Use?

Squared loss may be written:

$$L(\mathbf{p}, \mathbf{q}) = \sum_c (p_c - q_c)^2 \tag{11}$$

For $p_c = \mathsf{E}_{\mathcal{D}}[\pi(c, \mathbf{x}_n; \mathcal{D})]$ and $q_c = \pi(c, \mathbf{x}_n; \mathcal{D})$ this is almost variance.

Fix these choices of **p** and **q**: $L(\mathsf{E}_{\mathcal{D}}[\pi(c, \mathbf{x}_n; \mathcal{D})], \pi(c, \mathbf{x}_n; \mathcal{D}))$.

Consider loss function with the following restrictions:

1. $L(p, p) =$ constant.
2. $L(p, q)$ twice differentiable.
3. The second term in a Taylor approximation equals zero.

Examples: Log Loss, Squared Loss.

A Generalized Loss Function Strategy

Use Taylor approximation and then take expectation $\mathsf{E}_{\mathcal{D}}$:

$$\mathsf{E}_{\mathcal{D}}[L(\mathbf{p},\mathbf{q})] \quad \simeq \quad L(\mathbf{p},\mathbf{p}) + \frac{1}{2}\mathsf{E}_{\mathcal{D}}[(\mathbf{p}-\mathbf{q})' \left\{ \frac{\partial^2}{\partial \mathbf{q}^2} L(\mathbf{p},\mathbf{q})|_{\mathbf{q}=\mathbf{p}} \right\} (\mathbf{p}-\mathbf{q})].$$

For $L(\cdot,\cdot) =$ squared loss, the approximation is exactly variance.

For log loss, a reweighted variance emerges:

$$L(p,q) \quad \simeq \quad \text{constant} + \sum_c \frac{1}{p_c} \mathsf{E}_{\mathcal{D}}[p_c - q_c]^2. \qquad (12)$$

This is equivalent to reweighting A matrix in A-optimality formula.

Picking the Next Observation

Picking which observation to label next is an expectation computation

The expectation is over possible labeling (using current model):

$$
\begin{aligned}
E_{y_n}\left[\phi(\mathcal{D} \cup (\mathbf{x}_n, y_n))\right] &= \hat{\mathsf{P}}(y_n = 0 | \mathbf{x}_n)\phi(\mathcal{D} \cup (\mathbf{x}_n, 0)) \\
&+ \hat{\mathsf{P}}(y_n = 1 | \mathbf{x}_n)\phi(\mathcal{D} \cup (\mathbf{x}_n, 1)).
\end{aligned}
$$

ϕ is loss function with respect to a training set $\mathcal{D}$.

Computation time becomes $O(NK^4D^3)$ in our evaluation setting.

N is the number of candidates evaluated (we will use $N = 10$).

K is number of categories, D is number of predictors.

A Tour of Heuristic Active Learning Approaches

- Uncertainty Sampling
  - **entropy**-based uncertainty sampling
  - **margin**-based uncertainty sampling
- Query by Bagging
  - **QBB-MN** Query by Bagging – KL divergence measure
  - **QBB-AM** Query by Bagging – ensemble margin
- **CC** Classifier Certainty Method

Uncertainty Sampling Heuristic

Lewis and Gale, 1994:

- Pick examples classifier is "uncertain" about for labeling.
- Intuition: these examples should help clarify decision boundary.
- Measures of uncertainty include:
  - Shannon entropy of classification distribution
  - Margin for $\mathbf{x}_n = |\mathrm{P}(i|\mathbf{x}_n) - \mathrm{P}(j|\mathbf{x}_n)|$
    where i, j are the two most likely classes.
- These two measures are identical for binary classification.

Comparison Margin and Entropy Sampling Algorithms

- Shannon entropy sampling looks at probabilities for all categories
  - Picks examples with uniform distribution
- Margin sampling only depends on the two most likely categories
  - Other categories may potentially have zero probability mass.
- Differ when number of categories > 2:
  - low margin does not mean large entropy.

Query by Bagging Heuristic

- Based loosely on the Query by Committee algorithm.
- The algorithm forms an ensemble using the bagging technique.
- Picks for labeling the example with high ensemble disagreement.
- We evaluate two disagreement measures:
  - **QBB-AM** uses margin (Abe and Mamitsuka, 1998)
  - **QBB-MN** uses KL divergence (McCallum and Nigam, 1998)

Classifier Certainty Heuristic

MacKay, 1992. Roy and McCallum, 2001.

Minimize the certainty of predictions over the pool:

$$L(\hat{\mathsf{P}}, \hat{\mathsf{P}}) = -\sum_{\mathbf{x},c} \hat{\mathsf{P}}(c|\mathbf{x}) \log \hat{\mathsf{P}}(c|\mathbf{x}) \mathsf{P}(\mathbf{x})$$

where $\hat{\mathsf{P}}$ are the model's predictions.

This objective function can be minimized by any model with low entropy.

Summary of Methods Evaluated

- **Baseline**
  random instance selection
  bagging (interesting since it is used in QBB methods)

- **Loss Function Approaches**
  variance reduction (A-optimality)
  log loss reduction

- **Heuristics**
  CC Classifier Certainty
  QBB-MN Query by Bagging – KL divergence measure
  QBB-AM Query by Bagging – ensemble margin
  entropy-based uncertainty sampling
  margin-based uncertainty sampling

Evaluation Strategy

- Find data sets with many observations and varying numbers of
  - Predictors
  - Categories
- Split data into pool and test set (50/50).
- Perform 10 cross-fold validation
- Pick 20 random examples and let algorithms pick up to 300 examples.
- Repeat with 50, 100, and 200 starter examples.
- Due to size and other constraints, for three data sets pick $<$ 300.

The Data Sets

Data Set	Classes	Obs	Pred	Data Type
Art	20	20,000	5	artificial
ArtNoisy	20	20,000	5	artificial
ArtConf	20	20,000	5	artificial
Comp2a	2	1,989	6,191	document
Comp2b	2	2,000	8,617	document
LetterDB	26	20,000	16	char. image
NewsGroups	20	18,808	16,400	document
OptDigits	10	5,620	64	char. image
TIMIT	20	10,080	12	voice
WebKB	4	4,199	7,543	document

Accuracy After Training on Pool (Ceiling Accuracy)

Data Set	Accuracy
TIMIT	0.525
ArtNoisy	0.602
LetterDB	0.764
NewsGroups	0.820
ArtConf	0.844
WebKB	0.907
Art	0.919
Comp2a	0.885
Comp2b	0.889
OptDigits	0.964

Results - Accuracy

Data Set	random	bagging	variance	log loss
Art	0.809	**0.792**	**0.862**	**0.867**
ArtNoisy	0.565	**0.557**	**0.579**	**0.579**
ArtConf	0.837	0.830	0.842	0.840
Comp2a	0.821	**0.794**	0.805	0.821
Comp2b	0.799	0.793	0.807	0.796
LetterDB	0.609	**0.593**	**0.644**	**0.646**
NewsGroups	0.483	**0.422**	–	–
OptDigits	0.927	0.931	**0.937**	**0.944**
TIMIT	0.413	**0.397**	0.405	0.423
WebKB	0.830	**0.803**	–	–

	CC	QBB-MN	QBB-AM	entropy	margin
Art	**0.821**	**0.848**	**0.861**	**0.832**	**0.867**
ArtNoisy	0.567	**0.577**	0.571	**0.536**	**0.572**
ArtConf	0.845	0.843	**0.816**	**0.723**	**0.749**
Comp2a	**0.788**	0.814	0.818	0.826	0.818
Comp2b	0.796	0.804	0.808	0.805	0.800
LetterDB	**0.625**	**0.599**	**0.637**	**0.548**	**0.633**
NewsGroups	–	**0.464**	**0.444**	**0.356**	**0.438**
OptDigits	**0.942**	**0.941**	**0.949**	**0.951**	**0.952**
TIMIT	**0.395**	0.408	**0.438**	**0.327**	**0.440**
WebKB	–	**0.844**	**0.860**	**0.855**	**0.860**

Results - Number of Random Examples Needed to Give Similar Accuracy as Percentage of Stopping Point

Data Set	random	bagging	variance	log loss
Art	100	**73**	**>200**	**> 200**
ArtNoisy	100	**80**	**150**	**150**
ArtConf	100	83	108	100
Comp2a	100	**73**	87	140
Comp2b	100	87	113	93
LetterDB	100	**83**	**127**	**127**
NewsGroups	100	**77**	–	–
OptDigits	100	103	**117**	**143**
TIMIT	100	**80**	97	103
WebKB	100	**73**	–	–

	CC	QBB-MN	QBB-AM	entropy	margin
Art	**110**	**160**	**> 200**	**123**	**>200**
ArtNoisy	103	**140**	117	**53**	**117**
ArtConf	117	117	**92**	**42**	**42**
Comp2a	**60**	100	100	127	100
Comp2b	93	107	113	107	100
LetterDB	**113**	**83**	**120**	**60**	**120**
NewsGroups	–	**97**	**93**	**57**	**87**
OptDigits	**133**	**133**	**>200**	**>200**	**>200**
TIMIT	**77**	97	**140**	**30**	**127**
WebKB	–	**120**	**190**	**153**	**177**

Performance of Loss Function Methods

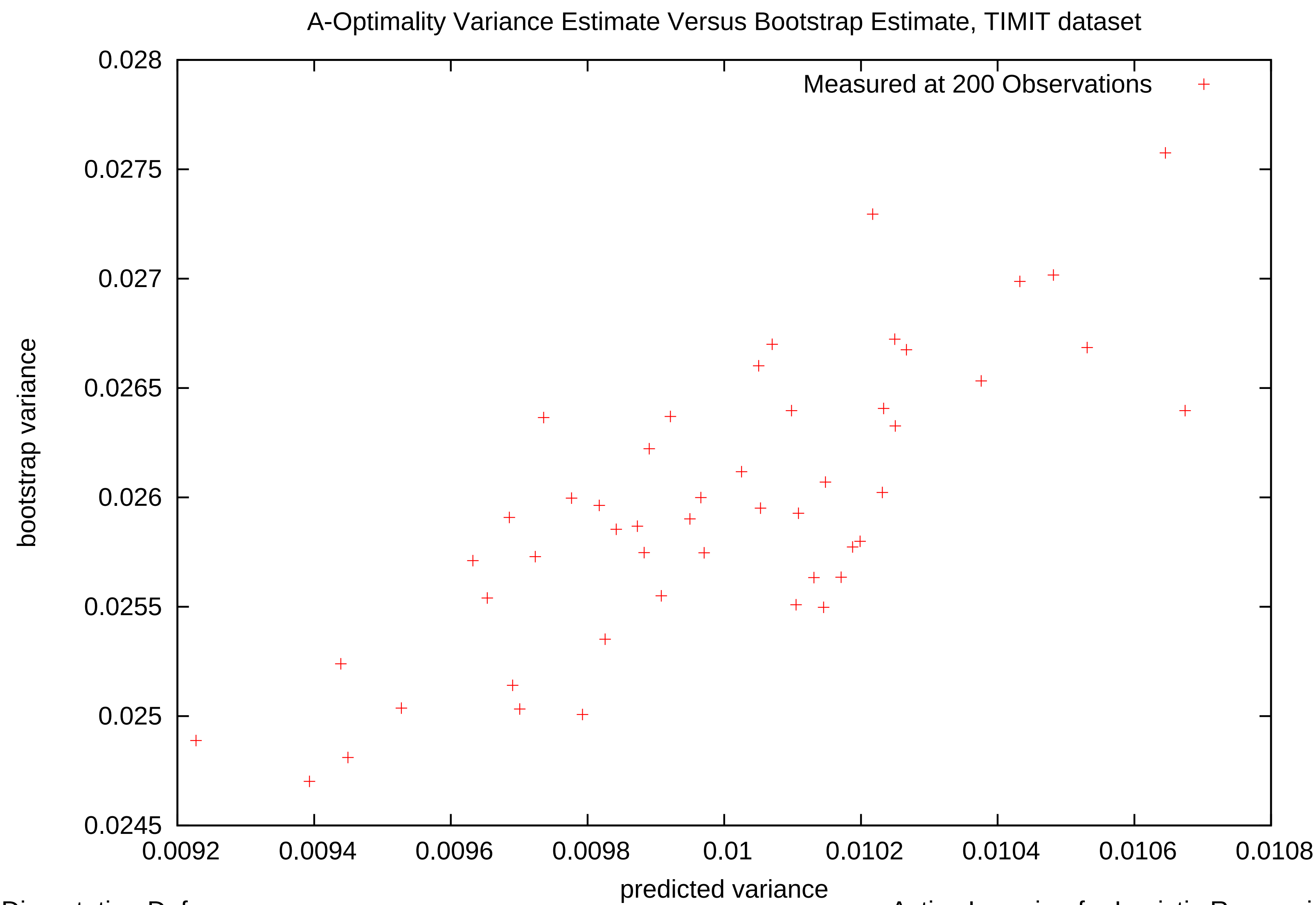
- Variance and Log Loss methods most robust methods tested:
  - Frequently outperform random training sets
  - Only methods to always match (or beat) random training sets
- Performance comes with computational cost:
  - Largest data sets took weeks to run.
  - Number of model parameters impediment in evaluation.

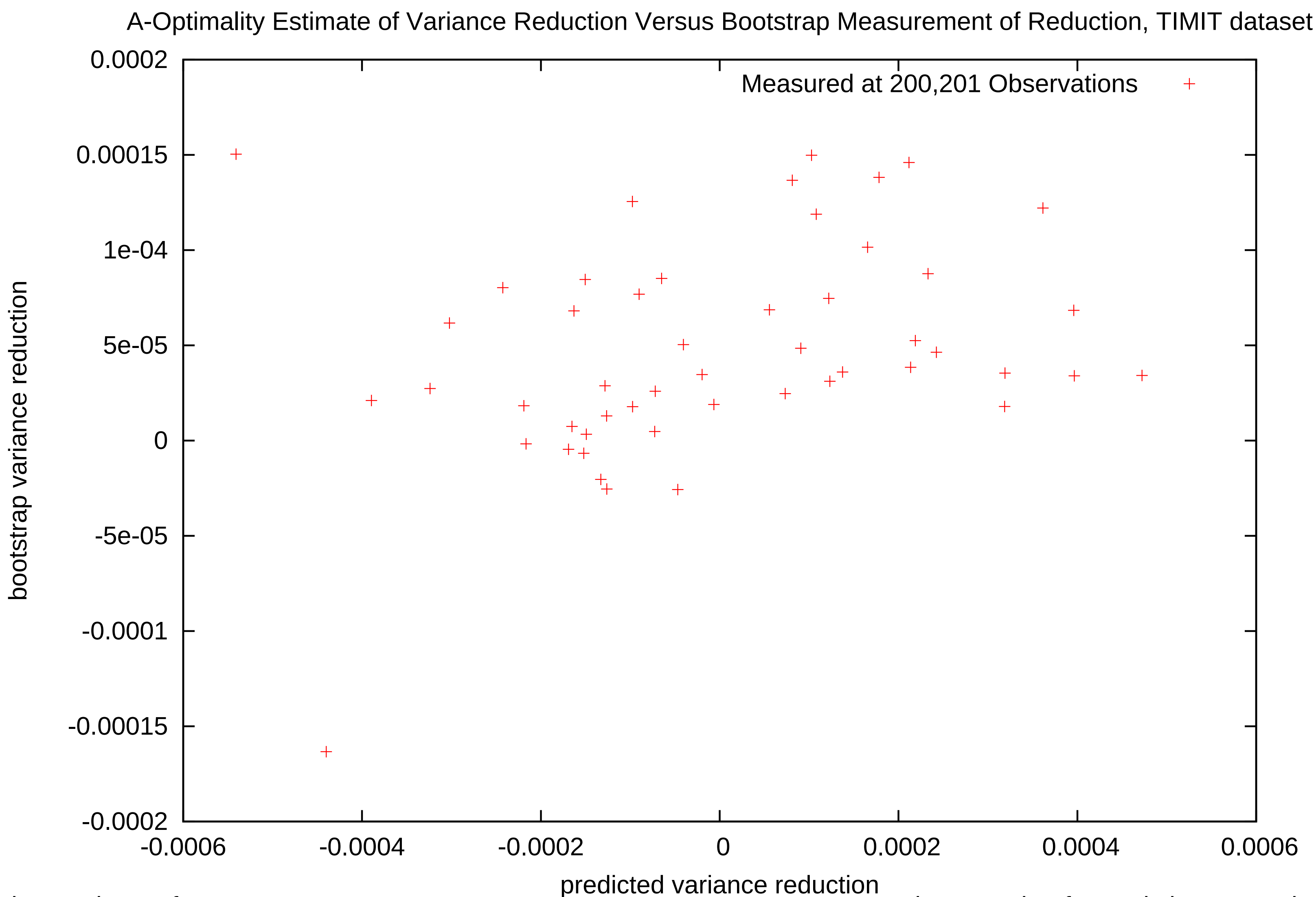
A Discussion of Variance and 0/1 Loss

- We minimize prediction variance as a means to decrease 0/1 loss.
- Recent theoretical analysis of 0/1 loss suggests this can be:
  - helpful when model is biased towards the correct classification.
  - harmful when model is biased towards a wrong classification.
- Our empirical evaluation suggests:
  - variance reduction for logistic regression often helpful
  - seldom if ever harmful
  - variance reduction not helpful in document classification
    (Most heuristics were also not helpful in binary document setting)

What Is the Role of Bias?

- Variance reduction did not help on several data sets, suggesting:
  - Bias reduction strategies could play a key part
  - Evaluated document data sets tested high for presence of bias
  - Document categorization may reflect qualities general to NLP
- Nonparametric techniques for estimating and reducing bias exist (Cohn, 1997)
- Decreasing bias could make variance reduction more powerful:
  - Use more powerful feature representations.
  - Explore more expressive models than logistic regression.


A-Optimality Variance Estimate Versus Bootstrap Estimate, TIMIT dataset
Measured at 200 Observations
bootstrap variance
predicted variance
0.028
0.0275
0.027
0.0265
0.026
0.0255
0.025
0.0245
0.0092
0.0094
0.0096
0.0098
0.01
0.0102
0.0104
0.0106
0.0108



A-Optimality Estimate of Variance Reduction Versus Bootstrap Measurement of Reduction, TIMIT dataset
Measured at 200,201 Observations
bootstrap variance reduction
predicted variance reduction
0.0002
0.00015
1e-04
5e-05
0
-5e-05
-0.0001
-0.00015
-0.0002
-0.0006
-0.0004
-0.0002
0
0.0002
0.0004
0.0006

Performance of Entropy Sampling

Entropy sampling does surprisingly poorly.

We attempt to correlate performance with “residual error” in data set.

$$\text{Residual Error} = \sum_{nc} \mathsf{E}[(y_{nc} - \mathsf{E}[c|\mathbf{x}_n])^2|\mathbf{x}_n, \mathcal{D}] + \text{Residual Bias}$$

Error defined above is training set independent error.

Approximated by training on entire pool and measuring on held out data.

Ranking of Data Sets by Residual Error

Data Set	Accuracy	Squared Error
TIMIT	0.525	0.616
ArtNoisy	0.602	0.52
LetterDB	0.764	0.352
NewsGroups	0.820	0.296
ArtConf	0.844	0.155
WebKB	0.907	0.143
Art	0.919	0.130
Comp2a	0.885	0.086
Comp2b	0.889	0.083
OptDigits	0.964	0.059

Entropy sampling underperforms on top 6 data sets in the evaluation.

Analysis of Margin Sampling

- Margin sampling fails on two data sets: ArtConf and NewsGroups
- Otherwise this method is very competitive and fast!
- These two data sets have hierarchical category structure.
- ArtConf has this property by construction.

NewsGroup Hierarchy of Topics

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey
talk.religion.misc alt.atheism soc.religion.christian	misc.forsale
sci.crypt sci.electronics sci.med sci.space	talk.politics.misc talk.politics.guns talk.politics.mideast

Suggested Improvements For Margin Sampling

- Penalize sampling of categories seen before:
  - Agglomerative clustering based on confusion matrix.
  - Sampling on higher level nodes.
- Alternative regime mixing random selection and active learning.
- Such changes should still facilitate fast margin sampling.

Results for Bagging

- Bagging is evaluated because it is essential to QBB methods.
- Bagging by itself is usually harmful to performance in evaluations.
- These results are specific to logistic regression.
- Results abound showing bagging benefits for decision trees.
- Factors effecting bagging performance in evaluation:
  - Relative stability of logistic regression compared to decision trees.
  - Small bag size used in evaluation.
  - Relatively small size of training set.

Results for QBB Methods and Classifier Certainty

- QBB-AM performance indistinguishable from margin sampling.
  - Recall, the method is defined as bagging plus margin sampling.
  - Performs badly on same two data sets as margin sampling.
- QBB-MN and Classifier Certainty underperform on two data sets.
- Hard to track sources of trouble for these latter two methods.

Summary of Evaluations

- Loss functions are most robust.
  - Only methods to consistently beat random training sets
  - These methods are also the slowest
- Of uncertainty approaches, margin sampling is preferred
  - The method only fails in well-defined circumstances.
  - It will likely be possible to improve the method.
- Alternative heuristics perform similarly
  - Performance of bagging by itself makes QBB methods suspect.
  - Use of multiple heuristics makes problems difficult to identify.

Dissertation Conclusions

- Development of Loss Function Methods:
  - These methods are most robust, but at computational cost
  - Results establish they are viable for many data sets
  - Best practice is to use either of these methods when possible
- Examination of Heuristics:
  - Identification of several settings where these methods fail
  - Identification of most promising of methods: margin sampling
  - Empirical findings suggests methods of improvement
- Identified challenging areas for active learning with heuristics:
  - learning in presence of hierarchically related categories
  - learning in presence of large residual squared error

Acknowledgments

Thanks to...

- Dissertation advisor: Lyle H. Ungar
- Members of the dissertation committee: Andreas Buja, Mark Liberman, Andrew McCallum, Fernando Pereira
- S. Ted Sandler and J. Ashley Burgoyne for help with data sets
- and many others who will be listed in the dissertation document