Bayesian Example Selection using BaBiES

Andrew I. Schein, S. Ted Sandler and Lyle H. Ungar

Department of Computer and Information Science
The University of Pennsylvania
Levine Hall
3330 Walnut Street
Philadelphia, PA 19104-6389
{ais,tsandler,ungar}@gradient.cis.upenn.edu

Abstract

Active learning is widely used to select which examples from a pool should be labeled to give best results when learning predictive models. It is, however, sometimes desirable to choose examples before any labeling or machine learning has occurred. The optimal experimental design literature has many theoretically attractive optimality criteria for example selection, but most are intractable when working with large numbers of predictive features. We present the BaBiES criterion, an approximation of Bayesian A-optimal design for linear regression using binary predictors, which is both simple and extremely fast. Empirical evaluations demonstrate that, in spite of selecting all examples prior to learning, BaBiES is competitive with standard active learning methods for a variety of document classification tasks.

1. Introduction

Recently, a large portion of machine learning literature has focused on *pool-based* scenarios where examples with classification labels are expensive to procure, but unlabeled data is abundant and inexpensive, residing in a so-called *pool.* We consider here the problem of *example selection* which attempts to gain the most prediction accuracy from machine learning algorithms while requiring the fewest number of labeled examples–thus reducing the cost of building machine learning systems. Random sampling from the pool provides a baseline for testing whether biased sampling strategies can improve machine learning accuracy.[1]

Active learning approaches to the pool based setting (Cohn et al., 1996; Lewis & Gale, 1994; Seung et al., 1992) take a trained machine learning algorithm and pick the next example from the pool for labeling according to a measure of expected benefit. Interleaving machine learning with labeling is intended to provide better measures of expected benefit since more information is available when each new example is picked for labeling. A field in statistics known as *optimal experimental design* (Fedorov, 1972; Chaloner & Verdinelli, 1995) focuses on the related problem of deciding what experimental conditions to use in order to learn the best model according to a decision theoretic cost. An Extension of the field of optimal design called sequential experimental design is analogous to pool-based active learning in that experiments are interleaved with computation that re-estimates the optimal design parameters. Nonsequential optimal experimental design has had no analogue in the machine learning literature until now.

In this paper we will take an optimality criteria from the optimal experimental design literature for a specific Bayesian model and apply it to pool-based machine learning of document topic labels. We call the approach Bayesian example selection, or more specifically, nonsequential Bayesian example selection, to contrast differences in approach to the active learning strategies. Empirical evaluation on three separate document classification domains using two separate formatting schemes demonstrate that a (nonsequential) Bayesian example selection approach can compete favorably with both active learning and random sampling methods of selecting examples.

The University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-04-08

[1] In this work all samples from the pool are taken without replacement, regardless of sampling strategy.

2. Background and Related Work

Current approaches to example selection usually formulate the problem in an *active learning* framework, where a machine learning algorithm is trained on a labeled subset of the pool and a utility measure scores pool observations for labeling. The variety of active learning algorithms differ primarily in their choice of utility functions. *Uncertainty sampling* (Lewis & Gale, 1994) takes the approach of selecting examples of least certain classification. For example, using a probabilistic classifier such as logistic regression or naive Bayes, uncertainty sampling would pick the observation who's predicted class probabilities yield the greatest entropy. The *query by committee* utility (Seung et al., 1992) measures the classification disagreement of a committee of classifiers, choosing an example with high disagreement. Cohn et al, (Cohn et al., 1996) measure the expected reduction in prediction variance of neural networks and other models, an active learning criterion that mirrors the nonsequential A-optimality method described shortly.

Selecting nonrandom examples for human labeling is just one of several competing approaches to learning in a pool-based environment. Latent variable models cast the unlabeled portion of the pool as "missing data" which can be learned along with a maximum likelihood model (Nigam et al., 2000). The co-training method (Blum & Mitchell, 1998) attempts to leverage two independent, but redundant, views of the data to grow the labeled set (using pool data) in an automatic fashion. Other instances of pool-based scenarios and proposals for pool-based learning abound and are growing in number.

The remainder of the paper is organized as follows: we describe A-optimality, a popular optimality criteria for linear regression and discuss it's applicability to common machine learning settings. Identifying computational challenges in optimizing the criteria we propose the BaBiES criteria, an approximation of A-optimality for learning in domains with sparse binary predictors. We then evaluate BaBiES using three different document classification domains using two separate data preparation schemes (leading to six evaluations in total), discuss the results and conclude.

3. Bayesian A-Optimal Example Selection

In pool-based example selection our goal is to partition a pool $\mathcal{P}$ into disjoint sets: $\mathcal{T}$, the training set and $\mathcal{R}$, the residual set to maximize machine learning model performance. Following the notation and decision theory presentation of (Chaloner & Verdinelli, 1995) we define a utility function $U(d, \beta, \mathcal{T}, \mathbf{y})$ for gauging model performance after selecting a training set $\mathcal{T}$ of fixed size. The vector β corresponds to the parameters of a trained model, y is a vector of responses for the training set, and d denotes a decision to be made after the model is trained. The size of the training set $|\mathcal{T}|$ is not given a symbol as it is assumed fixed throughout this exposition. The Bayesian solution to the example selection problem is to select:

$$\begin{aligned} U(\mathcal{T}^*) &= \max_{\mathcal{T} \subset \mathcal{P}} \int_{\mathcal{Y}} \max_{d \in \mathcal{D}} \int_{\beta} U(d, \beta, \mathcal{T}, \mathbf{y}) \qquad (1) \\ & \quad \cdot \mathsf{P}(\beta | \mathbf{y}, \mathcal{T}) \mathsf{P}(\mathbf{y} | \mathcal{T}) \, d\beta \, d\mathbf{y}. \end{aligned}$$

The intuition behind these dual expectations is that the prior over parameters allows for an estimation of response vector $\mathbf{y}$ which in turn allows us to compute expected updates for β.

Our goal is to develop example selection schemes for optimal *prediction accuracy* of a Bayesian logistic regression model in classification settings. As a first step in this paper we consider instead optimizing the prediction accuracy of the more tractable Bayesian linear regression model defined below:

$$\begin{aligned} y_j | \beta, \sigma_\epsilon^2 &\sim \mathcal{N}(x_j' \beta, \sigma_\epsilon^2 I) \text{ where} \qquad (2) \\ \beta &\sim \mathcal{N}(0, \sigma_p^2 I) \qquad (3) \end{aligned}$$

This is the standard linear regression with the additional assumption of a Gaussian prior over the parameter vector β performing the task of model shrinkage. Note the selection of two variance terms: σ_e^2 corresponding to the Bayes error rate of the model, and σ_p^2 defining the prior over the covariance matrix of the parameters. The use of a mean zero prior with isotropic variance is not necessary for the theory, but has proven useful in reducing overfitting in models with large numbers of predictors (*c.f.* ridge regression as presented in (Hastie et al., 2001)).

Given our goal of prediction error minimization we employ utility

$$U(\mathcal{T}) = -\int \left[(\beta - \hat{\beta})' A (\beta - \hat{\beta}) \right] \mathsf{P}(\mathbf{y}, \beta | \mathcal{T}) \, d\theta \, d\mathbf{y} \quad (4)$$

which is maximized when criterion:

$$\phi(\mathcal{T}) = \operatorname{tr} \left\{ A (X'X + \sigma_p^{-2} I)^{-1} \right\} \qquad (5)$$

is minimized. Equation (4) is the Bayesian A-optimality utility function for linear regression. The matrix $X'X$ is the Fisher information of the standard linear regression model, and depends on the training set only. The matrix A is a symmetric non-negative definite matrix determined by the pool as a whole. To understand the origins of the matrix A and Equation (4), note that prediction error can be decomposed into $\mathrm{Var}(x_j'\beta) + \epsilon_j$ where the ϵ_j corresponds to an irreducible error rate. The variance of the posterior β vector is given by $\sigma_e^2(X'X + \sigma_p^{-2}I)^{-1}$, and hence the prediction variance of a pool observation vector c is given by:

$$\mathrm{Var}(c'\beta) \quad = \quad \sigma_e^2 c'(X'X + \sigma_p^{-2}I)^{-1}c, \tag{6}$$

known as the $c-$optimality criterion. Note that here we ignore the effects of the irreducible error ϵ_j. Defining $A_i = x_i x_i'$ where x_i is a pool vector indexed by i, and $A = \sum_i A_i$ we derive an optimality for minimizing prediction variance over the entire pool:

$$\begin{aligned}
\sum_i \mathrm{Var}(x_i'\beta) &= \sigma_e^2 \sum_i x_i'(X'X + \sigma_p^{-2}I)^{-1}x_i & (7)\\
&= \sigma_e^2 \sum_i \mathrm{tr}\left\{A_i(X'X + \sigma_p^{-2}I)^{-1}\right\} & (8)\\
&= \sigma_e^2 \,\mathrm{tr}\left\{A(X'X + \sigma_p^{-2}I)^{-1}\right\} & (9)\\
&= \int \mathrm{tr}\left\{A(\beta - \hat{\beta})(\beta - \hat{\beta})'\right\} & (10)\\
&\quad \cdot \mathsf{P}(\mathbf{y}, \beta|\mathcal{T})\, d\theta\, d\mathbf{y} & \\
&= -1 \cdot \text{Equation (4)}. & (11)
\end{aligned}$$

4. The BaBiES Criterion

A challenge in applying the optimality criteria ϕ (Equation 5) to data sets with large numbers of predictors, such as document classification, is computing the inverse: $(X'X + \sigma_p^{-2}I)^{-1}$, an operation that is nearly cubic in the number of predictors. In pilot studies we found that by 2000 predictors, the inverse was too expensive to compute on a Pentium III computer when employed inside a greedy example selection algorithm. The numbers of predictors used in our empirical evaluation range from 3466 to 7543, and ideally we would like to work on domains with even greater numbers of predictors.

We present a heuristic approximation to computing ϕ (Equation 5) named BaBiES as an acronym for: **Ba**yesian **Bi**nary **E**xample **S**election. The BaBiES approximation applies in the special case where X is both sparse and binary, and the model to be learned is a Bayesian linear regression with a bias term. In this case the matrix $X'X$ is also sparse, with the individual $(X'X)_{ij}$ taking on the the number of times features i and j co-occur, and the diagonal values $(X'X)_{ii}$ consisting of the total number of times feature i occurs. We propose approximating the matrix $X'X$ using its diagonal, last row, and last column.[2] The approximation allows rewriting of Equations (5) in terms of pool counts P_w and training set feature counts T_w describing the number of times a predictor w occurs in the pool and training set respectively. The resulting objective function to minimize is:

$$\mathrm{BaBiES}(\mathcal{T}) \quad = \quad \sum_w \frac{P_w}{T_w + \frac{1}{3\sigma_p^2}}, \tag{12}$$

where the sum w is over predictors (*i.e.* word tokens in a document classification domain) rather than observations. Unlike the A-optimality objective ϕ, the BaBiES utility can be computed quickly, making it feasible to apply a greedy algorithm. The algorithm we use in evaluation, Greedy-BaBiES is outlined in Figure 1.

```
Inputs: $\mathcal{T}$ a set of seed training instances,
$k$ the desired final size $|\mathcal{T}|$
Output: The updated $\mathcal{T}$
while $|\mathcal{T}| < k$
  Select the example $x_i$ from the pool that reduces
  (12) the most.
  Add this example to $\mathcal{T}$, removing it from the pool
  (without updating the counts $P_w$).
return $\mathcal{T}$
```

Figure 1. The Greedy BaBiES Algorithm

Though we have derived and justified BaBiES using A-optimality as a starting point, there are several useful intuitions embedded within Equation 12 that are worth elucidating. Given a setting where binary predictors are independent, but occur with different marginal frequencies, there are two competing objectives that are attractive in sampling observations for machine learning training. First, we desire a training set from which to learn accurate parameters for the most common predictors since these predictors will be extremely valuable if they prove predictive of the response variable. Second, we hope to learn accurate parameters for as many predictors as possible since this will increase the probability of having one or more pieces of evidence to use in making a prediction for a test set observation. The BaBiES criterion captures both of these *desideratum:* the prior σ_p^2 determines the benefit of seeing a predictor for the first time in the training set, the numerator term gives high frequency predictors greater priority, while the fractional elements represent a diminishing returns in seeing a predictor more than once.

[2] The last row and column of $X'X$ encodes the bias term, which will always take the value '1'.

5. Evaluation

We evaluate the greedy BaBiES algorithm in the document classification domain, comparing it against various example selection strategies: uncertainty sampling (Lewis & Gale, 1994), query by committee (Seung et al., 1992), "maximum document-length" which adds the longest document to the training set (ties are broken randomly), and simple random selection, which serves as our baseline. We use a variant of query by committee described in (McCallum & Nigam, 1998) where committee disagreement is expressed in terms of KL-divergence from the mean.

We compare these selection techniques on three standard data sets for document classification: Ken Lang's 20-Newsgroups (Joachims, 1997), the WebKB collection (Craven et al., 1998), and Reuters-21578 (Lewis, 2003) corpus. For the Newsgroups data, we restrict our evaluation to the Comp.* subset, consisting of the five computer-related topics from the newsgroups hierarchy.

The WebKB data set consists of web pages culled from computer science department websites at various universities. The classification task is to predict whether a web page is a student, faculty, course, or project page. Finally, the Reuters data set consists of news feeds from the Reuters-21578 corpus, limited to those articles that are labeled with "earnings" and/or "acquisition" topics (the two most prevalent topics). For Reuters, the task is to predict whether an article should be labeled with the earnings topic.

All data sets are tokenized on consecutive alphabetic characters and consecutive numeric characters. Alphabetic strings are lowercased and numerical strings mapped to the special token "N." A count cut-off of five occurrences is used, so that a word-type must occur at least 5 times in the data set in order to be included in the vocabulary. Stop words are filtered from the vocabulary using using the Rainbow stoplist (McCallum, 1996). In the case of the Comp.* data sets, ascii-encoded binary data was manually separated from the articles. Each document was encoded as a binary-valued vector where each vector component encoded whether a specific word-type occurred in the document.

To better understand the impact that document length has on BaBiES and the other example selection algorithms, we create truncated data sets for each of the data sets listed above. The truncated sets are prepared analogously to the untruncated ones; only documents are restricted to twenty-five tokens in length (the first 25 unique tokens) and any document with fewer than twenty-five unique tokens is discarded. The purpose of the truncated document sets is to fully account for document length characteristics in the evaluation since document classification has the property that different observations (documents) have different numbers of distinct words. Other domains do not have this property and so we isolate length as a factor influencing performance of example selection strategies.

The classifier used in evaluation was a Bayesian logistic regression with a Gaussian shrinkage prior set to 1.0:

$$\begin{aligned} \mathsf{P}(y_j = c_i | x_j, \beta_i) &= \frac{\sigma(x_j' \beta_i)}{\sum_k \sigma(x_j' \beta_k)} \quad \text{where} && (13) \\ \sigma(\theta) &= \frac{1}{1+\exp(-\theta)} \quad \text{and} && (14) \\ \beta_i &\sim \mathcal{N}(0, I), \quad \forall i. && (15) \end{aligned}$$

The indices i, k range over categories and j indexes an observation.

The BaBiES prior σ_p^2 was set to 1.0. The committee size was set to five for query-by-committee.

In each of twenty trials, the data was randomly divided so that half the documents were held out for evaluation and the rest reserved for the pool. Twenty documents were pulled from the pool at random and labeled to create the initial training set. Using this training set as a seed, the performance of a particular example selection algorithm would be gauged with the following iterative procedure:

Step 1: the accuracy of the classifier trained on the current training set is measured and recorded.

Step 2: the selection algorithm at hand is given access to the training set, the pool—and in the case of uncertainty sampling and query-by-committee, one or more classifiers trained from the current training set. Using this information to inform its choice, the selection algorithm is then allowed to request that a particular document from the pool be labeled and added to the training set.

Step 3: if the total number of documents in the training set is less than 100× the number of classes, steps 1 and 2 are repeated.

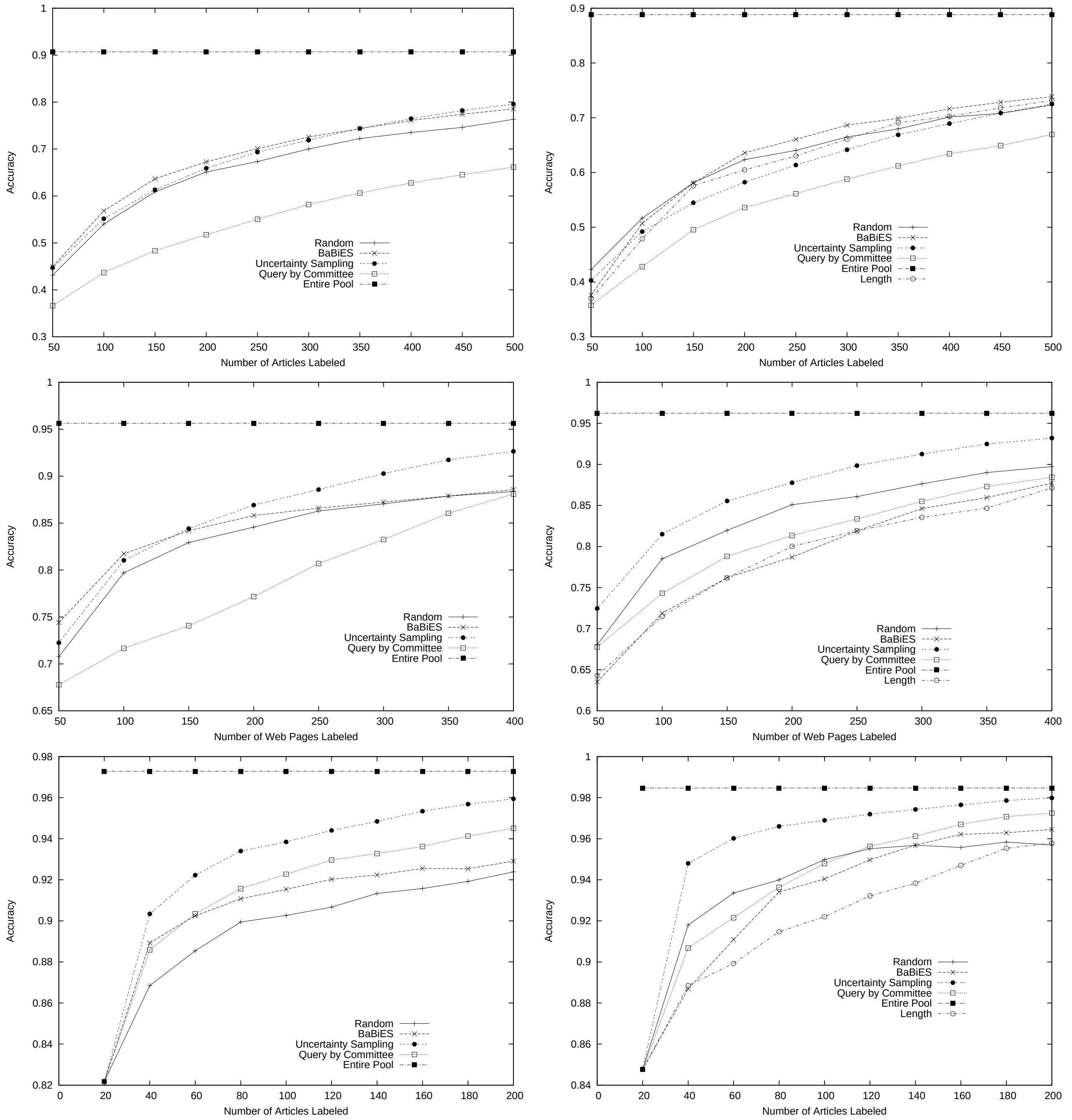

Figure 2. Evaluation accuracy on truncated (left column) and untruncated (right column) data sets. By row, from top to bottom the data sets are Comp.*, WebKB, and Reuters. Pool sizes for these experiments are equal to 1/2 the data set sizes given in Table 1.

Table 1. Statistics of the data sets used in evaluation include the number of categories (Classes), the number of observations in the data set before splitting into pool and test sets (Obs), the resulting number of observations after truncating documents to 25 unique tokens (Trun Obs), and the the number of tokens after applying the count cut-off (Toks). In the case of truncated documents the cut-off is applied only once: before the truncation has occurred.

	Classes	Obs	Trun Obs	Toks
Comp.*	5	4980	3592	7486
WebKB	4	4169	3730	7543
Reuters	2	5969	2745	3466

6. Experimental Results

Figure 2 shows performance results for random example selection, BaBiES, uncertainty sampling, query by committee, length (on the untruncated documents), and using the entire pool. When using the entire pool, the number of observations is fixed at a constant equal to one half of the data set size given in Table 1. Performance is measured by accuracy: the proportion of documents correctly labeled by the trained model, and measurements are taken over increments of the training set size $|\mathcal{T}|$. The accuracies reported are averages over twenty runs, and the standard deviations generally started at 0.06 early in the curves and went as low as 0.01 as more documents were added to the training set.

BaBiES performed significantly above random on the untruncated Comp.* and all of the truncated documents, with performance improvements in the last 60 added documents of the untruncated Reuters data set. In comparing the performance of BaBiES against picking the longest documents we see that BaBiES outperforms the length utility heuristic, demonstrating that BaBiES does not simply pick the longest document. However, close examination reveals that BaBiES does follow the performance trends of length on the untruncated documents succeeding when length does and performing less well when length fails to yield above-random results. These results suggest a modified BaBiES that accounts for length.

BaBiES approximates inverting the Fisher information matrix $X'X$ using diagonal information, and it would be interesting to determine how well BaBiES performs in comparison to different amounts and types of off-diagonal structure in $X'X$. At one extreme, we have the theoretical result of a diagonal Fisher information matrix if the predictors are mean centered and independent, since in this case the Fisher information matrix is proportional to the covariance matrix of X. However, for our data sets the predictors are not mean centered, and the Fisher information matrix can not be of full rank since the number of observations is far less than the number of predictors, for all data sets.

Table 2. Squared error of data set reconstruction using a number of principal components equal to 1% of the original number of unique tokens used in prediction. The numbers below are for untruncated documents.

Data Set	Comp.*	WebKB	Reuters
Squared Error	0.683	0.772	0.692

Table 2 attempts to answer the question of how correlation among the predictors affects the approximation by taking the (untruncated) data sets, projecting each observation into a number of principal components of X equal to 1% of the number of predictors (the number of predictors is shown in Table 1), and measuring the error induced in reconstructing the original data matrix X.

The relative orderings and magnitudes of the reconstruction error rates in Table 2 correlate with the performance of BaBiES, with WebKB having by far the greatest reconstruction error rate and poorest performance. These results suggest that BaBiES criterion is best suited to document classification data sets when the correlation structure among the predictors is easily modeled with fewer principal components. Experiments revealing negative BaBiES results on Reuters using an all-against-one evaluation predicting "earnings" articles among 135 Reuters topics confirm this finding (plots not shown).

Document truncation had a large impact on most of the methods evaluated. For BaBiES, document truncation led to performance improvements in every data set. Uncertainty sampling performs below random on the untruncated Comp.* data set, but above random when the data set is truncated. We conjecture that the untruncated version of the Comp.* set contains a region of high uncertainty with no utility to the trained model. Query by committee seemed the least affected by truncation, but also had the worst overall performance, failing to give above-random performance on all Comp.* and WebKB evaluations.

7. Summary

We proposed a new approach to selecting examples from a pool for training machine learning algorithms. The methodology can select examples for maximizing prediction performance before any human labeling or machine learning has occurred. The key ingredient of our method is an optimality criteria that describes with some accuracy the expected error induced by a particular training set, providing a means for greedy optimization. Since common optimality criteria in the experimental design literature are computationally intractable for high-dimensional observations, we develop BaBiES, a simple approximation which performs remarkably well on a variety of tasks. We expect that with more appropriate optimality criteria, better approximations, and numerical optimizations, example selection performance and robustness will improve.

References

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers.*

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, *Vol. 10, No. 3*, 273–304.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence*, *4*, 129–145.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A. K., Mitchell, T. M., Nigam, K., & Slattery, S. (1998). Learning to extract symbolic knowledge from the World Wide Web. *Proceedings of AAAI-98, 15th Conference of the American Association for Artificial Intelligence* (pp. 509–516). Madison, US: AAAI Press, Menlo Park, US.

Fedorov, V. V. (1972). *Theory of optimal experiments.* Academic Press, New York.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning.* Springer Series in Statistics. New York, NY: Springer-Verlag.

Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. *Proceedings of ICML-97, 14th International Conference on Machine Learning* (pp. 143–151). Nashville, US: Morgan Kaufmann Publishers, San Francisco, US.

Lewis, D. D. (2003). Reuters-21578 text categorization test collection distribution 1.0. http://www.daviddlewis.com/resources/ testcollections/reuters21578/.

Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval* (pp. 3–12). Dublin, IE: Springer Verlag, Heidelberg, DE.

McCallum, A. K. (1996). Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow.

McCallum, A. K., & Nigam, K. (1998). Employing EM in pool-based active learning for text classification. *Proceedings of ICML-98, 15th International Conference on Machine Learning* (pp. 350–358). Madison, US: Morgan Kaufmann Publishers, San Francisco, US.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*, 103–134.

Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. *Proceeedings of the Fifth Annual Workshop on Computational Learning Theory* (pp. 287–294).