ACTIVE LEARNING FOR LOGISTIC REGRESSION

Andrew Ian Schein

A DISSERTATION PROPOSAL

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2004

Lyle H. Ungar
Supervisor of Dissertation

Benjamin C. Pierce
Graduate Group Chairperson

ABSTRACT

ACTIVE LEARNING FOR LOGISTIC REGRESSION

Andrew Ian Schein

Supervisor: Lyle H. Ungar

In this dissertation, we develop and argue for principled experimental design approaches to active learning, specifically the $A-$optimality criterion applied to the logistic regression model. We claim that such approaches give greater accuracy and robustness than the currently popular heuristic methods of active learning. In the course of the dissertation we will develop the optimizations and techniques necessary to apply logistic regression A-optimality on large data sets. We will perform an evaluation of A-optimality on a variety of tasks, with special emphasis on document classification, in order to substantiate the method's attractive characteristics and demonstrate the scalability of the method in practice.

Our research is a refreshing contrast to a trend in empirical machine learning which foregoes the complex objective functions of experimental design for the lure of easily mastered heuristic methods. It is our experience that heuristic methods are unsuitable for a wide variety of data sets, often giving performance that is well-below that of random sampling. We intend to address situations where heuristic active learning fails using experimental design methods, and in doing so demonstrate the robustness of our approach.

A-optimality is general to a wide variety of methods including linear regression and multilayer backpropagation neural networks. The method is also applicable to the Markov random field class of models, and in this dissertation we derive special cases for binary and multinomial logistic regression classifiers in addition to the maximum entropy classifier. A-optimality has been applied to small data sets in the past, but in this dissertation we develop optimizations for computing A-optimality on the sort of large data sets common in modern machine learning applications. We hope that establishing the effectiveness of the method for probability models such as logistic regression will pave the way for robust and well-motivated active learning strategies for more complex “joint classification” problems such as part of speech tagging and parsing.

COPYRIGHT

Andrew Ian Schein

2004

Contents

Chapter 1

Introduction

In this dissertation, we develop and argue for experimental design approaches to active learning, particularly the $A-$optimality criterion applied to the logistic regression model. We claim that such approaches will give greater accuracy and robustness than the currently popular heuristic methods of active learning. In the course of the dissertation we will develop the optimizations and techniques necessary to apply logistic regression A-optimality on large data sets. We will perform an evaluation of A-optimality on a variety of tasks, with special emphasis on document classification, in order to substantiate the method's attractive characteristics and demonstrate the scalability of the method in practice.

1.1 Active Learning: a Definition

The last fifteen years of machine learning research has produced a volume of literature on *pool-based active learning* methods, where instead of receiving an i.i.d. sample from some underlying distribution, a learning algorithm may take an active role in selecting which examples from a pool of unlabeled data are labeled and added to the learner's training set. The pool is a bag of i.i.d. examples which may be sampled without replacement. The term pool-based active learning is used to distinguish this important type of active learning from other forms of active learning including methods that construct examples from R^n or other sets from first principles. Henceforth we will often use the term active learning to refer to pool-based active learning; since the dissertation does not treat the other forms, no confusion will arise.

The purpose of developing active learning methods is to achieve the best possible generalization error at the least cost, where cost is usually measured as a function of the number of examples labeled. Frequently we plot the tradeoff between number of examples labeled and generalization error through learning curves of the type introduced in Chapter 2. It is commonly believed that there should exist active learning methods that perform at least as well as random sampling from a pool on average, and should in most circumstances outperform random sampling. This belief is given theoretical justification under very specific assumptions [39, 18], but is also frequently contradicted by empirical evaluations.

1.2 Why Active Learning Is Hard

Active learning is hard because random sampling from the pool provides a very competitive baseline. As a rule of thumb, the generalization error rate of a machine learning algorithm decreases according to:

$$E_{\text{test}} = a + \frac{b}{n^{\alpha}} \tag{1.1}$$

where n is the training set size, and a, b and α depend on the task and learning algorithm [16, Chapter 9: page 492]. In the special case where the response variable is generated by a linear model (e.g. linear regression is the machine learning algorithm of choice, and all of its assumptions are satisfied) the generalization error over the pool can be derived:

$$E_{\text{test}} = a + \frac{a \cdot r}{n} \tag{1.2}$$

where E_{test} is the squared error, a is the "noise" of the task, and r is the rank of the pool data design matrix. A derivation of Equation 1.2 will be provided in the

Appendix of the final dissertation. For arbitrary machine learning algorithms there are usually no derived close form learning curves, although it is likely that for a broad array of the most popular learning algorithms α in Equation 1.1 is close to 1, given the trend in large-scale empirical evaluations which indicates that most algorithms perform similarly when compared on a broad array of tasks.

The very attractive baseline provided by random sampling from the pool is the primary challenge that active learning methods must overcome to justify their use. When labeling examples, an annotator can choose to select randomly from the pool and get a guaranteed expected generalization error decrease of: $O(\frac{1}{n^{\alpha}})$. In order for active learning to be accepted in industrial applications we must guarantee that the performance will offset the cost of implementing this biased sampling scheme and retraining the machine learning algorithm repeatedly. Particularly daunting is that active learning is most useful when applied to a new domain where we have few examples. In a new domain, we have little guarantee that heuristics that worked in the past will work again without tuning and tweaking.

1.3 A Recent Trend in Active Learning

There has been growing interest in active learning techniques and their application to actual data sets. A trend of the last ten years [1, 2, 14, 31, 22, 25, 30, 34, 44] has been to employ heuristic methods of active learning with no explicitly defined objective function. Uncertainty sampling [25], query by committee [39][1], and variants have proven particularly attractive because of their portability across a wide spectrum of machine learning algorithms.

It is our own experience that heuristic methods of active learning perform badly

[1] Query by Committee is a method with strong theoretical properties under limited circumstances [39, 18], but the overwhelming trend has been to apply the method in circumstances where the theory does not apply. Throughout the dissertation we will refer to query by committee as a heuristic method, leaving implicit the caveat that we refer to its use in circumstances where the appropriate theorems do not apply. Further discussion is given in Chapter 2.

on a wide variety of data sets, and that these methods do not measure up to the standards necessary to make active learning suitable for industrial applications. Further, it has been our experience in reproducing published results in the field that a wide variety of alternative evaluation decisions lead to weaker and occasionally worse than random performance of active learning methods. In Chapter 3 we show the even more surprising result that very well-trained models can diverge when started on a regime of heuristic active learning methods. What we learned from these experiments is that active learning is by no means a "solved problem," and that more robust methods, when identified, will be acknowledged as a much-desired contribution to the field.

1.4 Thesis

The purpose of this dissertation is to develop the experimental design methods, particularly the A-optimality theory for logistic regression, with the intention of demonstrating robustness on real-data in comparison to heuristic approaches for active learning. The biggest impediments to this aim are the issues of applying complex matrix operations to large data sets. We will develop the necessary optimizations to implement active learning on larger data sets than previously attempted with any A-optimality method. The technical achievements of the dissertation will facilitate a large scale evaluation which we expect will support our thesis that experimental design methods provide robustness advantages over heuristic active learning.

We hope to evaluate our method on a wide variety of data sets, but focus especially on document classification for several reasons. First, these data sets have reasonably "long" learning curves with palpable progress to be made even after 200 random observations have been sampled from the pool. Second, these document classification tasks can be viewed a precursor for other more expensive training set procurement problems in natural language processing. Part of speech and syntactic annotation are two examples where active learning is much desired [14, 22]. Finally, it is our belief that the majority of data sets from the commonly-used UC Machine Learning Irvine repository are not interesting in evaluation, because the learning rate for many of these data sets is so fast without any active learning, and the small size of many of the domains do not leave room for the "tricky" heteroscedastic noise structure that we show is so damaging to heuristic active learning methods (Chapter 3).

1.5 *A*-Optimality Explained

Logistic regression is a method that assigns probabilities to the class labels of observations. We choose as our objective for active learning of logistic regression the minimization of the following function:

$$\sum_{n,c} \mathrm{Var}\left[\hat{\mathsf{P}}(c|x_n)\right] = \sum_{n,c} \mathsf{E}\left[\hat{\mathsf{P}}(c|x_n) - \mathsf{P}(c|x_n)\right]^2, \tag{1.3}$$

where n indexes pool observations, c indexes the different categories of the classification task, P is the "true model" (defined shortly) and $\hat{\mathsf{P}}$ is our estimated probability model. We refer to this quantity as the prediction variance of the model measured over the pool. Algebraic manipulations of this quantity lead to the set of equations known as A-optimality. The allure of the method is that the squared difference between the limiting behavior of the model under i.i.d. sampling from an infinite pool $\mathsf{P}(\cdot|x)$, and the current model trained on a finite sample from the pool $\hat{\mathsf{P}}(\cdot|x)$ may be estimated using asymptotically correct approximations, allowing for a good estimate of model variance. This is what we mean when we say that A-optimality and other experimental design methods have an explicit objective function from which the criteria are derived.

1.6 The Thesis in Context

The last five years of active learning papers are mostly filled with proposals for new heuristic methods and evaluations of these methods. We intend to demonstrate that in situations where currently popular heuristics fail, experimental design, and A-optimality in particular, give robust performance.

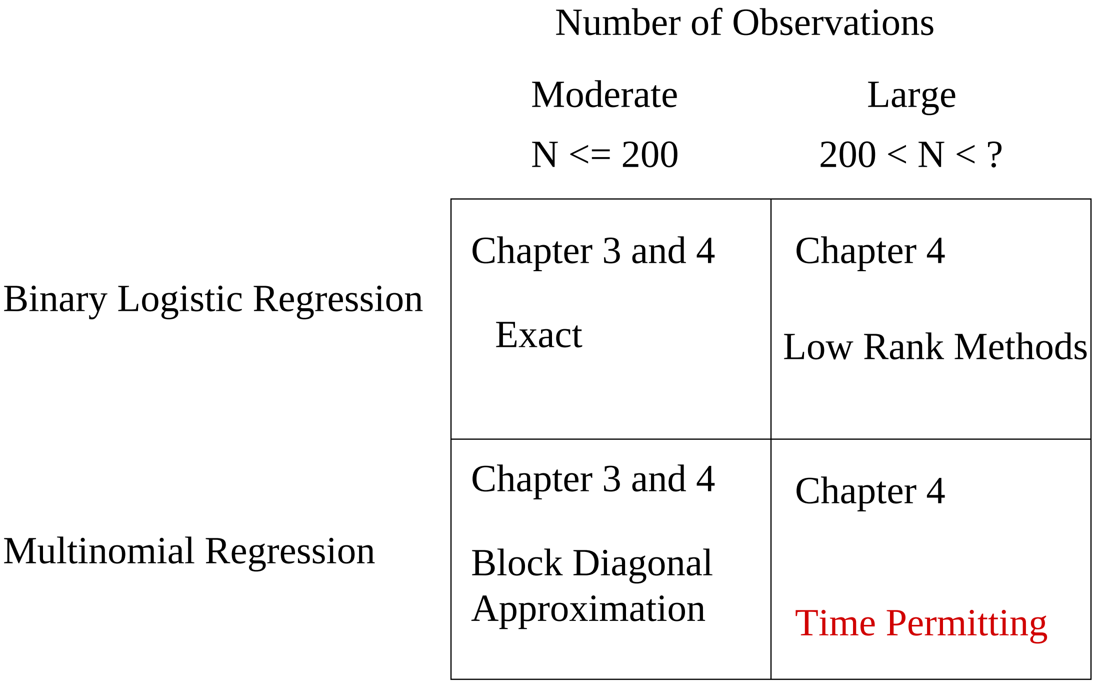
A-optimality has been explored previously in experimental design [9] as well as active learning of multilayer backpropagation neural networks [27, 11]. In experimental design the method has been applied to training logistic regression classifiers with a single predictive feature [8, 15]. Discussions with an expert in experimental design [48] confirmed our suspicion that the method has not been used in designing logistic regression experiments with more than one predictor, or at least that such work is not widely known. We conjecture that the reason for this state of affairs is that the experimental design community is concerned with answering questions in settings where it is believed that the model accurately reflect an underlying natural process. Such a belief regarding the veracity of a model form is hard to justify for logistic regression models with more than one variable. Chapter 3 describes one experiment type (with a single predictive feature) where A-optimality has been used for logistic regression. Our review of experimental design as a whole has lead us to believe the community focuses empirical evaluation on settings with very few predictors, regardless of the type of model employed. In contrast, our goal is to apply the technique in cases where the number of predictors is quite large.

Our current direction differs from the previous use of A-optimality in active learning of multilayer neural networks in both the underlying machine learning algorithm and (again) in the size of the data sets used in evaluation. Evaluations of A-optimality and backpropagation neural networks on real (as opposed to artificially generated data sets) occurred only once that we could find [11], and this data set size was relatively small by current standards. Discussions with an expert in application of A-optimality to backpropagation neural networks confirmed our conclusion that the method has not been evaluated on natural data more than once [13], or at least other evaluations are not well known. In contrast, our evaluations will be on many data sets, and these data sets will be significantly larger in both the number of observations in the training set and the number of predictive features used in the model.

Making A-optimality work on large data sets is a challenge, as we shall see throughout the dissertation. It is also quite likely the reason that the active learning community has seized on heuristic active learning techniques, most of which are actually a much newer in origin than the original proposals to use experimental design with backpropagation neural networks [27, 11]. One particular source of complexity in computing A-optimality is the number of predictive features D. Computing A-optimality takes $O(D^3)$ when done naively, and we will need this operation to occur in seconds rather than hours to perform evaluations in domains such as document classification.

1.7 Outline of Dissertation

The development of our techniques for applying A-optimality to real data set will proceed to cover the four blocks of Figure 1.1 and culminate in ever-more inclusive empirical evaluations. We begin by deriving the A-optimality equations for binary logistic regression and multinomial regression. We then proceed to apply various methods of computing the criterion. Using the Sherman-Woodbury-Morrison formula, we believe exact computation computation of the criterion should be practical when the training set size is small, for instance less than 200. This strategy is indicated by the left column of the Figure 1.1 diagram. As the training set gets bigger we propose using low rank approximations to the Fisher information matrix. We have yet to discover how much advantage low rank approximations will afford, hence the question mark in the right column of the Figure 1.1 diagram..



	Number of Observations	
	Moderate	Large
	N <= 200	200 < N < ?
Binary Logistic Regression	Chapter 3 and 4 Exact	Chapter 4 Low Rank Methods
Multinomial Regression	Chapter 3 and 4 Block Diagonal Approximation	Chapter 4 Time Permitting

Figure 1.1: The plan for attacking the methodological challenges of applying A-optimality to real data sets. In the left column we propose techniques that allow exact computation for A-optimality when the training set size is small. The right column represents the situation where the training set size is moderate, in which case we propose low rank approximations to the Fisher information matrix. The different rows represent our desire to find techniques for both logistic and multinomial regression.

The outline of the dissertation follows.

1. An Introduction

2. A Review of Active Learning Literature

   We review the different methods for pool-based active learning focusing especially on methods applicable to logistic regression.

3. Derivation of A-Optimality Criterion For Logistic Regression

   We motivate and derive the A-optimality criterion for active learning of both binary and multinomial logistic regression classifiers. We perform preliminary evaluations on small data sets for the binary case. In the final dissertation we will include comments on the Markov random field class of models as a whole.

4. Scaling Up Techniques for A-Optimality

   This chapter discusses the various techniques that make A-optimality for logistic regression tractable to run on large data sets. We explore the Sherman-Morrison-Woodbury formula as a method for when the number of training examples is relatively small ($\leq$ 200), but the number of predictors is potentially vast. We also explore the possibility of reduced rank approximations computed efficiently by exploiting special structure common to document classification and other natural language processing domains. The highlight of this chapter will be a set of evaluations that prove the methods are viable.

5. Evaluation

   In the final dissertation there will be a large-scale evaluation chapter.

6. Conclusions

1.8 Summary of Completed Work

To date, we have completed a derivation of the A-optimality criterion for binary and multinomial logistic regression and evaluated the binary case on small data sets. The results for binary logistic regression are detailed in Chapter 3 and in our submitted publication [36].

1.9 Timeline for Completion

We propose to complete all proposed work and defend the dissertation within approximately twelve months.

Chapter 2

Pool-Based Active Learning: A Review

In this chapter we introduce some of the core algorithms and concepts from the pool-based active learning and experimental design areas. Our main focus is on classification problems with noise, and active learning methods that can be used with logistic regression. We also touch on developments for linear regression and support vector machines in order to introduce some of the important concepts from the field of active learning as a whole.

Here, we fully develop the theory of A-optimality and give historical perspective on where it has been derived and how it has been applied in active learning. We demonstrate through literature summary that A-optimality has been derived for backpropagation neural networks, but empirical verification of the method in active learning applications using any machine learning algorithm on real data sets is lacking, with only one known evaluation on a natural (*i.e.* non-artificial) data set. We explicitly detail characteristics that delineate our present contribution from past research using A-optimality and other experimental design methods. We also give an overview of the most popular active learning methods, uncertainty sampling and query by committee, and note the various ways in which these two methods have been applied.

Following a convention that has developed in the active learning field we divide the "classical" active learning approaches of the early to mid 1990s into "objective function" and "heuristic" (or "algorithm independent") methods. The objective function methods include experimental design methods such as A, D, and $c-$optimality. The heuristic methods include uncertainty sampling and query by committee. In actuality, the line between having an explicit objective function and a heuristic can be blurred as heuristic approximations to objective functions are made for the benefit of expediency. An alternative view is that a heuristic approach is actually an objective function approach whose assumptions have not yet been exposed.

To begin, we need to define the term "design" and "design matrix" as used in the field of optimal experimental design. For regression experiments the design is encoded as the matrix X whose rows are the design point vectors x_n. The design matrix columns encode the independent variables (statistics terminology), predictors (alternative statistics terminology), or features (terminology prevalent among some machine learning researchers) while the design matrix rows encode separate observations. Implicitly the matrix X encodes a distribution η over x vectors: $\mathsf{P}_\eta(x)$. The response variables y (also termed "outcomes", "class labels", etc.) are not included in the design since they are not known until after an experiment is complete. In the case where some data is initially labeled, this knowledge can be encoded either as a prior over the parameters or by augmenting the notation to include known and unknown responses.

2.1 Objective Function Approaches

Objective function active learning methods such as D, c, and A-optimality explicitly quantify the differences between an ideal classifier and the currently learned model in terms of a loss function. Borrowing notation from Roy and McCallum [34] for the special case where the learning algorithm outputs a probability distribution, we may represent the objective function abstractly as:

$$E_{\mathsf{P}_{\mathcal{D}}} = \int_x L(\mathsf{P}(y|x), \hat{\mathsf{P}}_{\mathcal{D}}(y|x))\mathsf{P}(x), \tag{2.1}$$

where L is a loss function, $\mathsf{P}(y|x)$ are the probabilities associated with a model trained on the entire pool, and $\hat{\mathsf{P}}_{\mathcal{D}}(y|x)$ are the probabilities of a model trained on a partial representation of the pool where observations (x, y) follow distribution $\mathcal{D}$. Example loss functions for this case include cross entropy and squared error.

In many settings a model outputs something other than a probability, such as a real value, in which case the notation would need to be altered:

$$E_{\mathsf{P}_{\mathcal{D}}} = \int_x L(y(x;\beta), y(x;\hat{\beta}))\mathsf{P}(x) \tag{2.2}$$

where β and $\hat{\beta}$ are the true and estimated parameters estimates respectively. One such loss function is squared error.

2.1.1 A-Optimality for Linear Regression Models

To maintain chronological accuracy and develop the requisite algebraic methodology, we start with the classic design criteria of linear regression [17], with the familiar model of the data given by a Gaussian with isotropic noise model:

$$\mathbf{y}|_{\beta,\sigma^2} \sim \mathcal{N}(\beta' X, \sigma^2 I). \tag{2.3}$$

X is the design matrix, and its rows consist of the the predictors of the model. The vector β is the parameter vector of the model. Let x_n denote the (column) vector formed from the nth row of X. The maximum likelihood solution is equivalent to the least squares solution:

$$\arg\min_{\beta} \sum_n (y_n - \beta \cdot x_n)^2 \tag{2.4}$$

$$\hat{\beta} = (X'X)^{-1}X'\mathbf{y}. \tag{2.5}$$

The matrix $X'X$ is the observed Fisher information matrix of the linear regression.

The model is frequently regularized:

$$\arg\min_{\beta} \quad \sum_n (y_n - \beta \cdot x_n)^2 + \frac{1}{2\sigma_p^2}||\beta||^2 \tag{2.6}$$

$$\hat{\beta} = (X'X + \frac{1}{\sigma_p^2}I)^{-1}X'\mathbf{y} \tag{2.7}$$

in which case the Fisher information matrix becomes: $(X'X + \frac{1}{\sigma_p^2}I)$. The regularized variant is equivalent to a Bayesian linear regression where equation 2.3 is augmented with the assumption:

$$\beta \sim \mathcal{N}(\mathbf{0}, \sigma_p^2 I), \tag{2.8}$$

where the p in σ_p stands for "prior" to draw attention to the fact that it is a different parameter from the error variance σ^2 of Equation 2.3. We present linear regression in both Bayesian and non-Bayesian to show the duality and also, in the Bayesian case, to apply the decision theory framework in section 2.1.7.

Having defined the model of interest, linear regression, we contemplate now what objective function to apply obtaining good prediction accuracy. A large part of the experimental design literature has focused on two types of experimental goals: extremum performance and model identification problems. This dissertation is concerned with the quality of predictions over the pool where the pool is taken as an accurate representation of the distribution of a final test set. Therefore we are concerned with an extremum problem: minimizing an expected loss computed over the pool.

For linear regression, an experimental design objective function often used to obtain good prediction accuracy is:

$$\mathrm{Var}(\hat{y}) = \mathsf{E}[y(x, \hat{\beta}) - y(x, \beta)]^2. \tag{2.9}$$

That is, we minimize the difference between the current model and the actual model according to the squared loss. This quantity is known as the prediction variance of the model, and it is computed over the pool. Some facts we will need in deriving an optimality criteria from this objective function follow. First let $\hat{\beta}$ be the maximum likelihood (and therefore least squares) parameters for linear regression. Then $\hat{\beta} \sim \mathcal{N}(\beta, F^{-1})$, where $\beta,\ \mu$ are both vectors, and F is the Fisher information matrix [38], and $\mathrm{Var}(\hat{\beta}'x) = xF^{-1}x$ as a consequence of normality of $\hat{\beta}$. With this result in hand, we derive the loss incurred by making predictions over the pool of unlabeled data using current maximum likelihood values for the parameters β. We define $A_n = x_n x_n'$, $A = \sum_n A_n$ and compute:

$$\begin{aligned}
\sum_{n \in \text{Pool}} \mathrm{Var}(x_n' \hat{\beta}) &= \sum_{n \in \text{Pool}} x_n F^{-1} x_n \quad \text{by Normality} && (2.10) \\
&= \sum_{n \in \text{Pool}} \mathrm{tr}\left\{ x_n x_n' F^{-1} \right\} && (2.11) \\
&= \sum_{n \in \text{Pool}} \mathrm{tr}\left\{ A_n F^{-1} \right\} && (2.12) \\
&= \mathrm{tr}\left\{ A F^{-1} \right\}. && (2.13)
\end{aligned}$$

Equation 2.13 is referred to as A-optimality due to the A matrix that gives the method its name. Equation 2.10 is referred to as c-optimality; when the vectors x_n are renamed c_n, the naming convention becomes more apparent.

Before moving on, we give a formal definition of the Fisher information matrix computed over a likelihood function f:

$$I(\theta)_{ij} = -\mathsf{E}\left[\frac{\partial^2 \ln f(X|\theta)}{\partial \theta_i \partial \theta_j} \right]. \qquad (2.14)$$

For expediency, we will frequently denote the matrix $I(\theta)$ as F, making implicit the dependence on the parameters θ.

2.1.2 D-Optimality for Linear Regression Models

D-optimality concerns the model identification objective of designing experiments. Though our focus in this dissertation with classification accuracy means A rather than D-optimality is our primary focus, the reader will benefit from knowledge of this very popular experimental design criterion in placing the current active learning approaches in context. Furthermore, there are applications of active learning objective functions that follow in the spirit of D-optimality [46] so having a definition of D-optimality will help identify this trend and note its difference from the A-optimality "prediction variance" approach.

Since the maximum likelihood parameters $\hat{\beta}$ of a linear regression follows a normal distribution $\hat{\beta} \sim \mathcal{N}(\beta, F^{-1})$ [38], we may write out the distribution over parameters:

$$\mathsf{P}(\hat{\beta}|\beta, X, \sigma_p^2) \quad = \quad \left(\frac{1}{2\pi}\right)^{d/2} \frac{1}{\sqrt{|F^{-1}|}} \exp\left\{-\frac{1}{2}(\hat{\beta}-\beta)' F (\hat{\beta}-\beta)\right\} \tag{2.15}$$

A measure of parameter variance is given by the determinant: $|F|$, the inverse of volume of the parallelepiped encoded by the rows of the Fisher information matrix. Maximizing this determinant gives the D-optimality criterion. The D in the name D-optimality comes from "determinant."

In the Bayesian setting, we may derive the D-optimality criterion through the Shannon information measure of model uncertainty:

$$\int \mathsf{P}(y, \beta|X) \log \frac{\mathsf{P}(\beta|y, X)}{\mathsf{P}(\beta)} \, d\beta \, dy. \tag{2.16}$$

Noting that $\mathsf{P}(\beta)$ does not depend on the design X, we may reexpress the objective function in a more streamlined form:

$$\int \mathsf{P}(y, \beta|X) \log \mathsf{P}(\beta|y, X) \, d\beta \, dy. \tag{2.17}$$

Our goal is to maximize the expected information gain from the experiment which is equivalent to maximizing the Kullback-Leibler (KL) divergence between the prior and posterior models. Applied to linear regression, Equation 2.17 becomes [9]:

$$-\frac{k}{2}\log(2\pi) - \frac{k}{2} + \frac{1}{2}\log\det\left\{\sigma^{-2}F\right\}, \tag{2.18}$$

and once again we find that maximizing $|F|$ is the optimal solution. As a byproduct of these derivations we see that maximizing the expected information gain on the linear regression parameters is equivalent to minimizing model uncertainty.

In both D- and A-optimality for linear regression, we find that selecting examples is independent of the response values y, a fact exploited by Schein *et al.* [35] for selecting a training set before any labeling at all has occurred. In nonlinear models, we are not so lucky; the Fisher information depends on the response of the design matrix.

2.1.3 A-Optimality for Nonlinear Regression Models

A-optimality can be extended to a wide range of non-linear regression models, and a template is given in [9]. In Chapter 3 we derive a method for logistic regression. For now we explore the special case of backpropagation neural networks where the method has been applied in the past. In his Ph.D. dissertation [27] and companion publications [29, 28], MacKay derives the A-optimality and similar information-based objective functions for active learning of backpropagation neural networks inside a Bayesian setting. It was Cohn [11] who first evaluated A-optimality for backpropagation neural networks on "natural" data.

Neural networks may be trained using a variety of loss functions. Our discussion of backpropagation neural networks will consist solely of those networks fit with the least squares objective function, with its implicit Gaussian likelihood interpretation, *i.e.* we find parameter vector w that minimizes [4]:

$$\sum_n (y(x_n; w, \mathcal{A}) - t_n)^2 + \frac{1}{\sigma_p^2} \sum_d w_d^2 \tag{2.19}$$

where t_n is the observed training set output for observation n, and the second term in the summation provides model shrinkage as in the linear case. From this point we ignore the parameter $\mathcal{A}$ specifying the network architecture, and assume the architecture is fixed.

The objective function we choose to minimize through active data selection is:

$$\sum_{n \in \text{Pool}} (y(x_p; \hat{w}) - y(x_p; w))^2 \tag{2.20}$$

where we assume a true model with weights w, and compare it to the current model, with weights $\hat{w}$. As in the linear case, equation 2.20 is equivalent to a variance:

$$\sum_{n \in \text{Pool}} \text{Var}[y(x_n; \hat{w})]. \tag{2.21}$$

This objective function is related to the squared error by the following decomposition [20]:

$$\begin{aligned} E &= \mathsf{E}\left[(y(x;\hat{w}) - t)^2\right] && (2.22)\\ &= \mathsf{E}\left[t - \overline{y}(x;w))^2\right] + \left(\mathsf{E}_{\mathcal{D}}\left[y(x;\hat{w})\right] - \overline{y}(x;w)\right)^2 && (2.23)\\ &+ \mathsf{E}_{\mathcal{D}}\left[(y(x;\hat{w}) - \mathsf{E}_{\mathcal{D}}\left[y(x;\hat{w})\right])^2\right] && \\ &= \text{noise} + \left[\text{Bias}(y(x,\hat{w}))\right]^2 + \text{Var}\left[y(x;\hat{w})\right] && (2.24) \end{aligned}$$

where $\mathsf{E}_{\mathcal{D}}$ denotes an expectation over training sets, $\hat{y}(x;w)$ denotes the expected value of y under the true model parameters, and the expectation of the first term is an expectation of x, y over the true distribution. This is the squared bias/variance/noise decomposition of error for neural networks. A-optimality focuses on reduction of the variance term, ignoring the bias.

We employ a Taylor expansion to obtain an approximation:

$$y(x_p;\hat{w}) \simeq y(x_p;w) + c_p \cdot (\hat{w} - w) \tag{2.25}$$

where $c_p = \frac{\partial y(x_p;w)}{\partial w}$ is the gradient of y, which is dependent on x_p and w. Taking the variance of both sides we get:

$$\begin{aligned} \text{Var}[y(x_p;\hat{w})] &\simeq \text{Var}[y(x_p;w) + c_p \cdot (\hat{w} - w)] && (2.26)\\ &= \text{Var}[c_p \cdot (\hat{w} - w)] && (2.27)\\ &\simeq c_p' F^{-1} c_p. && (2.28) \end{aligned}$$

where the last step follows from the asymptotic normality of $(\hat{w} - w)$ [38]. Using the same algebraic manipulations developed in section 2.1.1, we arrive at an A-optimality for backpropagation neural networks:

$$\begin{aligned}\sum_{p \in \text{Pool}} \text{Var}[y(x_p;\hat{w})] &\simeq \sum_p c_p' F^{-1} c_p && (2.29)\\ &= \sum_p \text{tr}[c_p c_p' F^{-1}] && (2.30)\\ &= \sum_p \text{tr}[A_p F^{-1}] && (2.31)\\ &= \text{tr}[A F^{-1}] && (2.32)\end{aligned}$$

An approximation to the Fisher information matrix:

$$F_{jilk} = -\sum_n \frac{\partial y^n}{\partial w_{ji}} \frac{\partial y^n}{\partial w_{lk}} - \sum_n (y^n - t^n) \frac{\partial^2 y^n}{\partial w_{ji} \partial w_{lk}} \tag{2.33}$$

is frequently employed to speed up computations [27, 11]. The approximation is to drop the second term of the Fisher information matrix which is close to zero when the model is already accurate.

The method was evaluated on natural data by Cohn [11] who trained a neural network with 2 inputs a single layer of 20 hidden units and 2 outputs for a grand total of 80 parameters encoded in vector w. Hidden and output units were sigmoid, trained with the backprop procedure minimizing squared error. The method was evaluated by picking up to 100 observations. Despite an extensive search of the literature through document databases such as Researchindex, we could not find any other evaluation of nonlinear regression A-optimality on natural, as opposed to artificially generated, data in a pool-based active learning setting. This is a surprising fact given that Cohn's 1996 paper [11] and its earlier incarnation [10] are very well cited. Personal communication with Dr. Cohn, however, substantiates this assertion [13].

Through literature search, we were able to find some evaluations of A-optimality on artificial data [43, 19]. The examples that include noise in the data generation process use a homoscedastic noise generator in contrast to the perils of real data which often contain heterscedastic noise. The number of input units in these evaluations never exceed 4 and the number of hidden layer units never exceed 7. A single output unit was used in these evaluations. The largest number of parameters ever employed in an evaluation on artificial data that we could find was 35, and the evaluation was by Fukumizu [19].

Since this is a dissertation about applying A-optimality to logistic regression, a few words describing how our anticipated contributions differ from the previous work in backpropagation neural networks is in order. First, our emphasis is on applying A-optimality to a number of real data sets with very large numbers of features. As argued above, such empirical evaluation is lacking from the A-optimality active learning literature, regardless of the underlying machine learning algorithm. Second, the Fisher information matrix for logistic regression has a very different structure from backpropagation, requiring a different set of optimizations than has been used before. Third, our emphasis on large data sets will lead to optimizations of a different nature than previously used.

There are also many secondary motivations for applying A-optimality to logistic regression at this time. The convex structure of logistic regression's log-likelihood surface makes the method attractive in iterative re-training steps compared to the local solutions of backpropagation neural networks, and exploring this advantage is something worth pursuing. Logistic regression is a popular algorithm in some communities where backpropagation neural networks have not caught on. Finally, we note that successful logistic regression implementation of A-optimality may prove to be a stepping stone to a wider range of models, such as conditional random fields [24], while making the transition directly from backpropagation neural networks is less obvious and expedient.

2.1.4 An Information Theoretic Variant of A-Optimality

The derivation of A-optimality suggests a closely related information theoretic objective function [27, 29]. The intuition is the following. Since the A-optimality criterion is derived by adding up the variance terms of individual Gaussians that result from predictions over the pool, why not use instead the entropy of those individual Gaussians and add them up? Let $S(\mathsf{P}(y^n))$ denote the entropy from the prediction on observation n. The resulting objective function is:

$$
\begin{aligned}
S &= \sum_n S(\mathsf{P}(y^n)) && (2.34) \\
&= \frac{1}{2} \sum_n \log(c_n' F^{-1} c_n) + \text{constant} && (2.35)
\end{aligned}
$$

This quantity differs from A-optimality, since entropy is a nonlinear function of the variance term $(c_n' F^{-1} c_n)$. This is not the first information theory criterion we have seen: recall how we motivated D-optimality in information theoretic terms. Variants of Equation 2.35 have been applied in experimental design as well, and are reviewed in [9].

In this dissertation we are compelled to choose between A-optimality and Equation 2.35 as the emphasis of our studies. We have chosen to focus on A-optimality for now since this criterion is better known. It is likely that the techniques we will develop for implementation of A-optimality will facilitate evaluation of the information theoretic variant at a later date.

2.1.5 Classifier Certainty

For logistic regression and other probabilistic classifiers, several researchers have proposed minimizing the entropy of the algorithm's predictions [27, 28, 34][1]:

$$
-\sum_p \hat{\mathsf{P}}(y_p|x_p) \log \hat{\mathsf{P}}(y_p|x_p). \qquad (2.36)
$$

We call this method classifier certainty (CC). In a Bayesian setting for logistic regression, an analytical device improves probability estimates of this method [28], while in more general settings bagging has been used [34]. In any case, it is apparent that

[1] Some readers familiar with the language modeling literature will be used to "prediction entropy" as a measure of performance. However, in language modeling, it is actually cross-entropy that is measured, not prediction entropy for the reasons and perils outlined below.

minimization of 2.36 occurs when the predictions are close to 0 and 1, regardless of the true model $\mathsf{P}(y_p|x_p)$ or the true values y_p, and therefore it is not hard to find a situation where the method goes astray. We show such a situation in Chapter 3.

This is the first objective function approach we have presented that is not particularly specific to regression models, and in fact Roy and McCallum's evaluations were with the naive Bayes classifier [34]. As with other algorithm-agnostic methods, a strong case can be made that the classifier certainty method is more heuristic than explicit in its objectives.

2.1.6 Alternative Error-Based Objective Functions

In addition to variance-minimization techniques such as A-optimality, researchers have attempted to minimize other portions of the error decomposition of Equation 2.24. Cohn [12] explores minimization of the bias squared portion of error for locally weighted regression models. Sugiyama and Ogawa [42] minimize both bias and variance through a two-stage sampling approach. Both methods look promising, but empirical evaluation across diverse natural data sets is still lacking.

2.1.7 A Decision Theory Representation of Experimental Design

Frequently in the statistics literature on optimal experimental design, the authors choose to employ a decision theory exposition of the objective function and design criteria. In this section, we explain this style of exposition using the Bayesian setting and notation of Chaloner and Verdinelli [9], who follow the accounts of [32, 26].

Representing the model parameters by θ, the predictor vectors by x, response values by y, and a decision by d chosen from some set $\mathcal{D}$, risk is measured as the expectation of the utility function U:

$$U(\eta) = \int_y \max_{d \in \phi} \int_\Theta U(d, \theta, \eta, y) \mathsf{P}(\theta|y, \eta) \mathsf{P}(y|\eta) \, d\theta \, dy. \tag{2.37}$$

with solution:

$$U(\eta^*) \quad = \quad \max_{\eta \in \mathcal{H}} \int_y \max_{d \in \phi} \int_\Theta U(d, \theta, \eta, y) \mathsf{P}(\theta | y, \eta) \mathsf{P}(y | \eta) \, d\theta \, dy. \tag{2.38}$$

The interpretation of Equation 2.37 is that the model prior puts a probability over the response: $\mathsf{P}(y|\eta)$, which in turn can be used to put a posterior distribution over θ: $\mathsf{P}(\theta|y, \eta)$. With $\mathsf{P}(y|\eta)$ and $\mathsf{P}(\theta|y, \eta)$ in hand we may compute risk by taking an expectation over U. The decision d encodes a choice for X and also a terminal decision.

In cases where a portion of the pool already has associated outcomes $(x, y)_n$ prior to the experiment (the so-called sequential experimental design setting) this information may be encoded in the prior over θ or alternatively by expanding the notation to include previously acquired observations $(x, y)_n$ and future design points $x_{n'}$.

Equation 2.19 is an example of a utility function derived from a decision theory framework:

$$U(d, \beta, X, y) = -\mathsf{P}(y, \beta | X) \log \frac{\mathsf{P}(\beta | y, X)}{\mathsf{P}(\beta)} \tag{2.39}$$

where we substitute $\eta = X$ and $\theta = \beta$, and d does not appear.

The A-optimality criterion is derived for linear regression from the decision theory framework using [9]:

$$U(d, \beta, X, y) = -(\hat{\beta} - \beta)' A (\hat{\beta} - \beta). \tag{2.40}$$

2.2 Algorithm Independent Approaches

We now turn to algorithm independent approaches to active learning such as uncertainty sampling and query by committee. In the general classification setting this dissertation focuses on, little can be said that relates these approaches to explicit objective functions. Under a few assumptions, including at a minimum an assumption that classification is some noise free function of the predictors it may be possible to establish a relationship between each of these methods and an objective function.

The lack of principled motivation for these heuristic methods in more general settings has not stopped the empirical machine learning community from evaluating the methods on actual data sets [1, 2, 14, 31, 22, 25, 30, 34, 44]. In fact, by looking at the literature that has amassed around the heuristic methods one gains a sense of optimism for active learning as a whole. Our own experience with these methods paints a grim picture at times; the methods frequently produce results that are worse than random sampling from the pool. Traces of these negative results can be found within the empirical evaluations cited, but we wonder whether the literature as a whole might be biased towards positive results.

2.2.1 Uncertainty Sampling

Uncertainty sampling is a term invented by Lewis and Gale [25], though the ideas can be traced back to the query methods of Hwang *et al.* [23] and Baum [3]. We discuss the Lewis and Gale variant since it is widely implemented and general to probabilistic classifiers such as logistic regression. Essentially, the algorithm picks an observation with highly entropic predictions over the label categories as the one most useful for labeling.

The motivation for the approach is that observations with uncertain predicted labeling are more likely to be misclassified than observations with certain predicted labeling. As pointed out by Lewis and Gale, the method has several theoretical failings including: "underestimation of true uncertainty, and biases caused by non-representative classifiers" [25]. In Chapter 3 we demonstrate an additional failing of the uncertainty sampling: that even a model that has been trained with large representative data sets may diverge towards an inferior model when fed additional examples by this method.

2.2.2 Query by Committee

Query by committee was proposed by Seung, Opper and Sompolinksy [39], and then rejustified for the perceptron case by Freund *et al.* [18]. In its original formulation the method assumes:

- A noise-free classification task.

- A binary classifier with a Gibbs training [40] procedure.

Under these assumptions and a few others [39, 18] a procedure can be found that guarantees exponential decay in the generalization error:

$$E_g \sim e^{-nI(\infty)} \tag{2.41}$$

where $I(\infty)$ denotes a limiting information gain and n is the size of the training set. Compare Equation 2.41 to 1.1, to see the advantages of the method.

A description of the query by committee algorithm follows. A committee of k models M_i are sampled using the Gibbs training algorithm over the existing training set. The next training example is picked to minimize the entropy of the distribution over the model parameter posteriors. In the case of perceptron learning, this is achieved by selecting query points of prediction disagreement under various input distribution assumptions (stated explicitly in [39] and [18]). The method is repeated until enough training examples are found.

Alas, the assumptions of the method are frequently broken, and in particular the noise-free assumption does not apply to the data sets we hope to evaluate the method on. The noise-free assumption is critical to QBC, since the method depends on an ability to permanently discard a portion of version space (volume the parameters may occupy) with each query. Version space volume in the noisy case is analogous to the D-optimality score, since a determinant is essentially a volume measure. Generally the model variance, as measured through the D-optimality score of linear and non-linear models, does not decrease exponentially in the training set size even under optimal conditions.

The use of the query by committee method in situations where the assumptions do not apply is an increasing trend with the modifications of Abe and Mamitsuka [1] and McCallum and Nigam [30] who substitute bagging for the Gibbs training procedure setting the tone. The term "query by bagging" is becoming a catchphrase for algorithms that take a bagging approach to implementing the query by committee procedure. Measuring disagreement among committee members (members of the bag) may be performed either through the Jensen-Shannon divergence [1] or other KL-based information theoretic measures [30].

A description of the generalized query by committee algorithm called query by bagging [1] follows. A committee of models M_i is formed from the existing training set using the bagging procedure [6]. An observation is picked from the pool that maximizes disagreement from the pool. The procedure is repeated until enough training examples are chosen.

2.2.3 Beyond Classification: Heuristic Explorations

Uncertainty sampling and query by committee methods appear so general in their implementation that it is tempting to port the methods to more complex problems than the classification setting. Such has happened in the case of part of speech tagging, where the query by committee methods are generalized to apply to hidden Markov models [14]. In parsing, uncertainty sampling [22] and other heuristic approaches have been applied [44].

2.2.4 Heuristic Combinations

A recent trend in the pool-based active learning literature has been to take various approaches, usually uncertainty sampling or query by committee and try to improve performance through additional heuristics. McCallum and Nigam [30] augment query by committee by adding a density weighting factor to candidate observations in the pool, the motivation being the removal of outliers from serious consideration.

Another augmentation they employ is alternating active learning with the EM procedure in a naive Bayes model. Tang *et al.*, [44] combine clustering with uncertainty sampling to pick diverse sets of observations to label.

2.3 Large Margin Classifier Approaches

Since the year 2000, some attention has focused on novel approaches to support vector machine and perceptron active learning based on version space reduction [47, 37]. Recall, we saw version space arguments in the original motivations for the query by committee algorithm. These methods do not apply to logistic regression, but we regard them as interesting developments in the field of pool-based active learning.

2.4 Challenges: Model Misspecification and Broken I.I.D. Assumptions

Model misspecification is the phenomenon where the data do not match assumptions of the model. An example of misspecification is when the data are generated by a neural network with many hidden units, but the model employed is linear. Objective function methods including the experimental design methods are derived from an assumption that the response variable is generated by the model. How much misspecification may hurt the various active learning methods is unknown. MacKay [27] and Cohn [11] have both looked at this question on specific data sets. Our own evaluations presented in Chapter 3 on a single data set indicate that A-optimality for logistic regression is robust to this situation. The same evaluation indicates that both uncertainty sampling and classifier certainty do not handle this scenario well. This is a question that deserves more attention for A-optimality and all other active learning methods. Yue and Hickernall [49] tackled misspecification for linear models, and this is the only work we know of that has focused on correcting the problem.

A separate problem with active learning methods is that most of the theory of objective function approaches and intuitions of heuristic approaches rely on i.i.d. assumptions of the training set. In the nonlinear A-optimality case, one particular area of concern is the asymptotic approximation to variance, which relies on an i.i.d. assumption. A proper specification of the problem and its consequences are an interesting challenge.

2.5 Active Learning Evaluation Methodology

The largest evaluations of active learning were conducted using decision trees and variants of query by committee [1, 31] on UCI machine learning repository data [5]. Document classification [30] and other natural language processing domains are areas under frequent investigation [14, 22, 44, 2]. Evaluations typically plot accuracy per observations labeled through tabular or figure presentation. Figure 2.1 shows a typical evaluation. The horizontal line on top represents the performance of a method trained on the entire pool.

There are several variables of an evaluation that must be decided. First, it must be decided how many random examples to start out with before active learning will begin. Second, it must be decided whether to use a purely active learning approach to sampling (as performed in [30, 34]) or to mix active learning with random sampling. Also, some evaluations choose to sample more than one point at a time before retraining for computational expediency [31]. Our experience is that different choices for each of these variables may lead to very different conclusions about performance and robustness of a method.

Evaluations using active learning to select all examples are interesting because they represent a more challenging task; it seems much easier for a learner to far worse than random. Evaluations that mix random sampling with active learning represent how a conservative implementation is likely to work in the real world,

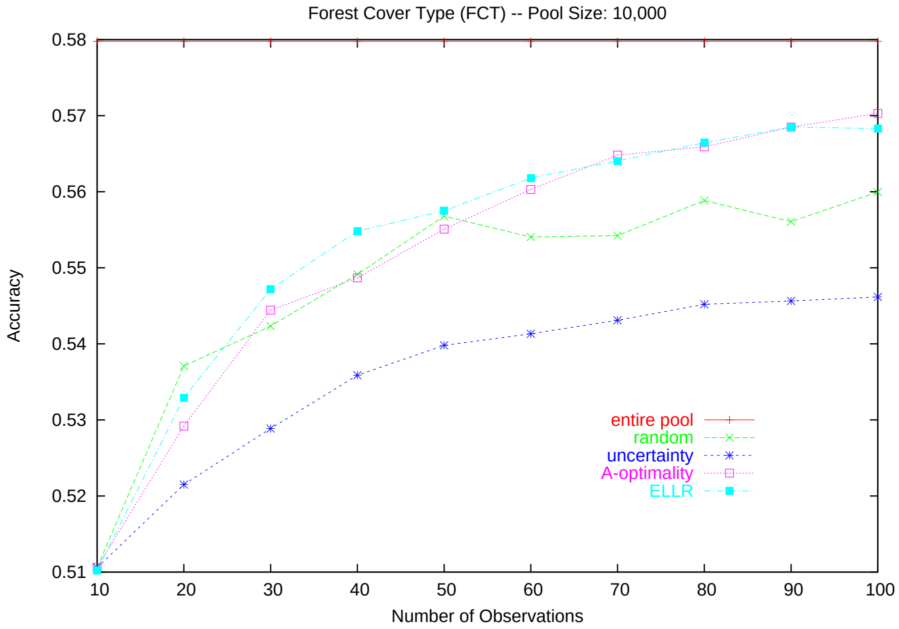


Figure 2.1: A Comparison of A-optimality, uncertainty sampling and classifier certainty (ELLR) on the Forest Cover Type data set from the UCI machine learning repository

since including more random samples decreases the probability that a method will perform worse than random, and in our experience often helps performance. A separate final decision that must be made in an evaluation is how many examples to pick before stopping the evaluation. Many researchers stop evaluations after $100 - 500$ queries, while some advocate using the entire pool when space permits large figures in exposition.

Past work in evaluation of active learning in the classification setting has typically focused on accuracy measures. An additional criteria that we will explore is calibration of the trained logistic regression probabilities. Calibration is an issue that is typically ignored, and it is our suspicion that most active learning methods will lead to a badly calibrated model. When the training data set is large enough, it is possible to re-calibrate a trained logistic regression model, but when training set sizes are small (several hundred examples) then a recalibration will not be very accurate. Calibration of the final model probabilities has importance in applications as well as in ensuring that the assumptions of the active learning procedures are intact after repeated queries.

2.6 Summary

The review gives a tour of the field serving the duel purposes of listing the various approaches and delineating our work from previous efforts employing experimental design in active learning. At times we express our own view of which areas deserve more attention. We maintain a degree of chronological accuracy; the experimental design methods were proposed as an answer to the active learning setting before most of the heuristic methods, and well before the heuristic methods caught on. Experimental design now seems to be unknown to much of the community due to the recent emphasis on heuristic methods and the recent entry of most members of the active learning community.

Chapter 3

A-Optimality for Logisitic Regression: Derivation and Preliminary Investigations

3.1 Introduction

In this Chapter we derive an objective function, called A-optimality, for measuring the expected benefit of labeling an example for logistic regression. In the design of experiments (DOE) literature [17, 9], other special cases of the same function have proven useful for designing linear regression and location/scale logistic regression experiments. We will evaluate A-optimality in training more general forms of logistic regression models in classification settings where the goal (or "experimental objective") is to build a classifier with the least generalization error.

In evaluating the quality and risks of an active learning method we look for two important traits:

1. The method must, in general, lead to attractive performance gains over random sampling from the pool.

2. The method must, in almost all cases, give performance that is at worst the

quality of random sampling from the pool.

Our empirical evaluations on four data sets demonstrate that A-optimality for logistic regression satisfies both of these *desiderata*.

3.2 From Scientific Modeling to Classification

Experimental design research for logistic regression has focused on the two parameter, single predictor model [8, 15]:

$$\begin{aligned} \mathsf{P}(y=1|x,\alpha,\mu) &= \frac{1}{1+\exp(-\alpha(x-\mu))}, && (3.1) \\ \alpha &\sim \mathcal{U}(a,b) && (3.2) \\ \mu &\sim \mathcal{U}(a',b') && (3.3) \end{aligned}$$

where the response variable y takes on the class labels $\{0,1\}$. This is a Bayesian model with a uniform distribution on the parameters. The parameter μ is called the 'location parameter', and it takes on the value of x s.t. $\mathsf{P}(y=1|x,\alpha,\mu)=0.5$. The parameter α, called the "scale parameter," encodes the change in probability with respect to x. Chaloner and Larntz [8] generalize several of the classic experimental design objective functions for linear regression to this class of models for answering such questions as: *For what value x does* $\mathsf{P}(y=1|x)=\gamma$, *where* $\gamma \in (0,1)$.

Note that the goal of the experiment described above is to learn something about x and therefore the underlying model that generates the response variable. In contrast, the goal of active learning is to build a model with the best generalization accuracy/error. This subtle difference in emphasis between the optimal experimental design community and the active learning community boils down to the observation that the optimal design researchers believe their models actually generate the response variable when deriving criteria for designing experiments, and therefore they can learn something about a natural process by conducting an experiment.

For our experiments in active learning, the logistic regression uses more than two parameters. The parametric form of the model is :

$$
\begin{aligned}
\mathsf{P}(y=1|x,\beta) &= \sigma(x'\beta), \text{ where} && (3.4)\\
\sigma(\theta) &= \frac{1}{1+\exp(-\theta)} && (3.5)\\
\beta &\sim \mathcal{N}(0,\sigma_p^2 I). && (3.6)
\end{aligned}
$$

The term $x'\beta$ denotes the dot product of two vectors: x, the predictors, and β, the model parameters.

In the two-parameter case, Equations 3.1 and 3.4 are equivalent, modulo the different choice of priors. Defining the shorthand $\sigma_n = \sigma(x_n'\beta)$ for observation n, the log-likelihood of the model we use for training classifiers is given by:

$$
\begin{aligned}
\mathcal{L} &= \left[\sum_n y_n \log \sigma_n + (1-y_n)\log(1-\sigma_n)\right] && (3.7)\\
&\quad -\frac{1}{2\sigma_p^2}||\beta||^2.
\end{aligned}
$$

The penalized log likelihood of Equation 3.7 can be viewed as a consequence of the prior on β, or equivalently the prior can be viewed as the Bayesian interpretation of the regularization term $\frac{1}{2\sigma_p^2}||\beta||^2$. In experimental design it is necessary to take the Bayesian viewpoint in order to start with a model with no training data and pick a training set. In active learning we do not necessarily begin biased sampling with the first observation; we can start with a random seed of examples. In this paper we take the regularization view of the model (c.f. [7]) rather than the Bayesian view. Our Gaussian priors over parameters build in no domain knowledge, but are instead a potential interpretation of the parametric form of the regularization.

3.3 A-Optimality for Binary Logistic Regression

We now derive an objective function for active learning of logistic regression classifiers. Denote by β the true parameter values and by $\hat{\beta}$ the maximum likelihood (ML) estimates of the model parameters. Since we believe by and large that the response distributions are not parameterized by the model, true parameters β can be viewed as the values the parameters take in the limit as the training set increases. A consequence of the representation of logistic regression in its exponential family form is that:

$$\hat{\theta} \;\sim\; \mathcal{N}(\theta, I(X, \hat{\theta})^{-1}) \;\text{ asymptotically,} \tag{3.8}$$

where $I(X, \hat{\theta})$ denotes the observed Fisher information matrix of the regularized logistic regression model:

$$I(X, \hat{\theta}) \;=\; \left[\sum_n x_n x_n' \sigma_n (1 - \sigma_n)\right] + \left[\sigma_p^2 I\right]^{-1} \tag{3.9}$$

and X denotes the training set predictor matrix, often called the design matrix. The observations x_n are vectors formed from the rows of the training set matrix X.

The objective function we seek to minimize when choosing which examples to label is defined as:

$$\sum_{n \in \text{Pool}} \text{Var}(\sigma(x_n' \hat{\beta})) = \sum_{n \in \text{Pool}} E[\sigma(x_n' \hat{\beta}) - \sigma(x_n' \beta)]^2. \tag{3.10}$$

In other words, we want the model predictions over the entire pool to be as close as possible to the predictions of the "true" model, in the squared loss sense.

We approximate Equation 3.10 using two steps of a Taylor expansion around $\sigma(x'\beta)$:

$$\begin{aligned} \sigma(x_n' \hat{\beta}) \;&\simeq\; \sigma(x_n' \beta) + c_n' (\hat{\beta} - \beta), \;\text{ where} && (3.11) \\ c_n \;&=\; (\frac{\partial}{\partial \beta_1} \sigma_n, \ldots, \frac{\partial}{\partial \beta_d} \sigma_n)' && (3.12) \end{aligned}$$

is the gradient vector for σ_n. Using the Taylor series approximation we have:

$$\begin{aligned} \text{Var}(\sigma(x_n' \hat{\beta})) \;&\simeq\; \text{Var}(c_n' (\hat{\beta} - \beta)) && (3.13) \\ &\simeq\; c_n' I(X, \hat{\beta})^{-1} c_n \;\text{ from (3.8)} && (3.14) \end{aligned}$$

Equation 3.14 is known as c-optimality, which minimizes the prediction variance over a single observation. Defining $A_n = c_n c_n'$ and $A = \sum_n A_n$ we derive a formula for minimizing the variance over the pool:

$$\begin{aligned}
\sum_{n \in \text{Pool}} c_n' I(X, \hat{\beta})^{-1} c_n &= \sum_{n \in \text{Pool}} \text{tr}\left\{ A_n I(X, \hat{\beta})^{-1} \right\} \\
&= \text{tr}\left\{ A I(X, \hat{\beta})^{-1} \right\} \qquad (3.15) \\
&\doteq \phi(X, \mathbf{y}). \qquad (3.16)
\end{aligned}$$

Equation 3.15 is the A-optimality objective function for logistic regression with the A matrix that gives the method its name. Frequently the A matrix will be notated in the literature as $A(\hat{\theta})$ in order to make explicit the dependency of the matrix on the model parameters (or equivalently the labeling of the training set) that we have left implicit in our own notation. We use instead the $\phi(X, \mathbf{y})$ notation to show the dependency of the criterion on the response value of the training set in what follows.

Equation 3.15 shows how to compute the utility of a labeled training set. We now need to derive a quantity that describes the expected benefit of labeling a new observation. We denote the labels of the training set by $\mathbf{y}$ and the training set predictors (encoded by a design matrix X) by $\mathcal{T}$. Then using the current estimated model $\hat{\mathsf{P}}(y|x)$, the expected benefit of labeling observation x_n is:

$$\begin{aligned}
\phi(\mathcal{T}, \mathbf{y}, x_n) &= \hat{\mathsf{P}}(y_n = 1|x_n)\phi(\mathcal{T} \cup \{x_n\}, \mathbf{y} \cup \{1\}) \\
&+ \hat{\mathsf{P}}(y_n = 0|x_n)\phi(\mathcal{T} \cup \{x_n\}, \mathbf{y} \cup \{0\}).
\end{aligned} \qquad (3.17)$$

Ignoring model-fitting, the worst-case computational cost associated with picking a new example is: $O(KND^2 + KD^3)$[1], where N is the number of pool examples used to create the A matrix, K is the number of candidates evaluated for inclusion in the training set and D are the number of predictors in the model. The N term may be reduced using Monte Carlo sampling from the pool.

[1] We assume the most naive of implementations for the matrix calculations.

3.4 Alternative Active Learning Methods for Evaluation

We evaluate A-optimality against two alternative methods: uncertainty sampling and classifier certainty.

3.4.1 Uncertainty Sampling

Uncertainty sampling (introduced in Chapter 2) in our implementation uses entropy of the current predictions to select the next example for labeling. The motivation for the approach is that observations with uncertain predicted labelings are more likely to be misclassified than observations with certain predicted labelings. As pointed out by Lewis and Gale, the method has several theoretical failings including: "underestimation of true uncertainty, and biases caused by nonrepresentative classifiers" [25]. In the Evaluation section we demonstrate an additional failing of the uncertainty sampling: that even a model that has been trained with large representative data sets may diverge towards an inferior model when fed additional examples by this method.

Using uncertainty sampling, the computational cost of picking an example from K candidates is: $O(KD)$ where D is the number of predictors.

3.4.2 Classifier Certainty

Classifier certainty (CC), advocated by [34], is a technique for more directly minimizing a loss function of interest than the uncertainty sampling heuristic. The technique is general to a large class of loss functions, however the authors demonstrate success with the log loss function:

$$L(X, \mathbf{y}) = -\sum_{x \in \text{Pool}} \mathsf{P}(y|x) \log \hat{\mathsf{P}}(y|x) \tag{3.18}$$

where $\mathsf{P}(y|x)$ denotes the probability of the hypothetical "true" model, and $\hat{\mathsf{P}}(y|x)$ denotes the probability using the current model. Unfortunately, the "true" probabilities $\mathsf{P}(y|x)$ are unknown and so an ad-hoc approximation of using the current model $\hat{\mathsf{P}}(y|x)$ is employed instead:

$$\hat{L}(X, \mathbf{y}) = -\sum_{x \in \text{Pool}} \hat{\mathsf{P}}(y|x) \log \hat{\mathsf{P}}(y|x). \tag{3.19}$$

Measuring the benefit of adding an observation is computed by an expectation similar to Equation 3.17. The method measures the expected decrease in prediction certainty over the pool after labeling an observation.

Using the current model probabilities $\hat{\mathsf{P}}(y|x)$ may cause problems for CC due to variance in predictions, causing an early bias that takes a partially trained model astray early in the learning curve. Roy and McCallum introduce bagging to cut down the variance affect in naive Bayes. MacKay proposes an objective function equivalent to Equation 3.19 for active learning of logistic regression classifiers, but with variance handled analytically (in a Bayesian framework) at the expense of implementation complexity [27]. A-optimality takes the alternative approach of making minimization of this variance the objective function for active learning.

Excluding the cost of model fitting, implementation of CC is at worst: $O(BKND)$, where B is the number of classifiers in the bag, N is the number of observations from the pool used to compute the benefit of adding an observation, D is the number of predictors, and K is the number of candidates evaluated for labeling. An approximation that is used for CC as in computing the A matrix of A-optimality is the sum over the pool; Monte Carlo sampling reduces this burden.

3.4.3 Method Relationships

The CC method can be interpreted as a cousin to uncertainty sampling. Uncertainty sampling chooses examples with the most entropic prediction values. CC on the other picks examples such that the *resulting* predictions of the pool remainder will become

Table 3.1: Descriptions of the data sets used in the evaluation. Included are counts of: the number of observations (OBS), the number of predictors (PRED) and the number of observations in the majority class (MAJ)

DATA SET	**OBS**	**PRED**	**MAJ**
FCT	20,000	54	10,210
WDBC	569	30	357
TD	711	21	400
SJGS	3190	36	1655

least entropic.

A-optimality can be interpreted as a method that combines components of both example uncertainty and prediction certainty. Example uncertainty plays a role in A-optimality because uncertain examples have a larger impact in defining $I(X, \beta)$ (see Equation 3.9), so uncertain examples have a tendency to decrease the function $\phi(X, \beta)$ when all other factors held constant. The probabilities of the predictions play a role as well since the elements of the A matrix have a tendency to decrease as Equation 3.19 decreases. As the entries of A decrease in magnitude so does the criterion $\phi(X, \mathbf{y})$ when all other factors are held constant.

3.5 Evaluation

We evaluate the method on four data sets chosen from the UC Irvine data repository [5]: Forest Cover Type (FCT), Wisconsin Diagnostic Breast Cancer (WDBC), Splice Junction Gene Sequence (SJGS), and Thyroid Domain (TD). The data sets were converted to a binary classification task by merging all but the most representative class label into a single class. Table 3.1 describes the data set characteristics after formatting while the individual processing steps are described below.

In all evaluations we train a binary logistic regression including a bias term using the regularization $\sigma_p^2 = 1$ until convergence. For computation of the A-optimality score we use the same prior (i.e. our model for active learning exactly matches the model we train). CC was trained without bagging. A bag size of 5 dramatically slowed down the evaluation, sometimes helping but also increasing the number of cases where CC performs worse than random sampling. We suspect that stability of logistic regression (compared to decision trees or naive Bayes) in combination with a small bag size was behind the case where bagging hurt performance. Since a bag size of 20 or 30 was impractical, we eliminated bagging from the experiments altogether.

3.5.1 Data Set Preparation

The Forest Cover Type (FCT) data set consists of measurements of 30x30 meter cells of forest land conducted by the US Forest Service. The task associated with the data set is to predict, using the measurements, which of 7 tree categories is growing in the cell. According to the data only one type of tree grows in each cell. The original number of records, $581,012$, is massive and so we randomly sampled the data set to reduce it to 20,000 observations. The number of predictors in the data set is 54. The lodgepole pine variety of tree happens to represent about 50% of the observations and so we merge all other tree types into a single category.

The Wisconsin Diagnostic Breast Cancer (WDBC) data set consists of evaluation measurements (predictors) and final diagnosis for 569 patients. The goal is to predict the diagnosis using the measurements. The number of predictors is 30.

The Thyroid Domain (TD) data set (called “thyroid-ann” in the repository) consists of patient evaluation measures and three classes: underactive thyroid, overactive thyroid and normal thyroid. We merge the underactive and overactive thyroid classes into a single class. The number of predicts is 21 and the number of observations is: 711.

The Splice Junction Gene Sequences (SJGS) consists of 3190 short sequences of DNA. The goal is to predict the presence of an intron/exon boundary (IE), an exon/intron boundary (EI) or no boundary. We merge the IE and EI classes into a single class. The sequences are converted into 6 predictors consisting of nucleotides plus the two other descriptors used in the data set, where the predictors are the number of times the nucleotide occurred in the sequence. We add the 30 interaction terms for these sequences to create a data set with 36 predictors.

3.5.2 Primary Evaluation Design

We perform evaluation over 100 train/test splits on each of the four data sets comparing the A-optimality criterion against uncertainty and CC methods. We decided on 100 repetitions since many of the data sets have a generalization accuracy in the high 50's, and such a low generalization accuracy can be associated with higher variance of the learning curve. Train/test splits were created by splitting the entire data sets (described in Table 3.1) in half at random.

On each of the 100 runs, 10 random examples were given as "seed examples" to each learner which proceeded to use their example scoring function to select the next 90 examples. The 10 seed examples contained at least 2 examples from each category label, ensuring a reasonable starting point for active learning. Though the pool sizes vary across data sets, 100 training examples is equal to less than half the pool in each case. At each iteration of observation selection, 10 candidates were chosen at random from the pool and scored according to the active learning scoring function. Ties in the scores of candidate observations were broken at random. We report accuracy as a measure of performance since in these data sets the marginal counts of the different classes are roughly equal (illustrated in Table 3.1).

3.5.3 Primary Evaluation Results

Evaluations took under 24 hours for the longest experiment of the four data sets, CT, to run. We used Monte Carlo sampling of 1000 observations from the pool in computing the A matrix and CC objective function in order to speed up the evaluation. This was only necessary for the CT data set since the pool sizes for the

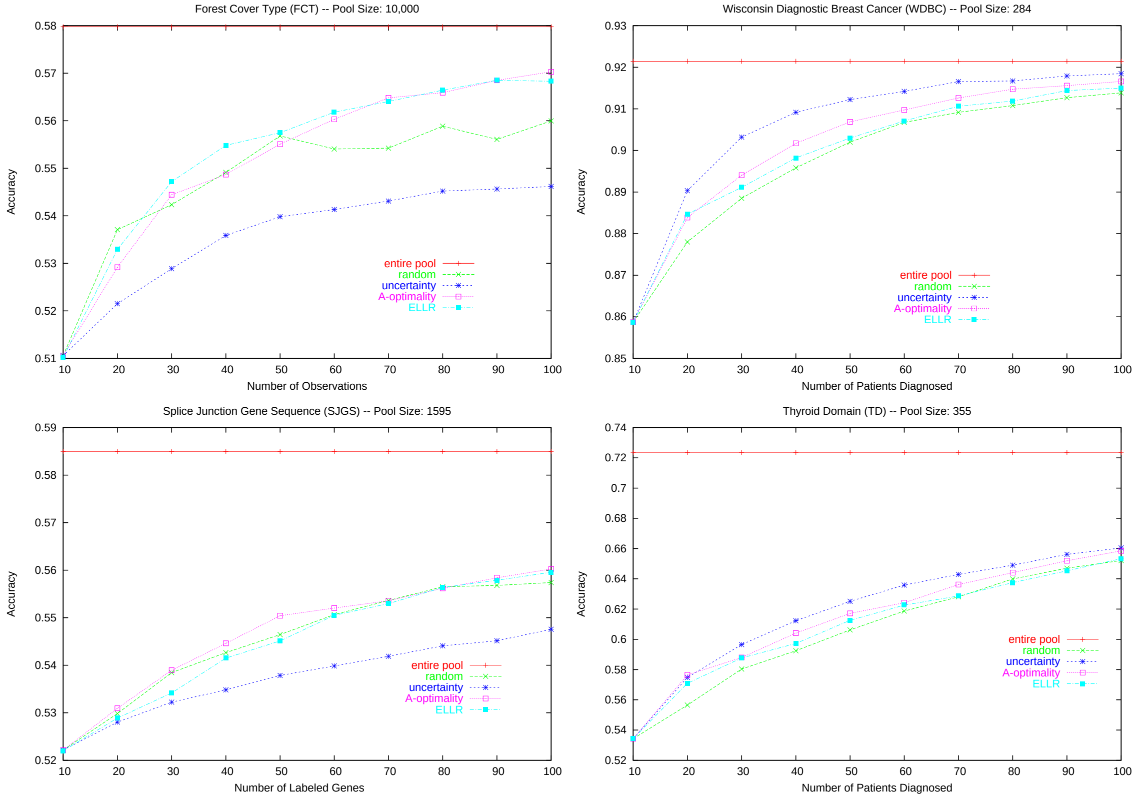


Figure 3.1: Empirical evaluation of $A-$optimality, uncertainty sampling, classifier certainty (denoted ELLR in the figures for historical reasons), and random sampling on four data sets.

other data sets were already relatively small.

Figure 3.1 shows the learning curves of the three active learning methods on all four data sets. Random sampling from the pool supplies a baseline. On all four data sets, A-optimality outperforms random noticeably, demonstrating that the method gives attractive performance. In contrast, uncertainty sampling diverges radically below random on the SJGS and FCT data sets. A-optimality and CC both appear to match or beat random performance in these evaluations. By the end of the learning curves the standard deviations of the accuracy were below 0.035 on all data sets for all methods, so at 100 trials the 95% confidence intervals had size less than 0.007 at the right hand side of the curve. The confidence intervals are larger for smaller training set sizes.

3.5.4 Limit Performance Evaluation

In addition to the primary evaluation, we explore the possibility that each of the active learning methods will cause a well-trained model to diverge towards a weaker performing model. We concocted an artificial data set shown in Table 3.2 consisting of two binary predictive features plus a bias term (a predictor that always takes the value 1). Feature 2 determines whether we enter a region of feature space with good predictive ability. When feature 2 is off, feature 1 determines the outcome with high probability. When feature 2 is on feature 1 has no predictive ability over the outcome. This is an example of a data set that does not match the distribution of a maximum likelihood logistic regression model.

In creating the training, pool, and test sets from Table 3.2, we used expected counts of each of the rows and outcomes of the table in what would otherwise be a random sample of 400 (train), 40 (pool), and 40 (test) respectively. The pool is sampled with replacement during evaluation to imitate having an infinite pool of data. Also, during evaluation we randomly break any ties in the estimated benefit of labeling examples to prevent deterministic outcomes due to ordering of observations in the pool. Using 40 examples in the test set with the expected values of each row/outcome computed from the table mimics the expected proportions of a very large sample from the table.

3.5.5 Limit Performance Results

Figure 3.2 shows the results of performing the limit performance evaluation averaged over 25 runs. Initializing the model with 400 observations, we add 600 additional observations according to the active learning criteria, with random sampling included as a baseline. Here we measure the sum of squared errors of prediction from the true class label. Examining Figure 3.2, we see that A-optimality performs on par with

Table 3.2: An artificial data set to test whether an active learning scheme can actually cause a well-trained model to diverge dramatically. Each row of the table is given equal weight in creating seed, pool and test sets that truly represent the underlying distribution of the data. The top two rows encode signal, the bottom two rows encode no signal.

OUTCOMES		**PREDICTORS**		
$\mathsf{P}(y=1\|x)$	$\mathsf{P}(y=0\|x)$	x_1	x_2	bias
0.1	0.9	0	0	1
0.9	0.1	1	0	1
0.5	0.5	0	1	1
0.5	0.5	1	1	1

random sampling. Uncertainty sampling has a tendency to pick the "noisy" examples which, in turn, drag the parameters associated with feature 1 and the bias feature from their optimal values. The squared error using the uncertainty sampling rises monotonically achieving a 3.0% increase at 1000 observations. CC also exhibits a noticeable increase in squared error, though not as large. Examining the choice of examples picked by CC we saw that observations from row 1 of Table 3.2 was grossly overrepresented while row 3 was grossly underrepresented.

3.6 Discussion of Evaluation

Based on the empirical evaluation we see that A-optimality is a very attractive objective function for active learning. The method always performs about as well as random sampling from the pool at a minimum. In most cases, A-optimality leads to substantial performance improvements. For instance, on the forest cover data set, the improvement of the method indicates that one example picked with A-optimality is worth almost two random examples. In contrast, uncertainty sampling has two data sets in the evaluation (FCT, SJGS) where the performance drops below random in a dramatic fashion. Comparisons A-optimality to CC on the natural data evaluations do not show a clear advantage of either method over the other.

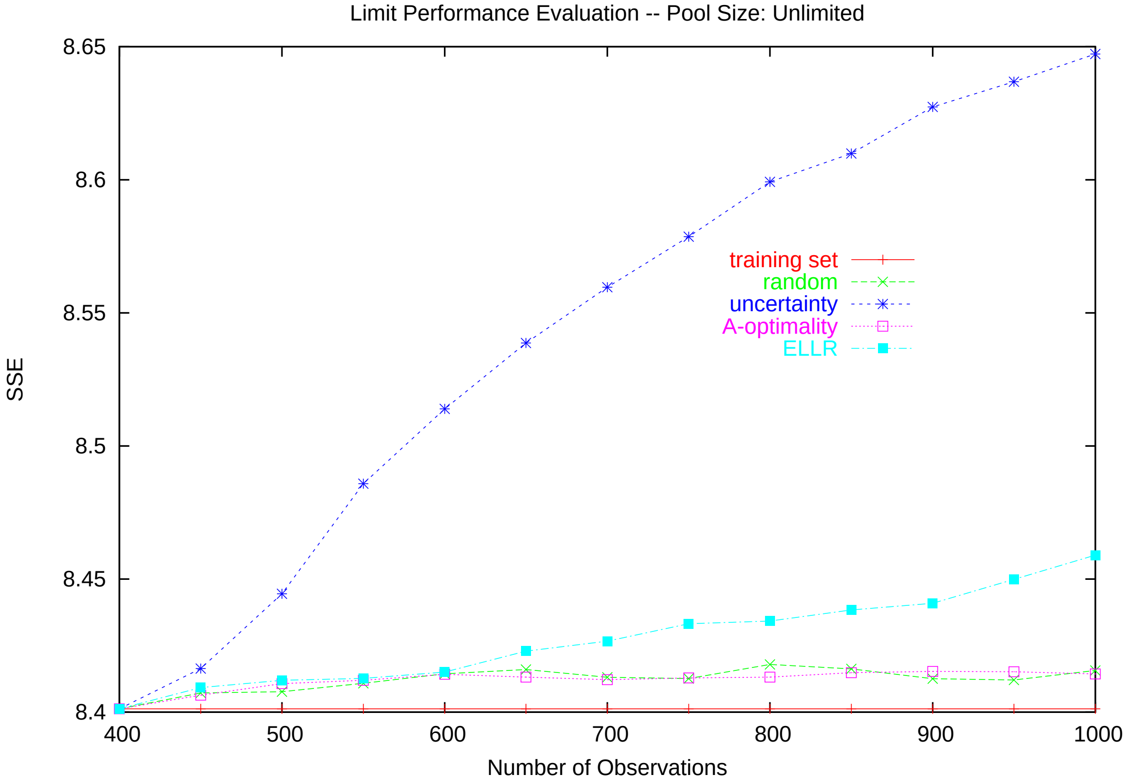


Figure 3.2: Sum of squared error (SSE) empirical evaluation of A-optimality, uncertainty sampling, CC (denoted ELLR in the figure for historical reasons), and random sampling on a data set consisting of an infinite pool sampled from Table 3.2.

One might wonder why CC performs worse than random sampling on the synthetic data when Equation 3.18 appears to more correctly reflect the true objective function of interest: matching the predictions of the true model. Equation 3.18 is minimized when the KL divergence $\mathrm{KL}(\mathsf{P}(y|x)||\hat{\mathsf{P}}(y|x))$ is minimized. However, the approximation made in Equation 3.19 is minimized when the predictions over the pool are closest to 0 and 1. Equation 3.19 may potentially bipass the true model $\mathsf{P}(y|x)$ in its effort to make its probabilistic predictions close to 0 and 1, and this helps or hurts performance depending on the quality of the initial model with respect to the complexity of the learning task.

In contrast, the A-optimality criterion of Equation 3.15 is much more conservative; it attempts to make the model predictions as close to the "true" model as possible, according to the squared loss function. The greatest impediment to applying A-optimality to arbitrary data sets is the computational cost of picking the next example. Parallelism, numerical optimization and numerical approximation can all play a role in reducing the computational cost of employing A-optimality. Future work will look at methods for making evaluations on larger data sets more practical.

In the evaluations of this paper, the single largest computational cost was model fitting. It was model fitting computational time that prevented us from employing bagging in the CC method. We expect this cost could be significantly diminished by seeding parameters with a previous solution rather than starting model fitting from scratch. If model fitting time could be eliminated as a bottleneck, we would expect computation of the A-optimality criterion to be the new bottleneck of the evaluation.

3.7 A-Optimality Derivation for Multinomial Regression

Now we consider the multi-category classification setting and seek methods for applying A-optimality. We begin with a description of the model, followed by derivation of A-optimality. Applying exact A-optimality proves burdonsome for many interesting data sets, and so we introduce an approximation of A-optimality that scales linearly in the number of categories.

3.7.1 Multinomial Regression

The multinomial regression model generalizes logistic regression to the multi-category case. The probability of a class label c given an observation vector x is parameterized as follows:

$$\mathsf{P}(y = c|x) \quad = \quad \frac{\exp(\beta_c \cdot x)}{\sum_{c'} \exp(\beta_{c'} \cdot x)}, \tag{3.20}$$

where there are separate parameter vectors β_c for each class c. In the two-category case, one of the parameter vectors β_0 can be set to $\mathbf{0}$ to reconstruct the standard logistic regression model.

As in the binary logistic regression presentation, we will consider the penalized log-likelihood in our derivation of an active learning criteria:

$$\mathcal{L} = \left[\sum_{n,c} 1_{n,c} \log \mathsf{P}(y = c|x)\right] + \frac{1}{2\sigma_p^2}||\beta||^2, \tag{3.21}$$

where n ranges over the observations in the training set and β is a single vector, formed by concatenating the individual β_c vectors together.

3.7.2 The Fisher Information Matrix

In deriving the Fisher information matrix we re-write the log-likelihood in a convenient form:

$$\begin{aligned}\mathcal{L} &= \left[\sum_{n,c} 1_{n,c} \log \mathsf{P}(y=c|x)\right] + \frac{1}{2\sigma_p^2}||\beta||^2 && (3.22)\\ &= \sum_{nc} 1_{n,c} \log \exp(\beta_c \cdot x) - \sum_n \log \sum_{c'} \exp(\beta_{c'} \cdot x) - \frac{1}{2\sigma_p^2}||\beta||^2 && (3.23)\\ & && (3.24)\end{aligned}$$

and take the first derivative:

$$\frac{\partial \mathcal{L}}{\partial \beta_{ci}} = \sum_{nc} x_{ni} - \sum_n \mathsf{P}(c|x_n)x_{ni} - \frac{\beta_{ci}}{\sigma_p^2}. \qquad (3.25)$$

In taking the second derivative $\frac{\partial \mathcal{L}}{\partial \beta_{ci}\beta_{c'j}}$ we must split the derivation into three cases.
Case I: $c = c'$ and $i = j$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \beta_{ci}\beta_{cj}} &= \frac{\partial \mathcal{L}}{\partial \beta_{ci}\beta_{c'j}}\left[\sum_{nc} x_{ni} - \sum_n \mathsf{P}(c|x_n)x_{ni} - \frac{\beta_{ci}}{\sigma_p^2}\right] && (3.26)\\ &= -\sum_n x_{ni}x_{nj}\mathsf{P}(c|x_n)(1-\mathsf{P}(c|x_n)) - \frac{1}{\sigma_p^2} && (3.27)\end{aligned}$$

Case II: $c = c'$ and $i \neq j$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \beta_{ci}\beta_{c'j}} &= \frac{\partial \mathcal{L}}{\partial \beta_{ci}\beta_{c'j}}\left[\sum_{nc} x_{ni} - \sum_n \mathsf{P}(c|x_n)x_{ni} - \frac{\beta_{ci}}{\sigma_p^2}\right] && (3.28)\\ &= -\sum_n x_{ni}x_{nj}\mathsf{P}(c|x_n)(1-\mathsf{P}(c|x_n)) && (3.29)\end{aligned}$$

Case III: $c \neq c'$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \beta_{ci}\beta_{c'j}} &= \frac{\partial \mathcal{L}}{\partial \beta_{ci}\beta_{c'j}}\left[\sum_{nc} x_{ni} - \sum_n \mathsf{P}(c|x_n)x_{ni} - \frac{\beta_{ci}}{\sigma_p^2}\right] && (3.30)\\ &= -\sum_n x_{ni}x_{nj}\mathsf{P}(c|x_n)\mathsf{P}(c'|x_n) && (3.31)\end{aligned}$$

Putting these three components of the Hessian together, we arrive at the Fisher information matrix:

$$F_{(ci)(c'j)} = \begin{cases} \sum_n x_{ni}^2 \mathsf{P}(c|x_n)(1-\mathsf{P}(c|x_n)) + \frac{1}{\sigma_p^2} & c = c' \text{ and } i = j \\ \sum_n x_{ni}x_{nj}\mathsf{P}(c|x_n)(1-\mathsf{P}(c|x_n)) & c = c' \text{ and } i \neq j \\ \sum_n x_{ni}x_{nj}\mathsf{P}(c|x_n)\mathsf{P}(c'|x_n) & c \neq c \end{cases} \qquad (3.32)$$

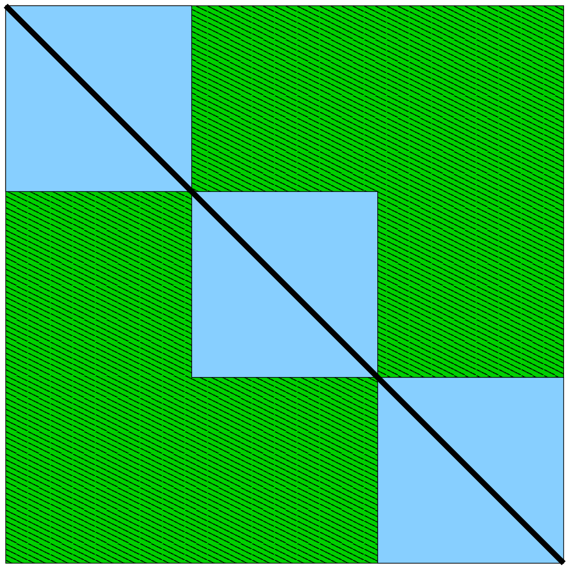


Figure 3.3: An illustration of the structure of the Fisher information matrix of the multinomial regression model in the 3-category case. The matrix consists of diagonal, block-diagonal and off-block-diagonal structure.

As an illustration, Figure 3.3 demonstrates the structure of the matrix for a 3-category case.

3.7.3 A-Optimality

As in previous derivations of A-optimality, we begin by stating the objective function of interest and than proceed to derive an optimality criterion. The objective function we wish to minimize through queries is:

$$\sum_{n,c} \mathrm{Var}(\hat{\mathsf{P}}(c|x_n)) = \mathsf{E}\left[\hat{\mathsf{P}}(c|x_n) - \mathsf{P}(c|x_n)\right]^2 . \tag{3.33}$$

As usual, we measure the sensitivity of the output with respect to the inputs through a gradient calculation. Define the vector $g_n(k)$ indexed by the category/predictor pair (c, i) as follows.

$$g_{nci}(k) = \frac{\partial}{\partial \beta_{ci}} \mathsf{P}(k|x_n) \tag{3.34}$$

$$
\begin{aligned}
&= \frac{\partial}{\partial \beta_{ci}} \frac{\exp(\beta_k \cdot x)}{\sum_{c'} \exp(\beta_{c'} \cdot x)} && (3.35)\\
&= \frac{\exp(\beta_k \cdot x)^2 x_{ni} - x_{ni} \exp(\beta_k \cdot x) \sum_{c''} \exp(\beta_{c''} \cdot x)}{\left(\sum_{c'} \exp(\beta_{c'} \cdot x)\right)^2} && (3.36)\\
&= \mathsf{P}(c|x_n)^2 x_{ni} - \mathsf{P}(c|x_n) x_{ni} && (3.37)
\end{aligned}
$$

when $k = c$ and

$$
\begin{aligned}
g_{nci}(k) &= \frac{\partial}{\partial \beta_{ci}} \mathsf{P}(k|x_n) && (3.38)\\
&= \frac{\partial}{\partial \beta_{ci}} \frac{\exp(\beta_k \cdot x)}{\sum_{c'} \exp(\beta_{c'} \cdot x)} && (3.39)\\
&= \frac{\exp(\beta_k \cdot x) \exp(\beta_c \cdot x) x_{ni}}{\sum_{c'} \exp(\beta_{c'} \cdot x)} && (3.40)\\
&= \mathsf{P}(c|x_n) \mathsf{P}(k|x_n) x_{ni} && (3.41)
\end{aligned}
$$

otherwise.

Then we have

$$
\operatorname{Var}\left[\hat{\mathsf{P}}(k|x_n)\right] \simeq g_n(k)' F^{-1} g_n(k) \qquad (3.42)
$$

from the asymptotic normality of the parameter vector β. Defining $A_n(k) = g_n(k) g_n(k)'$, $A(k) = \sum_n A_n(k)$ and $A = \sum_k A(k)$ we arrive at an A-optimality formula analogous to the backpropagation A-optimality in both form and motivation:

$$
\begin{aligned}
\sum_{n,k} \operatorname{Var}\left[\hat{\mathsf{P}}(k|x_n)\right] &\simeq \sum_{nk} g_n(k)' F^{-1} g_n(k) && (3.43)\\
&= \sum_{nk} \operatorname{tr}\left\{g_n(k) g_n(k)' F^{-1}\right\} && (3.44)\\
&= \sum_{nk} \operatorname{tr}\left\{A_n(k) F^{-1}\right\} && (3.45)\\
&= \sum_{k} \operatorname{tr}\left\{A(k) F^{-1}\right\} && (3.46)\\
&= \operatorname{tr}\left\{A F^{-1}\right\}. && (3.47)
\end{aligned}
$$

The A-matrix, as with the Fisher information matrix, has a block pattern.

The formula for A-optimality looks deceptively similar to the formula derived in Chapter 2 for multilayer neural networks; both the A and F matrices have definitions that are dependent on the details of the model from which they are derived.

3.7.4 Proposal: An Approximation

Both the Fisher information matrix and A matrix scale in memory usage by: $O(k^2d^2)$ where k is the number of categories and d is the number of predictors. For many of domains of interest, d is quite large, and as k reaches a moderate size the matrices will not fit in memory. We propose using only the block diagonal structure of both matrices, so that the scaling becomes $O(kd^2)$. In computing the Fisher information matrix, we compute only the block diagonal matrix. In computing the A matrix, compute only $g_{nci}(k)$ where $k = c$. Using the block diagonals is a heuristic but necessary approximation. It should do better than an even more obvious heuristic approximation which is to train k independent binary logistic regressions, and perform A-optimality on each regression separately. We may use this latter technique as a straw man.

3.8 Summary

We have presented a novel method for active learning of logistic regression classifiers based on the A-optimality objective function from the optimal experimental design literature. Empirical evaluations for the binary case on four data sets demonstrate that A-optimality gives attractive performance gains, and does not perform worse than random sampling. We believe that active learning methods must offer solid performance gains in addition to robust performance in the worst case in order to be accepted in industrial applications. The performance of A-optimality suggests that expected variance reduction objective functions for regression models contain both of these properties.

The final dissertation will include evaluation of the multi-category case in addition to the binary classification task. We will use a block diagonal approximation to the Fisher information matrix to achieve this.

Chapter 4

Scaling Up to Large Data Sets

In this chapter we explore methods for scaling A-optimality to large data sets. The notion of "large" can mean two things. First, there is the number of predictors in the model which determines the dimensions of F, and therefore impacts the computation time of inverting F. Large can also refer to the number of observations in the training set. Provided the pool is large enough to represent the underlying distribution, the size of the pool itself is not terribly important since we may perform Monte Carlo sampling to compute the A matrix in real time.

The methods presented in this chapter have not yet been tested, and so this chapter represents a good part of the "proposal part of the dissertation proposal."

There are five optimizations we will use to compute A-optimality. We begin with a list of the simpler optimizations. The first optimization is introduced in Chapter 3, where we perform block diagonal approximation to F when the number of categories is greater than 2. Another optimization is in re-training the model; we intend to seed parameters with a previous solution since the likelihood is convex and we believe that most parameters will be close to their final values already. A third optimization applies in sparse feature vector settings such as document classification. In a random sample of 200 out of a pool of several thousand documents, most predictors will only take on the value 0. Logistic regression sets the weights to zero of any such predictors, and therefore it is expedient to remove zero-constant predictors from consideration in computing A-optimality.

The next two optimizations are a little more complex, and are the primary focus of this chapter. We divide the computation of A-optimality into two cases: when the size of the training set is small, for instance < 200, and when the training set is moderate to large, ≥ 200. We introduce these methods using binary logistic regression, for simplicity, though the methods scale with little effort to multinomial regression when using the block diagonal approximation proposed in Chapter 3.

4.1 A Method for Small Training Sets

Reintroducing the notation:

$$\phi(X, \mathbf{y}) \doteq \operatorname{tr}\left\{AF^{-1}\right\} \tag{4.1}$$

In computing the A-optimality scoring function:

$$\begin{aligned} \phi(\mathcal{T}, \mathbf{y}, x_n) &= \hat{\mathsf{P}}(y_n = 1 | x_n)\phi(\mathcal{T} \cup \{x_n\}, \mathbf{y} \cup \{1\}) \\ &+ \hat{\mathsf{P}}(y_n = 0 | x_n)\phi(\mathcal{T} \cup \{x_n\}, \mathbf{y} \cup \{0\}). \end{aligned} \tag{4.2}$$

it is clear that computing $\operatorname{tr}\{AF^{-1}\}$ will be a limiting step.

This section explores the Matrix Inversion Lemma as a method for speeding up the computation of F^{-1} and therefore the A-optimality scores for the binary logistic regression model.

4.1.1 How the Lemma Applies

Lemma 1 *Let $F = (R^{-1} + X'DX)$ where R and D are both diagonal, and X is a $T \times D$ matrix with $T << D$. Then*

$$F^{-1} = R + RX\left(D^{-1} + XRX'\right)^{-1} XR, \tag{4.3}$$

an $O(T^3 + TD^2)$ *operation.*

The lemma is known by many names including the Sherman-Morrison-Woodbury formula [21].

We have yet to show that the Fisher information matrix F of a binary logistic regression can be constructed in such a way that the lemma may be applied. Let R^{-1} be the diagonal matrix with elements $\frac{1}{\sigma_p^2}$. Let D be a diagonal matrix with the n, n element equal to $\sigma_n(1 - \sigma_n)$. Let X be the design matrix of the logistic regression classifier. Then the Fisher information is defined as stated in in the lemma, as can be verified by reexamining Equation 3.9.

Once F^{-1} and A are computed, computing $\text{tr}\{AF^{-1}\}$ is an $O(D^2)$ operation.

4.2 Low Rank Methods for a Sparse Design Matrix

This dissertation hopes to focus special attention on the document classification domain. This domain is attractive because it serves as a preliminary step towards robust active learning methods for more complicated natural language processing (NLP) problems, such as part of speech tagging and parsing. A trait shared by many NLP problems is the use of mostly sparse data. For example, in document classification, the design matrix is represented by a document/term frequency matrix, and most terms do not appear in any given document. Sparse structure is noteworthy because it often leads to efficient matrix computations.

Even on data such as the document classification domain where the design matrix is a sparse document/term frequency matrix, F is generally dense and any interesting matrix operation (*e.g.* matrix multiplication, inversion) will take roughly $O(N^3)$. It is possible we can represent F as $Z'Z$ and then find the eigenvalues and eigenvectors of F through operations on Z. The advantage is that Z is derived analytically rather than numerically, and will be sparse in cases where the design matrix is. With the

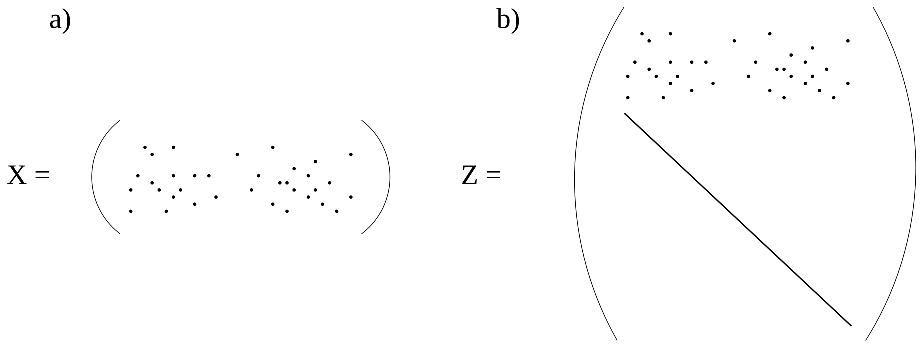


Figure 4.1: Illustration of the design matrix a) and the matrix z b)

eigenvectors and eigenvalues in hand, we produce a low rank approximation of F^{-1} for use in implementing A-optimality using the spectral decomposition of F [41].

This section begins by demonstrating a decomposition of F into $Z'Z$. We then review mathematical results related to the spectral theorem and matrix inversion.

A Decomposition of F

We start by showing a decomposition of F into $Z'Z$. From chapter 3 we are familiar with the definition of F as:

$$F = \left[\sum_n x_n x'_n \sigma_n (1 - \sigma_n)\right] + \frac{1}{\sigma_p^2}\mathbf{I}. \tag{4.4}$$

It will be convenient to rewrite this:

$$F = (X'DX + R) \tag{4.5}$$

where X is the design matrix, whose n'th row is formed from x'_n, D is a diagonal matrix with $D_{nn} = \sigma_n(1 - \sigma_n)$ and R is a diagonal matrix with $R_{dd} = \frac{1}{\sigma_p^2}$.

For document classification data sets, the design matrix encodes different documents (rows) and word token counts (columns), usually using sparse matrix data structure to save both time and space in computation. Figure 4.1 a) gives an illustration of this matrix. The design matrix dimensions are $T \times D$ where T is the training set size and D are the number of predictors, *i.e.* distinct word tokens in document classification domains.

Z is a matrix with dimensions $(T + D) \times D$ defined in the following way:

$$Z_{nd} = \begin{cases} X_{nd}\sqrt{\sigma_n(1-\sigma_n)} & n \leq T. \\ \frac{1}{\sigma_p^2} & n > T \text{ and } d = n - T + 1 \\ 0 & n > T \text{ and } d \neq n - T + 1 \end{cases} . \tag{4.6}$$

A visual illustration comparing X and Z is given in Figure 4.1. A more explicit visual representation of Z is given by the block matrix:

$$Z = \begin{pmatrix} X \\ -- \\ \frac{1}{\sigma_p^2} I \end{pmatrix} . \tag{4.7}$$

By algebra, $Z'Z$ equals F as defined in Equation 4.5. Furthermore, Z is about as sparse as F since the last D rows encode a diagonal matrix, and the top T rows share X's sparsity pattern.

Matrix Computations

The recent spate of progress in computing principal components without explicitly computing the covariance matrix [33, 45] is the pattern we will follow in developing a routine for computing eigenvectors and eigenvalues for $(Z'Z)$ using only Z. The eigenvectors will not be identical to principal component analysis (PCA) dimensions since (PCA) requires a first step of mean-centering Z. The procedure for computing the decomposition may prove to be the same, however. A quick experiment using Roweis's PCA model-fitting procedure has given promising results. In the final dissertation a complete report will be prepared. Using intelligent representation for the sparsity structure, row-rank PCA-fitting methods that already prove attractive on dense Z should prove even more attractive for computing one side of the singular value decomposition when Z is sparse.

Low Rank Inversion

Theorem 1 Spectral Decomposition. *Let X be a symmetric matrix. Then X may be decomposed $X = Q'\Lambda Q$, where Q are orthonormal eigenvectors of X and Λ is a diagonal matrix encoding corresponding eigenvalues.*

By convention, we assume that the eigenvalues are ordered left to right by decreasing magnitude.

Corollary 1 Inversion. *The inverse of a symmetric matrix X may be computed from the spectral decomposition: $Q\Lambda^{-1}Q'$.*

We propose using only the dominant eigenvalues and eigenvectors in computing the inversion of F, producing a low rank approximation. With F^{-1} computed $\text{tr}\{AF^{-1}\}$ is an $O(D^2)$ rather than $O(D^3)$ operation.

4.3 Summary

We have proposed two methods to achieve our goal of applying A-optimality to large data sets. Many of the published active learning evaluations use only $100-200$ samples from the pool, and we believe this scenario could be handled through the Sherman-Woodbury-Morrison formula. Low rank approximations to F^{-1} also look promising.

Chapter 5

Evaluations

This is a placeholder for what will become a chapter of evaluations.

We hope to evaluate on several large document classification data sets, in addition perhaps to running the methods on a good portion of the UC Irvine repository. We intend to compare against several alternatives including uncertainty sampling, and query-by-committee using the Jensen-Shannon divergence as a measure of disagreement.

In addition to various classification accuracy scores such as Accuracy, F-score, micro and macro F score, we hope to measure the quality of the calibration of the finally trained model.

Chapter 6

Conclusion

The claim of this dissertation is that experimental design methodologies and particularly A-optimality offer performance advantages over heuristic methods. A series of matrix optimizations outlined in the proposal will aid in performing the evaluations necessary to substantiate the claim. The claim is necessarily restricted to classification using logistic regression which is the focus of our derivations and evaluations within the dissertation, however these evaluations will lead to optimism for experimental design in more general settings. Empirical advantages of A-optimality for active learning are not currently known due to lack of substantial evaluation using any of the machine learning algorithms where A-optimality applies.

There are several technical obstacles to be overcome in achieving our aim of large scale evaluation: primarily they involve making A-optimality tractable to compute on large data sets. A data set can have many observations and/or many predictive features, and both of these definitions of large will be tackled in this dissertation.

Bibliography

[1] N. Abe and H. Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the 15th International Conference on Machine Learning (ICML1998)*, pages 1–10, 1998.

[2] M Banko and E Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39'th Annual ACL Metting (ACL2001)*, 2001.

[3] Eric B. Baum. Neural net algorithms that learn in polynomial time from examples and queries. *IEEE Transactions on Neural Networks*, 2(1), 1991.

[4] Christopher M. Bishop. *Neural Networks for Pattern Recognitiion.* Oxford University Press, 1995.

[5] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[6] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[7] S. Le Cessie and J. C. Van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.

[8] Kathryn Chaloner and Kinley Larntz. Optimal bayesian design applied to logistic regression experiments. *Journal of Statistical Planning and Inference*, 21:191–208, 1989.

[9] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, Vol. 10, No. 3:273–304, 1995.

[10] David A. Cohn. Queries and exploration using optimal experimental design. In *Advances in Neural Information Processing Systems 6*, 1994.

[11] David A. Cohn. Neural network exploration using optimal experimental design. *Neural Networks*, 9(6):1071–1083, 1996.

[12] David A. Cohn. Minimizing statistical bias with queries. In *Advances in Neural Information Processing Systems 9*. MIT Press, 1997.

[13] David A. Cohn. Personal communication, 2004.

[14] Ido Dagan and Sean P. Engelson. Committee-based sampling for training probabilistic classifiers. In *International Conference on Machine Learning*, pages 150–157, 1995.

[15] Robert Davis and Armand Prieditis. Designing optimal sequential experiments for a bayesian classifier. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3), 1999.

[16] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*, volume November. John Wily & Sons, Inc., New York, second edition, 2000.

[17] Valeri V. Fedorov. *Theory of Optimal Experiments*. Academic Press, New York, 1972.

[18] Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.

[19] Kenji Fukumizu. Active learning in multilayer perceptrons. In *Advances in Neural Information Processing Systems 8*, pages 295–301. MIT Press, 1996.

[20] S Geman, E Bienenstock, and R Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

[21] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, 1983.

[22] Rebecca Hwa. Sample selection for statistical parsing. *Computational Linguistics*, 2004. to appear.

[23] Jenq-Neng Hwang, Jai J. Choi, Seho Oh, and Robert J. Marks II. Query-based learning applied to partially trained multilayer perceptrons. *IEEE Transactions on Neural Networks*, 2(1), 1991.

[24] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc., 2001.

[25] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In W. Bruce Croft and Cornelis J. van Rijsbergen, editors, *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 3–12, Dublin, IE, 1994. Springer Verlag, Heidelberg, DE.

[26] David V Lindely. *Bayesian Statistics–a Review.* SIAM, Philadelphia, 1972.

[27] David J. C. MacKay. *Bayesian Methods for Adaptive Models.* PhD thesis, California Institute of Technology, 1991.

[28] David J. C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):698–714, 1992.

[29] David J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):589–603, 1992.

[30] Andrew McCallum and Kamal Nigam. Employing em in pool-based active learning for text classification. In *Proceedings of the 15th International Conference on Machine Learning (ICML1998)*, 1998.

[31] Prem Melville and Raymond J. Mooney. Diverse ensembles for active learning, 2004. Submitted for Publication.

[32] H. Raiffa and R Schlaifer. *Applied Statistical Decision Theory*. The MIT Press, 1961.

[33] Sam Roweis. Em algorithms for pca and spca. In *Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 626–632. MIT Press, 1998.

[34] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proc. 18th International Conf. on Machine Learning*, pages 441–448. Morgan Kaufmann, San Francisco, CA, 2001.

[35] Andrew I. Schein, S. Ted Sandler, and Lyle H. Ungar. Bayesian Example Selection using BaBiES. *Under Review*, 2004.

[36] Andrew I. Schein and Lyle H. Ungar. A-Optimality for Active Learning of Logistic Regression Classifiers. *Under Review*, 2004.

[37] Greg Schohn and David Cohn. Less is more: Active learning with support vector machines. In *Proc. 17th International Conf. on Machine Learning*, pages 839–846. Morgan Kaufmann, San Francisco, CA, 2000.

[38] G. A. F. Seber and C. J. Wild. *Nonlinear Regression*. Wiley series in probability and statistics. Wiley, 1989.

[39] H. S. Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Computational Learning Theory*, pages 287–294, 1992.

[40] H. S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Physical Review A*, 45(8), 1992.

[41] Gilbert Strang. *Linear Alegbra and Its Applications.* Saunders HBJ, third edition edition, 1988.

[42] Masashi Sugiyama and Hidemitsu Ogawa. Incremental active learning for optimal generalization. *Neural Computation*, 12(12):2909–2940, 2000.

[43] Hiroyuki Takizawa, Taira Nakajima, Hiroaki Kobayashi, and Tadao Nakamura. An active learning algorithm based on existing training data. *IEICE Transactions on Information and Systems*, E83-D(1), 2000.

[44] Min Tang, Xiaoqiang Luo, and Salim Roukos. Active learning for statistical natural language parsing. In *ACL 2002*, 2002.

[45] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 21(3):611–622, 1999.

[46] Simon Tong and Daphne Koller. Active learning for parameter estimation in bayesian networks. In *NIPS*, pages 647–653, 2000.

[47] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, 2002.

[48] Isabella Verdinelli. Personal communication, 2004.

[49] Rong-Xian Yue and Fred J. Hickernell. Robust designs for fitting linear models with misspecification. *Statistica Sinica*, pages 1053–1069, 1999.