Likelihood-based Data Squashing: A Modeling Approach to Instance Construction.

David Madigan, Nandini Raghavan, & William DuMouchel
AT&T Labs - Research
{*madigan,raghavan,dumouchel*}*@research.att.com*

Martha Nason & Christian Posse
Talaria, Inc.
{*mnason,posse*}*@talariainc.com*

Greg Ridgeway
University of Washington
greg@stat.washington.edu

September 28, 1999

Abstract

Squashing is a lossy data compression technique that preserves statistical information. Specifically, squashing compresses a massive dataset to a much smaller one so that outputs from statistical analyses carried out on the smaller (squashed) dataset reproduce outputs from the same statistical analyses carried out on the original dataset. Likelihood-based data squashing (LDS) differs from a previously published squashing algorithm insofar as it uses a statistical model to squash the data. The results show that LDS provides excellent squashing performance even when the target statistical analysis departs from the model used to squash the data.

1 Introduction

Massive datasets containing millions or even billions of observations are increasingly common. Such data arise, for instance, in large-scale retailing, telecommunications,

astronomy, computational biology, and internet logging. Statistical analyses of data on this scale present new computational and statistical challenges. The computational challenges derive in large part from the multiple passes through the data required by many statistical algorithms. When data are too large to fit in memory, this becomes especially pressing. A typical disk drive is a factor of $10^5 - 10^6$ times slower in performing a random access than is the main memory of a computer system (Gibson *et al.*, 1996). Furthermore, the costs associated with transmitting the data may be prohibitive. The statistical challenges are many: what constitutes "statistical significance" when there are 100 million observations? how do we deal with the dynamic nature of most massive datasets? how can we best visualize data on this scale?

Much of the current research on massive datasets concerns itself with *scaling up* existing algorithms - see, for example, Bradley *et al.* (1998) or Provost and Kolluri (1999). In this paper we focus on the alternative approach of *scaling down* the data. Most of the previous work in this direction has focused on sampling methods such as random sampling, stratified sampling, duplicate compaction (Catlett, 1991), and boundary sampling (Aha *et al.*, 1991, Syed *et al.*, 1999). Recently DuMouchel *et al.* (1999) [DVJCP] proposed an approach that instead *constructs* a reduced dataset. Specifically their data squashing algorithm seeks to compress (or "squash") the data in such a way that a statistical analysis carried out on the squashed data provides the same outputs that would have resulted from analyzing the entire dataset. Success with respect to this goal would deal very effectively with the computational challenges mentioned above - the entire armory of statistical tools could then work with massive datasets in a routine fashion and using commonplace hardware.

DVJCP's approach to squashing is model-free and relies on moment-matching. The squashed dataset consists of a set of pseudo data points chosen to replicate the moments of the "mother-data" within subsets of a partition of the mother-data. DVJCP explore various approaches to partitioning and also experiment with the order of the moments. On a logistic regression example where the mother-data contains 750,000 observations, a squashed dataset of 8,443 points outperformed a simple random sample of 7,543 points by a factor of amost 500 in terms of mean square error with respect to the regression coefficients from the mother-data. DVJCP provide a

theoretical justification of their method by considering a Taylor series expansion of an arbitrary likelihood function. Since this depends on the moments of the data, their method should work well for any application in which the likelihood is well-approximated by the first few terms of a Taylor series, at least within subsets of the partitioned data. The empirical evidence provided to date is limited to logistic regression.

In this paper we consider the following variant of the squashing idea: suppose we declare a statistical model in advance. That is, suppose we use a particular statistical model to squash the data. Can we thus improve squashing performance? Will this improvement extend to models other than that used for the squashing? We refer to this approach as "likelihood-based data squashing" or LDS.

LDS is similar to DVJCP's original algorithm (or DS) insofar as it first partitions the dataset and then chooses pseudo data points corresponding to each subset of the partition. However the two algorithms differ in how they create the partition and how they create the pseudo data points. For instance, in the context of logistic regression with two continuous predictors, Figure 1 shows the partitions of the two-dimensional predictor space generated by the two algorithms for a single value of the dichotomous response variable. The DS algorithm partitions the data along certain marginal quantiles, and then matches moments. The LDS algorithm partitions the data using a likelihood-based clustering and then selects pseudo data points so as to mimic the target sampling or posterior distribution. Section 2 describes the algorithm in detail.

In what follows, we explore the application of LDS to logistic regression, variable selection for logistic regression, and neural networks.

Note that both the DS and LDS algorithms produce pseudo data points with associated weights. Use of the squashed data requires software that can use these weights appropriately.

2 The LDS Algorithm

We motivate the LDS algorithm from a Bayesian perspective. Suppose we are computing the distribution of some parameter θ posterior to three data points d_1, d_2, and

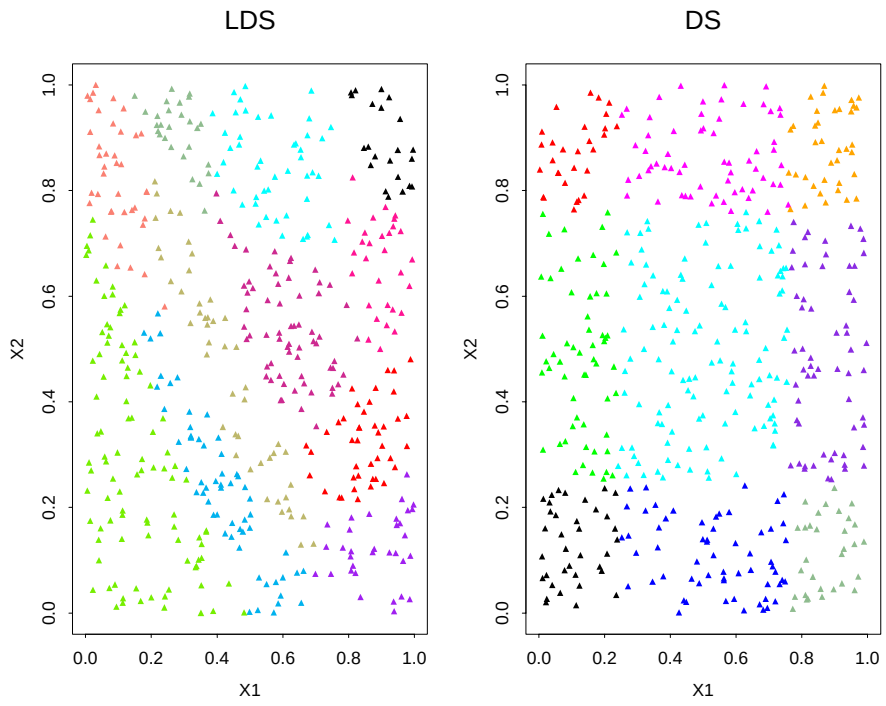

Figure 1: *Data partitions created by LDS and DS*

d_3 (the mother-data). We have:

$$Pr(\theta \mid d_1, d_2, d_3) \propto Pr(d_1 \mid \theta) Pr(d_2 \mid \theta) Pr(d_3 \mid \theta) Pr(\theta).$$

Now suppose $Pr(d_1 \mid \theta) \approx Pr(d_2 \mid \theta)$, at least for the values of θ with non-trivial posterior mass. Then one can *construct* a pseudo data point d^* such that

$$(Pr(d^* \mid \theta))^2 \approx Pr(d_1 \mid \theta) Pr(d_2 \mid \theta).$$

A squashed dataset comprising d^* with a weight of 2 and d_3 with a weight of 1 (see Table 1) will approximate the analysis posterior to the entire mother-data.

In practice, for every mother-data point d_i, LDS first evaluates $Pr(d_i \mid \theta)$ at a set of k values of θ, $\{\theta_1, \ldots, \theta_k\}$ to generate a *likelihood profile* $(Pr(d_i \mid \theta_1), \ldots, Pr(d_i \mid \theta_k))$ for each d_i. Then LDS clusters the mother-data points according to these likelihood profiles. Finally LDS constructs one or more pseudo data points from each cluster and assigns weights to the pseudo data points that are functions of the cluster sizes.

Note that since LDS clusters the mother data points according to their likelihood profiles, the resultant clusters typically bear no relationship to the kinds of clusters

Table 1: *Simple example of squashing when* $Pr(d_1 \mid \theta) \approx Pr(d_2 \mid \theta)$. *LDS constructs the pseudo data point* d^* *so that* $Pr(d_1 \mid \theta)Pr(d_2 \mid \theta)Pr(d_3 \mid \theta) \approx (Pr(d^* \mid \theta))^2 Pr(d_3 \mid \theta)$.

Mother-data		Squashed-data	
Instance	*Weight*	*Instance*	*Weight*
d_1	1	d^*	2
d_2	1		
d_3	1	d_3	1

that would result from a traditional clustering of the data points. Figure 1, for example, shows LDS constructing several clusters containing data points with disparate (x_1, x_2) coordinates. Figure 2 shows the LDS clusters in the context of simple linear regression though the origin (i.e., a model with a single parameter). In this case, the likelihood profiles for each data point d_i represent the likelihoods for d_i with a variety of lines defined by a set of slopes $\{\beta_1, \ldots, \beta_k\}$. The left-hand panel shows mother-data generated from a bivariate normal distribution with zero correlation (i.e., noise) whereas the right-hand panel shows mother-data generated from a model with a true slope of 1. Both plots demonstrate substantial symmetries about the origin - the likelihood of any point (x, y) is the same as that of $(-x, -y)$ for all β_i. Both plots also have a cluster centered on the origin. Since all the lines pass through the origin, points near the origin should have similar likelihoods for all lines. The right-hand panel exhibits distinctive radial clusters, since likelihood in this context is a function of the distance from the data point to the line.

2.1 Detailed Description

Let observations $y = (y_1, \ldots, y_n)$ be realized values of random variables $Y = (Y_1, \ldots, Y_n)$. Suppose that the functional form of the probability density function $f(y; \theta)$ of Y is specified up to a finite number of unknown parameters $\theta = (\theta_1, \ldots, \theta_p)$. Denote by $l(\theta; y)$ the log likelihood of θ, that is, $l(\theta; y) = \log f(y; \theta)$ and denote by $\hat{\theta}$ the value of θ that maximizes $l(\theta; y)$.

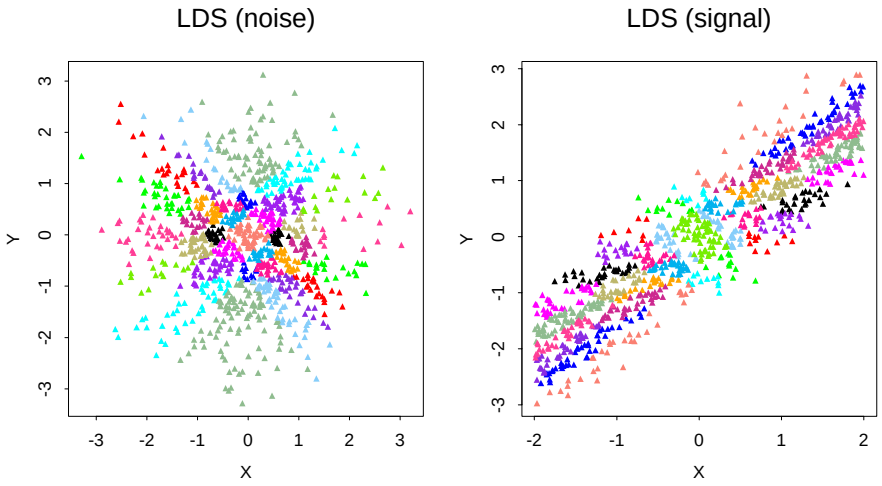

Figure 2: *Data partitions created by LDS and DS*

The base version of LDS (base-LDS) proceeds as follows:

[SELECT] *Select Values of* θ. Select a set of k values of θ according to a central composite design centered on $\check{\theta}$. $\check{\theta}$ is an estimate of $\hat{\theta}$ generally based on at most one pass through the mother-data. A central composite design (Box *et al.*, 1978) chooses $k = 1 + 2p + 2^p$ values of θ: one central point ($\check{\theta}$), $2p$ "star" points along the axes of θ, and 2^p "factorial points" at the corners of a cube centered on $\check{\theta}$. Figure 3 illustrates the design for $p = 3$. This design is a basic standard in response surface mapping (Box and Draper, 1987). Section 3 below addresses the exact locations of the star and factorial points.

[PROFILE] *Evaluate the Likelihood Profiles.* Evaluate $l(\theta_j; y_i)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, k$. In a single pass through the mother-data, this creates a likelihood profile for each observation.

[CLUSTER] *Cluster the Mother-Data in a Single Pass.* Select a sample of $n' < n$ datapoints from the mother-data to form the initial cluster centers. For the remaining $n - n'$ datapoints, assign each datapoint y_i to the cluster c that

minimizes:

$$\sum_{j=1}^{k} \left(l(\theta_j; y_i) - \bar{l}_c(\theta_j;) \right)^2$$

where $\bar{l}_c(\theta_j;)$ denotes the average of the log likelihoods at θ_j for those data points in cluster c.

[CONSTRUCT] *Construct the Pseudo Data.* For each of the n' clusters, construct a single pseudo datapoint. Consider a cluster containing m datapoints, $(y_{i_1}, \ldots, y_{i_m})$. Let y_i^* denote the corresponding pseudo datapoint. The algorithm initializes y_i^* to $\frac{1}{m} \sum_k y_{i_k}$ and then optionally refines y_i^* by numerically minimizing:

$$\sum_{j=1}^{k} \left((m \times l(\theta_j; y_i^*)) - \sum_{k=1}^{m} l(\theta_j; y_{i_k}) \right)^2 .$$

The results reported in this paper do not include this optional step.

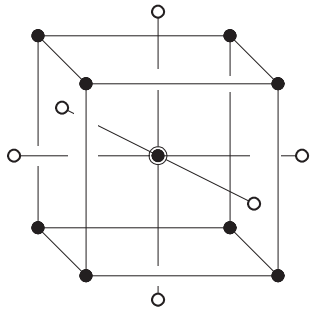

Figure 3: *Central composite design for three variables*

As described, the algorithm requires two passes over the mother-data: one to estimate $\check{\theta}$, and one to evaluate the likelihood profiles and perform the clustering. The first pass can be omitted in favor of an estimate of $\check{\theta}$ based on a random sample, although this can adversely affect squashing performance - see Section 6 below.

There exist a variety of elaborations of the base algorithm, some of which we discuss in what follows. For large p, the central composite design will choose an unnecessarily large set of values of θ at the SELECT phase. The literature on experimental design (see, for example, Box *et al.*, 1978) provides a rich array of fractional factorial designs that efficiently scale with p. The clustering algorithm in base-LDS

can also be improved; Zhang *et al.* (1996) describe an alternative that could readily provide a replacement for the CLUSTER phase. Other elaborations include using alternative clustering metrics at the CLUSTER phase, varying both the number of pseudo points and the construction algorithm at the CONSTRUCT phase, and iterating the entire LDS algorithm. Some but not all of these elaborations require extra passes over the mother-data.

3 Evaluation: Logistic Regression

To evaluate the performance of LDS we conducted a variety of experiments with datasets of various sizes. In each case our primary goal was to compare the parameter estimates based on the mother-data with the corresponding estimates based on the squashed data. To provide a baseline we also computed estimates based on a simple random sample. We provide results both for simulated data and for the AT&T data from DVJCP. Following DVJCP we report results in the form of residuals from the mother-data parameter estimates, that is, (reduced-data parameter estimate - mother-data parameter estimate). The residuals are standardized by the standard errors estimated from the mother-data and are averaged over all the parameters in the pertinent model.

Note that reproducing parameter estimates represents a more challenging target than reproducing predictions since the former requires that we obtain high quality estimates for *all* the parameters. Section 3.4 below shows that accurate parameter estimate replication does result in high quality prediction replication.

3.1 Small-Scale Simulations

Implementation of base-LDS requires an initial estimate $\check{\theta}$ of $\hat{\theta}$ and a choice of locations for the k values of θ used in the central composite design. We carried out extensive experimentation with small-scale simulated mother-data in order to understand the effects of various possible choices on squashing performance.

For the initial estimate $\check{\theta}$ of $\hat{\theta}$ we considered three possibilities: $\hat{\theta}_{\mathrm{SRS}}$, $\hat{\theta}_{\mathrm{ONE}}$, and $\hat{\theta}$. $\hat{\theta}_{\mathrm{SRS}}$ is a maximum likelihood estimator of θ based on a 10% random sample, $\hat{\theta}_{\mathrm{ONE}}$ is an approximate maximum likelihood estimator of θ based on a single step of the

standard logistic regression Newton-Raphson algorithm (this requires a single pass through the mother-data), and $\hat{\theta}$ is the maximum likelihood estimator of θ based on the mother-data.

In the central composite design, let d_F denote the distance of the 2^p "factorial points" from $\check{\theta}$ and let d_S denote the distance of the $2p$ "star" points from $\check{\theta}$, both distances in standard error units. Here we considered $d_F = \{0.1, 0.5, 1, 3\}$ and $d_S = \{0.1, 0.5, 1, 3\}$.

In each case, the mother-data consisted of 1000 observations generated from the following logistic regression model:

$$\log \frac{Pr(Y=1)}{1-Pr(Y=1)} = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 \qquad (1)$$

with $X_1 \equiv 1$, $X_2, X_3, X_4, X_5 \sim U(0,1)$ and $\beta_1, \ldots, \beta_5 \sim U(0, 0.5)$.

For each of 100 simulated mother-datasets from this model, LDS generated 48 squashed datasets corresponding to the 48 ($3 \times 4 \times 4$) design settings. Parameter estimates based on each of these, as well as on an SRS sample were computed. The LDS and SRS datasets were of size 100.

Figure 4 shows boxplots of the standardized residuals of the parameter estimates. The residuals are with respect to the parameter estimates from the mother-data, and are standardized by the standard errors of the estimates from the mother-data.

Several features are immediately apparent:

- With appropriate choices for d_F, LDS outperforms random sampling for all three settings of $\check{\theta}$. Note that the results are shown on a $\log_{10}$ scale; for instance, for LDS-MLE with $d_S = 0.1$ and $d_F = 0.1$, LDS outperforms SRS by a factor of about 10^5.

- Squashing performance improves as the quality of $\check{\theta}$ improves from $\hat{\theta}_{SRS}$ to $\hat{\theta}_{ONE}$ to $\hat{\theta}$.

- There is a dependence between the size of d_F and the quality of $\check{\theta}$. For $\check{\theta} = \hat{\theta}_{\mathrm{SRS}}$, $d_F = 3$ is the optimal setting amongst the four choices. For $\check{\theta} = \hat{\theta}_{\mathrm{ONE}}$, several choices of d_F yield equivalent performance. For $\check{\theta} = \hat{\theta}$, $d_F = 0.1$ is the optimal setting amongst the four choices.

- The choice of d_S has a relatively small effect on squashing performance.

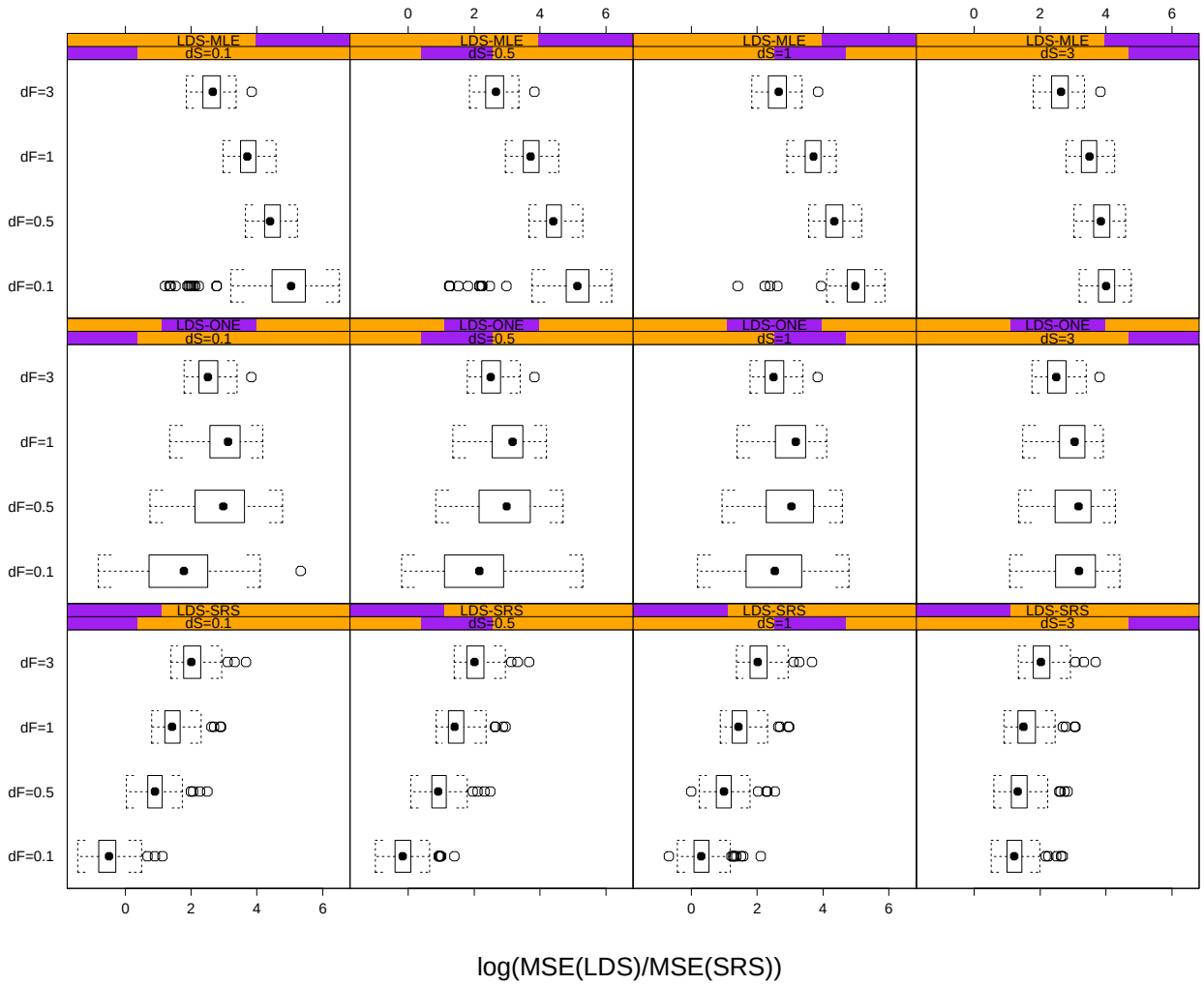

Figure 4: *Small Scale Simulation Results. Each boxplot shows a particular setting of* $\check{\theta}$, d_F, *and* d_S. *The horizontal axes show the log-ratio of the mean square error from random sampling to the mean square error from LDS.*

Since $\check{\theta}$ defines the center of the design matrix where LDS evaluates the likelihood profiles, it is hardly surprising that performance degrades as $\check{\theta}$ departs from $\hat{\theta}$. It is evidently more important to cluster datapoints that have similar likelihoods in the region of the maximum likelihood estimator (which with large datasets will be close to the posterior mean) than to cluster datapoints that have similar likelihoods in regions of negligible posterior mass. What is perhaps somewhat surprising is the extent to which the design points need to depart from $\check{\theta}$ when $\check{\theta} \neq \hat{\theta}$. In that case it is best to evaluate the likelihood profiles at a diffuse set of values of θ most of which are far out in the tails of θ's posterior distribution. In fact, choosing d_S and d_F as large as 10 still gives acceptable performance when $\check{\theta} \neq \hat{\theta}$. This implies that when LDS doesn't have a very good estimate of $\hat{\theta}$, it needs to ensure a very broad coverage of the likelihood surface.

3.2 Medium-Scale Simulations

Here we consider the performance of LDS in a somewhat larger-scale setting. In particular, we simulated mother-datasets of size 100,000 from the logistic regression model specified by (1) again with $X_1 \equiv 1$, $X_2, X_3, X_4, X_5 \sim U(0,1)$ and $\beta_1, \ldots, \beta_5 \sim U(0, 0.5)$. Figure 5 shows the results for different choices of $\check{\theta}$.

Clearly setting $\check{\theta} = \hat{\theta}_{\text{SRS}}$ yields substantially poorer squashing performance than either $\check{\theta} = \hat{\theta}_{\text{ONE}}$ or $\check{\theta} = \hat{\theta}$. However, Section 6 below describes how this can be alleviated with an iterative version of LDS that achieves squashing performance comparable to that for $\check{\theta} = \hat{\theta}$, but starting with $\check{\theta} = \hat{\theta}_{\text{SRS}}$.

Note that even with 100,000 observations the five parameters in the model specified by (1) are often not all significantly different from zero. Experiments with models in which either all of the parameters are indistinguishable from zero or all of the parameters are significantly different from zero yielded LDS performance results that are similar to those reported here. For simplicity we only report the results from model (1).

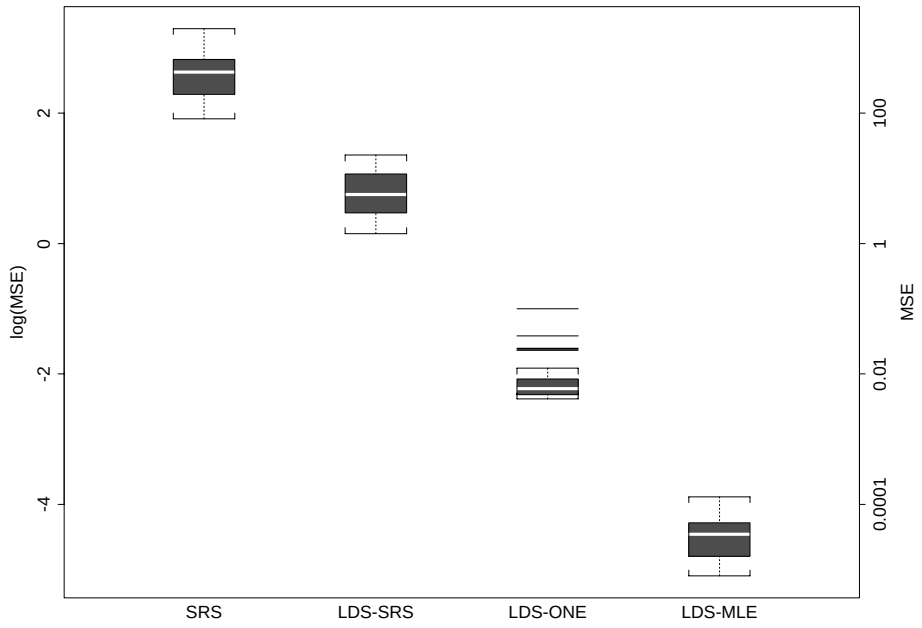

Figure 5: *Performance of Base-LDS for 30 repetitions of the medium-scale simulated data. "SRS" refers to the performance of a 1% random sample. "LDS-SRS" refers to base-LDS with* $\check{\theta} = \hat{\theta}_{\mathrm{SRS}}$ *(i.e., a maximum likelihood estimator of* θ *based on a 1% random sample), "LDS-ONE" refers to base-LDS with* $\check{\theta} = \hat{\theta}_{\mathrm{ONE}}$ *(i.e., a maximum likelihood estimator of* θ *based on a single pass through the mother-data), and "LDS-MLE" refers to base-LDS with* $\check{\theta} = \hat{\theta}$ *(i.e., the maximum likelihood estimator of* θ *based on the mother-data). For LDS-SRS and LDS-ONE we set* $d_F \equiv d_S \equiv 3$ *whereas for LDS-MLE we set* $d_F \equiv d_S \equiv 0.25$*. Note that the vertical axis is on the log scale.*

Table 2: *Performance of Base-LDS for the AT&T data.* k *is the number of evaluations of the likelihood per data point.* $\frac{SRS}{LDS}$ *is the average MSE for simple random sampling (154.04 in this case) divided by the MSE for LDS (i.e., the improvement factor over simple random sampling). HypRect*$(\frac{1}{2})$ *shows the most comparable results from DVJCP (Note that HypRect*$(\frac{1}{2})$ *uses 8,373 observations as compared with 7,450 observations in the other rows).*

k	$\check{\theta}$	d_F	d_S	*MSE*	$\frac{SRS}{LDS}$
85	$\hat{\theta}_{\text{ONE}}$	5	5	0.023	6697
149	$\hat{\theta}_{\text{ONE}}$	**5**	**5**	**0.019**	**8107**
DS HypRect$(\frac{1}{2})$				0.24	642
SRS (10 replications)				154.04	1

3.3 Larger-Scale Application: The AT&T Data

DVJCP describe a dataset of 744,963 customer records. The binary response variable identifies customers who have switched to another long-distance carrier. There are seven predictor variables. Five of these are continuous and two are 3-level categorical variables. Thus for logistic regression there are 10 parameters. As before we consider 1% random and squashed samples. With 10 parameters, the central composite design requires 1,024 factorial points, 20 star points, and 1 central point for a total of 1,045 points. This would incur a significant computational effort. In place of the fully factorial component of the central composite design, we evaluated two fractional factorial designs, a resolution V design requiring 128 factorial points and a resolution IV design requiring 64 points (Box *et al.*, 1978, p.410). In brief, a Resolution V design does not confound main effects or two-factor interactions with each other, but does confound two-factor interactions with three-factor interaction, and so on. A Resolution IV design does not confound main effects and two-factor interactions but does confound two-factor interactions with other two-factor interactions. Table 2 describes the results.

LDS outperforms SRS by a wide margin and also provides better squashing performance than DS in this case.

Table 3: *Comparison of predictions for the AT&T data using logistic regression with all 10 main effects. For each reduced dataset the $N = 744,963$ predictive residuals are defined as (Probability based on reduced dataset) - (Probability based on the mother-data) $\times$ 10,000. Each row of the table describes the distribution of the corresponding residuals for a given reduction method.*

Method	*Mean*	*StDev*	*Min*	*Max*
Random Sample	-41	193	-870	679
LDS	**0.4**	**2**	**-5**	**11**
HypRect($\frac{1}{2}$)	-2	9	-37	34

If the actual parameter estimates from the mother-data are used for $\check{\theta}$ in the first step of the algorithm (i.e. setting $\check{\theta} = \hat{\theta}$), then it is possible to reduce the MSE to 0.01 (k=149). At the other extreme setting $\check{\theta} = \hat{\theta}_{\text{SRS}}$ increases the MSE disimproves to 1.04 (k=149).

3.4 Prediction

Our primary goal so far has been to emulate the mother-data parameter estimates. A coarser goal is to see how well squashing emulates the mother-data predictions. Following DVJCP we consider the AT&T data where each observation in the dataset is assigned a probability of being a *Defector*. We used the parameter estimates from a 1% random sample and from a 1% squashed dataset to assign this probability and the compared these with the "true" probability of being a *Defector* from the mother-data model. For each observation in the mother-data, we compute (Probability based on reduced dataset) - (Probability based on the mother-data), multiplied by 10000 for descriptive purposes. Table 3 describes the results. LDS performs about two orders of magnitude better than simple random sampling and also outperforms the comparable model-free HypRect($\frac{1}{2}$) method from DVJCP.

4 Evaluation: Variable Selection

The preceding results demonstrate that using a particular logistic regression model to squash a dataset allows one to accurately retrieve the parameter estimates for that model with a 1% squashed sample. However, the utility of the algorithm is enhanced by its ability to facilitate other analyses that an analyst might have performed on the mother-data. Since variable selection is a widely used modeling step in regression analysis, we consider the following question: would a variable selection algorithm applied to the squashed data select the same model that the algorithm would select when applied to the mother-data? In what follows we examine all possible subsets of the predictor variables ("all-subsets") and score the competing models using the Bayesian Information Criterion (BIC, Schwarz, 1978). BIC is a penalized log-likelihood evaluated at the MLE:

$$BIC = -2l(\hat{\theta}; y) + p \log(n)$$

where n is the number of datapoints and p is the dimensionality of θ.

For the AT&T data, all-subsets applied to the mother-data, a 1% random sample, and a 1% squashed dataset all select the full model. However the rank correlation between the BIC scores for the mother-data and the BIC scores for the squashed data is 0.9995 as opposed to 0.9922 for the mother-data-SRS comparison.

For the simulated medium-scale mother-data with 100,000 datapoints and 5 predictors (see Section 3.2), a 1% LDS-squashed sample with $\check{\theta} = \hat{\theta}$ selected the correct model in each of 30 replications. By comparison, a 1% SRS selected the correct model in 10 of the 30 replications. Table 4 shows some results.

These results suggest that it is possible to achieve a 100-fold reduction in computational effort for variable selection for certain model classes. This would facilitate the application of expensive variable selection algorithms such as all-subsets or Bayesian model averaging to massive data. Furthermore, the costs associated with transmitting a dataset over a network could be greatly reduced if variable selection is the target activity. Note that for linear and certain non-linear regression models Furnival and Wilson (1974) and Lawless and Singhal (1978) describe a highly efficient approach to variable selection that does not require maximum likelihood estimation for each individual model.

Table 4: *LDS for logistic regression variable selection. "LDS Correct" shows the percentage of the n replications in which LDS selected the correct model (i.e., the model selected by the mother-data). "SRS Correct" shows the percentage of the n replications in which a simple random sample selected the correct model.*

Model: $\text{logit}(Y) = \sum \beta_i X_i$	N	P	n	*LDS Correct*	*SRS Correct*
$\beta_1 = 0.1, \beta_2 = 0.25, \beta_3 = 0.5, \beta_4 = 0.75, \beta_5 = 1.0$	100,000	5	30	100%	33%
$\beta_i \sim \text{unif}(0, 1)$	100,000	5	30	100%	27%
$\beta_i \sim \text{unif}(0, 0.5)$	100,000	5	30	100%	23%

5 Evaluation: Neural Networks

The evaluations thus far have focused on logistic regression. Here we consider the application of LDS (still using a logistic regression model to perform the squashing) to neural networks. We simulated data from a feed-forward neural network with two input units, one hidden layer with three units, and a single dichotomous output unit (Venables and Ripley, 1997). The left-hand panel of Figure 6 compares the test-data misclassification rate using a neural network model based on the mother-data (10,000 points) with the test-data misclassification rate based on either a simple random sample of size 1,000 (black dots) or an LDS squashed dataset of size 1,000 (red dots). In either case, predictions are based on a holdout sample of 1,000 generated from the same neural network model that generated the mother-data. The results are for 30 replications. It is apparent that LDS consistently reproduces the misclassification rate of the mother-data. The right-hand panel of Figure 6 compares the predictive residuals (i.e., (Probability based on reduced dataset) - (Probability based on the mother-data)) for the two methods. Table 5 shows the results in a format comparable with Table 3. These predictive results are not as good as those for the logistic regression analysis of the AT&T data (Table 3), but here the application is to different a model class to that used for the squashing and LDS substantially outperforms simple random sampling nonetheless.

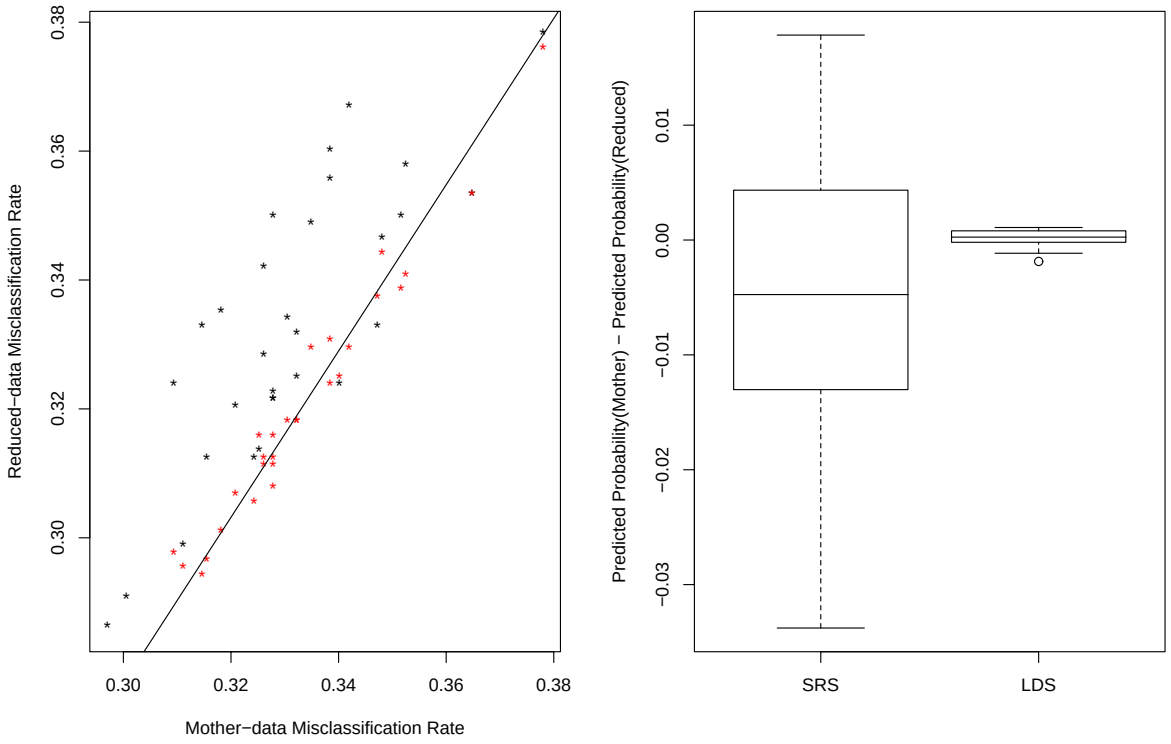

Figure 6: *Comparison of neural network predictions for random sampling and LDS. The left-hand panel shows the misclassification rates for the mother-data predictions versus the reduced-data predictions. The right-hand panel shows the predictive residuals. Both panels reflect performance on 1,000 hold-out datapoints generated from from the same neural network model that generated the mother-data. The figure is based on 30 replications.*

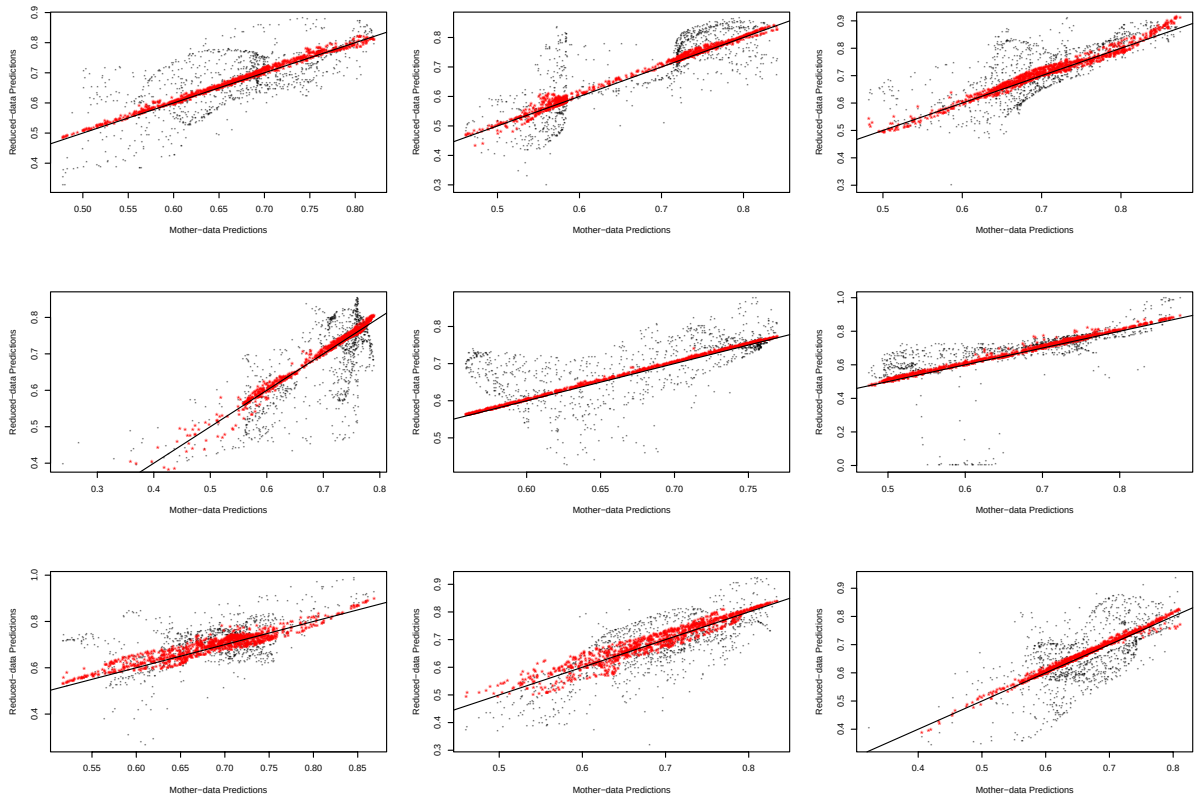

Figure 7: *Comparison of neural network predictions for random sampling and LDS. In each scatterplot, the red dots represent LDS predictions, whereas the black dots represent predictions based on random sampling. The horizontal axis shows the predicted probabilities from the neural network fitted to the mother-data. The vertical axis shows the equivalent predicted probabilities from neural network model fitted to the reduced datasets. The points on the diagonal line are where the predictions agree. The figure shows 9 replications.*

Table 5: *Comparison of neural network predictions for random sampling and LDS. For each reduced dataset the 1,000 residuals from the hold-out data are defined as (Probability based on reduced dataset) - (Probability based on the mother-data). Each row of the table describes the distribution of the corresponding residuals for a given reduction method. The results are averaged over 30 replications.*

Method	*Mean*	*StDev*	*Min*	*Max*
Random Sample	-0.005	0.08	-0.29	0.25
LDS	0.0002	0.02	-0.06	0.07

Figure 7 shows the individual predictions for nine of the replications with LDS predictions (red dots) superimposed on SRS predictions (black dots). Points on the diagonal line represent predictions where the reduced-data prediction and the mother-data prediction agree. The variability of the prediction from random sampling is apparent. Note that for both LDS and SRS, the back-propagation algorithm used to fit the neural network is itself a source of variability since convergence to local log-likelihood maxima frequently occurs.

6 Iterative LDS

Except where noted, the evaluations reported thus far utilize a single pass through the mother-data to compute $\check{\theta}$. In the case of logistic regression, $\check{\theta}$ is the output of the first step of the standard Newton-Raphson algorithm for estimating $\hat{\theta}$. In fact, this provides a remarkably accurate estimate of $\hat{\theta}$ and results in squashing performance close to that provided by setting $\check{\theta} = \hat{\theta}$.

For those cases where there does not exist a high-quality, one-pass estimate of $\hat{\theta}$, and furthermore many passes through the data are required for an exact estimate of $\hat{\theta}$, iterative LDS (ILDS) provides an alternative approach. ILDS works as follows:

1. Set $\check{\theta} = \hat{\theta}_{\text{SRS}}$, an estimate of $\hat{\theta}$ based on a simple random sample from the mother data.

2. Squash the mother-data using LDS (this requires one pass through the moth-

Table 6: *"Cooling" schedule for ILDS*

Iteration	d_F	d_S
1	3	3
2	3	3
3	2	2
4	0.5	0.5
>= 5	0.25	0.25

erdata).

3. Use the squashed data to estimate $\hat{\theta}_{\text{LDS}}$.

4. Set $\check{\theta} = \hat{\theta}_{\text{LDS}}$ and go to (2).

In practice, this procedure requires three or four iterations to achieve squashing performance similar to the performance achievable when $\check{\theta} = \hat{\theta}$ with each iteration requiring a pass through the mother data.

Figure 8 shows the MSE reduction achievable with seven iterations. This is based on a 1% squashed sample from mother-data generated from model (1) with N=100,000 and 30 repetitions. Based on the experiments reported in Section 3.1, we reduced d_F and d_S as the iterations proceeded. Table 6 shows the schedule for results in Figure 8. Generally the performance is not sensitive to the particular schedule although it is important not to reduce d_F and d_S too quickly.

7 Discussion

There are many possible refinements to LDS:

- The clustering algorithm in base-LDS assigns each datapoint y_i to the cluster c that minimizes:

$$\sum_{j=1}^{k} \left(l(\theta_j; y_i) - \bar{l}_c(\theta_j;) \right)^2$$

where $\bar{l}_c(\theta_j;)$ denotes the average of the log likelihoods at θ_j for those data points in cluster c. Note that this approach is independent of the method

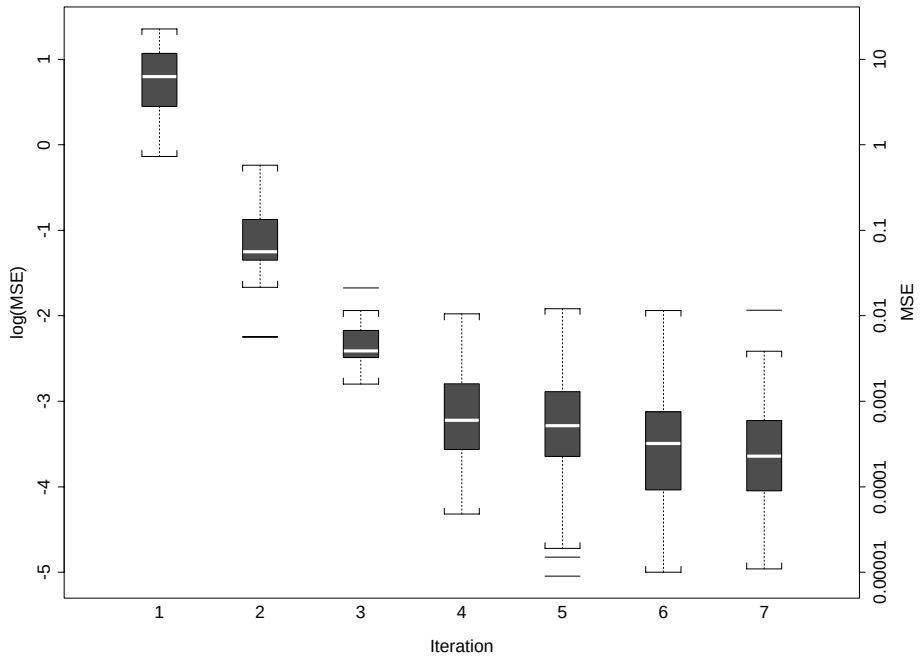

Figure 8: *Squashing performance of ILDS. The first iteration sets $\check{\theta}$ equal to a maximum likelihood estimator of θ based on a 1% random sample. Subsequent iterations set $\check{\theta}$ to the maximum likelihood estimator based on the squashed 1% sample from the previous iteration.*

subsequently used to select the pseudo-data points. An obvious alternative is to instead assign each datapoint y_i to the cluster c that minimizes:

$$\sum_{j=1}^{k} \left(l(\theta_j; y_c^*) - \bar{l}_c(\theta_j;) \right)^2$$

where y_c^* is the current pseudo-point for cluster c. However, as with the similar optional step in the CLUSTER phase of base-LDS, our initial results suggest that the impact on squashing performance is negligible.

- LDS selects a single pseudo-data point per cluster. In contrast DVJCP's approach constructs multiple points per cluster choosing the points to match moments in the mother-data. It is possible to combine both approaches. That is, use DVJCP's moment matching approach to construct points in the LDS-derived clusters. Other approaches include sampling multiple points per cluster or selecting multiple points to minimize the criterion described in the previous point.

- Breiman and Friedman (1984) proposed a squashing methodology they called "delegate sampling." The basic idea is to construct a tree such that datapoints at the leaves of the tree are approximately uniformly distributed. Delegate sampling then samples datapoints from the leaves in inverse proportion to the density at the leaf and assigns weights to the sampled points that are proportional to the leaf density. In principle, this could be combined with either LDS or DS.

Our evaluations of LDS assume that the same response variable is used in both the squashing and the subsequent analysis. When this is not the case we would expect DS to outperform LDS.

Statistical methods that depend strongly on local data characteristics such as trees and non-parametric regression may be particularly challenging for squashing algorithms. A concern is that minor deviations in the location of the squashed data points may result in substantial changes to the fitted model. In this case, a constructive approach to squashing may be more promising than methods based on partitioning.

We have yet to evaluate LDS with a large number of input variables (i.e., large p). In the neural network context, preliminary experiments suggest that the squashing performance of base-LDS for neural networks does degrade as the number of units in the input layer increases. Including interaction terms in the logistic regression model used for the squashing alleviates the problem somewhat.

LDS Software in both C and R is available from `madigan@research.att.com`.

Acknowledgements

We thank Robert Bell, Simon Byers, Daryl Pregibon, Werner Stuetzle, and Chris Volinsky for helpful discussions.

References

Aha, D.W., Kilber, D., and Albert, M.K. (1991). Instance-based learning algorithms. *Machine Learning*, **6**, 37–66.

Box, G.E.P., Hunter, W.G., and Hunter, J.S. (1978). *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building.* John Wiley & Sons, New York, NY, USA,

Box, G.E.P. and Draper, N.R. (1987). *Empirical Model Building and Response Surfaces.* John Wiley & Sons, New York, NY, USA,

Bradley, P.S., Fayyad, U., and Reina, C. (1998). Scaling clustering algorithms to large databases. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 9–15.

Breiman, L. and Friedman, J. (1984). Tool for large data set analysis. In: *Statistical signal processing*, Edward J. Wegman, James G. Smith, Eds., New York : M. Dekker, 191–197.

Catlett, J. (1991). Megainduction: A test flight. In: *Proceedings of the Eighth International Workshop on Machine Learning*, 596–599.

DuMouchel, W., Volinsky, C., Johnson, T., Cortes, C., and Pregibon, D. (1999). Squashing flat file flatter. In: *Proceedings of the Fifth ACM Conference on Knowledge Discovery and Data Mining*, 6–15.

Furnival, G.M. and Wilson, R.W. (1974). Regression by leaps and bounds. *Technometrics*, **16**, 499–511

Gibson, G.A., Vitter, J.S., and Wilkes, J. (1996). Report of the working group on storage I/O issues in large-scale computing. *ACM Computing Surveys*, **28**.

Lawless, J. and Singhal, K. (1978). Efficient screening of nonnormal regression models. *Biometrics*, **34**, 318–327.

Provost, F. and Kolluri, V. (1999). A survey of methods for scaling up inductive algorithms. *Journal of Data Mining and Knowledge Discovery*, **3**, 131–169.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6**, 461–464.

Syed, N.A., Liu, H., and Sung, K.K. (1999). A study of support vectors on model independent example selection. In: *Proceedings of the Fifth ACM Conference on Knowledge Discovery and Data Mining*, 272–276.

Venables, W.N. and Ripley, B.D. (1997). *Modern Applied Statistics with S-PLUS.* Springer-Verlag, New York.

Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: An efficient data clustering method for large databases. *SIGMOD.*