

# Data squashing by empirical likelihood

Art Owen\*

Stanford University

January 2000

## Abstract

Data squashing was introduced by DuMouchel, Volinsky, Johnson, Cortes & Pregibon (1999). The idea is to scale data sets down to smaller representative samples instead of scaling up algorithms to very large data sets. They report success in learning model coefficients on squashed data. This paper presents a form of data squashing based on empirical likelihood. This method reweights a random sample of data to match certain expected values to the population. The computation required is a relatively easy convex optimization. There is also a theoretical basis to predict when it will and won't produce large gains. In a credit scoring example, empirical likelihood weighting also accelerates the rate at which coefficients are learned. We also investigate

---

\*Department of Statistics, Sequoia Hall, Stanford CA 94025, owen@stat.stanford.edu, Tel 650 725-2232, Fax 650 725-8977

the extent to which these benefits translate into improved accuracy, and consider reweighting in conjunction with boosted decision trees.

## 1 Introduction

A staple problem in data mining is the construction of classification rules from data. Some data warehouses are so large, that it becomes impractical to train a classification rule using all available data. Instead a sample of the available data may be selected for training. For instance, the Enterprise Miner from the SAS Institute features the SEMMA process, an acronym in which the leading “S” stands for “sample”.

DuMouchel et al. (1999) introduce “data squashing” to improve upon sampling. Instead of scaling up algorithms to large data sets, one scales down the data to suit existing algorithms. And instead of relatively passive sampling from a large data set, they construct a data set in a way that should make it suitable for training algorithms on.

Suppose that the original data consist of  $N$  pairs  $(X_i, Y_i)$ ,  $i = 1, \dots, N$ . Here  $X_i$  is a vector of predictor variables and  $Y_i$  is a variable to be predicted from  $X_i$ . In data squashing, one constructs a much smaller data set  $(x_i, y_i)$ ,  $i = 1, \dots, n \ll N$ , assigning weights  $w_i$ . There is not necessarily any connection between points like  $x_1$  and  $X_1$  with the same index. Indeed a value like  $x_1$  might not correspond to  $X_i$  for any  $i$ . The idea is that training an algorithm on  $n$  weighted  $(x_i, y_i)$  pairs can be much faster than training on all  $N$  original data points. Large speed gains may be expected when the

squashed data fit in main memory.

Here is an outline of this paper. Section 2 describes data squashing, presents a version using empirical likelihood weights, and points out connections between data squashing, numerical integration, and variance reduction techniques used in Monte Carlo simulation and survey sampling. Finding the empirical likelihood weights reduces to a very tractable convex optimization problem. Empirical likelihood squashing also has theoretical underpinnings that predict when it will and won't work, as outlined in Section 2.

Section 3 describes a credit scoring problem. The data values in it have been simulated and distorted by obfuscating transformations, and the variable names and data source have been hidden for confidentiality. But I am assured that it remains a good test case for algorithms.

Section 4 applies logistic regression to small data samples, with and without empirical likelihood reweighting. The reweighting accelerates the rate at which coefficients are learned. Section 5 replaces logistic regression with boosted decision trees. Section 6 presents our conclusions. We are less pleased with the results of squashing than are DuMouchel et al. (1999), though we describe the sort of problem where we expect squashing to add the most value. Our different conclusions could be due of differences in the algorithms, differences in the way the results are assessed, or simply because the data sets are different.

We conclude this section with some more references. Madigan, Raghavan, DuMouchel, Nason, Posse & Ridgeway (2000) offer a likelihood based form of squashing, geared to exploit a user-specified statistical model. Bradley,

Fayyad & Reina (1998) have goals similar to those of DuMouchel et al. (1999) and Madigan et al. (2000), but instead of representing the data by a weighted set of points, they employ mixture models. The elements in the mixtures include Gaussian distributions, multinomial distributions, and products thereof. Rowe (1983) describes some earlier work in this direction, but the more recent cited work is much more ambitious as befits the greater computational power available today.

## 2 Data Squashing

We begin by outlining the data squashing method of DuMouchel et al. (1999). Then we cast some older methods in a new light, as special cases of data squashing.

Our notation differs somewhat from the original. DuMouchel et al. (1999) do not distinguish predictor and response during squashing, deferring that distinction to the training stage. This allows the same squashed data set to be used for multiple prediction problems. They also choose weights  $w_i$  so that  $\sum_{i=1}^n w_i = N$ , where we scale so that the average weight is 1, that is  $\sum_{i=1}^n w_i = n$ . Many training algorithms are not affected by this scaling, and in any case it is simple to alternate between these conventions.

DuMouchel et al. (1999) choose  $(w_i, x_i, y_i)$  triples as outlined here. The first step is to group the  $(X_i, Y_i)$  vectors into regions. They suggest several ways to construct regions. In the simplest method, the points  $(X_i, Y_i)$  in a region are those that share values for every discrete variable, and also share

values for discretized versions of every continuous variable. For the points in each region, some low order moments of the non-categorical variables are computed. Then for each region, a set of points  $(x_i, y_i)$  and corresponding weights  $w_i$  are chosen, so that the weighted moments on the squashed data match, or nearly match, the unweighted moments on the original data.

For  $m = 1, \dots, M$ , let  $g_m(X_i, Y_i)$  be a function of the  $(X, Y)$  pairs. A moment within a region corresponds to taking for  $g_m$  a product of powers of non-categorical variables, multiplied by a function that is one inside that region and zero outside of it. Let  $Z_{mi} = g_m(X_i, Y_i)$  and  $z_i = g_m(x_i, y_i)$ . Ideal weights would provide a perfect match, with

$$\frac{1}{n} \sum_{i=1}^n w_i g_m(x_i, y_i) = \frac{1}{N} \sum_{i=1}^N g_m(X_i, Y_i) \equiv \bar{Z}_m, \quad (1)$$

and

$$\sum_{i=1}^n w_i = n. \quad (2)$$

Given enough moments and regions, ideal weights are not possible, and Du-Mouchel et al. (1999) minimize

$$\sum_{m=1}^M \omega_m \left( \frac{1}{n} \sum_{i=1}^n w_i g_m(x_i, y_i) - \bar{Z}_m \right)^2 \quad (3)$$

instead. Here  $\omega_m > 0$  with larger values for the lower order moments. The value of (3) is minimized over  $w_i$ ,  $x_i$ , and  $y_i$ , for  $i = 1, \dots, n$ . For a scalar

valued variable, like a person's age, or the number of children in a household, the squashed data value need not match any of the sample values. But it is not allowed to go outside the range of the data. Thus the squashed data may have records with 2.2 children but should not have records with  $-3$  children.

## 2.1 Sampling as squashing

Some issues in data mining echo those of sampling. Two good references on sampling are Cochran (1977) and Lohr (1999).

Simple random sampling can be cast as a trivial version of squashing. Let  $(x_i, y_i)$  for  $i = 1, \dots, n$  be a subset of  $n$  distinct  $(X_i, Y_i)$  pairs, such as a simple random sample (without replacement) of them. Take  $M = 1$  and  $g_1(X, Y) \equiv 1$  and  $w_i = 1$ .

In stratified sampling, the population is partitioned into strata  $h = 1, \dots, H$  containing  $N_h$  elements each. A sample of  $n_h$  values is taken from stratum  $h$  and the weight  $nN_h/(Nn_h)$  makes (1) hold for functions  $g_m$  that are indicators of the strata.

The regression estimator is used in sampling theory to incorporate a known value of some population mean. Suppose that  $(1/N) \sum_{i=1}^N g_m(X_i, Y_i) = \bar{Z}_m$  are known for  $m = 1, \dots, M$ . Then form the weights

$$w_i = 1 + (\bar{Z} - \bar{z})S_{zz}^{-1}(z_i - \bar{z}), \quad (4)$$

where  $z_i = (g_1(x_i, y_i), \dots, g_m(x_i, y_k))'$ ,  $\bar{Z} = (\bar{Z}_1, \dots, \bar{Z}_m)'$ , and

$$S_{zz} = \frac{1}{n} \sum_{i=1}^n (z_i - \bar{z})(z_i - \bar{z})'$$

$$\bar{z} = \frac{1}{n} \sum_{i=1}^n z_i.$$

The regression weights satisfy (1) for  $m = 1, \dots, M$ . The regression estimator can be shown to subsume stratification by introducing indicator variables  $z_m$ . Regression and stratification can also be combined in several ways.

The regression estimator is also widely used in Monte Carlo simulation. There it is known as the method of control variates. Two general references are (Bratley, Fox & Schrage 1987, Ripley 1987). Hesterberg (1995) has a good presentation of the reweighting approach to control variates.

## 2.2 Empirical likelihood squashing

The problem with regression weights is that they can take negative values  $w_i < 0$ , and these may be unusable in some training algorithms. If one insists that  $w_i \geq 0$ , then either there are no solutions to (1) and (2), or else there is an  $n - M - 1$  dimensional family of solutions. When there are no solutions, one might either increase  $n$  or remove some of the moments from consideration.

Suppose that there is an  $n - M - 1$  dimensional family of solutions. It is natural to pick the one that is somehow closest to having equal weights.

The empirical likelihood weights are those that maximize  $\prod_{i=1}^n w_i$  subject to  $\sum_{i=1}^n w_i = n$  and  $\sum_{i=1}^n w_i z_i = n\bar{Z}$ . Owen (1990) describes how to compute these weights. It reduces to minimizing a convex function over a convex domain, which can be taken to be  $M$  dimensional Euclidean space. An Splus function available in <http://www-stat.stanford.edu/~owen> computes the empirical likelihood weights.

Empirical likelihood provides one way of picking the weights  $w_i$  that are closest to equality. One can also use other distance measures, such as the Kullback-Liebler distance  $\sum_{i=1}^n w_i \log(w_i)$  or the Hellinger distance  $\sum_{i=1}^n (w_i^{1/2} - 1)^2$ . Empirical likelihood weights have an advantage in that their computation is slightly simpler than the alternatives. Minimizing the Euclidean distance  $\sum_{i=1}^n (w_i - 1)^2$  is simpler still, but reduces to the regression weights (4) that may be negative (Owen 1991).

### 2.3 Benefits of weighting

Stratification, or more generally regression weighting, has the advantage of reducing the variance of associated estimators. Let  $h(X_i, Y_i)$  be a function of the data cases. Let  $\bar{H} = (1/N) \sum_{i=1}^N h(X_i, Y_i)$  and  $\bar{h} = (1/n) \sum_{i=1}^n h(x_i, y_i)$ .

From a simple random sample, the estimate of  $\bar{H}$  is  $\bar{h}$  which has variance approximately  $\sigma_H^2/n$ , where

$$\sigma_H^2 = \frac{1}{N} \sum_{i=1}^N (h(X_i, Y_i) - \bar{H})^2.$$

The main error in this approximation is a multiplicative factor  $1 - n/N$  which

we take to be virtually one for data squashing.

When  $M = 1$ , the effect of regression weighting is to reduce the variance to  $(1 - \rho^2)\sigma_H^2/n$  where  $\rho$  is the correlation between  $h(X_i, Y_i)$  and  $g_1(X_i, Y_i)$ . For  $M \geq 1$  the reduction factor is  $1 - R^2$  where  $R^2$  is proportion of variance of  $h(X_i, Y_i)$  explained by a linear regression on  $Z_{1i}, \dots, Z_{Mi}$ . Owen (1991) shows that empirical likelihood reduces the asymptotic variance of estimated means by the same factor that regression estimators do.

A training method that estimates means more accurately can often be shown to predict more accurately. In the simplest cases, like linear regression, the prediction is constructed as a smooth function of some sample moments. In more complicated settings, like maximum likelihood estimation a parameter vector  $\theta$  is defined by equations

$$\frac{1}{N} \sum_{i=1}^N \frac{\partial}{\partial \theta} \log f(X_i, Y_i; \theta) = 0, \quad (5)$$

and the estimate  $\hat{\theta}$  solves the equations

$$\frac{1}{n} \sum_{i=1}^n w_i \frac{\partial}{\partial \theta} \log f(x_i, y_i; \hat{\theta}) = 0. \quad (6)$$

Qin & Lawless (1994) show that empirical likelihood weights produce variance reduction for  $\hat{\theta}$  compared to an unweighted estimate. The extent of the reduction depends on how well  $Z_i$  are correlated with the derivatives being averaged in (5). Baggerly (1998) shows that the same reduction holds for other distance measures such as Kullback-Liebler and Hellinger.

## 2.4 Diminishing returns

Better parameter estimates translate directly into better prediction rules, but generally there are diminishing returns. For example, consider a logistic regression in which

$$P(Y = 1 | X) = (1 + \exp(-\beta_0 - X\beta))^{-1},$$

for a parameter vector  $\theta = (\beta_0, \beta)$ . Suppose for simplicity that the logistic regression is in fact accurate, so that knowing  $\theta$  means knowing the Bayes rule. In ordinary sampling the estimate  $\hat{\theta}$  approaches  $\theta$  with an error of order  $n^{-1/2}$ . For weighted misclassification losses, the loss using  $\hat{\theta}$  is then typically  $O(n^{-1})$  above the Bayes loss (Wolff, Stork & Owen 1996). The reason is that at the Bayes rule the derivative with respect to  $\theta$  of the expected misclassification is zero. We expect an error of approximate form  $B_0 + An^{-1}$  if no squashing is used, and of the approximate form  $B_0 + A'n^{-1}$  if regression or empirical likelihood squashing is used, with a fixed list of  $M$  functions. Generally we can expect that  $A' \leq A$ , but if the Bayes error  $B_0$  dominates the estimation error  $An^{-1}$ , then regression or empirical likelihood squashing will bring only a small benefit.

If the logistic model fails to hold, then instead of taking  $B_0$  to be the Bayes error, take it to be the best error rate available within the logistic family.

The squashing method of DuMouchel et al. (1999) adjusts more than the weights. It also estimates new values  $x_i$  and  $y_i$ . Since these are not sampled,

we cannot quote results like those for empirical likelihood squashing.

But we can suppose that searching for  $x_i$ ,  $y_i$  and  $w_i$  should have the effect of matching (or approximately matching) many more than  $M$  functions for the same value of  $n$ . This is similar to the way in which a Gauss quadrature rule that adjusts both the location and weights in numerical integration (Davis & Rabinowitz 1984), integrates higher order polynomials than one that only adjust the weights or the locations. By matching more function values, it should be possible to come closer to the Bayes error rate, but not of course to reduce the Bayes error rate. Thus we could reasonably expect an error of the form  $B_0 + A''n^{-r}$  where either  $r > 1$ , or  $r = 1$  and  $A'' \leq A'$ .

Thus we expect squashing, in its various forms, to be effective in cases where the Bayes error is dominated by sampling or approximation errors. In particular, settings with a zero Bayes error may benefit enormously from squashing.

## 2.5 When to expect benefits

It is reasonable to expect better model coefficients from squashing, albeit with eventually diminishing gains in prediction accuracy. In order to realize gains in the coefficients, they must be related to quantities that are correlated with the values of  $g_m(X_i, Y_i)$ . More precisely, if the vector  $\partial \log f(X_i, Y_i; \theta) / \partial \theta$  is well approximated by a linear combination of  $g_m(X_i, Y_i)$  then we can expect an improved estimate of  $\theta$ .

It helps to distinguish between local and global features of the data. A

logistic regression uses global features of the data. It is reasonable to expect that these features could be highly correlated with judiciously chosen global features  $g_m(X_i, Y_i)$ .

A nearest neighbor method uses local features, such as averages of  $Y_i$  over small regions (determined by  $X$  values). It is not reasonable to expect one of these local averages to be correlated with global features of the data. Therefore squashing with global features  $g_m$  will not help nearest neighbors much, and for an improvement, one must consider ways to employ a large number of local functions  $g_m$ .

A method like a classification tree would seem a priori to be intermediate. The first split is a global feature of the data. The final splits made, at least in a large tree, are very local features. Thus squashing with global  $g_m$  should help on the first splits but not the later ones.

### 3 Example data

This data set inspired by a real commercial problem, but the problem has been disguised (from me) in order to preserve confidentiality.

The training data have 92000 rows and 46 columns. The data arise from a credit scoring problem, but their source is not known, and the data set has been transformed and obfuscated, as described below. Each row of data describes one credit case, and the rows are presented in random order. Each column contains one variable. The response variable is in column 41, and is a 0 or 1 describing bad and good credit outcomes respectively. It may have

been possible to attribute a dollar value to each bad or good outcome, but such dollar values were not in the data I received, and indeed may not have existed.

The data have roughly 85% good cases although this is not necessarily the percentage good in the population. Variables 2 through 40 and 42 through 46 are predictor variables describing the credit history of the case.

The original data values have been transformed. The original values for a given predictor were put into a vector  $v$  of 92000 elements. The transformed values are  $z = [(v - \min(v))/(\max(v) - \min(v))]^p$  where the power  $p$  was chosen at random, independently for each predictor variable. Missing values remained missing after transformation, and did not contribute to the  $\min(v)$  and  $\max(v)$ .

Column 1 is a score variable used to predict the response. It was constructed with the knowledge of what all the input variables mean. An unknown and possibly proprietary algorithm was used to generate this column. This custom-built score serves as a benchmark against which to compare the performance of training methods.

Missing values in the original data were stored as  $-9999.0$ . The missing values are interpreted as “not available” or “not believed”. There are 309262 missing values, about 8.5% of the predictor values. Column 19 was almost 97% missing. Dropping that column and 5 other columns that were more than 10% missing, left the data with 78165 missing values. Values were imputed for the other missing entries as described below. The result is 38 remaining predictors.

Prior to building prediction models with this data, a transformation was applied to each column of predictor values. The non-missing values  $X_{ij}$  were replaced by  $X_{ij}$  raised to a power  $p'$  chosen from among the values  $\{.1, .2, .5, 1, 2, 5, 10\}$ . The value  $p'$  was chosen to maximize a normalized separation of means,

$$\frac{|\hat{E}_j(X_{ij}^p | Y_i = 1) - \hat{E}_j(X_{ij}^p | Y_i = 0)|}{\sqrt{\hat{V}_j(X_{ij}^p | Y_i = 1)/n_{1j} + \hat{V}_j(X_{ij}^p | Y_i = 0)/n_{0j}}}.$$

Here  $\hat{E}_j$  and  $\hat{V}_j$  denote means and variances over pairs  $(X_i, Y_i)$   $i = 1, \dots, N$  with nonmissing  $X_{ij}$ , and  $n_{yj}$  is the number of  $Y_i = y$  for which  $X_{ij}$  is not missing.

Each missing value  $X_{ij}^p$  was simply replaced by an imputed value,  $(\hat{E}_j(X_{ij}^p | Y_i = 1) + \hat{E}_j(X_{ij}^p | Y_i = 0))/2$ . The idea was to replace the missing values by ones that were as neutral as possible regarding the classification at hand. There were 24430 observations with one or more imputed values.

## 4 Logistic regression

The first classification method to be applied was simple logistic regression. The training data contain  $N = 92000$  cases in randomized order. Therefore a simple random sample is obtained by taking  $x_i = X_i$  and  $y_i = Y_i$  for  $i = 1, \dots, n$ . Logistic regression was fit to the first  $n$  cases for  $n = 1000, 2000, 4000, 8000$ , and  $92000$ . Both weighted and unweighted logistic regressions were run. For  $n = N$ , all the weights are 1.0, making the weighted

and unweighted analyses identical.

The weights were chosen so that for each  $j = 1, \dots, 38$  and each  $y \in \{0, 1\}$ , the weighted mean of those  $x_{ij}$  with  $y_i = y$  matched the unweighted mean of those  $X_{ij}$  with  $Y_i = y$ . The reason for such a choice is as follows. Some simple global classifiers are based solely on response group conditional means, variances and covariances of predictors, so it is reasonable to expect these conditional means to carry some relevant information. There are too many predictor variables to allow use of all of the conditional second moments.

The conditional moments can be matched by imposing equation (1) with  $M = 76$  functions

$$g_{2j-1}(X_i, Y_i) = (X_{ij} - \mu_{0j})(1 - Y_i), \quad (7)$$

$$g_{2j}(X_i, Y_i) = (X_{ij} - \mu_{1j})Y_i, \quad (8)$$

for  $j = 1, \dots, 38$ , taking  $\bar{Z}_j = 0$  and

$$\mu_{yj} = \frac{\sum_{i=1}^N X_{ij} 1_{Y_i=y}}{\sum_{i=1}^N 1_{Y_i=y}}. \quad (9)$$

The weights for  $n = 2000$  are shown in Figure 1. The smallest weight is 0.35 and the largest is 3.25. As  $n$  increases the weights become more nearly equal to one. For  $n = 500$  it was not possible to reweight the data to match the conditional moments, using only positive weights. This is why  $n = 1000$  is the smallest sample size we use.

Figure 2 shows how the Euclidean distance between estimated coefficient vectors and the full data coefficient vector decreases as  $n$  increases. The decrease is faster for the empirical likelihood weighted estimates. In terms of accuracy in estimating coefficients, empirical likelihood weighting increases the effective sample size by roughly 4.

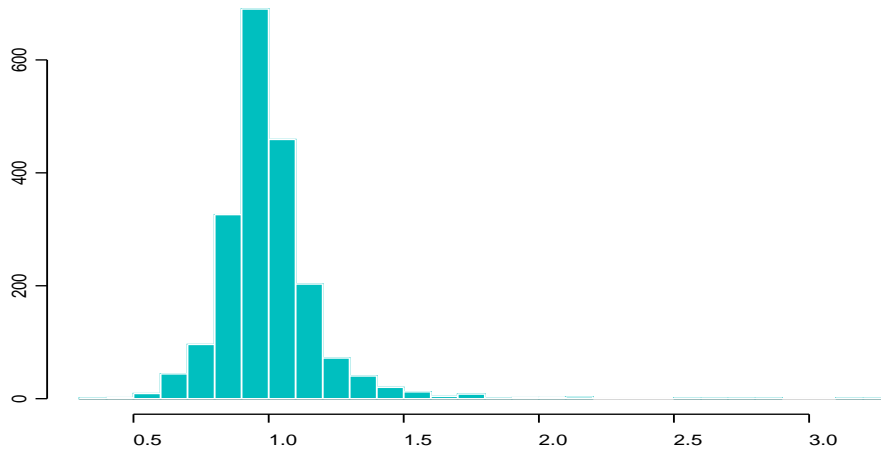


Figure 1: Shown are the empirical likelihood weights for the credit scoring data for  $n = 2000$ .

Increased accuracy in coefficient estimation leads to increased accuracy in classification, but with diminishing returns. Figure 3 shows receiver operating characteristic (ROC) curves for several classifiers, described below, on this data. An ROC curve can be plotted for any classifier that produces a score function  $\phi(X)$  from the predictors. The interpretation of the score

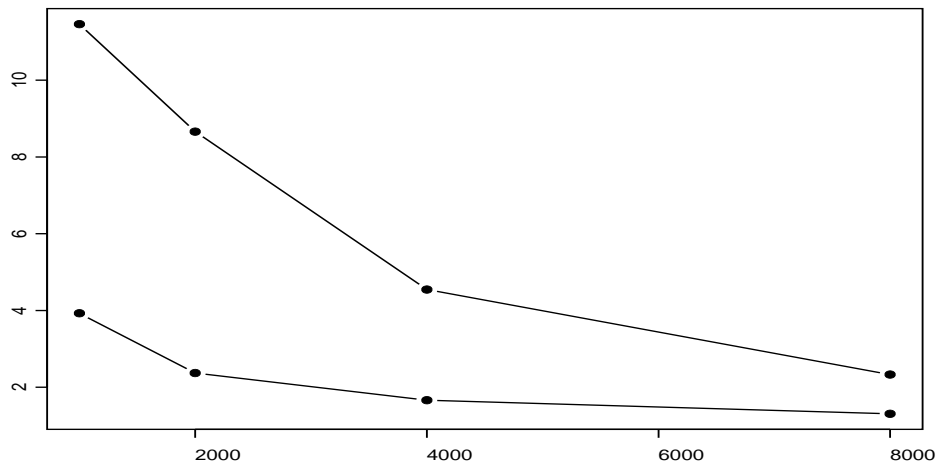


Figure 2: Shown are the distances between the logistic regression coefficients for all 92,000 sample points and those based on subsamples. The lower line is for weighted logistic regressions using empirical likelihood weights.

function is that larger values of  $\phi(X)$  make  $Y = 1$  more likely. A point is classified as  $Y = 1$  if and only if  $\phi(X) > \phi_0$ , where the threshold  $\phi_0$  is chosen to trade off the error rates of false positive and false negative predictions. The ROC curve plots the proportion of the good ( $Y = 1$ ) cases with  $\phi(X) > \phi_0$  versus the proportion of the bad ( $Y = 0$ ) cases having  $\phi(X) > \phi_0$ . As  $\phi_0$  decreases from  $\infty$  to  $-\infty$  the ROC curve arcs from  $(0, 0)$  to  $(1, 1)$ .

The top ROC curve in Figure 3 corresponds to the customized score vector supplied with the data. The other solid lines correspond to empirical likelihood weighted logistic regressions on  $n$  points for  $n = 1000, 2000, 4000, 8000, 92000$ . These lines increase with increasing  $n$ . The dashed lines correspond to unweighted logistic regression for  $n = 1000, 2000, 4000, 8000$ . At  $n = N = 92000$ , the weighted and unweighted ROC curves are the same. There is a reference point at  $(0.2, 0.8)$ . This describes a hypothetical classification in which the rule accepts 80% of the good cases and only 20% of the bad ones. The custom rule is nearly this good.

ROC curves tend to make performance differences among classifiers look very small. Part of the reason is that the underlying probabilities are plotted over ranges from 0% to 100%, while important distinctions among real classifiers can be much smaller than this. For example the difference between 75% and 80% acceptance of good cases, while small on a plot like this, is likely to be of practical importance.

Despite this, it is clear that there are diminishing returns as  $n$  increases, whether weighted or unweighted. Logistic regression on 8000 cases produces an ROC curve that essentially overlaps the logistic regression on all 92000

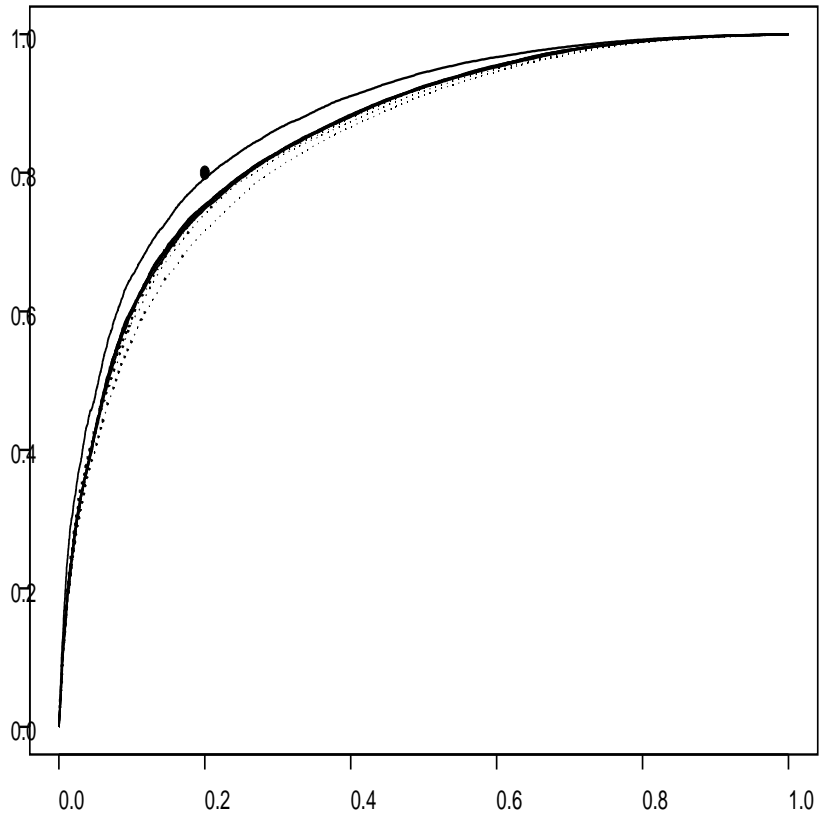


Figure 3: Shown are ROC curves for logistic regressions and a proprietary score. The percent of good cases classified as good is plotted against the percent of bad cases classified as good. For example, the point at (0.2, 0.8) describes an unrealized setting in which 80% of good cases would be accepted, along with only 20% of bad cases. The solid curves, from top to bottom are for: a proprietary score, empirical likelihood weighted logistic regression on samples of sizes 92000, 8000, 4000, 2000, and 1000. The dashed curves, from top to bottom are for unweighted logistic regression on 8000, 4000, 2000, and 1000 cases. The curves overlap significantly, as described in the text.

cases. Empirical likelihood weighting produces such overlap at a smaller sample, perhaps  $n = 2000$ . Although the coefficients keep getting better, performance tends to converge to a limit. It is reasonable to expect that better squashing techniques would get logistic regressions as good as the full data logistic regression at even smaller sample sizes than empirical likelihood weighted logistic regression does.

The ROC curves in Figure 3 are computed on  $N$  points including the  $n$  points used for training. But, there is little risk of overfitting here. The sample sizes  $n$  are all either very large compared to 39 or small compared to  $N$  (or both). As evidence that these logistic regressions do not overfit, notice that logistic regression on all 92000 cases has not produced an ROC curve much better than one on 4000 cases.

## 5 Boosted Trees

Logistic regression is by now a fairly old classification technique. More modern classification methods can also make use of observation weights. We also considered boosted classification trees. Boosted classification trees make predictions by combining a very large number of typically small classification trees. In the extreme, the individual trees have only one split. Taking a weighted sum of such stumps produces an additive model.

Friedman (1999*a*) and Friedman (1999*b*) describe Multiple Additive Regression Tree, or MART, modeling for constructing boosted tree classifiers. This builds on earlier work by Friedman, Hastie & Tibshirani (1999) which

built in turn on Freund & Schapire (1996).

ROC curves were obtained for MART using samples of size 1000, 2000, 4000, 8000 and 92000, using both empirical likelihood weighted and un-weighted analyses. When plotted, these ROC curves tend to be very hard to distinguish from each other as well as from those of logistic regression and the customized method. As in Figure 3 the curves separate the most visually, over the interval between 0.1 and 0.2 on the horizontal axis. Over that range they are roughly parallel with some crossings among close curves.

	0.01	0.05	0.10	0.20	0.50	0.80	0.90	0.95	0.99
Custom	0.217	0.479	0.651	0.792	0.9448	0.99217	0.99761	0.99895	0.999793
Mart	0.190	0.485	0.634	0.774	0.9433	0.99172	0.99733	0.99891	0.999871
Logistic	0.163	0.431	0.604	0.754	0.9244	0.99026	0.99720	0.99881	0.999858
Mart 1	0.132	0.405	0.557	0.732	0.9276	0.98890	0.99651	0.99854	0.999754
Mart 2	0.139	0.394	0.540	0.709	0.9202	0.98750	0.99550	0.99813	0.999690
Mart 4	0.188	0.456	0.626	0.770	0.9361	0.99150	0.99689	0.99877	0.999832
Mart 8	0.189	0.477	0.636	0.774	0.9419	0.99147	0.99707	0.99889	0.999871
Mart 1w	0.143	0.430	0.585	0.745	0.9238	0.98980	0.99656	0.99844	0.999651
Mart 2w	0.178	0.432	0.598	0.750	0.9274	0.98830	0.99571	0.99829	0.999625
Mart 4w	0.170	0.431	0.599	0.753	0.9326	0.98950	0.99624	0.99846	0.999754
Mart 8w	0.183	0.477	0.633	0.775	0.9435	0.99163	0.99720	0.99885	0.999819

Table 1: ROC values for boosted trees. Shown are the heights of 11 ROC curves, corresponding to 11 methods as described in the text. The ROC curves are evaluated at horizontal values given in the top row.

Table 1 show numerical values from these ROC curves. Values smaller than 0.5 are given to 3 significant places, while values close to 1 are given so that their difference from 1 may be computed to 3 significant places. In the region over (0.10, 0.20) both weighted and unweighed MART models tend to

do better on larger sample sizes. The use of weights sometimes helps and sometimes hurts, but does not seem to make much difference. MART models respond to both global and local features of the data. We anticipated that weighting might help the global portion but not the local one. It does not appear that weights greatly accelerate MART.

We also investigated boosted trees using an evaluation copy of Mineset. We were unable to obtain results better than logistic regression for this data, and there did not appear to be any benefit to using empirical likelihood weights, even when boosting stumps (which are global in nature).

## 6 Discussion

The results for empirical likelihood based data squashing are not as encouraging as those in the original paper by DuMouchel et al. (1999). Here we outline the differences, and then describe where more positive results might be expected.

First, they based their comparisons primarily on the quality of estimated logistic regression coefficients. Like them, we get good results for coefficients, but find diminishing returns for classification performance. They also compare predicted probabilities from squashed models to predicted probabilities from the full data set. Such probabilities are deterministic functions of coefficients and so they won't show diminishing returns the way that misclassification rates do.

A second difference is that we report results on some local methods in

addition to global ones, and found little benefit there. This is an area where more ambitious squashing as described in DuMouchel et al. (1999) might be able to make a big improvement.

Thirdly, it is reasonable to expect that optimistic results are entirely appropriate on one data set and not on another. More data sets will need to be investigated. Their data set had only 7 predictors while we used 38. As a consequence they were able to look at interactions, where we did not consider our sample sizes large enough for that. Nor is it reasonable to match all interaction moments in our case. Both data sets were of comparable total size, because they had 744963 records compared to our 92000.

We should point out that the original motivation for squashing is speed, although much of this article stresses accuracy. The reason is that essentially the same speed gains can be achieved by sampling. So for squashing to represent a gain over sampling, it should be more accurate for the same  $n$ .

The diminishing returns suggest that for some small  $n$  squashing could be much better than sampling, but for larger  $n$  the practical value will disappear. This suggests that squashing will be most useful on problems where even when one fills computer memory with data, one is undersampling.

Here are some settings which maximize the promise of squashing. First, problems with near zero Bayes error might benefit more from squashing. Secondly, while in classification one only needs to compute a score on the right side of a threshold, in other problems one must predict a numerical value (e.g. profit versus profitable). Here the diminishing returns might set in much later. Third, when the records have only 7 or 38 predictors a very

large  $n$  will fit in memory. But when the records have many thousands or millions of predictors, much smaller values of  $n$  will fit in memory and there could be more to gain from some form of squashing.

Finally, the squashing described in DuMouchel et al. (1999) might serve as a good data obfuscation device. An organization could release a squashed training data set and a squashed test set for researchers to evaluate learning methods, without ever releasing a single confidential data record.

## Acknowledgements

I thank Bruce Hoadley for valuable discussions on data mining and Jerome Friedman for making available an early version of his MART code. This work was supported by NSF grants DMS-9704495 and DMS-0072445.

## References

- Baggerly, K. A. (1998), ‘Empirical likelihood as a goodness-of-fit measure’, *Biometrika* **85**(3), 535–547.
- Bradley, P. S., Fayyad, U. & Reina, C. (1998), Scaling clustering algorithms to large databases, *in* ‘Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD)’, pp. 9–15.
- Bratley, P., Fox, B. J. & Schrage, L. E. (1987), *A Guide to Simulation (Second Edition)*, Springer-Verlag.

- Cochran, W. G. (1977), *Sampling Techniques (3rd Ed)*, John Wiley & Sons.
- Davis, P. J. & Rabinowitz, P. (1984), *Methods of Numerical Integration (2nd Ed.)*, Academic Press, San Diego.
- DuMouchel, W., Volinsky, C., Johnson, T., Cortes, C. & Pregibon, D. (1999), Squashing flat files flatter, *in* ‘Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining (KDD)’. To appear.
- Freund, Y. & Schapire, R. (1996), Experiments with a new boosting algorithm, *in* ‘Machine learning: proceedings of the thirteenth international conference’, pp. 148–156.
- Friedman, J. (1999*a*), Greedy function approximation: a stochastic boosting machine, Technical report, Stanford University, Department of Statistics.
- Friedman, J. (1999*b*), Stochastic gradient boosting, Technical report, Stanford University, Department of Statistics.
- Friedman, J., Hastie, T. & Tibshirani, R. (1999), Additive logistic regression: a statistical view of boosting, Technical report, Stanford University, Department of Statistics.
- Hesterberg, T. (1995), ‘Weighted average importance sampling and defensive mixture distributions’, *Technometrics* **37**(2), 185–194.

- Lohr, S. (1999), *Sampling: Design and Analysis*, Duxbury Press, Pacific Grove, CA.
- Madigan, D., Raghavan, N., DuMouchel, W., Nason, M., Posse, C. & Ridgeway, G. (2000), 'Likelihood-based data squashing: a modeling approach to instance construction', *Journal of Data Mining and Knowledge Discovery* . To appear.
- Owen, A. (1990), 'Empirical likelihood ratio confidence regions', *The Annals of Statistics* **18**, 90–120.
- Owen, A. (1991), 'Empirical likelihood for linear models', *The Annals of Statistics* **19**, 1725–1747.
- Qin, J. & Lawless, J. (1994), 'Empirical likelihood and general estimating equations', *The Annals of Statistics* **22**, 300–325.
- Ripley, B. D. (1987), *Stochastic Simulation*, John Wiley & Sons.
- Rowe, N. C. (1983), Rule-based statistical calculations on a database abstract, PhD thesis.
- Wolff, G., Stork, D. & Owen, A. (1996), 'Empirical error-confidence curves for neural-network and Gaussian classifiers', *International Journal of Neural Systems* **7**(3), 263–271.