Chapter 1

DATA SQUASHING: CONSTRUCTING SUMMARY DATA SETS

William DuMouchel
AT&T Labs Research
Florham Park, NJ 07932 USA
dumouchel@research.att.com

Abstract A "large dataset" is here defined as one that cannot be analyzed using some particular desired combination of hardware and software because of computer memory constraints. DuMouchel et al. (1999) defined "data squashing" as the construction of a substitute smaller dataset that leads to approximately the same analysis results as the large dataset. Formally, data squashing is a type of lossy compression that attempts to preserve statistical information. To be efficient, squashing must improve upon the common strategy of taking a random sample from the large dataset. Three recent papers on data squashing are summarized and their results are compared.

Keywords: Sampling, substitute dataset, moment matching

1. Introduction: The need for general-purpose data summaries of large datasets

One of the chief obstacles to effective data mining is the clumsiness of managing and analyzing data in very large files. The process of model search and model fitting often require many passes over a large dataset, or random access to the elements of a large dataset. Many statistical fitting algorithms assume that the entire dataset being analyzed fits into computer memory, restricting the number of feasible analyses. Here we define "large dataset" as one that cannot be analyzed using some particular desired combination of hardware and software because of computer memory constraints. There are two basic approaches to this problem: either switch to a different hardware/software/analysis strategy or else

substitute a smaller dataset for the large one. Here we assume that the former strategy is unavailable or undesirable and consider ways of constructing a smaller substitute dataset. This latter approach was named data squashing by DuMouchel et al. (1999). Formally, data squashing is a form of lossy compression that attempts to preserve statistical information. Suppose that the original or "mother" dataset is a matrix Y having N rows or entities and n columns or variables. The squashed dataset is a matrix X having M rows and $n+1$ columns, where $M << N$. The extra column in X is a column of weights, $w_i, i = 1, \ldots, M$, where $w_i > 0$ and $\sum_i w_i = N$. It is assumed that M is small enough so that X can be processed by the desired hardware/software, and that the software can make appropriate use of the weight variable. The n-dimensional distribution of the rows of X weighted by the w_i is intended to approximate the distribution of the rows of Y well enough that statistical analysis of X is an acceptable substitute for the desired analysis of Y.

There are two trivial forms of data squashing that can often be used as comparison or baseline methods. The first is simple random sampling, in which X consists of a random sample of M rows of Y, each given weight $w_i = N/M$. The biggest disadvantage of this strategy is the inaccuracy introduced by sampling variance. Dividing a sample size by 100 multiplies most variances of estimates by 100 as well. With very large initial sample sizes, this may not be a problem for simple estimates such as overall means or proportions or sample correlations. However, for many business purposes, the detection of small differences, or the detection of trends in a small subset of the overall population, is crucial to the success of the data mining project. In such cases the equivalent of throwing away 99% of the data will be unacceptable. The second trivially easy data squashing method might be called unique row extraction, in which X consists of the set of unique rows of Y, and w_i is the multiplicity of the i-th row of X in Y. If the resulting X is small enough to fit in memory, then we have what might be called perfect or lossless squashing. (One might round each quantitative element of Y slightly before extracting the unique rows, so that the rounded values are still considered fully informative for the purposes of statistical analyses, thereby reducing M, the number of unique rows, and thus X.) For a nontrivial application of squashing, we must have a situation where the X from unique row extraction is too large to analyze with the desired hardware and software and where also the analysis results from simple random samples of size small enough to be so analyzed are deemed to be too variable for the purposes of the analyses.

This chapter summarizes the results of three recent papers on data squashing. In addition to DuMouchel et al. (1999), we consider Madi-

gan et al. (1999) and Owen (1999). All these papers are experimental in nature, and none of them make a conclusive case that the data squashing concept is a significant contribution to the practice of data mining. But they do show that for certain analysis goals data squashing can be at least two orders of magnitude more efficient than random sampling. The three papers describe very different methodologies and theoretical rationales for data squashing. Comparing and contrasting them helps to define the limits of data squashing and to suggest future directions for data squashing research.

2. Three data squashing methods

DuMouchel et al. (1999) presents a theoretical framework and justification for data squashing involving a Taylor series representation for the likelihood function from an arbitrary modeling problem. The Taylor series describes the local behavior of the log likelihood function for each fixed parameter value as the continuous variables in X vary. As such, the theory assumes that each column of X must have a restricted range to achieve an accurate approximation, leading to a strategy of defining multivariate bins in every dimension and repeating the data squashing independently within each bin. The conclusion is that if a low-order Taylor series can approximate the contribution to the log likelihood within each bin, then a strategy of choosing X and w_i to match low-order moments within each bin will allow the squashed dataset to approximately duplicate the corresponding analysis of Y. See DuMouchel et al. (1999) for details of this theoretical justification. An implementation of this method involves separate construction of a weighted set of points for each region of Y defined by fixed values of the categorical variables and fixed ranges of all continuous variables. These points might be constructed to match, for example, the means and covariance matrix of the points in the corresponding rows of Y. The construction involves a search in the constrained $m(Q+1)$-dimensional space defined by the m weights and mQ variable values if there are m points to construct and Q continuous variables. The constraints come about because the weights should be positive and the continuous variable values should lie within their respective fixed ranges.

The model likelihood-based method of Madigan et al. (1999) attempts to build a squashed dataset that approximates a specific likelihood function directly, rather than rely on the general Taylor series argument for approximating all possible likelihood functions. In (Madigan et al. 1999) the authors choose logistic regression for a fixed response variable as the specific model around which the squashing is structured. The resulting

squashed dataset may not be as useful as a generically squashed dataset for all possible analyses, but may be more accurate for analysis models similar to logistic regression, for example other classification methods involving the same response variable. The method of approximation involves matching a likelihood profile, that is a vector of values of the log likelihood function at K values in the p-dimensional parameter space of the logistic regression coefficients. The K parameter vectors must be chosen so that a smooth function of p variables can be approximately identified by the corresponding K function values. This is achieved by using the techniques of quadratic response surface estimation from the theory of statistical design of experiments. Whereas DuMouchel et al. (1999) explicitly partitions the data space into bins based on compact regions of the original variables, Madigan et al. (1999) partitions the same data space into regions defined by clustering the likelihood profile vectors contributed by each data point in the mother dataset. Data points having very similar likelihood profiles are deemed equivalent and are merged into a single squashed data point by taking their mean. The corresponding w_i is the number of points so merged. The computations require two one-pass algorithms involving the mother dataset, one to get an approximate estimate of the all-data logistic regression coefficients, and one to perform an approximate clustering of the N likelihood profiles.

In spite of the use of the word "likelihood", the empirical likelihood method of Owen (1999) has more in common with the moment matching method of DuMouchel et al. (1999) than with the model likelihood-based method of Madigan et al. (1999). The methodology of Owen (1999) also directly matches moments of the original variables in the Y dataset to moments in the X dataset. However, Owen (1999) avoids the computationally intensive constrained nonlinear optimizations that Madigan et al. (1999) uses to find solutions. Instead, Owen (1999) starts out with a simple random sample to get the X-values and reweights these sampled points to fit the required moments. The estimation of the w_i involves the maximization of the product of the wi among all weight vectors that satisfy the moment equalities, an algorithm called empirical likelihood estimation and described in Owen (1990). The result is a greatly reduced computational effort. The corresponding downside is primarily that a simple random sample is not an efficient choice of rows of X, so that M must necessarily be large to match a given number of moments.

Table 1.1. Characteristics of the three squashing investigations

	Method	Moment Matching within Bins (DuMouchel et al. 1999)	Model Likelihood-Based (Madigan et al. 1999)	Empericial Likelihood Moment Match. (Owen 1999)
1	Response-variable specific?	No	Yes	Yes [1]
2	Type of functions matched	Moments of raw data of	Likelihood function profile	Moments of raw data
3	No. of matched functions	$\approx Mn$	149 M (10 coefs)	$\approx 2n$
4	Achieve exact match?	No	No	Yes
5	Generate pseudovalues ? and weights for rows?	Generate both	Generate both	Sample pseudovalues, generate weights
6	Computational techniques	Moment computations, constrained nonlinear least squares	One-pass Log. Regr., Likelihood profiles, one-pass clustering	Moment computations, empirical likelihood maximization
7	Computational Effort	High	Medium	Low
8	Largest Y-matrix used in examples	745,000 rows 8 columns	745,000 rows 8 columns	92,000 rows 39 columns
9	Analyses investigated	Logistic Regression (10 coefs and 48 coefs)	Log.Regr.(5 & 10 coefs) Log. Regr. variable sel. Neural Network classif.	Log. Regr. (39 coefs) Boosted decision trees
10	Reduction factor (range)	43 – 341	100	11.5 – 92
11	Efficiency vs. SRS for Log. Regr. coefs.	Up to 656 (10 coefs) Up to 86 (48 coefs)	$\approx 10,000$ (5 coefs) Up to 8,100 (10 coefs)	About 4 (39 coefs)
12	Efficiency for alternate analyses	NA	> 1000? (variable sel.) 16 (neural network)	≈ 1 (boosted trees) ≈ 1
13	Investigate/discuss classification accuracy?	No	Not much	Yes
14	Simulation-based choice of tuning constants?	No	Yes	No
15	Factors investigated	Binning strategies, reduction factor, no. of moments fitted	Likelihood profile settings, initial coefficient estimator	Reduction factor

For example, Owen (1999) reports an example in which it was not possible to reweight a sample of 500 points to match just 78 moments using the empirical likelihood algorithm. Perhaps because of this problem, Owen (1999) chooses to estimate only very low-order moments and also only estimates M points globally, rather than repeat the estimation and moment matching separately within many disjoint regions of the n-dimensional space.

3. Detailed comparison of the three squashing investigations

Table 1.1 provides a comparison of the three squashing methodologies for each of 15 characteristics. This section discusses each row of Table 1.1 in turn.

1. Response-variable specific? As discussed above, the original algorithm (DuMouchel et al. 1999) makes no assumption as to which variable is a response, although the example evaluations of squashing's accuracy all use the same response variable. It is assumed but not proven that similar accuracy would be attained for models using other response variables with the same squashed dataset. On the other hand, the algorithm in (Madigan et al. 1999) is very clearly tuned to the particular response variable defined by the likelihood function used to create likelihood profiles. It is assumed that analyses of the squashed dataset involving other response variables would match those of the mother dataset much more poorly. Owen's discussion in (Owen 1999) focuses on a particular response variable, in particular the way preliminary data transformations and imputation of missing values are carried out, and in his choice of moments to match. However, the empirical likelihood squashing methodology seems perfectly general. If the moments and cross-moments being matched are symmetrically defined as to all the variables, there is no reason to think that the squashing would preferentially work better for any one response variable.

2. Type of functions matched. As discussed above, DuMouchel et al. (1999) and Owen (1999) match moments of the raw data, while Madigan et al. (1999) operates in a sort of dual parameter space, matching points with similar likelihood profiles.

3. No. of matched functions. As discussed in more detail in the paper, the algorithm of DuMouchel et al. (1999) estimates weighted pseudopoints separately within each of many bins. The number of pseudopoints constructed within each bin rises proportional to the log of the number of mother data points in the bin. For more populated bins, the number of moments that the construction attempts to match rises to

"use up" the degrees of freedom available within the bin. The result is that a bin having only one pseudopoint will match the means of each variable only, while bins with many pseudopoints may involve higher order moments even including all 4th-order moments and cross moments. As a rough approximation, the number of moments is about Mn, if there are M pseudopoints and n variables. Since the implementation described by Owen (1999) does not involve separate estimations within bins, and further focuses on estimating the moments that naïve Bayes classifiers would use, there are only $2n$ moments being fit, irrespective of M. In the Madigan et al. (1999) method, there are K values per likelihood profile and the clustering attempts to match them separately for each of M pseudopoints. The product MK is often about $Mp2$, since it takes about $p2$ design points to estimate a quadratic response surface in p dimensions. In their 10-parameter example, Madigan et al. (1999) uses $K = 149$.

4. Achieve exact match? While the DuMouchel et al. (1999) and Madigan et al. (1999) methods involve a great many functions of the data in the matching process, there is no attempt to achieve an exact or globally optimal solution to the matching equations. On the other hand, Owen (1999) requires an exact solution to its specified optimization problem, and therefore requires a much larger M to match a given number of functions. The results in rows 11 and 12 of Table 1.1 seem to indicate that the former strategy is more effective.

5. Generate pseudovalues and weights for rows? As discussed above, DuMouchel et al. (1999) and Madigan et al. (1999) estimate both the values of X and the weights, while Owen (1999), obtains the values of X from a random sample and estimates the weights only. Although this may be inefficient in requiring a larger M, it saves much computation and also ensures that only actual mother data values enter into the squashed dataset. (No families with 2.2 children!)

6. Computational techniques. As discussed above, the DuMouchel et al. (1999) method must collect very many moments and cross moments from the binned mother data and then solve constrained nonlinear least squares problems to match the moments. The Owen (1999) algorithm collects fewer moments and uses the empirical likelihood method to match these moments by reweighting a random sample. The likelihood profile method of Madigan et al. (1999) avoids iterative computations entirely with two one-pass algorithms, the more laborious second one requiring the computation of a likelihood profile for each point and then immediately assigning it to one of M clusters.

7. Computational effort. For the same size M and n, we estimate that DuMouchel et al. (1999) requires the most computational effort to produce the squashed dataset, while Owen (1999) requires the least.

8. Largest Y-matrix used in the examples. DuMouchel et al. (1999) and Madigan et al. (1999) each use the same dataset having about 745,000 rows and 8 columns. The dataset in (Owen 1999) has 92,000 rows and 39 columns. All three squashing methods scale up linearly with respect to the number of rows. Scaling up with the number of columns is more problematical. In unreported preliminary calculations, the efficiency of the DuMouchel et al. (1999) and the Madigan et al. (1999) methods drops off as n increases up to 8, so that, for example, one cannot say how well a 39-column implementation of either of these two squashing methods would work.

9. Analyses investigated. All three papers use logistic regression as the primary analysis for examples. DuMouchel et al. (1999) provides results for both a main-effects model having 10 coefficients and a second-order model having 48 coefficients. Madigan et al. (1999) uses two main-effects models having 5 and 10 coefficients, respectively. In addition, Madigan et al. (1999) investigates the behavior of all-subsets logistic regression variable selection for simulated datasets having 100,000 rows and 5 coefficients, as well as the behavior of a neural network (Venables and Ripley 1997), having two input units, one hidden layer with three units, and a single dichotomous output unit. In their neural network example, the squashed dataset is constructed using a logistic regression profile likelihood, after which its ability to duplicate the neural network of the mother dataset is evaluated. Owen (1999) presents a main-effects logistic regression with 39 coefficients. It also uses the same squashed datasets to estimate boosted decision trees (Friedman 1999a;b) and compare them to those estimated from the mother dataset.

10. Reduction factor (range). The reduction factor is the ratio N/M of the number of rows in Y to the number of rows in X. Examples in DuMouchel et al. (1999) range from 43 to 341, and in Owen (1999) the reduction factor ranges from 11.5 to 92. All the examples in Madigan et al. (1999) have reduction factor equal 100, except for the neural network example reduction factor, which is 10.

11. Efficiency vs. SRS for Log. Regr. coefs. The statistical efficiency achieved for the purpose of estimating p regression coefficients, compared to that expected from random sampling, is defined as

$$\mathit{eff} = Np/(M \sum_{j=1}^{p} (b_j - \beta_j)^2/\sigma_j^2)$$

where b_j is the estimate of the j-th regression coefficient based on the squashed dataset, β_j is the estimated coefficient from the mother dataset, and σ_j is the standard error of the coefficient based on estimation from the mother dataset. If the squashed dataset is a random sample from the mother dataset, the expected value of *eff* is 1. Each of the three papers presents various results with various values of *eff*, depending on different reduction factors, binning algorithms, likelihood profile settings, and so forth. The values in row 11 of Table 1.1 are the maximum reported efficiencies for the indicated value of p. Here we see dramatic differences in the accuracy of the three methods of creating squashed datasets. The model likelihood-based squashing of Madigan et al. (1999) is far more accurate for logistic regression coefficients than the other two methods, and the moment matching within bins of DuMouchel et al. (1999) is more accurate than the empirical likelihood method of Owen (1999). However, in general accuracy decreases with increasing number of coefficients p, and dataset dimension n, so caution in interpreting these numbers is warranted. It is probable that the likelihood profile method benefits "unfairly" from being based on the very model that it is being evaluated on.

12. Efficiency for alternate analyses. For the two papers that presented analyses other than logistic regression, the efficiency of squashing drops dramatically. Owen was unable to detect any consistent estimation advantage of empirical likelihood squashing over simple random sampling for the boosted decision tree analyses. The efficiency of boosting in a three-dimensional neural network simulated dataset with a reduction factor of 10 was about 16 using the likelihood profile method. The definition of efficiency for the ability to replicate a neural network was not based on coefficients but was based on the average squared difference in predictions between the neural network based on the mother dataset and that from the squashed dataset.

13. Investigate/discuss classification accuracy? DuMouchel et al. (1999) and Madigan et al. (1999) focus almost entirely on evaluating data squashing based on accuracy of parameter estimation or comparison of predictions based on Y versus those based on X. This corresponds with the stated goal of data squashing of being able to duplicate the results of an analysis of the large dataset, as opposed to how well one can predict or classify new data. In a high noise environment, prediction or classification error may be quite large even if there is no error in estimating model parameters. Owen (1999) emphasizes this point and the related one that in such cases there may be diminishing returns in any improvement over a simple random sample, if the only goal is to predict or classify new data. The paper shows that ROC curves, for example,

may hardly change even though squashing provides a 4 to 1 improvement in the efficiency of parameter estimation compared to a simple random sample.

14. Simulation-based choice of tuning constants? Madigan et al. (1999) includes the results of preliminary experiments comparing ways of choosing the configuration of parameter values at which the likelihood profile is evaluated. This may have enabled that method to be better tuned than the other two methods.

15. Factors investigated. DuMouchel et al. (1999) reports results from several choices of bin definitions, number of moments fitted, and reduction factor, for both the main effects and second-order logistic regression model. However, these factors were not set up in a factorial design as were the likelihood profile settings comparisons reported in Madigan et al. (1999). Only the reduction factor was varied in Owen (1999).

4. Discussion

The good performance of the model likelihood-based squashing is impressive, making us eager to better understand its limits. Can it scale up to larger numbers of parameters and variables? The length of a likelihood profile vector would need to be greatly increased to handle either the 48-parameter model of DuMouchel et al. (1999) or the 39-parameter model of Owen (1999). Would the one-pass clustering of the profile vectors still work so well? It is remarkable that replacing each cluster by its average data vector leads to very accurate squashing. DuMouchel et al. (1999) assessed the seemingly similar strategy of replacing each Y-matrix bin by the mean of all the data falling in the bin. The results were horrible (far worse than for simple random samples) for both partitions used, one having 394 bins and the other having 3,710 bins in the eight-dimensional data space. This failure is presumably because the set of bin means has much reduced variance for each variable compared to Y, resulting in some severely biased regression coefficients. Yet even crude one-pass clusters in the 149-dimensional likelihood profile space allow the use-the-mean rule to retain almost full accuracy for coefficient estimation. An important question for future research is whether the likelihood profile method can be extended to models for the joint distribution of all variables, avoiding the need to specify a response variable.

Although the method of Owen (1999) seemed to be inefficient compared to the other squashing methods, perhaps it can be modified to improve performance and still retain the benefits of quick computation and having the elements of X be legitimate elements of Y. Reweighting

a stratified sample from Y may greatly improve the performance of the empirical likelihood method. Some as yet unpublished experimentation of our own indicates that taking a random point from each Y-data bin works much better than taking the bin means. The initial weights are the bin sizes, and these weights can be iteratively improved to allow matching of moments or other functions, either by empirical likelihood estimation or by quadratic programming methods. Along these same lines, a method similar to stratified sampling and having about the same goals as data squashing is called delegate sampling, proposed by Breiman and Friedman (1984).

DuMouchel et al. (1999) and Owen (1999) did not address the problem of missing data, while Owen (1999) addressed it in an ad-hoc manner, dropping some variables having lots of missing data, and devising an imputation scheme for filling in missing values in the other variables. For some purposes, it might be desired to have the same distribution of missing data in the squashed data set as in the mother dataset, for example if one wanted to be able to build models for missingness. The original concept of data squashing in DuMouchel et al. (1999) was as a generic data description module, independent of whatever subsequent analyses are planned. It is assumed that an analyst exploring the squashed dataset wants to see everything, warts and all, including the prevalence and patterns of missing data. This highlights the need for squashing techniques that smoothly handle both categorical and continuous variables. The likelihood profile method has a potential weakness here, in that Madigan et al. (1999) does not state what should be done if the clustering of likelihood profiles leads to a pooling of data having different categorical values. (It never happened in their examples.)

Concerning the general research methodology of all three papers, they all used logistic regression as their primary analysis example. This is not because the authors felt a great need for new ways to fit logistic regression to huge datasets. To the contrary, widely available programs like SAS proc logistic (SAS Institute 1998) can handle very large datasets because they do not keep the entire dataset in memory and quickly compute the coefficients in just a few passes over the data. This is convenient for data squashing research, since it allows easy computation of the results on the large dataset to evaluate each squashing technique. Logistic regression, especially a second-order logistic regression that estimates quadratic and interaction terms such the 48-coefficient model in DuMouchel et al. (1999) is viewed as an easy-to-work-with proxy for other highly nonlinear methods for which there may be no available software that avoids the need to keep all the data in memory. To evaluate the performance of data squashing on such latter methods, we must

necessarily restrict the mother dataset to be of manageable size. An interesting research question is how to extrapolate the efficiency measurements in a squashing evaluation. If, for a certain squashing method, $eff = 100$ when $N = 105$ and $M = 103$, what does that say about *eff* when $N = 107$ and $M = 104$? Of course, there can be many alternative definitions of efficiency besides *eff* as defined in the discussion of row 11 of Table 1.1. Regression coefficients from multiparameter models are very sensitive to near collinearities in the data, making the value of *eff* somewhat unstable in such datasets. Such instability could be viewed as a proper challenge for a good squashing technique to meet, since many of the nonlinear techniques such as neural networks or decision trees that one might want to apply to the squashed dataset also have similar instabilities due to their dependence on local properties of the data. A more stable and also more generally applicable measure of squashing efficiency is one that focuses on stability of predictions:

$$pred.eff = \frac{\sum_{i=1}^{N}(p_i^{SRS} - p_i^Y)^2}{\sum_{i=1}^{N}(p_i^X - p_i^Y)^2},$$

where each p_i denotes a prediction of the mean response for row i of the mother dataset, and the subscripts X, Y, and SRS denote predictions based on parameters estimated from the squashed dataset of size M, the mother dataset, and a simple random sample of size M, respectively. For further accuracy in estimating *pred.eff*, the numerator and denominator could each be estimated from the mean of several samples. Note that this measure focuses on the ability of the squashed data to replicate analyses of the mother data, not the ability to predict new data. There will be a different value of *pred.eff* for each combination of predictive method and response variable that is evaluated.

In summary, these papers show that data squashing can be a great improvement over random sampling when the number of variables is not too great or the response variable is fixed. Unfortunately, the need for squashed datasets is greatest when there are tens, hundreds or even thousands of variables. In such diverse datasets usually there are many potential modeling projects involving many choices of response variables. It would be extremely valuable to have a single dataset of manageable size, produced by an enterprise-wide data warehouse resource but available for easy analysis throughout the organization, with the assurance that even reasonably complex relationships among the variables are quite likely to be replicable in the original dataset. We can only hope that the papers reviewed here stimulate more research into extending the reach of data squashing methods.

Bibliography

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

J. Friedman. Greedy function approximation: A stochastic boosting machine. Technical report, Department of Statistics, Stanford University, 1999a.

J. Friedman. Stochastic gradient boosting. Technical report, Department of Statistics, Stanford University, 1999b.

D. Madigan, N. Raghavan, W. DuMouchel, M. Nason, C. Posse, and G. Ridgeway. Likelihood-based data squashing: A modeling approach to instance construction. Technical report, AT&T Labs Research, 1999.

A. Owen. Empirical likelihood ratio confidence regions. *The Annals of Statistics*, 18:90–120, 1990.

A. Owen. Data squashing by empirical likelihood. Technical report, Department of Statistics, Stanford University, 1999.

SAS Institute. SAS Users Manual, 1998.

W.N. Venables and B.D. Ripley. *Modern Applied Statistics with S-Plus*. Springer-Verlag, 1997.