Squashing Massive Data Sets:
An Overview of Existing Methods and Ideas for Further Research

Erlend Berg[*], Xeni Dimakos[†] & Ragnar B. Huseby [‡]

Abstract

Data squashing was introduced by DuMouchel et al. (1999). It is a method for reducing ("squashing") a massive data set to a smaller data set that can be handled by traditional statistical methods. The squashed data are constructed such that some weighted empirical moments are approximately equal in the squashed and original data sets. The squashed data set is generally not a subsample of the original data set.

In this paper we review data squashing. We also discuss several aspects of the method that are not covered in DuMouchel et al. (1999) and present ideas for further research.

[*] Norwegian Computing Center, P.O.Box 114 Blindern, N-0314 Oslo, Norway. Email: erlend.berg@nr.no

[†] Department of Mathematics, University of Oslo, P.O.Box 1053 Blindern, N-0316 Oslo, Norway. Email: xeni@math.uio.no

[‡] Norwegian Computing Center, P.O.Box 114 Blindern, N-0314 Oslo, Norway. Email: ragnar.huseby@nr.no

Contents

1 Introduction

Suppose that we have a data set with so many records that traditional statistical methods of inference are computationally impractical or even infeasible. What do we do?

In simple cases the problem can be solved by increasing computer memory and processing capacity. When this is not sufficient, one idea is to develop specialized methods to deal with very large data sets in reasonable computation time. (For information on data mining, see e.g. Berry and Linoff (1997) or Aas et al. (1999).)

Alternatively, one could imagine extracting a smaller data set from the large, and then employing traditional methodology on the smaller set. It is this data extraction approach that will be discussed here.

Consider a density $f(\mathbf{X};\boldsymbol{\theta})$ with $\boldsymbol{\theta}$ unknown. Suppose we have a large number N of i.i.d. observations $\mathbf{x}_1,\ldots,\mathbf{x}_N$. We wish to find 'observations' $\mathbf{y}_1,\ldots,\mathbf{y}_M$, $M \ll N$, with associated weights $w_1,\ldots,w_M$, such that inferences about $\boldsymbol{\theta}$ on the basis of $\mathbf{y}_1,\ldots,\mathbf{y}_M$ and $w_1,\ldots,w_M$ are similar to inferences made on the basis of the original data $\mathbf{x}_1,\ldots,\mathbf{x}_N$.

Possible ways of determining $\mathbf{y}_1,\ldots,\mathbf{y}_M$ and $w_1,\ldots,w_M$ are:

1. Let $\mathbf{y}_1,\ldots,\mathbf{y}_M$ be a random sample from $\mathbf{x}_1,\ldots,\mathbf{x}_N$, and set $w_i = N/M$ for all i. We refer to this method as *Simple Random Sampling* (SRS). Of course, SRS works even if the form of $f(\cdot)$ is unknown.

2. If all the variables are discrete, then a simple way to *aggregate* them is to create one point $\mathbf{y}_i$ for every combination of values of the variables that occur in the data, and let the weight be the number of original data points with that particular combination. If some variables are discrete and others continuous, we may regionalize the data using the discrete data and applying other techniques within each such region.

3. When the form of $f(\cdot)$ is known, we may use a method called *Likelihood-based Data Squashing* (LDS). In LDS, each $\mathbf{y}_i$ will be the 'center' of a cluster of observations $\{\mathbf{x}_j\}_{j\in C}$. Each cluster consists of observations of similar likelihood. The weights $\{w_i\}$ will be the number of observations in each cluster. The LDS technique is summarized in Appendix B.

4. If $f(\cdot)$ is unknown (but smooth), an alternative to SRS is to determine $\mathbf{y}_1,\ldots,\mathbf{y}_M$ and $w_1,\ldots,w_M$ by requiring the extracted data set to have approximately the same likelihood as the original data set. This method, *Data Squashing* (DS), is the main topic of this document. As for LDS, the squashed set is not generally a subset of the original set: A record in the squashed set is not necessarily equal to any of the records in the original set.

The rest of this document is organized as follows. Section 2 summarizes the original paper, gives some simple examples and discusses some related issues. Section 3 contains some comments and discussions pertaining to DS and DS-type algorithms. Section 4 is a informal collection of ideas for further research in the field.

2 Data Squashing by Moment Matching

2.1 Summary of Data Squashing

Data squashing as presented in DuMouchel et al. (1999) consists of three steps:

- *Group the data in regions.* If some of the variables are discrete (categorical), they are used to bin the data. Each of the resulting regions can be sub-divided in e.g. hyper-rectangles or data spheres (Johnson and Dasu, 1998).
- *Calculate empirical moments for each region.* The number of moments that need to be calculated increases with the number of squashed data points to be generated.
- *Generate pseudo data points and weights.* For each region, a set of pseudo points is created so that their weighted moments match those of the original data. The sum of the weights of the generated data in a region should equal the number of original data points in that region.

The goal is that likelihood-based methods applied to the squashed data and original data should give the same inferences, *regardless of the choice of statistical model.*

Suppose that the original data set has columns (variables)

$$A^1, \dots, A^C, X^1, \dots, X^Q.$$

The As are categorical variables, while the Xs are continuous. Assume that the N original data points are the results of N independent draws from the unknown density

$$f(a_1, \dots, a_C, x_1, \dots, x_Q; \boldsymbol{\theta}).$$

Our goal is to find squashed data $\mathbf{y}_i = (B_{i1}, \dots, B_{iC}, Y_{i1}, \dots, Y_{iQ})$ and weights w_i such that the likelihoods of the real and squashed data sets are equal for any $f(\cdot)$ and $\boldsymbol{\theta}$. That is, we want

$$\prod_{i=1}^{M} f(B_{i1}, \dots, B_{iC}, Y_{i1}, \dots, Y_{iQ}; \boldsymbol{\theta})^{w_i} = \prod_{j=1}^{N} f(A_{j1}, \dots, A_{jC}, X_{j1}, \dots, X_{jQ}; \boldsymbol{\theta}). \quad (1)$$

We now regionalize the data using the categorical variables, so that we have one region for every observed combination of categorical data. Optionally, DuMouchel et al. (1999) suggest that the regions are divided further in the continuous variables. They describe two different techniques for subgrouping the data, *hyper-rectangles* and *data spheres.* Data spheres are also described in Johnson and Dasu (1998).

Let R be the number of regions and N_r be the number of original data points within region r so that $\sum_{r=1}^{R} N_r = N$. For region r we create a squashed data set with $M_r \ll N_r$ points.

We enforce (1) separately in each region. Because of the way we have created the regions, the categorical variables of the original data are constant within them,

$$A_{jc} = A_{j'c} = A_c, \quad \forall c \in \{1, \dots, C\}$$

for any pair of points j, j' in the same region r. We set the categorical variables of the squashed data equal to the categorical variables of the original data,

$$B_{ic} = A_c, \quad \forall c \in \{1, \dots, C\},$$

where i are the indices of the squashed data points in region r.

Taking logarithms, we now have

$$\sum_{i=1}^{M_r} w_i \ln\{f(A_{r1}, \dots, A_{rC}, Y_{i1}, \dots, Y_{iQ}; \boldsymbol{\theta})\} \\ = \sum_{j=1}^{N_r} \ln\{f(A_{r1}, \dots, A_{rC}, X_{j1}, \dots, X_{jQ}; \boldsymbol{\theta})\}. \quad (2)$$

We now replace $\ln\{f(\cdot; \boldsymbol{\theta})\}$ by its Taylor expansion on both sides of the equation. (See Appendix A for a brief review of Taylor expansions.) Since the Taylor coefficients depend only on f, $\boldsymbol{\theta}$, and $(A_1, \dots, A_C)$, which are the same for the original and squashed data, and since we expand around the same point $\mathbf{x}$, the Taylor coefficients will be the same on both sides of the equation. Hence, we may write

$$\sum_{i=1}^{M_r} w_i \sum_{k=1}^{K} g_k \prod_{q=1}^{Q} (Y_{iq} - x_q)^{p_{kq}} \approx \sum_{j=1}^{N_r} \sum_{k=1}^{K} g_k \prod_{q=1}^{Q} (X_{jq} - x_q)^{p_{kq}}$$

for each region r. In the approximation above, $(p_{k1}, \dots, p_{kQ})$, $k = 1, \dots, K$ are Q-vectors of non-negative integers corresponding to the needed powers in Taylor's formula. We change the order of summation to obtain

$$\sum_{k=1}^{K} g_k \sum_{i=1}^{M_r} w_i \prod_{q=1}^{Q} (Y_{iq} - x_q)^{p_{kq}} \approx \sum_{k=1}^{K} g_k \sum_{j=1}^{N_r} \prod_{q=1}^{Q} (X_{jq} - x_q)^{p_{kq}}. \quad (3)$$

If this is to hold for any smooth f, and hence for arbitrary Taylor coefficients g_k, the sums above must be approximated term by term:

$$\sum_{i=1}^{M_r} w_i \prod_{q=1}^{Q} (Y_{iq} - x_q)^{p_{kq}} \approx \sum_{j=1}^{N_r} \prod_{q=1}^{Q} (X_{jq} - x_q)^{p_{kq}} \quad k = 1, \dots, K, \quad (4)$$

for each region r.

Since each k is associated with a moment about $\mathbf{x}$, the above approximations may be interpreted as follows: Fix a moment k and a region r. Calculate the weighted sum of this moment for the squashed data in r and the (unweighted) sum of the same moment for the original data in r. *These two sums should match, for every moment k and region*

r. DuMouchel et al. (1999) emphasize this idea of moment matching and discuss the equivalent equation of likelihoods only as motivation.

The above moment matching criteria are what we use to determine the squashed data points $\{\mathbf{Y}_i\}$ and their weights $\{w_i\}$. In addition, to obtain interpretable results, we require positive weights and 'non-extrapolating' squashed data:

$$\begin{aligned} & w_i \geq 0 \quad \forall i \\ & \min_j X_{jq} \leq Y_{iq} \leq \max_j X_{jq} \quad j \in \{1,\ldots,N_r\}, i \in \{1,\ldots,M_r\} \quad \forall q, \forall r. \end{aligned} \tag{5}$$

The $\{(\mathbf{y}_i, w_i)\}$ that achieve the closest approximations (4) may be determined by the use of a least squares criterion. That is, calculate, once and for all, the moments

$$z_k = \sum_{j=1}^{N_r} \prod_{q=1}^{Q} (X_{jq} - x_q)^{p_{kq}}$$

of the original data. Then, DuMouchel et al. (1999) suggest, determine the squashed data and weights by minimizing the objective function

$$S(\mathbf{Y}, \mathbf{w}) = \sum_{k=1}^{K} u_k \left(z_k - \sum_{i=1}^{M_r} w_i \prod_{q=1}^{Q} (Y_{iq} - x_q)^{p_{kq}} \right)^2, \tag{6}$$

where $\{u_k\}$ are optimization weights. The optimization weights determine the moments that are matched with the highest precision. They might for instance give equal weight to every moment, or require lower-order moments to be matched more closely than higher-order moments, as is done in DuMouchel et al. (1999).

Unless where stated explicitly, we shall in the following assume that the original data consist of only continuous variables. We do this without loss of generality, since the method is performed independently in each region. In each region, the categorical variables are constant. They may therefore be regarded as part of a density function of the continuous variables alone.

2.2 The Number of Equations and Order of the Expansion

Let K be the number of equations needed to find a set of squashed points. Assume that the original data have dimension Q so that the full data set is $\mathbf{X}_j = (X_{j1}, \ldots, X_{jQ})$, $j = 1, \ldots, N$. For simplicity we assume that there is only one region. In order to determine M squashed points $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{iQ})$, $i = 1, \ldots, M$ with weights $w_1, \ldots, w_M$ we need $K \geq M(Q+1)$ equations.

Consider the Taylor expansion that leads to (4). The rth order expansion gives $\binom{Q+r-1}{r}$ equations. So, in order to obtain K or more equations, the Taylor expansion must be taken up to and including order ν, where

$$\nu = \min\{m : \sum_{r=0}^{m} \binom{Q+r-1}{r} \geq M(Q+1)\}.$$

As only $M(Q+1)$ equations are required to solve for our squashed points, we often do not need all terms (combination of derivatives) of order ν. A relevant question is therefore which equations from the νth level that should be included in our set of equations.

(Since we will in practice be performing an optimization, and not solving a system of equations, we *may* use all νth level equations. But this may lead to unnecessarily heavy computation, since the number of Taylor coefficients increases sharply from one order to the next.)

As an illustration, assume that $Q = 4$ and that we want to generate $M = 5$ squashed data points. Then at least $K = 25$ equations are needed. Taylor expansion up to order 2 generates 15 equations. Taylor expansion of order 3 generates 20 additional equations. We need at least 10 of these, but which of them are we to choose? (DuMouchel et al. (1999) seem to give preference to the marginal moments.) It is not clear how our selection will influence the resulting squashed points.

How many squashed data points should we create in each region? The algorithm itself gives no clear answer. DuMouchel et al. (1999) use the somewhat arbitrary formula

$$M_r = \max(1, \alpha \log_2 N_r),$$

with $\alpha > 0$.

2.3 Some Simple Examples

The different complexities of the examples below are the result of varying three parameters of the massive data and the squashing procedure. The first is Q, the dimension of the original data. The second is the number of regions into which we subdivide the data. And the third is the number of Taylor terms we employ. The complexity of the system increases with all of these.

Example 1 In the simplest cases, we deal with continuous, scalar data $X_1, \ldots, X_N$, that is $Q = 1$.

(a)

Consider first using only one region (i.e. no regionalization), and matching Taylor terms up to and including order 1. This yields $K = 2$ equations. Hence, we can only generate one pseudo point $\{Y, w\}$. The equations (4) become :

$$\begin{aligned} w &= \sum_{j=1}^{N} 1 = N, \qquad (k=1); \\ w(Y-x) &= \sum_{j=1}^{N} (X_j - x), \qquad (k=2); \end{aligned}$$

with solution

$$Y = \frac{1}{N}\sum_{j=1}^{N} X_j, \qquad w = N.$$

(b)

To generate more than one pseudo point we can either group the original data in regions or include more terms in the Taylor expansion. Assume now that we want to generate two pseudo points, Y_1 and Y_2 with weights w_1 and w_2. With no assumptions on the distribution of $X_1, \ldots, X_N$, it is natural to let the two regions R_1 and R_2 consist of positive and negative values of the data points. That is $R_1 = \{X_j \leq 0\}$ and $R_2 = \{X_j > 0\}$. Our equations are now

$$\begin{aligned} w_r &= N_r = \sum_{j=1}^{N} I(X_j \in R_r), \qquad (k = 1); \\ w_r(Y_r - x) &= \sum_{X_j \in R_r} (X_j - x), \qquad (k = 2), \end{aligned}$$

for $r = 1, 2$. Hence, we find

$$Y_1 = \frac{1}{\sum I(X_j \leq 0)} \sum_{j=1}^{N} X_j I(X_j \leq 0), \qquad w_1 = \sum_{j=1}^{N} I(X_j \leq 0) \tag{7}$$

$$Y_2 = \frac{1}{\sum I(X_j > 0)} \sum_{j=1}^{N} X_j I(X_j > 0), \qquad w_2 = \sum_{j=1}^{N} I(X_j > 0). \tag{8}$$

(c)

The alternative way to generate two pseudo points is to match moments up to and including the third order, that is, use $K = 4$. In this case the equations become

$$\begin{aligned} w_1 + w_2 &= N \\ \sum_{i=1}^{2} w_i(Y_i - x) &= \sum_{j=1}^{N}(X_j - x) \\ \sum_{i=1}^{2} w_i(Y_i - x)^2 &= \sum_{j=1}^{N}(X_j - x)^2 \\ \sum_{i=1}^{2} w_i(Y_i - x)^3 &= \sum_{j=1}^{N}(X_j - x)^3. \end{aligned}$$

Observe that even in this very simple case a general solution to this set of equations is not available. To get an idea of what a solution would look like, assume that the empirical mean and skewness of $X_1, \ldots, X_N$ equal zero and that we Taylor expand around the empirical mean (or zero). In this case we have

$$\begin{aligned} w_1 + w_2 &= N \\ w_1 Y_1 + w_2 Y_2 &= 0 \\ w_1 Y_1^2 + w_2 Y_2^2 &= N\widehat{\sigma}^2 > 0 \\ w_1 Y_1^3 + w_2 Y_2^3 &= 0, \end{aligned}$$

with $\widehat{\sigma}^2 = \sum X_j^2/N$. A solution is given by $w_i = N/2$ and $Y_1 = -Y_2 = \widehat{\sigma}$. A simple numerical example shows how the results differ from the pseudo points defined by (7)–(8). Let $N = 4$, $X_1 = -X_4 = 3$ and $X_2 = -X_3 = 1$. Hence $\overline{X} = 0$, $\widehat{\sigma}^2 = 5$ and $\sum X_j^3 = 0$ From (7)–(8) we find $w_1 = w_2 = 2, Y_1 = -Y_2 = 2$, while the latter procedure yields $w_1 = w_2 = 2, Y_1 = -Y_2 = \sqrt{5}$ so the pseudo points are more spread.

Example 2 Assume that we have a data set in $Q = 2$ continuous variables, and that we use one region only. We want to match the moments of squashed and original data up to and including order 1.

How many squashed data points may we create? (Note that the number of original data points does not really matter, except to impose an upper limit.) If we want M squashed data points, there are $M(Q+1)$ unknowns. Taylor expansion up to order 1 implies $K = 3$ so we can have at most $M = 1$ squashed data point.

(In general the number of wanted squashed points will determine a minimum number of moments required. For an increase in the number of moments matched, one may generally choose between adding more squashed points or 'improving' the existing squashed points.)

The equations (4) become

$$\begin{aligned} w &= \sum_{j=1}^{N} 1 = N, \qquad (k = 1); \\ w(Y_1 - x_1) &= \sum_{j=1}^{N} (X_{j1} - x_1), \qquad (k = 2); \\ w(Y_2 - x_2) &= \sum_{j=1}^{N} (X_{j2} - x_2), \qquad (k = 3). \end{aligned}$$

In larger and possibly over-determined cases, recall that we will use some kind of least squares technique to solve the system. Then the fact that the right hand sides of the equations are constants and may be calculated once and for all, is helpful. In the simple case above, however, we easily solve to find

$$\begin{aligned} w &= N; \\ Y_1 &= \frac{1}{N} \sum_{j=1}^{N} X_{j1}; \\ Y_2 &= \frac{1}{N} \sum_{j=1}^{N} X_{j2}. \end{aligned}$$

Example 3 In the final example we set the number of continuous variables to two, $Q = 2$. We still work with only one region. To get more than one squashed point, we need to increase the order of the Taylor expansion to two so all moments of order ≤ 2 will be matched. This will give us $K = 6$ equations. In order for the system not to be under-determined, we need $K \geq M(1+Q)$, so $M \leq \frac{6}{3} = 2$. So we can have at most $M = 2$ squashed points.

(Observe that we could have chosen to create only one squashed point. The equation system would have been over-determined. But since we use a minimum-distance solution concept, a solution could have been found.)

The 6 equations in 6 unknowns become

$$\begin{aligned}
\sum_{i=1}^{2} w_i &= \sum_{j=1}^{N} 1 = N, && (k=1);\\
\sum_{i=1}^{2} w_i(Y_{i1} - x_1) &= \sum_{j=1}^{N} (X_{j1} - x_1), && (k=2);\\
\sum_{i=1}^{2} w_i(Y_{i2} - x_2) &= \sum_{j=1}^{N} (X_{j2} - x_2), && (k=3);\\
\sum_{i=1}^{2} w_i(Y_{i1} - x_1)(Y_{i2} - x_2) &= \sum_{j=1}^{N} (X_{j1} - x_2)(X_{j2} - x_2), && (k=4);\\
\sum_{i=1}^{2} w_i(Y_{i1} - x_1)^2 &= \sum_{j=1}^{N} (X_{j1} - x_1)^2, && (k=5);\\
\sum_{i=1}^{2} w_i(Y_{i2} - x_2)^2 &= \sum_{j=1}^{N} (X_{j2} - x_2)^2, && (k=6).
\end{aligned}$$

A more elegant notation is to write only the exponent vectors p_k in the Taylor expansion for which these equations arise;

$$\begin{aligned}
p_1 &= (0 \quad 0)^\top,\\
p_2 &= (1 \quad 0)^\top,\\
p_3 &= (0 \quad 1)^\top,\\
p_4 &= (1 \quad 1)^\top,\\
p_5 &= (2 \quad 0)^\top,\\
p_6 &= (0 \quad 2)^\top.
\end{aligned}$$

Note that these are, as they should be, all 6 possible non-negative integer vectors of $Q = 2$ elements summing to a number ≤ 2.

2.4 On Applying DS to Regression Data

Suppose for simplicity that there are no categorical variables, so $\mathbf{X} = (X_1, \ldots, X_Q)$. Suppose also that the first variable X_1 is to be regressed on the others, $X_2, \ldots, X_Q$. In this context the parameters of interest $\boldsymbol{\theta}$ are the regression coefficients.

Many important problems involving huge data sets are regression problems, yet it is not obvious that the DS technique as described is immediately applicable. In a regression context, we wish to maximize the likelihood corresponding to the *conditional* density

$$\prod_{j=1}^{N} g(X_1|X_2,\dots,X_Q,\boldsymbol{\theta}),$$

and not the joint density

$$\prod_{j=1}^{N} f(X_1,\dots,X_Q|\boldsymbol{\theta})$$

which has been used to until now.

But

$$\prod_{j=1}^{N} f(X_{j1},\dots,X_{jQ}|\boldsymbol{\theta}) = \prod_{j=1}^{N} g(X_{j1}|X_{j2},\dots,X_{jQ},\boldsymbol{\theta}) \prod_{j=1}^{N} h(X_{j2},\dots,X_{jQ}),$$

since $X_2,\dots,X_Q$ does not depend on $\boldsymbol{\theta}$. Hence, taking logarithms,

$$\sum_{j=1}^{N} \ln(f(X_{j1},\dots,X_{jQ}|\boldsymbol{\theta})) = \sum_{j=1}^{N} \ln(g(X_{j1}|X_{j2},\dots,X_{jQ},\boldsymbol{\theta})) + \sum_{j=1}^{N} \ln(h(X_{j2},\dots,X_{jQ})).$$

So our squashed data $\{(\mathbf{Y}_i, w_i)\}$ satisfy

$$\begin{aligned}\sum_{i=1}^{N} w_i \ln(g(Y_{i1}|Y_{i2},\dots,Y_{iQ},\boldsymbol{\theta})) + \sum_{i=1}^{N} w_i \ln(h(Y_{i2},\dots,Y_{iQ}))& \\ \approx \sum_{j=1}^{N} \ln(g(X_{j1}|X_{j2},\dots,X_{jQ},\boldsymbol{\theta})) + \sum_{j=1}^{N} \ln(h(X_{j2},\dots,X_{jQ})).&\end{aligned}$$

For fixed $\{\mathbf{X}_i\}$ and $\{(\mathbf{Y}_i, w_i)\}$, the second sum on either side of the approximation does not depend on $\boldsymbol{\theta}$. So maximizing the conditional likelihood of the squashed data is approximately equivalent to maximizing the conditional likelihood of the original data. This means that estimated regression coefficients of the squashed data will approximate those that would be obtained from the original data. Hence, data squashing "works" for such regression data.

2.5 Significance of the Taylor Expansion Center

When only one squashed point is generated per region, it is always the region mean. Hence, the squashed point is independent of the Taylor expansion center. With more points per region, the influence of the center is less clear. In example 1 (c), where two points are generated in one region, assuming the mean and third central moment of the original data to be zero, a simple calculation shows that here too the pseudo points are independent of the expansion center.

Moreover, if $\mathbf{x} = (x_1, \ldots, x_Q)$ is the Taylor expansion center, we may substitute $Y'_{iq} = Y_{iq} - x_q$ and $X'_{jq} = X_{jq} - x_q$ in (4). If we find $Y'_{iq} = f(\mathbf{X}')$ for some $f(\cdot)$, then $Y_{iq} = f(\mathbf{X} - \mathbf{x}) + x_{iq}$. This illustrates that matching moments about $\mathbf{x}$ is equivalent to matching moments about zero in another region defined by $X'_{jq} = X_{jq} - x_q$ and then inverse-transforming.

2.6 Implementing Data Squashing

DuMouchel et al. (1999) suggest to minimize (6) using the Newton-Raphson method with second order derivatives.

The cost of computing the squashed data points can be broken down into the cost associated with the regionalization, computation of moments for the full data set and the Newton-Raphson iterative procedure to find the squashed points and weights. For the regionalization and computation of moments, the CPU demand is proportional to NQ and NMQ respectively.

These steps are minor compared to the computationally much more intensive Newton-Raphson method. Denoting by K_r and M_r the number of equations in (4) and the number of squashed points in region $r = 1, \ldots, R$, we have that $K_r = O(M_r Q)$. (We have assumed here that the requirement $K \geq M(Q+1)$ is enforced with equality.) Each iteration in the Newton-Raphson procedure is dominated by evaluations that involve $O(\sum_{r=1}^{R} M_r K_r) = O(\sum_{r=1}^{R} M_r^2 Q)$ operations for regions $r = 1, \ldots, R$. Hence, the number of squashed points and the length of each record are the two most important factors for the CPU time.

By using $M_r = \max(1, \alpha \log_2 N_r)$ as in DuMouchel et al. (1999) the evaluations involve

$$O(\alpha^2 Q \sum_{r=1}^{R} (\log_2(N_r))^2) = O(\alpha^2 Q R (\log_2(N/R))^2)$$

operations. The computational cost increases quadratically in $\log_2 N$. DuMouchel et al. (1999) refer to this as “scaling well in N”. The CPU for the optimization step increases linearly in Q. It should be noted that the regionalization allows to use parallel computing, as the squashed points are found independently for each region.

A software implementation of the DS algorithm is currently under development at the Norwegian Computing Center.

3 Some General Remarks

3.1 Moment Generating Functions and Characteristic Functions

The moment generating function of a random scalar X is defined as

$$M_X(t) = \mathrm{E}(e^{tX}).$$

The moments (about zero) of X are $M_X^{(r)}(0) = \mathrm{E}X^r$. For a data set $X_1, \ldots, X_N$ the estimated moment generating function is $\widehat{M}_X(t) = \sum_{i=1}^N e^{tX_i}/N$ with rth derivative in $t = 0$ equal to the rth empirical moment, i.e. $\widehat{M}_X^{(r)}(0) = \sum_{i=1}^N X^r/N$.

A squashed data set $Y_1, \ldots, Y_M$ should of course have a similar moment generating function as the original data $X_1, \ldots, X_N$, and we should have $\widehat{M}_Y(t) = \widehat{M}_X(t)$. One possibility is to require that $\widehat{M}_Y^{(r)}(0) = \widehat{M}_X^{(r)}(0)$ for $r = 1, 2, \ldots$ with increasing precision for increasing r. In fact, equating the estimated derivatives of the moment generating function, amounts to solving the equations $\sum_{i=1}^N X_i^r = (N/M) \sum_{j=1}^M Y_j^r$, $r = 1, 2, \ldots$, which are exactly the equations obtained for scalar data from Taylor expansion about zero and using equal weights $w_j = N/M$, $j = 1, \ldots, M$.

More generally, equating $\widehat{M}_{X-a}^{(r)}(a) = \widehat{M}_{Y-a}^{(r)}(a)$, $r = 1, 2, \ldots$ we obtain the same equations as with Taylor expansion centered at a point a, assuming equal weights.

Generally, if data $Y_1, \ldots, Y_M$ have weights $v_1, \ldots, v_M$ that sum to one, a natural generalization of the estimated moment generating function is $\widehat{M}_Y(t) = \sum_{j=1}^M v_j e^{tY_j}$. Equating the derivatives in $t = 0$ we get $\sum_{i=1}^N X_i^r = \sum_{j=1}^M N v_j Y_j^r$. Again, with $w_j = N v_j$ this is the equations obtained from Taylor expansion/moment matching argument. In conclusion, the moment matching criterion in DuMouchel et al. (1999) is exactly what would have been obtained if the initial requirement had been equation of the moment generating functions (through the derivatives). Observe, that the argument also holds for multidimensional data.

Another function that completely describes the distribution is the characteristic function

$$\phi_X(t) = \mathrm{E}e^{itX} = \mathrm{E}\cos(tX) + i\mathrm{E}\sin(tX).$$

For the derivatives we find that $\phi_X^{(r)}(t) = \mathrm{E}\{(iX)^r e^{itX}\}$ and hence $\phi_X^{(r)}(0) = i^r \mathrm{E}X^r = i^r M_X^{(r)}(0)$. Equating the estimated derivatives of the characteristic function of the squashed and original data set is equivalent to equating estimated moment generating functions.

3.2 What DS Really Is

The "core" of the DS technique is the following: There is a density $f(x; \boldsymbol{\theta})$ from which the massive data set $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ is a realization. We approximate the log-density

$\ln(f(x))$ by a finite sum

$$\ln(f(x)) \approx \sum_{k=1}^{K} a_k(\mathbf{x}, \boldsymbol{\theta}, f) b_k(\mathbf{x}, x). \tag{9}$$

Here we have used a sloppy notation to indicate that $a_k(\cdot)$ is dependent on the form of the density function $f(\cdot)$ whereas $b_k(\cdot)$ is not. The form of the sum is essential. *We must be able to split each term in two factors. The "coefficient" term a_k must not depend on x. The "expansion" term b_k must not depend on $\boldsymbol{\theta}$ or $f(\cdot)$.* Both may in general depend on the original data set $\mathbf{x}$, although in the original version of data squashing, only b_k does.

We now require the log-likelihood of the original data set to approximate the log-likelihood of the weighted squashed data set as in (2), giving

$$\sum_{i=1}^{M} w_i \sum_{k=1}^{K} a_k(\mathbf{x}, \boldsymbol{\theta}, f) b_k(\mathbf{x}, y_i) = \sum_{j=1}^{N} \sum_{k=1}^{K} a_k(\mathbf{x}, \boldsymbol{\theta}, f) b_k(\mathbf{x}, x_j).$$

Now comes the crucial part of the technique, and the reason for the requirements on the terms: We change the order of summation. We obtain

$$\sum_{k=1}^{K} a_k(\mathbf{x}, \boldsymbol{\theta}, f) \sum_{i=1}^{M} w_i b_k(\mathbf{x}, y_i) = \sum_{k=1}^{K} a_k(\mathbf{x}, \boldsymbol{\theta}, f) \sum_{j=1}^{N} b_k(\mathbf{x}, x_j).$$

We require this to be valid for any unknown $\boldsymbol{\theta}$ and $f(\cdot)$, hence the equality must be valid term by term. Then for each k, $a_k(\cdot)$ cancels, leaving

$$\sum_{i=1}^{M} w_i b_k(\mathbf{x}, y_i) = \sum_{j=1}^{N} b_k(\mathbf{x}, x_j), \qquad k = 1, \ldots, K.$$

This is the system we use to determine w_i and y_i.

So in principle any log-density decomposition of the form (9) may be used. Taylor and Fourier series are two examples of decompositions of this kind.

3.3 When is DS Useful ?

DuMouchel et al. (1999) state that DS is preferable if there is no acceptable traditional solution from one or two passes over the data. In a sense this is correct, but it is not the only consideration. For the data squashing to be worthwhile, the total computational expense of squashing *and analysis* must be smaller than direct inference based on the original data set. Also, since SRS is a faster way of reducing a data set, the squashed dataset should be superior to SRS (or versions of stratified sampling). Moreover, the statistical method that we intend to apply to the squashed set must be able to handle weights.

A question that has not been adequately addressed is the *practical use and importance* of DS. When is SRS or other sampling techniques insufficient and when can the cost of data squashing be defended?

A non-manageable massive data set typically consist of more than a million records. For many types of statistical analysis, a data set of about 10.000 or even 100.000 records is manageable. Unless the data shows a very large inter-record variability, it seems that such a SRS sample will be sufficient to do inference on means, trends or variances of the data. For many applications, for instance regression, data may be aggregated and the analysis performed on the aggregated data without loss of information.

In the contexts of *tail probabilities* and *threshold considerations*, SRS is known to be inadequate and data squashing could be a good idea. While an SRS of 10.000 records would typically represent an original set of one million records very well, this is not the case for very small samples. If the computation is heavy, a small sample is nevertheless desired. In this case, the ability of DS to squeeze more information into the sample could be of importance.

3.4 Comparing Different DS Techniques

Working with extending and improving the DS technique of DuMouchel et al. (1999), we must consider how different DS techniques could be compared. It is difficult to imagine that the “goodness” of differently generated squashed data set can be compared without using them for a specific purpose. This means that any result will depend on the applied method. DuMouchel et al. (1999) imply that squashing typically reduces the variability in the data, and that this aspect is important for many methods.

Moreover, the results obtained from the squashed sets need to be compared to some “truth”. With simulated data this could be the model (parameters) used to fit the data. Besides, with simulated data we could estimate under the true model using the squashed data set, and this way eliminate the “model-dependent” part of the comparison. Also, the approach used in DuMouchel et al. (1999) seems reasonable. Here, the results obtained from fitting the full data set to the chosen model is used as the “truth” in the comparisons. In the logistic regression example in DuMouchel et al. (1999), residuals are defined as “(estimated coefficients from squashed data - true coefficients)/std”. MSE, the average squared value of these residuals, is compared to the reduction factor N/M. A squashing technique is deemed do work well if MSE $\ll N/M$.

We should always keep in mind that the data squashing techniques should be compared to *clever* sampling strategies. In DuMouchel et al. (1999) DS is only compared to SRS. Since regionalization is a part of the DS routine, comparing it to stratified sampling techniques seems more natural.

4 Ideas for Research

4.1 Dependencies in the Data

In DuMouchel et al. (1999) the records of the massive data set are assumed to be i.i.d. and the density must be "smooth", but otherwise no distributional or independence assumptions are made within each record. However, data squashing is also of interest for data sets that exhibit dependencies, either horizontally within the records or vertically between the records.

4.1.1 Time Series

An interesting case of horizontal dependencies occurs when the records of the data set are time series. Data squashing for time series is considered in Dimakos (2000b) and here we review some of the results presented in that paper.

A typical source of massive data sets is consumer data such as monetary transactions, purchases or telephone calls. Typically, each record in the massive data set represents a customer whose behavior is registered over a certain time period. For time series, autocorrelations or autocovariances are of particular importance. In DS, squashed data points are found by matching empirical moments, including a certain set of autocovariance estimates as the following argument shows.

Assume that the massive data set of interest consists of i.i.d. time series $\mathbf{X}_j = (X_{j1}, \ldots, X_{jQ})$, $j = 1, \ldots, N$. Without loss of generality we will assume that the times series have zero mean, so that $\mathrm{E}(X_{jq}) = 0 \; \forall j, \forall q$. For a stationary time series $\mathbf{X}_j = (X_{j1}, \ldots, X_{jQ})$ with zero mean the autocovariance at lag k may be estimated by either

$$\widehat{\gamma}_k(\mathbf{X}_j) = \sum_{q=1}^{Q-k} X_{jq} X_{j(q+k)}/Q, \quad j = 1, \ldots, N$$

or by the covariance across the records of a pair of columns that are lag k apart, i.e. by

$$\widehat{\delta}_k^q(\mathbf{X}) = \sum_{j=1}^{N} X_{jq} X_{j(q+k)}/N, \quad q = 1, \ldots, Q-k. \tag{10}$$

Both estimates above are partial in the sense that they do not make use of all the available information as does the best estimate of the autocovariance which is

$$\widehat{\gamma}_k(\mathbf{X}) = \sum_{j=1}^{N} \sum_{q=1}^{Q-k} X_{jq} X_{j(q+k)}/(QN) = \sum_{j=1}^{N} \widehat{\gamma}_k(\mathbf{X}_i)/N = \sum_{q=1}^{Q-k} \widehat{\delta}_k^q(\mathbf{X})/Q. \tag{11}$$

Matching all the $(Q+1)(1+Q/2)$ moments corresponding to Taylor expansion of order

2, we must minimize

$$\begin{aligned} S(\mathbf{Y},\mathbf{w}) &= \sum_{k=1}^{(Q+1)(1+Q/2)} u_k(\sum_{j=1}^{N}\prod_{q=1}^{Q} X_{jq}^{p_{kq}} - \sum_{i=1}^{M} w_i \prod_{q=1}^{Q} Y_{iq}^{p_{kq}})^2 \\ &= \sum_{k=1}^{Q+1} u_k(\sum_{j=1}^{N}\prod_{q=1}^{Q} X_{jq}^{p_{kq}} - \sum_{i=1}^{M} w_i \prod_{q=1}^{Q} Y_{iq}^{p_{kq}})^2 \\ &+ \sum_{k=0}^{Q-1}\sum_{q=1}^{Q-k} u'_{kq}(\sum_{j=1}^{N} X_{jq}X_{j(q+k)} - \sum_{i=1}^{M} w_i Y_{iq} Y_{i(q+k)})^2. \end{aligned} \tag{12}$$

The relabeled optimization weights u'_{kq} depend on the ordering of the exponent vectors $\mathbf{p}_{Q+2}, \ldots, \mathbf{p}_{(Q+1)(1+Q/2)}$. In (12) the second order terms are reordered according to the lags $k = 0, \ldots, Q-1$. Also, it is clear that we are matching the partial autocovariance estimate $\widehat{\delta}_k^q(\mathbf{X})$ in (10) with the corresponding weighted autocovariance estimate in the squashed data set. From (12) we see that these autocovariance estimates are included for all lags $k = 0, \ldots, Q-1$ and all columns $q = 1, \ldots, Q-k$.

The function $S(\mathbf{Y},\mathbf{w})$ has a global minimum of zero at the solution of the original set of equations (4), provided that such a solution exists. Hence, if a global minimum exists, it also holds that $\sum_{q=1}^{Q-k}\sum_{j=1}^{N} X_{jq}X_{j(q+k)} = \sum_{q=1}^{Q-k}\sum_{i=1}^{M} w_i Y_{iq} Y_{i(q+k)}$ for $k = 0, \ldots, Q-1$. This implies that the best estimate of the autocovariance $\widehat{\gamma}_k(\mathbf{X})$ is matched with the corresponding weighted estimate in the squashed data set if there is a global minimum. Otherwise, there is a possibility that the autocovariances are only approximately matched.

In DuMouchel et al. (1999), the same optimization weights are used for all second order terms except for the variance (lag zero) terms, which are given a larger weight. From (12) it is clear that for time series the optimization weights should be equal for all terms representing the same lag, and also that it is possible to emphasize or scale down the importance of certain lags by adjusting the corresponding weights.

Another issue when considering data squashing for time series is the increase in the horizontal dimension that is typically associated with time series. Time series of length 50 or 100 are not particularly long, but still considerably longer than the records considered in DuMouchel et al. (1999). With a large Q it is not possible to use hyper-rectangles in the regionalization. Instead we need to use data spheres or another regionalization technique that does not suffer the same curse of dimensionality. DuMouchel et al. (1999) suggest to collapse several regions into larger regions. Another possibility is to generate regions using a data set consisting of moments or other characteristics of each record, rather than using the data directly. Specifically, for each record $\mathbf{X}_j = (X_{j1}, \ldots, X_{jQ})$, $j = 1, \ldots, N$ we may find a set of moments, percentiles etc. $\mathbf{m}_j = (m_j^1, m_j^2, \ldots, m_j^P)$, $j = 1, \ldots, N$ with $P \ll Q$. The data set $\mathbf{m}_1, \ldots, \mathbf{m}_N$ is then grouped using for instance hyper-rectangles. The regionalization for $\mathbf{X}_1, \ldots, \mathbf{X}_N$ is defined by assigning $\mathbf{X}_j$ to the same region as $\mathbf{m}_j$.

Also the minimization to find the squashed points will suffer from an increase in the horizontal dimension. First of all, the CPU time for each iteration increases linearly in Q. Secondly, optimization is very difficult in high dimensions and at best requires many iterations. This implies that if data squashing is to be feasible for reasonably sized time series, effort is needed to improve the computations.

4.1.2 Vertical Dependencies

A simple kind of vertical dependence is Markov dependence, where record $\mathbf{X}_j$ only depends on the previous record $\mathbf{X}_{j-1}$. For instance, data sets with this structure arise when doing MCMC. In this case we are able to write down the log-likelihood which is

$$l_x(\mathbf{x}_1, \ldots, \mathbf{x}_N; \boldsymbol{\theta}) = \sum_{j=1}^{N} \log f(\mathbf{x}_j | \mathbf{x}_{j-1}, \boldsymbol{\theta}).$$

It follows that the coefficients in (4) are of the form $g_{kj} = g(\boldsymbol{\theta}, \mathbf{a}, j)$. Hence we cannot change the order of summation as was done in (3) and equate term by term without calculating these coefficients.

For other dependency structures we may not be able to write down the full likelihood. Even if we are not able to show that moment matching arises from equating Taylor expansions of the log-likelihoods, we may still calculate moments and do moment matching. However, the interpretation and properties of such a squashing are still unexplored.

4.2 Alternative Expansions

One idea for a research project is to look for alternative decompositions (9). The ideal decomposition would satisfy the following requirements. (Some of these are necessary.)

- The decomposition converges toward the real log-density.
- The convergence is such that we may cut the sum at any point we want; i.e. the first terms in the sum should be the most important.
- No matter how many terms we include, the sum is always a valid log-density. (I.e., the corresponding density is positive and integrates to 1.)
- The factors $b_k(\cdot)$ are interpretable (e.g. moments).
- There is no need for regionalization.
- It is "democratic" in the sense that the added accuracy obtained from using more terms in the sum is spread evenly over the area of interest and not concentrated around one point.
- And more?

Quite possibly, there is no such perfect decomposition. But it is also possible that there is at least one that is more apt to our use than Taylor expansion.

4.2.1 Fourier Series Expansion

Let $f(x)$ be a real function with period 2π so that $f(x) = f(x + 2\pi)$, with $x \in \mathbb{R}$. The Fourier series expansion of $f(x)$ is

$$f(x) = \frac{A_0}{2} + \sum_{n=1}^{\infty}(A_n \cos(nx) + B_n \sin(nx)),$$

where

$$A_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(nx)dx \quad n = 0, 1, 2, \ldots$$

and

$$B_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(nx)dx \quad n = 1, 2, \ldots.$$

More generally if f is defined on $[-a, a]$ the expansion is as above but with nx substituted by $n\pi x/a$ and the integral in the Fourier coefficients is taken over $[-a, a]$ and divided by a rather than π.

Using that $e^{i\theta} = \cos\theta + i\sin\theta$, the Fourier expansion may also be written in the more general complex form

$$f(x) = \sum_{-\infty}^{\infty} C_n e^{inx}.$$

If, as in our case, the function $f(x)$ is real, the pairs of coefficients (C_n, C_{-n}) are complex conjugates for all n. And the coefficients in the real and complex forms of the Fourier expansion are related by

$$C_n = (A_n - iB_n)/2 \qquad \text{and} \qquad C_{-n} = (A_n + iB_n)/2.$$

Now consider using the Fourier expansion in place of the Taylor expansion in (18). Assume that the log-likelihood fulfills the Fourier requirements. For notational simplicity, we suppress the indexing on regions. We then find

$$\sum_{k=1}^{M} w_k \sum_{-\infty}^{\infty} C_n e^{inY_k} = \sum_{j=1}^{N}\sum_{-\infty}^{\infty} C_n e^{inX_j}.$$

Changing the order of summation we obtain

$$\sum_{-\infty}^{\infty} C_n \sum_{k=1}^{M} w_k e^{inY_k} = \sum_{-\infty}^{\infty} C_n \sum_{j=1}^{N} e^{inX_j}$$

which should be compared to (3). Equating term by term, the squashed points are obtained from

$$\sum_{k=1}^{M} w_k e^{inY_k} = \sum_{j=1}^{N} e^{inX_j}, \quad |n| = 0, 1, 2, \ldots. \tag{13}$$

Equating the real and imaginary terms separately, we obtain the equivalent system

$$\sum_{k=1}^{M} w_k \cos(nY_k) = \sum_{j=1}^{N} \cos(nX_j), \quad n = 0, 1, 2, \ldots \quad , \tag{14}$$

$$\sum_{k=1}^{M} w_k \sin(nY_k) = \sum_{j=1}^{N} \sin(nX_j), \quad n = 1, 2, \ldots . \tag{15}$$

As an illustration consider generating one squashed point Y (example 1 a). The equations to solve are then

$$we^{inY} = \sum_{j=1}^{N} e^{inX_j}, \qquad |n| = 0, 1, 2, \ldots .$$

or

$$w\cos(nY) = \sum_{j=1}^{N} \cos(nX_j), \qquad n = 0, 1, 2, \ldots \quad ,$$

$$w\sin(nY) = \sum_{j=1}^{N} \sin(nX_j), \qquad n = 1, 2, \ldots .$$

Inserting $n = 0$ we find $w = N$ as expected. Setting $n = 1$ yields

$$\cos(Y) = \frac{\sum_j \cos(X_j)}{N} \qquad \text{and} \qquad \sin(Y) = \frac{\sum_j \sin(X_j)}{N},$$

With $X_1 = -X_4 = 3$ and $X_2 = -X_3 = 1$ as in example 1 (c) and $n = 1$ we find $Y = 1.80$ and $Y = 0$ from the two equations respectively.

As stated above the characteristic function of a random variable X is $\phi_X(t) = \mathrm{E}(e^{itX})$. A natural estimate of this function is $\widehat{\phi}_X(t) = \sum_j e^{itX_j}/N$. Hence, with $w_k = N/M$ we see that (13) may be written as $\widehat{\phi}_Y(n) = \widehat{\phi}_X(n)$. To summarize, we have showed that term by term equation based on Taylor expansion is equivalent to matching estimated derivatives $\widehat{M}_Y^{(r)}(0) = \widehat{M}_X^{(r)}(0)$ of the moment generating function in $t = 0$, while (if $w_k = N/M$) using Fourier expansion is equivalent to matching the estimated characteristic function for $|t| = 0, 1, \ldots$.

4.2.2 The Discrete Wavelet Transform

Wavelets are an alternative to using Taylor expansion of the log-likelihood. The Wavelet expansion or representation of a signal $f(t)$ in continuous time is

$$\begin{aligned} f(t) \approx & \sum_k s_{J,k}\phi_{J,k}(t) + \sum_k d_{J,k}\psi_{J,k}(t) \\ & + \sum_k d_{J-1,k}\psi_{J-1,k}(t) + \cdots + \sum_k d_{1,k}\psi_{1,k}(t). \end{aligned}$$

Here J is the number of multi-resolution components and k ranges from 1 to the number of coefficients in the specified component. The coefficients $s_{J,k}, d_{J,k}, \ldots, d_{J-1,k}$ are called the wavelet transform coefficients. The functions $\phi_{J,k}(t)$ and $\psi_{J,k}(t)$ are the father and mother wavelet functions. Except for special cases there is no analytic formula for computing a wavelet function.

Another part of the wavelet tool-box that might prove useful for data squashing is *wavelet shrinkage.* Assume that a signal (or function) $f(t)$ is observed at discrete locations $x_1, \ldots, x_N$ and denote the corresponding signal values by $f_1, \ldots, f_N$. The discrete wavelet transform (DWT) maps the discrete signals to a vector of wavelet coefficients $w_1, \ldots, w_N$. By using a wavelet shrinkage procedure, it is possible to determine the coefficients that represent the main features of the signal. This may be regarded as de-noising or data compression. Assume that the wavelet shrinkage determines $K < N$ coefficients that represent the main features. By the inverse discrete wavelet transform (IDWT) the corresponding signal, $\widehat{f}_1, \ldots, \widehat{f}_K$ can be reconstructed. Under certain assumptions on f it is also possible to find locations $y_1, \ldots, y_K$ so that $\widehat{f}_j = f(y_j)$, $j = 1, \ldots, K$. Hence, with scalar data $X_1, \ldots, X_N$ and with a fixed distribution f, this approach can be used to generate squashed points. However, we have not yet fully considered the computational aspects of this approach. The DWT is faster than the fast Fourier transform (FFT) and should be applicable also to massive data sets. The shrinkage procedure seems more problematic for a massive dataset. Wavelet shrinkage is available in S-Plus WAVELETS and possibly also from free software. We have not seen examples of applications to massive data sets.

In contrast to the other DS techniques we consider, the wavelet approach automatically determines not only the location of the squashed points, but also *the number of points.* Of course the wavelet shrinkage method requires some input (a threshold value that determines the smoothness of the wavelet approximation and an estimate of the scale of the noise), but is nevertheless more data driven.

4.3 Regression and Density Estimation for Function Approximation

4.3.1 Polynomial Regression

Assume that our original data set consists of scalars $X_1, \ldots, X_N$. If the data follows a distribution $f(x)$ the log-likelihood can be written $L(\mathbf{X}) = \sum_j \log(f(X_j)) = \sum_j g(X_j)$. Suppose that g is unknown, but that we have measurements $Z_j = g(X_j)$, $j = 1, \ldots, N$ and that we want to use these to estimate g (or the full log-likelihood).

With polynomial regression, we would minimize

$$\sum_{j=1}^{N}(Z_i - \sum_{k=1}^{K} b_k p_k(X_j))^2 \tag{16}$$

with respect to the coefficients $b_1, \ldots, b_K$ for polynomials (or functions) p_k. Using the resulting approximation, the estimated log-likelihood is

$$\widehat{L}(\mathbf{X}) = \sum_{j=1}^{N} \widehat{g}(X_j) = \sum_{j=1}^{N}\sum_{k=1}^{K} b_k(\mathbf{x},\mathbf{z})p_k(X_j).$$

Applying the above approximation to squashing yields the following equations:

$$\sum_{i=1}^{M}\sum_{k=1}^{K} w_i b_k(\mathbf{x},\mathbf{z})p_k(Y_i) = \sum_{j=1}^{N}\sum_{k=1}^{K} b_k(\mathbf{x},\mathbf{z})p_k(X_j).$$

Changing the order of summation and requiring term by term equality, the coefficients cancel (and hence, need not be found) and the equations to solve in order to obtain the squashed points $Y_1,\dots,Y_M$ are

$$\sum_{i=1}^{M} w_i p_k(Y_i) = \sum_{j=1}^{N} p_k(X_j).$$

With polynomials $p_k(x) = x^k$ this is exactly the equations obtained from Taylor expansion of $L(\mathbf{X})$ about zero in one dimension, see (4). An an example, approximating the density by a linear regression ($K = 1$) is equivalent to a Taylor expansion of order 1. Observe also the similarity with the argument of Section 3.2.

4.3.2 Density Estimation Methods

We now consider applying ideas from density estimation to data squashing. Assume that we want to approximate a density $g(x)$. The standard way of doing this is kernel smoothing. The estimated density is

$$\widehat{g}(x) = \frac{1}{Nh}\sum_{j=1}^{N} K(\frac{x-X_j}{h}),$$

for a kernel function $K(\cdot)$ and a smoothing parameter h.

A technique that combines ideas from density estimation and regression, is local polynomial regression. Rather than the minimizing (16), we minimize

$$\sum_{j=1}^{N}(Z_j - \sum_{k=1}^{K} b_k(X_j - x)^k)^2 K_h(x - X_j) \tag{17}$$

for a kernel K_h and smoothing parameter h. With $h = 0$ we get interpolation while $h = \infty$ (Fan and Gijbels, 1996) amounts to linear regression.

Unfortunately, there are fundamental problems that prevent applying these ideas to data squashing in practice. First of all, kernel methods are used to estimate the density itself, and not the log-density. Most kernels are developed for positive functions only, whereas log-densities generally take on negative values as well. In data squashing it is the log-density that is expanded. Secondly, in contrast to the previously considered "expansion-like" techniques, the estimation above provide no ordering of the terms and the "outer sum" is over all terms in the original dataset. With Taylor (and Fourier) the number of terms determines the degree of approximation, and we simply include the number of terms needed to solve for $w_1,\dots,w_M,Y_1,\dots,Y_M$.

4.4 Moment Matching vs. Histogram Matching

An alternative approach to data squashing by moment matching is histogram matching. Consider for simplicity a massive data set of scalars $X_1, \ldots, X_N$ and assume that we are able to produce a frequency histogram based on the data set by dividing the data in regions or bins. A reasonable requirement for a squashed data set is that its histogram should resemble that of the original data. The straightforward way of doing this is to have one squashed point per bin in the histogram. The squashed point is taken as the bin center and the corresponding weight is the bin frequency (bar height).

However, with the squashed points taken as the empirical mean of the original data in the bin, we obtain the same data set as by doing moment matching with one squashed point per region and the regions equal to the histogram bins. For multidimensional data $\mathbf{X}_1, \ldots, \mathbf{X}_N$ the argument is similar. With moment matching the squashed point in region r is equal to the element-wise mean $\mathbf{Y}_r = (\sum_{j \in r} X_{j1}, \ldots, \sum_{j \in r} X_{jQ})/N_r$. This is also a reasonable measure for the center in each histogram "bar".

With histogram matching, the number of squashed points is increased by dividing the data into more regions, but still with one squashed point per region. However, Taylor expansion is a tool for determining more than one point in each region.

4.5 Unidentically Distributed Data

Data squashing as proposed in DuMouchel et al. (1999) does not apply to data sets in which the records are not identically distributed. When the records of the data set follow individual distributions the log-likelihood of the massive data set is

$$l_x(\mathbf{x}_1, \ldots, \mathbf{x}_N; \boldsymbol{\theta}) = \sum_{j=1}^{N} \log f_j(\mathbf{x}_i; \boldsymbol{\theta}_j),$$

where f_j denotes the distribution of record j. Taylor expanding we get

$$l_x(\mathbf{x}_1, \ldots, \mathbf{x}_N; \boldsymbol{\theta}) \approx \sum_{j=1}^{N} \sum_{k=1}^{K} g_{jk} \prod_{q=1}^{Q} (X_{jq} - a_q)^{p_{kq}}$$

and it is no longer possible to do the trick of changing the order of summation as g depends on j through f_j and $\boldsymbol{\theta}_j$. Hence, moment matching can not be derived as a result of matching log-likelihoods. Technically it is of course still possible to calculate and match moments, but the matched quantities are not interpretable as means, variances etc.

Nevertheless, massive data sets will often consist of records that are not i.i.d. Currently there are no available methods that deal with this other than ad hoc solutions such as splitting the data sets to subsets of data that are approximately identically distributed.

4.6 Moment Matching for Finding SRS Weights

Compared to data squashing, sampling techniques have the property that the reduced data set is a subset of the original data set. In some applications, this will be desired or even required.

For instance, if each data record of the massive data set is a time series, selecting a subset of the original data would secure that each pseudo point itself is a time series. With data squashing this is not the case. However, depending on the size of the sample and the characteristics of the massive data set, a simple random sample might be less representative than a squashed data set in the sense that features like moments and percentiles may be quite different from the same features in the full data set.

An appealing approach is to combine the ideas of sub-sampling and moment matching to determine weights for a SRS. First the points of the reduced data set $\mathbf{Y}_1, \dots, \mathbf{Y}_M$ are determined using SRS from the full data set. Then the weights $w_1, \dots, w_M$ are found by minimizing (6) with respect to the weights, keeping the points fixed.

Owen (1999) suggests a slightly different approach and determine the weights by minimizing $\prod_i w_j$ and requiring (4) and that the weights should be positive and sum to M. The approach is a special case of what is called *empirical likelihood squashing.*

4.7 Transformation Prior to Squashing

Transforming the original data before doing data squashing can secure that all data points are within a certain bounded region, or that all moments of a certain order exists (see Section 4.9). For each data point X_{iq} one may find $X'_{iq} = h_q(X_{iq})$, for some transformation function $h_q(\cdot)$ that could be specific to each column q or equal for all of them. In principle, the transformation could be any function, or h could in some sense be based on the data.

It is of interest to try several such transformations on original data, and compare the squashed points (transformed back) with squashed points obtained without the transformation procedure. It is also not clear how transformation would affect the weights.

4.8 Links to Experimental Design

There are parallels between data squashing as considered in this note and in DuMouchel et al. (1999) and the general field of experimental design (Box et al., 1978). In experimental design one typically has K variables or covariates $V_1, \dots, V_K$ that influence the outcome of a certain experiment. These covariates are used as explanatory variables in the model fitting that follows the actual experiment. If each variable V_k has n_k possible levels $k = 1, \dots, K$, the total number of possible covariate patterns is $p = \prod_k n_k$. With many covariates or many levels within each covariate, p can be considerable. Typically it is to expensive to do one experiment or obtain one measure for each of the p covariate settings. In such situations $m < p$ settings are selected so that the range of covariates are covered in a way that makes it possible to determine the main effects, and with increasing m also second and third order effects. Roughly speaking, the aim is that the chosen covariate patterns are spread in such a way that they cover the "covariate space".

The similarity with a data set that requires squashing is clear. The number of data records exceeds the capacity of the available computing power and statistical methods we want to apply, so we do not use all the records but find a smaller data set. While the aim in experimental design is to pick covariate patterns or records so that it is possible to estimate certain effects, the idea of data squashing is that moments are matched. Also, in experimental design the chosen covariate patterns often needs to be a subset of the full range of patterns, while DS produces pseudo points that are not a subset of the original points.

We think that is would be interesting to consider applying ideas from experimental design for data reduction and compare it to data squashing.

4.9 Instabilities of Moment Estimation

For many probability densities, some or all theoretical moments do not exist. However, the density may be infinitely many times differentiable and its Taylor expansion exist. Of course, empircal moments may still be calculated and matched and interpreted as quantities that describe the full data set.

Dimakos (2000a) applies data squashing to generalized Pareto distribution. Depending on the model parameters, this distribution can have finite or infinite moments. The results indicate that data squashing outperforms stratified random sampling for the less heavy tailed distributions, but is associated with a larger bias for the more heavy tailed distributions. However, the experiments show that squashing works even if the underlying theoretical higher order moments are infinite.

It is possible to correct for infinite moments in the optimization by adjusting the optimization weights. If the weights are very small, the effect of instabilities are probably minimal. On the other hand, assigning large weights to moments that do not exist might cause problems. Further work is required to understand these issues.

4.10 The Importance of Regionalization

The categorical variables define regions in a unique way. But the continuous variables may be used to regionalize the data further. This may influence both the computation speed and the squashed points and weights. Also, to construct a certain number of squashed points, one may either use few regions and match moments of a high order, or many regions and lower order moments.

This is clearly an important aspect of the moment-based DS algorithm, but DuMouchel et al. (1999) do not say much about it.

4.11 Bayesian Ideas

Several extensions of DS to Bayesian settings seem interesting. Similarly, one could imagine matching expansion of posterior distributions rather than likelihoods.

Also, it should be possible to explore the effect of putting a prior on the squashed points.

Acknowledgments

Erlend Berg and Ragnar B. Huseby were supported by The Research Council of Norway (NFR) under grant no. 121144/420 (Knowledge, data and decisions). Xeni K. Dimakos was supported by The Research Council of Norway (NFR) under grant no. 110673/420 (Numerical Computations in Applied Mathematics).

We wish to thank Peter Green, Arnoldo Frigessi and Bård Storvik who contributed with valuable comments and interesting ideas.

A Taylor Series Expansion

The nth degree Taylor polynomial of $f(x)$ about $x = a$ is

$$\begin{aligned} P_n(x) &= \sum_{k=0}^{n} \frac{f^{(k)}(a)}{k!}(x-a)^k \\ &= f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n. \end{aligned}$$

The smoother the function f, the lower order n one needs to approximate it well. If $f(\cdot)$ is infinitely many times differentiable, then (under some conditions) there is a neighborhood of x around a for which

$$\lim_{n\to\infty} P_n(x) = f(x).$$

Introducing $\mathbf{x} = (x_1, \ldots, x_Q)^\top$ and $\mathbf{a} = (a_1, \ldots, a_Q)^\top$, and writing $f_i = \frac{\partial f}{\partial x_i}$, the mth-degree Taylor polynomial for a function $f : \mathbb{R}^Q \to \mathbb{R}$ around $\mathbf{a}$ is

$$\begin{aligned}P_m(\mathbf{x}) &= f(\mathbf{a}) \\ &+\sum_{q=1}^{Q} f_q(\mathbf{a})(x_q - a_q) \\ &+\frac{1}{2!}\sum_{q,r=1}^{Q} f_{qr}(\mathbf{a})(x_q - a_q)(x_r - a_r) + \dots \\ &+\frac{1}{m!}\sum_{q_1,q_2,\dots,q_m=1}^{Q} f_{q_1 q_2 \dots q_m}(\mathbf{a})(x_{q_1} - a_{q_1})(x_{q_2} - a_{q_2})\cdots(x_{q_m} - a_{q_m}).\end{aligned}$$

The ith sum in this polynomial has Q^i terms. In total, then, $P_m(\mathbf{x})$ has $\sum_{i=0}^{m} Q^i$ terms. We see that the polynomial is a weighted sum of all possible products the form $\prod(x_q - a_q)$ of order $\leq m$. Hence we may write

$$P_m(\mathbf{x}) = \sum_{k=1}^{K} g_k \prod_{q=1}^{Q} (x_q - a_q)^{p_{kq}} \tag{18}$$

for suitable constants g_k and exponent vectors $p_k = (p_{k1}, \dots, p_{kQ})^\top$. The set of vectors p_k in the polynomial $P_m(\mathbf{x})$ are all possible Q-dimensional vectors of non-negative integer elements that sum to a number $\leq m$. The number of terms in the polynomial,

$$K = \sum_{r=0}^{m} \binom{Q + r - 1}{r},$$

will be the number of possible such vectors. To each vector p_k, there is a corresponding constant g_k involving only factorials and evaluations of differentials.

B Likelihood-based Data Squashing

This section summarizes the basic idea of Madigan et al. (2000).

Suppose that a density $f(y; \theta)$ is specified up to θ and that we have a massive data set $y_1, \dots, y_N$ of observations from $f(\cdot)$. Let $l(\theta; y_i)$ be the likelihood of θ given the observation y_i, and let $\hat{\theta}$ be the maximum likelihood estimator of θ given $y_1, \dots, y_N$.

The idea behind the paper is as follows: Imagine that two data points y_1, y_2 have similar likelihood functions,

$$l(\theta, y_1) \approx l(\theta, y_2),$$

at least in a reasonable neighborhood of $\hat{\theta}$. Consider finding an artificial “intermediate point” y^* such that

$$l(\theta, y^*)^2 \approx l(\theta, y_1) l(\theta, y_2)$$

for reasonable θ. Then the idea is to “squash” y_1 and y_2 into y^* and assigning double exponential weight to y^*. The authors suggest clustering original data with similar likelihoods. The procedure is as follows.

1. Find a crude estimate $\check{\theta}$ of $\hat{\theta}$ based on a single pass through the data. Select k values of θ,

$$\theta_1, \dots, \theta_k,$$

around and including $\check{\theta}$, using e.g. a central composite design.

2. Evaluate

$$l(\theta_j, y_i)$$

for each $i = 1, \dots, n$ and $j = 1, \dots, k$.

3. Select $n' \ll n$ data points as initial cluster centers. Pass through the remaining $n - n'$ points and assign them to the cluster that minimizes

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

where $\bar{l}_c(\theta_j)$ is the mean of $l(\theta_j, \cdot)$ evaluated in all points previously assigned to cluster c.

4. When all points have been assigned to a cluster, select the squashed data points to be

$$y_i^* = \frac{1}{m_c} \sum_{j=1}^{m_c} y_{ij},$$

the means of each cluster. Choose the weights to be m_c, the number of original data points assigned to the corresponding cluster.

5. Optionally, refine y_i^* by an iteration procedure to make the squashed data point represent the mean *likelihood* of each cluster instead of the data mean.

Logistic regression is used in the examples: Regression coefficients from analyzing squashed data sets is compared to results from analyzing SRS data sets, and the true coefficients.

Quality of neural network models trained on squashed data is compared to models trained on all data and SRS data.

LDS consistently outperformed SRS in quality of output. But the paper contains little information about the computational properties of LDS compared to SRS.

References

Aas, K., Huseby, R. B., and Thune, M. (1999). Data mining - A survey.

Berry, M. J. A. and Linoff, G. (1997). *Data Mining techniques.* Wiley, New York.

Box, G. E., Hunter, W. G., and Hunter, J. S. (1978). *Statistics for experimenters.* Wiley, New York.

Dimakos, X. D. (2000a). Data squashing for tail inference in the generalized Pareto distribution. Norwegian Computing Center, Report no. 957.

Dimakos, X. D. (2000b). A note on data squashing for time series. Norwegian Computing Center, Report no. 955.

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

Fan, J. and Gijbels, I. (1996). *Local polynomial modelling and its applications.* Chapman and Hall, London.

Johnson, T. and Dasu, T. (1998). Comparing massive high dimensional data sets. In *Proc. of the 4h Intl. Conference on Knowledge Discovery and Data Mining (KDD)*, pages 229–233.

Madigan, D., Raghavan, N., DuMouchel, W., Nason, M., Posse, C., and Ridgeway, G. (2000). Likelihood-based data squashing: A modeling approach to instance construction. *Data Mining and Knowledge Discovery.* To appear.

Owen, A. (1999). Data squashing by empirical likelihood. http://www-stat.stanford.edu/˜owen/reports/.