A Note on Data Squashing for Time Series

Xeni K. Dimakos*

Abstract

We explore data squashing as introduced in DuMouchel et al. (1999) when the records in the massive data set are iid time series. We focus on autocovariance estimates and their role the squashing. Our findings imply that the optimization weights that are used to find the squashed data points should reflect the dependency structure of the time series and that it could be advantageous to change the objective function that is minimized to find the squashed points.

Key words: Autocorrelation; Data mining; Data squashing; Knowledge Discovery; Massive data; Moment matching; Time series;

1 Introduction

Data squashing was proposed by DuMouchel et al. (1999) as a way of dealing with massive data sets. Rather than scaling up the computing power and statistical methods, the data set itself is scaled down. From the massive data set a new and considerably smaller data set, called the squashed data set, is generated. The size of the squashed data set is so that any advanced statistical method can be applied. The new data set is not a subsample of the original data. The idea is to generate a data set that improves upon sampling with respect to accuracy in inference by finding the new data points so that a set of empirical moments on the original and squashed data are approximately equal.

Data squashing makes the assumption that the records of the massive data set are independent, but makes no assumption on the structure of each data record. In this paper we are interested in data sets with records of dependent variables. In particular, we consider data squashing when each record in the massive data set is a time series. We start by giving a brief review of data squashing. As autocovariances

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

are of particular interest for times series data, we explore the role of autocovariance estimation in data squashing. We find that there are several ways to include information on autocovariances in the squashing. This leads us to suggest several modifications of the original approach which are natural for time series. The paper ends with a discussion of our findings and other issues of relevance for applying data squashing to time series.

2 Method

We consider a data set with N records $\mathbf{X}_i = (X_{i1}, \ldots, X_{iQ}), i = 1, \ldots, N$ with N very large. We assume that all the variables are continuous and that the data set can be represented as a $N \times Q$ matrix $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)'$. A data set is often called massive when N is of the order 10^5 or 10^6. In practice, the size of a data set is always measured relatively to the statistical methods that we are interested in applying. Hence, a data set that is small in absolute size, might be "massive" in the sense that it is computationally infeasible to analyze if the statistical methods we apply are complex and computationally demanding.

The idea of data squashing is to generate a new and smaller data set of records $\mathbf{Y}_j = (Y_{j1}, \ldots, Y_{jQ}), j = 1, \ldots, M$ so that $M \ll N$. To each record a non-negative weight w_j is associated and the weights are generated so that $\sum_{j=1}^{M} w_j = N$. The standard way of reducing a large data set, is to generate a subsample using simple random sampling or another sampling technique. In contrast, the data points generated by data squashing are new pseudo points on the space spanned by the original data set, and not sub sampled from the data set itself.

The $M(Q+1)$ squashed points and weights are regarded as unknown variables that are to be determined so that $K \geq M(Q+1)$ empirical weighted moments on the squashed data set are equal to, or "match", the corresponding empirical unweighted moments on the full data set. Defining an empirical moment about $\mathbf{a} = (a_1, \ldots, a_Q)$ through exponent vectors $\mathbf{p}_1, \ldots, \mathbf{p}_K$ of non-negative integers, the squashed points and the weights are determined so that

$$\sum_{j=1}^{M} w_j \prod_{q=1}^{Q} (Y_{jq} - a_q)^{p_{kq}} = \sum_{i=1}^{N} \prod_{q=1}^{Q} (X_{iq} - a_q)^{p_{kq}} \quad k = 1, \ldots, K. \tag{1}$$

By letting one exponent vector be a vector of zeros, i.e. $\mathbf{p}_l = \mathbf{0}$ for one $l \in \{1, \ldots, K\}$, the weights are constrained so that $\sum_{j=1}^{M} w_j = N$. This constraint allows us to refer to (1) as moment matching as the scaling factor that would be present when equating empirical moments cancels.

DuMouchel et al. (1999) suggest to determine the new points and weights by using

a Newton-Raphson procedure to minimize

$$S(\mathbf{Y}, \mathbf{w}) = \sum_{k=1}^{K} u_k \left(\sum_{i=1}^{N} \prod_{q=1}^{Q} (X_{iq} - a_q)^{p_{kq}} - \sum_{j=1}^{M} w_j \prod_{q=1}^{Q} (Y_{jq} - a_q)^{p_{kq}} \right)^2, \qquad (2)$$

for optimization weights u_k, $k = 1, \ldots, K$ under the constraint that the weights should be positive and the points should not extrapolate, i.e. $w_j \geq 0$, $\forall j$ and $\min_i X_{iq} \leq Y_{jq} \leq \max_i X_{iq}$, $j = 1, \ldots, M, q = 1, \ldots, Q$. There will rarely be a solution for which (2) equals zero which implies that the exact matching in (1) is only acheivable in special cases. In practice, one tends to assign unequal weights so that precision in lower order moments is preferred.

DuMouchel et al. (1999) derive the moment matching criterion (1) by equating the Taylor expansion of the data log-likelihood $\sum_{i=1}^{N} \log f(\mathbf{x}_i; \boldsymbol{\theta})$ and the weigthed log-likelihood $\sum_{j=1}^{M} w_j \log f(\mathbf{y}_j; \boldsymbol{\theta})$ of the squashed data, assuming the same distribution $f(\cdot, \boldsymbol{\theta})$ for the original and squashed points. The argument shows that the nth order of the Taylor approximation (includig nth order derivatives) results in $\binom{Q+n-1}{n}$ terms that are matched in (1). These are all the moments determined by the set of exponents for which $\sum_q p_{kq} = n$. For example, the constant term in the Taylor expansion corresponds to $\mathbf{p}_1 = \mathbf{0}$ in (1) and the first order of the expansion corresponds to $\mathbf{p}_k = (p_{k,k-1} = 1, p_{kl} = 0, l \neq k-1), k = 2, \ldots, Q+1$ etc.

Matching the empirical moments on the original and the squashed data set can also be motivated by considering the associated theoretical moments. Each column q in the massive data set can be regarded as N independent observations of a stochastic variable $X^q, q = 1, \ldots, Q$. These stochastic variables need not be independent and are distributed according to individual distributions that can be characterized by moments $\mathrm{E}(X^q - a)^k$ while the covariances $E(X^q - a_q)(X^{q+k} - a_{q+k})$ describe the relationshipe between the variables. In doing moment matching we require the squashed data set to provide approximately the same estimates of these theoretical moments as the massive data set.

An important feature of data squashing is that the massive data set can be grouped into regions and data squashing preformed independently for each region. Grouping is advantageous because it allows to match a smaller number of moments to obtain the same number of squashed point. To generate M squashed points with no grouping, we could match $K = M(Q+1)$ moments. If the data is grouped into R regions with M/R records in each, it sufficies to match K/R moments within each region. Hence, there are less unknowns to be determined which generally facilitates optimization. In terms of the Taylor expansion argument, it is easy to see that the approximation is improved for smaller regions compared to the full domain. Moreover, grouping opens for parallel computing. Observe that when regions are employed, the weights within each region are constrained so that they sum to the number of original points in each region.

DuMouchel et al. (1999) suggest to group the data in hyper-rectangles or data spheres. By independently splitting each column in the data set into r bins, a total of r^Q hyper-rectangles are generated. Data spheres (DuMouchel et al., 1999; Johnson and Dasu, 1998) are generated by partitioning the data set into l layers according to the distance from the points to some center. The data set is also partitioned into $2Q$ pyramids denoted P_{q+} and $P_{q-}, q = 1, \ldots Q$. A record $\mathbf{X}_i$ belongs to pyramid P_{qs} if $X_{iq} = \max\{|X_{i1}|, \ldots |X_{iQ}|\}$ and $s = \text{sign}(X_{iq})$. The data are usually standardized before the data spheres are found. In combination the layers and pyramids define $2Ql$ data spheres.

In summary data squashing consists of three steps: (i) grouping the data into regions and determining the number of squashed points that is to be found for each region, (ii) calculation of empirical moments for the massive data sets and (iii) the optimization where the squashed points and weights are found for each region. The generated squashed data set is then used for inference.

3 Autocovariances in Data Squashing

In DuMouchel et al. (1999) the records of the massive data set are assumed to be iid, while no assumptions are made on the structure of each data record $\mathbf{X}_i, i = 1, \ldots, N$. In their application of the data squashing the variables $X_{iq}, q = 1, \ldots Q$ are explanatory variables.

A common source of massive data sets is the collection of data describing customer behaviour such as monetary transactions, purchases or telephone calls or data that arise from industrial production monitoring. Each record in the massive data set presents one data generating item, i.e. a customer or production unit. When the data are collected over a certain time period, there will often be horizontal dependencies and trends in the data.

To gain insight to data squashing for dependent data, we will consider the simplified situation where $\mathbf{X}_i, i = 1, \ldots, N$ are iid time series of length Q. Without loss of generality we will assume that the times series have zero mean, so that $\text{E}(X_{iq}) = 0, \forall i, \forall q$.

For time series autocorrelations or autocovariances are of particular interest. For a stationary time series $\mathbf{X}_i = (X_{i1}, \ldots, X_{iQ})$ with zero mean the lag k autocovariance can be estimated by

$$\widehat{\gamma}_k(\mathbf{X}_i) = \sum_{q=1}^{Q-k} X_{iq} X_{i(q+k)}/Q, \quad i = 1, \ldots, N. \tag{3}$$

When we have N iid time series $\mathbf{X}_1, \ldots, \mathbf{X}_N$ the lag k autocovariance may also be estimated by the covariance across the records of a pair of columns that are lag k

apart, i.e. by

$$\widehat{\delta}_k^q(\mathbf{X}) = \sum_{i=1}^{N} X_{iq} X_{i(q+k)}/N, \quad q = 1, \ldots, Q-k. \tag{4}$$

Both the estimate (3) and (4) are sub-optimal in the sense that the do not use all the available information in the data, as does the estimate

$$\widehat{\gamma}_k(\mathbf{X}) = \sum_{i=1}^{N} \sum_{q=1}^{Q-k} X_{iq} X_{i(q+k)}/(QN) = \sum_{i=1}^{N} \widehat{\gamma}_k(\mathbf{X}_i)/N = \sum_{q=1}^{Q-k} \widehat{\delta}_k^q(\mathbf{X})/Q. \tag{5}$$

We will refer to (3) and (4) as local autocovariance estimates in contrast to the global estimate (5). For iid time series $(\mathbf{Y}_1, \ldots, \mathbf{Y}_M)$ with associated weights $w_1, \ldots, w_M$ we define the corresponding weighted autocovariance estimates

$$\begin{aligned} \widehat{\gamma}_k(\mathbf{Y}_j) &= \sum_{q=1}^{Q-k} Y_{jq} Y_{j(q+k)}/Q, \\ \widehat{\delta}_k^q(\mathbf{Y}) &= \sum_{j=1}^{M} w_j Y_{jq} Y_{j(q+k)} / \sum_{j=1}^{M} w_j \end{aligned} \tag{6}$$

and

$$\widehat{\gamma}_k(\mathbf{Y}) = \sum_{j=1}^{M} \sum_{q=1}^{Q-k} w_j Y_{jq} Y_{j(q+k)}/(Q \sum_{j=1}^{M} w_j). \tag{7}$$

Suppose that we are to squash a massive data set consisting of N iid time series by matching the Q first order and $(Q+1)Q/2$ second order moments about zero as well as the constraint on the weights, a total of $(Q+1)(1+Q/2)$ terms. According to (2) the squashed points and the weights are to minimize

$$\begin{aligned} S(\mathbf{Y}, \mathbf{w}) &= \sum_{k=1}^{(Q+1)(1+Q/2)} u_k (\sum_{i=1}^{N} \prod_{q=1}^{Q} X_{iq}^{p_{kq}} - \sum_{j=1}^{M} w_j \prod_{q=1}^{Q} Y_{jq}^{p_{kq}})^2 \\ &= S_0(\mathbf{Y}, \mathbf{w}) + S_1(\mathbf{Y}, \mathbf{w}) + S_2(\mathbf{Y}, \mathbf{w}) \end{aligned} \tag{8}$$

with

$$
\begin{aligned}
S_0(\mathbf{Y},\mathbf{w}) &= u_1(N-\sum_{j=1}^{N} w_j)^2 \\
S_1(\mathbf{Y},\mathbf{w}) &= \sum_{k=1}^{Q} u_{k+1}(\sum_{i=1}^{N} X_{ik} - \sum_{j=1}^{M} w_j Y_{jk})^2 \\
S_2(\mathbf{Y},\mathbf{w}) &= \sum_{k=Q+2}^{(Q+1)(1+Q/2)} u_k(\sum_{i=1}^{N}\prod_{q=1}^{Q} X_{iq}^{p_{kq}} - \sum_{j=1}^{M} w_j \prod_{q=1}^{Q} Y_{jq}^{p_{kq}})^2 \\
&= \sum_{k=0}^{Q-1}\sum_{q=1}^{Q-k} u'_{kq}(\sum_{i=1}^{N} X_{iq}X_{i(q+k)} - \sum_{j=1}^{M} w_j Y_{jq}Y_{j(q+k)})^2. \qquad (9)
\end{aligned}
$$

The relabeled optimization weights u'_{kq} in (9) depend on the ordering of the exponent vectors $\mathbf{p}_{Q+2},\ldots,\mathbf{p}_{(Q+1)(1+Q/2)}$. The terms in $S_2(\mathbf{Y},\mathbf{w})$ are reordered according to the lags $k=0,\ldots,Q-1$. From (9) it is clear that since $\sum_{j=1}^{M} w_j = N$ we are matching the local autocovariance estimates $\widehat{\delta}_k^q(\mathbf{X})$ and $\widehat{\delta}_k^q(\mathbf{Y})$ in (4) and (6) 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 exactly at the solution of the original set of equations (1), provided that such a solution exists. If there is such a solution, it also holds that $\sum_{q=1}^{Q-k}\sum_{i=1}^{N} X_{iq}X_{i(q+k)} = \sum_{q=1}^{Q-k}\sum_{j=1}^{M} w_j Y_{jq}Y_{j(q+k)}$ for $k=0,\ldots,Q-1$ and hence it follows that $\widehat{\gamma}_k(\mathbf{X}) = \widehat{\gamma}_k(\mathbf{Y})$. This demonstrates that the best estimate of the autocovariance (5) is only matched with the corresponding weighted estimate (7) on the squashed data if there is a solution to the original set of equations, otherwise only the local autocovariance estimates (4) and (6) are matched. When only a local minimum is found, the local autocovariance estimates are matched approximately, inducing an approximative match of the global autocovariance estimates.

4 Choice of Optimization Weights

The ordering of the second order terms in (9) arises from the natural ordering of autocovariances according to lags. In principle, data squashing imposes no ordering on the second order terms. In practice, one typically distinguishes between the variance terms that are often called the pure terms, and the cross terms, that is the autocovariances. However, for time series it is natural to also classify cross terms according to the lag they represent. The same argument applies to the optimization weights in (2) and (9). Apriori, we are free to use any set of optimization weights, keeping in mind that our choice influences the characteristics of the squashed data set. In the application in DuMouchel et al. (1999), emphasis is put on matching the means and variances, as well

as on the constraint on the weights. These terms are assigned indices $k = 1, \ldots, 2Q+1$ and given weights $u_k = 1000$. The sum of the optimization weights for the second order cross terms and higher order terms sum to one, i.e. $\sum_{k=2Q+3}^{K} u_k = 1$, with decreasing weight size for increasing order of the terms.

When $\mathbf{X}_i$ is a time series, then we suggest that all terms representing the same lag should be assigned equal weight, that is to let $u'_{kq} = u'_k$ in (9). In Section 3 we showed that the best estimates of the autocovariance (5) and (7) are matched by matching separately the estimates (4) and (6) that are based on pairs of columns of the data set. We are not able to see when assigning different optimization weights to terms that represent the same lag can be defended as this implies that some column pairs are considered to be more important than others in describing the autocovariance at a certain lag. With such a weighting strategy we are further away from matching the global estimates of the autocovariance than when equal weights is applied within each lag.

Since the optimization weights determine which moments that are matched most accurately, the optimization weights can be used to incorporate in the squashing procedure which moments that are considered to be most crucial for describing the data set. When we have prior information on the dependency structure of the time series that indicates that accuracy in matching autocovariances of certain lags is important, we can give high weight to the terms that represent these lags and smaller weight to the other lags. When there is no prior information we would tend to let the weights be decreasing in increasing lags so that $u'_{kq} = u'_k$ and $u'_1 \geq u'_2 \geq \cdots \geq u'_{Q-1}$ in (9). This way we prefer precision in matching short lags to precision in matching higher lag autocovariances.

5 An Alternative Objective Function

In Section 3 we showed that in DuMouchel et al. (1999) the squashed data points are generated by matching the local autocovariance estimates $\widehat{\delta}_k^q(\mathbf{X})$ and $\widehat{\delta}_k^q(\mathbf{Y})$ for all lags $k = 0, \ldots, Q-1$ and all time points $q = 1, \ldots, Q-k$. Provided that the match is close enough and that the same optimization weights are assigned to all local estimates representing the same lag, also the the global estimates $\widehat{\gamma}_k(\mathbf{X})$ and $\widehat{\gamma}_k(\mathbf{Y}, \mathbf{w})$ are matched.

Assume now that our primary interest is that the global autocovariance estimate $\widehat{\gamma}_k(\mathbf{Y}, \mathbf{w})$ on the squashed data set match the corresponding quantity on the massive data set for all lags. Of course, the same argument applies to the terms in $S_1(\mathbf{Y}, \mathbf{w})$, rather than matching columnwise means, the overall mean could be matched. As our interest is in the second order terms, this issue is not considered here. From this

perspective it seems natural to replace $S_2(\mathbf{Y}, \mathbf{w})$ by

$$\widetilde{S}_2(\mathbf{Y}, \mathbf{w}) = \sum_{k=0}^{Q-1} u'_k (\sum_{q=1}^{Q-k} \sum_{i=1}^{N} X_{iq} X_{i(q+k)} - \sum_{q=1}^{Q-k} \sum_{j=1}^{M} w_j Y_{jq} Y_{j(q+k)})^2 \tag{10}$$

in (8). When doing this replacement it follows automatically that only one optimization weight per lag is applied, as suggested in Section 4. Introducing $s_k^q(\mathbf{X}) = \sum_{i=1}^{N} X_{iq} X_{i(q+k)}$ and $s_k^q(\mathbf{Y}) = \sum_{j=1}^{M} w_j Y_{jq} Y_{j(q+k)}$ we have that

$$\begin{aligned}
\widetilde{S}_2(\mathbf{Y}, \mathbf{w}) &= \sum_{k=0}^{Q-1} u'_k \{\sum_{q=1}^{Q-k} (s_k^q(\mathbf{X}) - s_k^q(\mathbf{Y}))\}^2 \\
&= \sum_{k=0}^{Q-1} u'_k \sum_{q=1}^{Q-k} (s_k^q(\mathbf{X}) - s_k^q(\mathbf{Y}))^2 \\
&\qquad + \sum_{k=0}^{Q-1} u'_k \sum_{q=1}^{Q-k} \sum_{r \neq q} (s_k^q(\mathbf{X}) - s_k^q(\mathbf{Y}))(s_k^r(\mathbf{X}) - s_k^r(\mathbf{Y})) \\
&= S_2(\mathbf{Y}, \mathbf{w}) + R(\mathbf{X}, \mathbf{Y}, \mathbf{w}).
\end{aligned}$$

Here we let $u'_{kq} = u'_k$ also in $S_2(\mathbf{Y}, \mathbf{w})$ to make the expressions comparable. Observe that the term $R(\mathbf{X}, \mathbf{Y}, \mathbf{w})$ is zero if $\widehat{\delta}_k^q(\mathbf{X}) = \widehat{\delta}_k^q(\mathbf{Y})$ for $k = 0, \ldots, Q-1, q = 1, \ldots, Q-k$ and is the term that encourages the local structure. By using the expression $\widetilde{S}_2(\mathbf{Y}, \mathbf{w})$ we are effectively reducing the number of terms in the outer sum to Q. Considering the original equations in (1) we now have $1 + 2Q$ equations compared to $(Q+1)(1+Q/2)$ when using $S_2(\mathbf{Y}, \mathbf{w})$.

We think it might be advantageous to use $\widetilde{S}_2(\mathbf{Y}, \mathbf{w})$ rather than $S_2(\mathbf{Y}, \mathbf{w})$, because our primary aim is that the squashed data set preserves the global autocovariance structure of the massive data set. We are less interested in the local features, and it seems to us unnecessary to use effort to match these. Even if we are increasing the number of solutions, we hope that $\widetilde{S}_2(\mathbf{Y}, \mathbf{w})$ will give a smoother objective function that facilitates the optimization and that the local minimas will be less variable with respect to inference.

6 Discussion

In this paper we have given a brief review of data squashing and considered application of it to time series data. In particular, we have focussed on how autocovariances are estimated and used in the squashing. We have demonstrated that the squashed points are generated so that local autcovariance estimates that are based on pairs

of columns are matched, as opposed to matching global estimates that incorporate all available information. This observation led us to suggest an alternative matching criterion for the second order moments, in which the autocovariance estimate that uses all available information for each time lag is matched. Furthermore, we have focussed on the optimization weights. We argue that the optimization weights should be equal for terms that represent the same time lag when several local autocovariance estimates are matched for each lag. Our suggested matching strategy avoids this problem as only one estimate per lag is matched. The natural ordering and interpretation of the second order moments that follows when the records are times series also suggest that the optimization weights could incorporate prior information on the dependency structure of the time series. By varying the magnitude of the weights, accuracy in matching of autocovariances at certain important lags are preferred.

It remains to test our suggested guidelines for choosing the optimization weights and the alternative objective function. The suggested modifications should be compared to data squashing as outlined in DuMouchel et al. (1999) for a set of simulated time series data sets.

There are several aspects of data squashing that are of particular importance for time series data but that are not considered in this paper. The most important of these concerns the curse of dimensionality associated with time series data. A massive data set of time series typically has a higher horizontal dimensionality than indicated in DuMouchel et al. (1999) and the application therein. With the computational approach suggested in DuMouchel et al. (1999) data squashing does not scale so well with the dimension of the records Q. In their analysis of the computational complexity DuMouchel et al. (1999) found that for the regionalization and computation of moments, the CPU demand is proportional to NQ and NK respectively, and hence increase linearly in both N and Q assuming $K = M(Q + 1)$ However, these steps are minor compared to the computationally much more intensive optimization, in which each iteration is dominated by evaluations for which the CPU increases linearly in Q. This implies that data squashing for normal length time series, may require days of running unless effort is made to improve the optimization for instance by using clever starting points or finding smoother objective functions. Keeping in mind that data squashing always competes with using the largest possible sample for which analysis is feasible, it is clear that unless such efforts are made data squashing will only be the method of our choice if the gain in accuracy in inference compared to sub sampling justifies the gross computational expense and long waiting time.

Also the regionalization that is performed prior to finding the squashed points is affected by an increase in the record dimension. With large Q it is not feasible to generate r^Q hyper-rectangles. This suggests that we need to use data spheres or another regionalization technique that does not suffer from the same curse of dimensionality.

Compared to a squashed data set, a sub sampled data set has the advantage that the records have the same structure as found in the massive data set. For instance, if

the original records are time series with a certain dependency structure, then so are the records of the reduced set. Therefore, it seems a good idea to combine the idea of moment matching with sampling to determine weights for a simple random sample. For a fixed simple random sample $\mathbf{Y}_1, \ldots, \mathbf{Y}_M$ Owen (1999) suggests to determine the weights by maximizing $\prod_j w_j$ under the constraint that the weights are positive, sum to N and that (1) holds for a chosen K. The technique is called empirical likelihood squashing and seems to us a very appealing idea for time series data.

Data squashing is a new and unexplored. Upon presenting data squashing to new audiences we have experienced to be met with a skepticism and disbelief as to how it can improve upon the accuracy of sampling to such a degree that the increased computational complexity is worthwhile. Application of data squashing to simulated as well as real data is needed in order to understand and explore the properties of the method. Also, there are clearly many theoretical aspects of data squashing that are not yet well enough understood. Some important issues are outlined in Berg et al. (2000).

It remains to see if data squashing will become the method of choice when working with massive data sets. Still, we consider it a good alternative to constantly increasing computer memory and processing capacity or the very unappealing alternative of settling for poor and insufficient statistical methods. This way data squashing also serves as a reminder of the usefulness of well established sampling techniques.

Acknowledgments

This work was supported by The Research Council of Norway (NFR) under grant no. 110673/420 (Numerical Computations in Applied Mathematics). The author wishes to thank Ragnar B. Huseby and Erlend Berg at the Norwegian Computing Center, who took the initiative to work on data squashing and introduced me to the area. They both contributed to the paper through important discussions and useful comments. Arnoldo Frigessi gave valuable comments that contributed to improving the final paper.

References

Berg, E., Dimakos, X. D., and Huseby, R. B. (2000). Squashing massive data sets: An overview of existing methods and ideas for further research.

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*, pp. 6–15.

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)*, pp. 229–233.

Owen, A. (1999). Data squashing by empirical likelihood.