Chapter 3
Statistical Leveraging Methods in Big Data

Xinlian Zhang, Rui Xie, and Ping Ma

Abstract With the advance in science and technologies in the past decade, big data becomes ubiquitous in all fields. The exponential growth of big data significantly outpaces the increase of storage and computational capacity of high performance computers. The challenge in analyzing big data calls for innovative analytical and computational methods that make better use of currently available computing power. An emerging powerful family of methods for effectively analyzing big data is called statistical leveraging. In these methods, one first takes a random subsample from the original full sample, then uses the subsample as a surrogate for any computation and estimation of interest. The key to success of statistical leveraging methods is to construct a data-adaptive sampling probability distribution, which gives preference to those data points that are influential to model fitting and statistical inference. In this chapter, we review the recent development of statistical leveraging methods. In particular, we focus on various algorithms for constructing subsampling probability distribution, and a coherent theoretical framework for investigating their estimation property and computing complexity. Simulation studies and real data examples are presented to demonstrate applications of the methodology.

Keywords Randomized algorithm · Leverage scores · Subsampling · Least squares · Linear regression

3.1 Background

With the advance in science and technologies in the past decade, big data has become ubiquitous in all fields. The extraordinary amount of big data provides unprecedented opportunities for data-driven knowledge discovery and decision making. However, the task of analyzing big data itself becomes a significant

X. Zhang · R. Xie · P. Ma (✉)
Department of Statistics, University of Georgia, Athens, GA, USA
e-mail: xinlian.zhang25@uga.edu; ruixie@uga.edu; pingma@uga.edu

W. K. Härdle et al. (eds.), *Handbook of Big Data Analytics*, Springer Handbooks of Computational Statistics, https://doi.org/10.1007/978-3-319-18284-1_3

challenge. Key features of big data, including large volume, vast variety and high velocity, all contribute to the challenge of the analysis. Among these, the large volume problem is of great importance. On one hand, the number of predictors for big data may be ultra-large, and this is encountered frequently in genetics and signal processing study. The ultra-high dimension of predictors is referred to as the curse of dimensionality. One of the most pressing needs and continuous efforts in alleviating the curse of dimensionality is to develop new techniques and tools to achieve dimension reduction and variable selection with good properties (Bhlmann and van de Geer 2011; Friedman et al. 2001). On the other hand, we often encounter cases in which sample size is ultra-large. When the sample size reaches a certain scale, although it is considered as preferable in the classical regime of statistical theory, the computational costs of many statistical methods become too expensive to carry out in practice. The topic of this chapter focuses on analyzing big data with ultra-large sample sizes.

A Computer Engineering Solution A computer engineering solution to the big data problem is to build more powerful computing facilities. Indeed, in the past decade, high performance computing platforms such as supercomputers and cloud computing have been developed rapidly. However, none of these technologies by themselves can fully solve this big data problem. The supercomputers are precious computing resources and cannot be allocated to everyone. As for cloud computing, it does possess the advantage of large storage capacities and relatively cheap accessibility. However, problems arise when transferring big data over limited Internet uplink bandwidth. Not to mention that the transferring process also raises new privacy and security concerns. More importantly, the exponential growth of the volume of big data significantly outpaces the increase of the storage and computational capacity of high performance computers.

Computational Capacity Constrained Statistical Methods Given fixed computational capacity, analytical and computational methods need to be adapted to this constraint. One straightforward approach is *divide-and-conquer*. In this approach, one divides the large dataset into small and manageable pieces and performs statistical analysis on each of the small pieces. These results from small pieces are then combined together to provide a final result for the full sample. One notable feature of this procedure is the significant reduction in computing time in a distributed computing environment. However, divide-and-conquer method has its own limitations. On one hand, the efficiency of divide-and-conquer methods still relies on the parallel computing environment, which is not available at all times; on the other hand, it is challenging to develop a universal scheme for combining the results from smaller pieces to form a final estimate with good statistical properties. See Agarwal and Duchi (2011), Chen and Xie (2014), Duchi et al. (2012), Zhang et al. (2013) for applications of this approach.

Fundamentally novel analytical and computational methods are still much needed to harness the power and capture the information of these big data. Another emerging family of methods to tackle the super-large sample problem is the family of *statistical leveraging methods* (Drineas et al. 2006, 2010; Ma et al. 2013, 2014; Ma and Sun 2015; Mahoney 2011). The key idea is to draw a manageable small subsample from the full sample, and perform statistical analysis on this subsample. For the rest of this chapter, we focus on the statistical leveraging methods under a linear model setup. Consider the following linear model

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta}_0 + \epsilon_i, \qquad i = 1, \ldots, n \tag{3.1}$$

where y_i is the response, $\mathbf{x}_i$ is the p-dimensional *fixed* predictor, $\boldsymbol{\beta}_0$ is the $p \times 1$ coefficient vector, and the noise term is $\epsilon_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$. To emphasize, we are dealing with big data in cases where sample size n is ultra large, and $n \gg p$.

Written in vector-matrix format, the linear model in (3.1) becomes

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}_0 + \boldsymbol{\epsilon}, \tag{3.2}$$

where $\mathbf{y}$ is the $n \times 1$ response vector, $\mathbf{X}$ is the $n \times p$ *fixed* predictor or design matrix, $\boldsymbol{\beta}_0$ is the $p \times 1$ coefficient vector, and the noise vector is $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 I)$. In this case, the unknown coefficient $\boldsymbol{\beta}_0$ can be estimated through a least squares (LS) procedure, i.e.,

$$\hat{\boldsymbol{\beta}}_{\mathrm{LS}} = \mathrm{argmin}_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2,$$

where $|| \cdot ||$ represents the Euclidean norm on $\mathbb{R}^n$. If the predictor matrix $\mathbf{X}$ is of full column rank, the LS estimator $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ can be expressed as

$$\hat{\boldsymbol{\beta}}_{\mathrm{LS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{3.3}$$

The general statistical leveraging method in linear model (Drineas et al. 2012, 2006; Mahoney 2011) is given below.

As a by-product, Step 1 of Algorithm 1 provides a random "sketch" of the full sample. Thus visualization of the subsamples obtained enables a surrogate visualization of the full data, which is one of the unique features of subsample methods.

Successful application of statistical leveraging methods relies on both effective design of subsampling probabilities, through which influential data points are sampled with higher probabilities, and an appropriate way to model the subsampled data. So we will review a family of statistical algorithms that employ different subsampling probabilities as well as different modeling approaches for the subsampled data.

Algorithm 1: Statistical leveraging in linear model

```
Step 1(subsampling): Calculate the sampling probability ${\pi_i}_{i=1}^n$ based on
  leveraging-related methods with the full sample ${y_i, \mathbf{x}_i}_{i=1}^n$, use ${\pi_i}_{i=1}^n$ to take a random
  subsample of size $r > p$ and denote the subsample as ${y_i^*, \mathbf{x}_i^*}_{i=1}^r$.
Step 2(model-fitting): Fit the linear model to the subsample ${y_i^*, \mathbf{x}_i^*}_{i=1}^r$ via a weighted LS
  with weights $1/\pi_i$, and return the estimator $\tilde{\boldsymbol{\beta}}$.
```

For the rest of this chapter, we first describe the motivation and detailed layout for statistical leveraging methods. Then corresponding asymptotic properties for these estimators are demonstrated. Furthermore, synthetic and real-world data examples are analyzed. Finally, we conclude this chapter with a discussion of some open questions in this area.

3.2 Leveraging Approximation for Least Squares Estimator

In this chapter, the idea of statistical leveraging is explained with a focus on the linear model. We mainly tackle the challenge of designing subsampling probability distribution, which is the core of statistical leveraging sampling. Various leveraging sampling procedures are discussed from different perspectives. A short summary of all discussed approaches for modeling the subsampled data concludes this section.

3.2.1 Leveraging for Least Squares Approximation

For linear model in (3.2), the LS estimator is $\hat{\boldsymbol{\beta}}_{\text{LS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. The predicted response vector can be written as $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$, where $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is the named as hat matrix for its purpose of getting $\hat{\mathbf{y}}$. The ith diagonal element of $\mathbf{H}$, $h_{ii} = \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i$, where $\mathbf{x}_i^T$ is the ith row of $\mathbf{X}$, is the *statistical leverage* of ith observation. The concept of "statistical leverage" is historically originated from regression diagnostics, where the statistical leverage is used in connection with analyses aiming at quantifying the extent of how influential an observation is for model prediction (Chatterjee and Hadi 1986; Hoaglin and Welsch 1978; Velleman and Welsch 1981). If $\text{rank}(\mathbf{H}) = \text{rank}(\mathbf{X}) = p$ (assuming that $n \gg p$), then we have $\text{trace}(\mathbf{H}) = p$, i.e. $\sum_{i=1}^n h_{ii} = p$. A widely recommended rule of thumb in practice for "large" leverage score is $h_{ii} > 2p/n$ (Hoaglin and Welsch 1978). Also note that $\text{Var}(e_i) = \text{Var}(\hat{y}_i - y_i) = (1 - h_{ii})\sigma^2$. Thus if the ith observation is a "high leverage point," then the value of y_i has a large impact on the predicted value $\hat{y}_i$ and the corresponding residual has a small variance. To put it another way, the fitted regression line tends to pass closer to the high leverage data points.

As an illustration, we provide a toy example in Fig. 3.1. For $i = 1, \ldots, 10{,}000$, we simulate $y_i = -1 + x_i + \epsilon_i$, where x_i is generated from t-distribution with 2 degree of freedom and $\epsilon_i \overset{\text{i.i.d.}}{\sim} N(0, 4)$. The left panel of Fig. 3.1 displays the scatterplot for the full sample and associated fitted LS regression line. The right panel displays the scatterplot for a subsample of size 20 drawn using sampling probabilities constructed from leverage scores (h_{ii}), i.e.

$$\pi_i = \frac{h_{ii}}{\sum_{i=1}^n h_{ii}} = \frac{h_{ii}}{p}, \quad i = 1, \ldots, n. \tag{3.4}$$

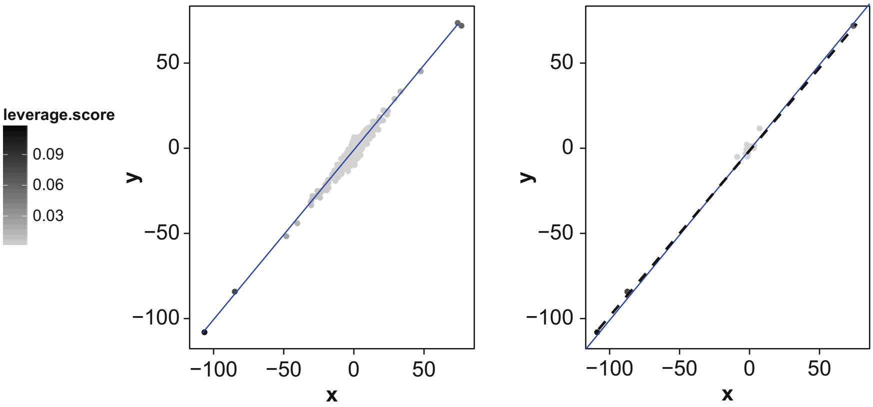

Fig. 3.1 Illustration of motivation of statistical leveraging. The left panel displays the scatterplot of full sample and the corresponding fitted LS regression line. In the right panel, only 20 points were chosen using sampling probabilities constructed from leverage scores (h_{ii}) are plotted. The solid line is the fitted LS regression line with full dataset and the dashed line is the fitted weighted LS regression line from 20 sampled points. In both panels, the color of points corresponds to the leverage score, the higher the leverage score, the darker the color

The solid line is the fitted LS regression line with full sample and dashed line is the fitted weighted LS regression (as described in Step 2 of Algorithm 1) line with 20 subsampled data points. As shown in the left panel of Fig. 3.1, there are several points on the upper right and lower left corner with relatively higher leverage scores, and they are close to the fitted regression line of the full sample. The right panel of Fig. 3.1 implies that fitted weighted LS regression line from the 20 subsampled points (dashed line) is good enough to recover the LS regression line using the full sample.

It is worth noting that one cannot take the subsample by simply using 20 points with highest leverage scores in a deterministic way. The reason is that once the observations are ordered with respect to corresponding leverage scores, then the joint distribution of certain part of the data, e.g. the top 20 observations, will be different from the distribution of original data. Estimators constructed from these deterministic subsamples are biased compared to both the true parameter and the LS estimators from the full sample (Coles et al. 2001). However, if the subsamples are collected in a random fashion, with the probabilities formed by normalizing leverage scores of all observations, the estimates based on the random samples are still guaranteed to be asymptotically unbiased to the true parameter as well as the LS estimators of the full sample. See Sect. 3.3 for more details.

Figure 3.1 also shows how the statistical leveraging methods provide a way of visualizing large datasets.

Following Algorithm 1, the statistical leveraging methods involve computing the exact or approximating statistical leverage scores of the observations, constructing subsample probabilities according to those scores as in Eq. (3.4), sampling data points randomly, and estimating the weighted LS estimator from subsampled data as the final estimate for the full data. We refer to this sampling procedure as *leverage-based sampling* and denote the estimator from this particular procedure as *basic leveraging estimator (BLEV)*. Statistical analysis of the leveraging method is provided in Drineas et al. (2006, 2010), Ma et al. (2013), Raskutti and Mahoney (2014).

As an extension of BLEV, Ma et al. (2014, 2013) proposed *shrinked leveraging-based sampling*, which takes subsample using a convex combination of an exact or approximate leverage scores distribution and the uniform distribution, thereby obtaining the benefits of both. For example, we let

$$\pi_i = \lambda \frac{h_{ii}}{p} + (1-\lambda)\frac{1}{n}, \quad i = 1, \ldots, n, \tag{3.5}$$

where $0 < \lambda < 1$.

We refer to the estimator resulting from sampling data points using (3.5), and estimating the weighted LS estimation on the sampled data points as *shrinked leveraging estimator (SLEV)*. In particular, we use the notation SLEV(λ) to differentiate various levels of shrinkage whenever needed.

3.2.2 A Matrix Approximation Perspective

Most of the modern statistical models are based on matrix calculation, so in a sense when applied to big data, the challenges they face can be solved through using easy-to-compute matrices to approximate the original input matrix, e.g., a low-rank matrix approximation. For example, one might randomly sample a small number of rows from an input matrix and use those rows to construct a low-rank approximation to the original matrix. In this way, it is not hard to construct "worst-case" inputs for which *uniform* random sampling performs very poorly (Drineas et al. 2006; Mahoney 2011). Motivated by this idea, a substantial amount of efforts have been devoted to developing improved algorithms for matrix-based problems that construct the random sample in a *nonuniform* and data-dependent approach (Mahoney 2011), such as least-squares approximation (Drineas et al. 2006, 2010), least absolute deviations regression (Clarkson et al. 2013; Meng and Mahoney 2013), and low-rank matrix approximation (Clarkson and Woodruff 2013; Mahoney and Drineas 2009).

In essence, these procedures are composed of the following steps. The first step is to compute exact or approximate (Drineas et al. 2012; Clarkson et al. 2013), statistical leverage scores of the design matrix. The second step is to use those scores to form a discrete probability distribution with which to sample columns and/or rows randomly from the input data. Finally, the solution of the subproblem is used as an approximation to the solution of the original problem. Thus, the leveraging methods can be considered as a special case of this approach. A detailed discussion of this approach can be found in the recent review monograph on randomized algorithms for matrices and matrix-based data problems (Mahoney 2011).

3.2.3 The Computation of Leveraging Scores

The key for calculating the leveraging scores lies in the hat matrix $\mathbf{H}$, which can be expressed as $\mathbf{H} = \mathbf{U}\mathbf{U}^T$, where $\mathbf{U}$ is a matrix with its columns formed by any orthogonal basis for the column space of $\mathbf{X}$, e.g., the Q matrix from a QR decomposition or the matrix of left singular vectors from the thin singular value decomposition (SVD). Thus, the leverage score of the ith observation, i.e. the ith diagonal element of $\mathbf{H}$, h_{ii}, can also be expressed as

$$h_{ii} = ||\mathbf{u}_i||^2, \tag{3.6}$$

where $\mathbf{u}_i$ is the ith row of $\mathbf{U}$. Using Eq. (3.6), the leverage scores h_{ii}, for $i = 1, 2, \dots, n$ can be obtained. In practice, as a surrogate, $\tilde{h}_{ii}$ are computed as the approximated leverage score in some cases.

The theoretical and practical characterization of the computational cost of leveraging algorithms is of great importance. The running time of the leveraging algorithms depends on both the time to construct the sampling probabilities, $\{\pi_i\}_{i=1}^n$, and the time to solve the optimization problem using the subsample. For uniform sampling, the computational cost of subsampling is negligible and the computational cost depends on the size of the subsample. For statistical leveraging methods, the running time is dominated by computation of the exact or approximating leverage scores. A naïve implementation involves computing a matrix $\mathbf{U}$, through, e.g., QR decomposition or SVD, spanning the column space of $\mathbf{X}$ and then reading off the Euclidean norms of rows of $\mathbf{U}$ to obtain the exact leverage scores. This procedure takes $O(np^2)$ time, which is at the same order with solving the original problem exactly (Golub and Van Loan 1996). Fortunately, there are available algorithms, e.g., Drineas et al. (2012), that compute relative-error approximations to leverage scores of $\mathbf{X}$ in roughly $O(np \log p)$ time. See Drineas et al. (2006) and Mahoney (2011) for more detailed algorithms as well as their empirical applications. These implementations demonstrate that, for matrices as small as several thousand by several hundred, leverage-based algorithms can be competitive in terms of running time with the computation of QR decomposition or the SVD with packages like LAPACK. See Avron et al. (2010), Meng et al. (2014) for more details on this topic. In the next part of this section, another innovative procedure will provide the potential for reducing the computing time to the order of $O(np)$.

3.2.4 An Innovative Proposal: Predictor-Length Method

Ma et al. (2016) introduced an algorithm that allows a significant reduction of computational cost. In the simple case of $p = 1$, and no intercept, we have that $h_{ii} = x_i^2$, i.e. the larger the absolute value of an observation, the higher the leverage score. The idea for our new algorithm is to extend this idea to the case of $p > 1$ by using simply Euclidean norm of each observation to approximate leverage scores and indicate the importance of the observation. That is, we define sampling probabilities

$$\pi_i = \frac{\|\mathbf{x}_i\|}{\sum_{i=1}^{n} \|\mathbf{x}_i\|}, \quad i = 1, \ldots, n. \tag{3.7}$$

If the step 1 of Algorithm 1 is carried out using the sampling probabilities in Eq. (3.7), then we refer to the corresponding estimator as the *predictor-length estimator (PL)*. It is very important to note that the computational cost for this procedure is only $O(np)$, i.e. we only need to go through each observation and calculate its Euclidean norm.

For illustration, in Figs. 3.2 and 3.3, we compare the subsampling probabilities in BLEV, SLEV(0.1), and PL using predictors generated from normal distribution and t-distribution. Compared to normal distribution, t-distribution is known to have heavier tail. So it is conceivable that observations generated from normal distribution tend to have more homogeneous subsampling probabilities, i.e. the circles in Fig. 3.2 are of similar sizes, whereas observations generated from t-distribution tend to have heterogeneous subsampling probabilities, i.e. high probabilities will be assigned to only a few data points, represented in Fig. 3.3 as that a few relatively huge circles on the upper right corner and down left corner. From Fig. 3.3, we clearly observe that the subsampling probabilities used to construct BLEV are much more dispersive than SLEV(0.1) especially in the case of t. It is interesting to note that the probabilities of observations for PL roughly lie in between that of BLEV and SLEV(0.1).

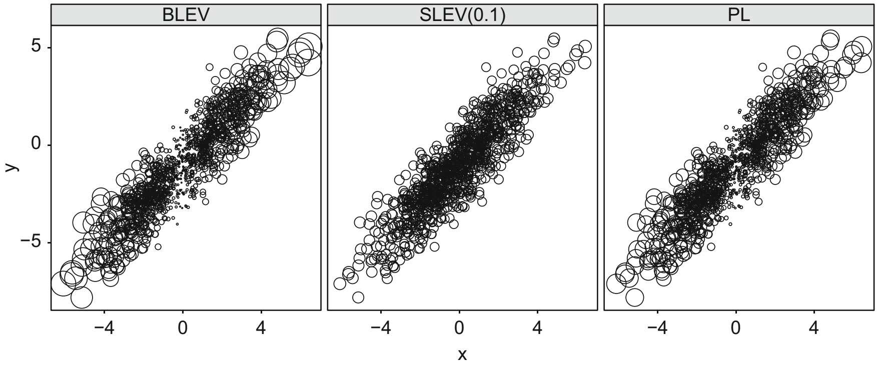

Fig. 3.2 Illustration of different subsampling probabilities with predictor generated from normal distribution. For $i = 1, \ldots, 1000$, $y_i = -1 + x_i + \epsilon_i$, where x_i is generated i.i.d. from $N(0, 4)$ and $\epsilon_i \sim N(0, 1)$. Each circle represents one observation, and the area of each circle is proportional to its *subsampling probability* under each scheme. The area of point with maximum probability in all three probability distributions are set to 100 times that of the point with minimum probability

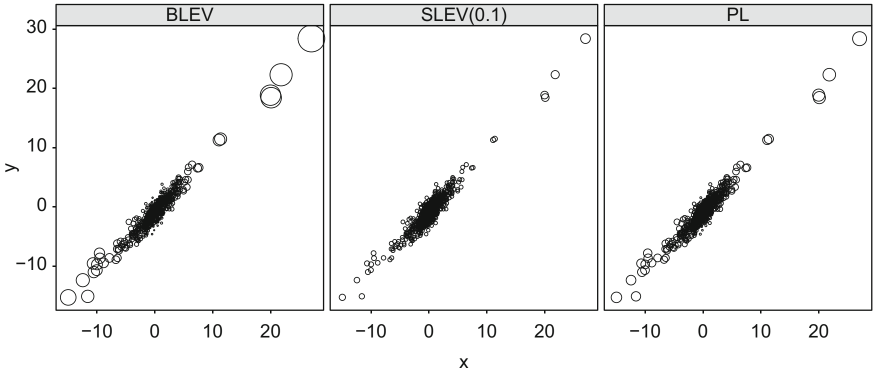

Fig. 3.3 Illustration of different subsampling probabilities with predictor generated from t-distribution. For $i = 1, \ldots, 1000$, $y_i = -1 + x_i + \epsilon_i$, where x_i is generated i.i.d. from t-distribution with df=2 and $\epsilon_i \overset{\text{i.i.d.}}{\sim} N(0, 1)$. Each circle represents one observation, and the area of each circle is proportional to its *subsampling probability* under each scheme. The area of point with maximum probability in all three probability distributions are set to 100 times that of the point with minimum probability

3.2.5 More on Modeling

As mentioned in Step 2 of Algorithm 1, after subsampling, the basic framework of statistical leveraging requires to *rescale, i.e. weight* subsamples appropriately using the same probability distribution as used for subsampling. The purpose for this *weighted LS* step is essentially to construct unbiased estimator of coefficient vector for the linear regression analysis (Drineas et al. 2012, 2006; Mahoney 2011). However, *unweighted leveraging estimators*, in which the modeling of sampled data is carried out by plain least square, is also considered in the literature. That means, in Step 2 of Algorithm 1, instead of solving weighted LS problem, we solve for an *unweighted LS estimator*. Ma et al. (2013) first proposed several versions of unweighted methods that suggest potential improvement over BLEV. The asymptotic properties of unweighted estimators will be discussed later.

3.2.6 Statistical Leveraging Algorithms in the Literature: A Summary

Based on the different combinations of subsampling and modeling strategies we list several versions of the statistical leveraging algorithms that are of particular interest in practice.

- **Uniform Subsampling Estimator (UNIF)** is the estimator resulting from *uniform subsampling* and *weighted LS estimation*. Note that when the weights are uniform, then the weighted LS estimator is the same as the unweighted LS estimator.
- **Basic Leveraging Estimator (BLEV)** is the estimator resulting from *leverage-based sampling* and *weighted LS estimation* on the sampled data, which is originally proposed in Drineas et al. (2006), where the empirical statistical leverage scores of X were used to construct the subsample and weight the subsample optimization problem.
- **Shrinked Leveraging Estimator (SLEV)** is the estimator resulting from sampling using probabilities in (3.5) and *weighted LS estimation* on the sampled data. The motivation for SLEV will be further elaborated in Sect. 3.3. Similar ideas are also proposed in the works of importance sampling (Hesterberg 1995).
- **Unweighted Leveraging Estimator (LEVUNW)** is the estimator resulting from *leverage-based sampling* and *unweighted LS estimation* on the sampled data. The sampling and reweighing steps in this procedure are done according to different distributions, so the results for the bias and variance of this estimator might differ from the previous ones.
- **Predictor Length Estimator (PL)** is the estimator resulting from sampling using probabilities in (3.7) and *weighted LS estimation* on the sampled data.

3.3 Statistical Properties of Leveraging Estimator

In this section, we provide an analytic framework for evaluating the statistical properties of statistical leveraging. We examine the results for bias and variance of leveraging estimator discussed in previous sections.

The challenges for analyzing the bias and variance of leveraging estimator come from two parts. One is the two layers of randomness in the estimation, randomness in the linear model and from the random subsampling; the other is that the estimation relies on random sampling through a nonlinear function of the inverse of random sampling matrix. A Taylor series analysis is used to overcome the challenges so that the leveraging estimator can be approximated as a linear combination of random sampling matrices.

3.3.1 Weighted Leveraging Estimator

We start with bias and variance analysis of leveraging estimator $\tilde{\boldsymbol{\beta}}$ in Algorithm 1. The estimator can be written as

$$\tilde{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{y}, \tag{3.8}$$

where $\mathbf{W}$ is an $n \times n$ diagonal matrix. We can treat $\tilde{\boldsymbol{\beta}}$ as a function of $\mathbf{w} = (w_1, w_2, \ldots, w_n)^T$, the diagonal entries of $\mathbf{W}$, denoted as $\tilde{\boldsymbol{\beta}}(\mathbf{w})$. Randomly sampling with replacement makes $\mathbf{w} = (w_1, w_2, \ldots, w_n)^T$ have a scaled multinomial distribution,

$$\mathbf{Pr}\left[w_1 = \frac{k_1}{r\pi_1}, w_2 = \frac{k_2}{r\pi_2}, \ldots, w_n = \frac{k_n}{r\pi_n}\right] = \frac{r!}{k_1!k_2!\ldots, k_n!}\pi_1^{k_1}\pi_2^{k_2}\cdots\pi_n^{k_n},$$

with mean $E[\mathbf{w}] = \mathbf{1}$. To analyze the statistical properties of $\tilde{\boldsymbol{\beta}}(\mathbf{w})$, Taylor series expansion is performed around the vector $\mathbf{w}_0$, which is set to be the all-ones vector, i.e., $\mathbf{w}_0 = \mathbf{1}$. As a result, $\tilde{\boldsymbol{\beta}}(\mathbf{w})$ can be expanded around the full sample ordinary LS estimator $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$, as we have $\tilde{\boldsymbol{\beta}}(\mathbf{1}) = \hat{\boldsymbol{\beta}}_{\mathrm{LS}}$. Then we come up with the following lemma, the proof of which can be found in Ma et al. (2013).

Lemma 1 *Let $\tilde{\boldsymbol{\beta}}$ be the output of the Algorithm 1, obtained by solving the weighted LS problem of (3.8), where $\mathbf{w}$ denotes the probabilities used to perform the sampling and reweighting. Then, a Taylor expansion of $\tilde{\boldsymbol{\beta}}$ around the point $\mathbf{w}_0 = \mathbf{1}$ yields*

$$\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_{LS} + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Diag\,\{\hat{\mathbf{e}}\}\,(\mathbf{w} - \mathbf{1}) + R_W, \tag{3.9}$$

where $\hat{\mathbf{e}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ is the LS residual vector, and where R_W is the Taylor expansion remainder.

Given Lemma 1, we establish the expression for conditional and unconditional expectations and variances for the weighted sampling estimators in the following Lemma 2.

Conditioned on the data $\mathbf{y}$, the expectation and variance are provided by the first two expressions in Lemma 2; and the last two expressions in Lemma 2 give similar results, except that they are not conditioned on the data $\mathbf{y}$.

Lemma 2 *The conditional expectation and conditional variance for the algorithmic leveraging procedure Algorithm 1, i.e., when the subproblem solved is a weighted LS problem, are given by:*

$$\mathbf{E}_{\mathbf{w}}\left[\tilde{\boldsymbol{\beta}}|\mathbf{y}\right] = \hat{\boldsymbol{\beta}}_{LS} + \mathbf{E}_{\mathbf{w}}\left[R_W\right]; \tag{3.10}$$

$$\mathbf{Var}_{\mathbf{w}}\left[\tilde{\boldsymbol{\beta}}|\mathbf{y}\right] = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\left[Diag\,\{\hat{\mathbf{e}}\}\,Diag\left\{\frac{1}{r\boldsymbol{\pi}}\right\}\,Diag\,\{\hat{\mathbf{e}}\}\right]\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} + \mathbf{Var}_{\mathbf{w}}\left[R_W\right], \tag{3.11}$$

where $\mathbf{W}$ specifies the probability distribution used in the sampling and rescaling steps. The unconditional expectation and unconditional variance for the algorithmic leveraging procedure Algorithm 1 are given by:

$$\mathbf{E}\left[\tilde{\boldsymbol{\beta}}\right] = \boldsymbol{\beta}_0; \tag{3.12}$$

$$\mathbf{Var}\left[\tilde{\boldsymbol{\beta}}\right] = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} + \frac{\sigma^2}{r}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Diag\left\{\frac{(1-h_{ii})^2}{\pi_i}\right\}\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} + \mathbf{Var}\left[R_W\right]. \tag{3.13}$$

The estimator $\tilde{\boldsymbol{\beta}}$, conditioning on the observed data $\mathbf{y}$, is approximately unbiased to the full sample LS estimator $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$, when the linear approximation is valid, i.e., when the $\mathbf{E}\left[R_W\right]$ is negligible; and the estimator $\tilde{\boldsymbol{\beta}}$ is unbiased relative to the true β_0 of the parameter vector β.

For the first term of conditional variance of Eq. (3.11) and the second term of unconditional variance of Eq. (3.13), both of them are inversely proportional to the subsample size r; and both contain a sandwich-type expression, the middle of which involves the leverage scores interacting with the sampling probabilities.

Based on Lemma 2, the conditional and unconditional expectation and variance for the BLEV, PLNLEV, and UNIF procedures can be derived, see Ma et al. (2013). We will briefly discuss the relative merits of each procedure.

The result of Eq. (3.10) shows that, given a particular data set $(\mathbf{X}, \mathbf{y})$, the leveraging estimators (BLEV and PLNLEV) can approximate well $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$. From the statistical inference perspective, the unconditional expectation result of Eq. (3.12) shows that the leveraging estimators can infer well $\boldsymbol{\beta}_0$.

For the BLEV procedure, the conditional variance and the unconditional variance depend on the size of the $n \times p$ matrix $\mathbf{X}$ and the number of samples r as p/r. If one chooses $p \ll r \ll n$, the variance size-scale can be controlled to be very small. The sandwich-type expression containing the leverage scores $1/h_{ii}$, suggests that the variance could be arbitrarily large due to small leverage scores. This disadvantage of BLEV motivates the SLEV procedure. In SLEV, the sampling/rescaling probabilities approximate the h_{ii} but are bounded from below, therefore preventing the arbitrarily large inflation of the variance. For the UNIF procedure, since the variance size-scale is large, e.g., compared to the p/r from BLEV, these variance expressions will be large unless r is nearly equal to n. Moreover, the sandwich-type expression in the UNIF procedure depends on the leverage scores in a way that is not inflated to arbitrarily large values by very small leverage scores.

3.3.2 Unweighted Leveraging Estimator

In this section, we consider the unweighted leveraging estimator, which is different from the weighted estimators, in that the sampling and reweighting are done according to different distributions. That is, modifying Algorithm 1, no weights are used for least squares. Similarly, we examine the bias and variance of the unweighted leveraging estimator $\tilde{\boldsymbol{\beta}}_{\mathrm{LEVUNW}}$. The Taylor series expansion is performed to get the following lemma, the proof of which may be found in Ma et al. (2013).

Lemma 3 *Let $\tilde{\boldsymbol{\beta}}_{LEVUNW}$ be the output of the modified Algorithm 1, obtained by solving the unweighted LS problem of (3.3), where the random sampling is performed with probabilities proportional to the empirical leverage scores. Then, a Taylor expansion of $\tilde{\boldsymbol{\beta}}_{LEVUNW}$ around the point $\mathbf{w}_0 = r\boldsymbol{\pi}$ yields*

$$\tilde{\boldsymbol{\beta}}_{LEVUNW} = \hat{\boldsymbol{\beta}}_{WLS} + (\mathbf{X}^T\mathbf{W}_0\mathbf{X})^{-1}\mathbf{X}^T Diag\left\{\hat{\mathbf{e}}_w\right\}(\mathbf{w} - r\boldsymbol{\pi}) + R_{LEVUNW}, \qquad (3.14)$$

where $\hat{\mathbf{e}}_w = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{WLS}$ *is the LS residual vector,* $\hat{\boldsymbol{\beta}}_{WLS} = (\mathbf{X}^T\mathbf{W}_0\mathbf{X})^{-1}\mathbf{X}\mathbf{W}_0\mathbf{y}$ *is the full sample weighted LS estimator,* $\mathbf{W}_0 = Diag\{r\boldsymbol{\pi}\} = Diag\{rh_{ii}/p\}$*, and* R_{LEVUNW} *is the Taylor expansion remainder.*

Even though Lemma 3 is similar to Lemma 1, the point about which the Taylor expansion is calculated, and the factors that left multiply the linear term, are different for the LEVUNW than they were for the weighted leveraging estimators due to the fact that the sampling and reweighting are performed according to different distributions.

Then the following Lemma 4, providing the expectations and variances of the LEVUNW, both conditioned and unconditioned on the data $\mathbf{y}$, can be established given the Lemma 3.

Lemma 4 *The conditional expectation and conditional variance for the LEVUNW procedure are given by:*

$$\begin{aligned}
\mathbf{E}_\mathbf{w}\left[\tilde{\boldsymbol{\beta}}_{LEVUNW}|\mathbf{y}\right] &= \hat{\boldsymbol{\beta}}_{WLS} + \mathbf{E}_\mathbf{w}\left[R_{LEVUNW}\right];\\
\mathbf{Var}_\mathbf{w}\left[\tilde{\boldsymbol{\beta}}_{LEVUNW}|\mathbf{y}\right] &= (\mathbf{X}^T\mathbf{W}_0\mathbf{X})^{-1}\mathbf{X}^T Diag\{\hat{e}_w\}\mathbf{W}_0 Diag\{\hat{e}_w\}\mathbf{X}(\mathbf{X}^T\mathbf{W}_0\mathbf{X})^{-1}\\
&\quad + \mathbf{Var}_\mathbf{w}\left[R_{LEVUNW}\right].
\end{aligned}$$

where $\mathbf{W}_0 = Diag\{r\boldsymbol{\pi}\}$*, and where* $\hat{\boldsymbol{\beta}}_{\mathrm{WLS}} = (\mathbf{X}^T\mathbf{W}_0\mathbf{X})^{-1}\mathbf{X}\mathbf{W}_0\mathbf{y}$ *is the full sample weighted LS estimator. The unconditional expectation and unconditional variance for the LEVUNW procedure are given by:*

$$\begin{aligned}
\mathbf{E}\left[\tilde{\boldsymbol{\beta}}_{LEVUNW}\right] &= \boldsymbol{\beta}_0;\\
\mathbf{Var}\left[\tilde{\boldsymbol{\beta}}_{LEVUNW}\right] &= \sigma^2(\mathbf{X}^T\mathbf{W}_0\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}_0^2\mathbf{X}(\mathbf{X}^T\mathbf{W}_0\mathbf{X})^{-1}\\
&\quad + \sigma^2(\mathbf{X}^T\mathbf{W}_0\mathbf{X})^{-1}\mathbf{X}^T Diag\{I - P_{\mathbf{X},\mathbf{w}_0}\}\mathbf{W}_0 Diag\{I - P_{\mathbf{X},\mathbf{w}_0}\}\\
&\quad \times\mathbf{X}(\mathbf{X}^T\mathbf{W}_0\mathbf{X})^{-1} + \mathbf{Var}\left[R_{LEVUNW}\right] \qquad (3.15)
\end{aligned}$$

where $P_{\mathbf{X},\mathbf{w}_0} = \mathbf{X}(\mathbf{X}^T\mathbf{W}_0\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}_0$.

The estimator $\tilde{\boldsymbol{\beta}}_{\mathrm{LEVUNW}}$, conditioning on the observed data $\mathbf{y}$, is approximately unbiased to the full sample *weighted* LS estimator $\hat{\boldsymbol{\beta}}_{\mathrm{WLS}}$, when $\mathbf{E}_\mathbf{w}\left[R_{\mathrm{LEVUNW}}\right]$ is negligible; and the estimator $\tilde{\boldsymbol{\beta}}_{\mathrm{LEVUNW}}$ is unbiased relative to the "true" value $\boldsymbol{\beta}_0$ of the parameter vector $\boldsymbol{\beta}$.

Note that the unconditional variance in Eq. (3.15) is the same as the variance of uniform random sampling, since the leverage scores are all the same. The solutions to the weighted and unweighted LS problems are identical, since the problem being solved, when reweighting with respect to the uniform distribution, is not changed. Moreover, the variance is not inflated by small leverage scores. The conditional variance expression is also a sandwich-type expression. The center of the conditional variance, $\mathbf{W}_0 = Diag\{rh_{ii}/n\}$, is not inflated by very small leverage scores.

3.4 Simulation Study

In this section, we use some synthetic datasets to illustrate the efficiency of various leveraging estimators.

One hundred replicated datasets of sample size $n = 100{,}000$ were generated from

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta}_0 + \epsilon_i,$$

where coefficient $\boldsymbol{\beta}_0$ was set to be $(\mathbf{1}_{10}, \mathbf{0.2}_{30}, \mathbf{1}_{10})^T$, and $\epsilon_i \overset{\text{i.i.d.}}{\sim} N(0, 3)$, the predictor $\mathbf{x}_i$ was generated from three different distributions: multivariate normal distribution (denoted as Normal here and after), multivariate t-distribution with $df = 2$ (denoted as T2 here and after), and multivariate Cauchy distribution (denoted as Cauchy here and after). Compared to normal distribution, t-distribution has a heavy tail. Tail of Cauchy distribution is even heavier compared to that of the t-distribution. For the multivariate normal distribution, the mean vector was set to be $\mathbf{1}_{50}$ and the covariance matrix to be Σ, the (i, j)th element of which was $\Sigma_{i,j} = 3 \times (0.6)^{|i-j|}$. For the multivariate t-distribution, we set the non-centrality parameter as $\mathbf{1}_{50}$, the covariance matrix as Σ and the degree of freedom as 2. For the multivariate Cauchy distribution, we used $\mathbf{1}_{50}$ for position vector and Σ defined above for dispersion matrix.

3.4.1 *UNIF* and *BLEV*

We applied the leveraging methods with different subsample sizes, $r = 2p, \ldots, 10p$, to each of 100 datasets, and calculated squared bias and variance of the leveraging estimators to the true parameters $\boldsymbol{\beta}_0$. In Fig. 3.4, we plotted the variance and squared bias of $\tilde{\boldsymbol{\beta}}_{\text{BLEV}}$ and $\tilde{\boldsymbol{\beta}}_{\text{UNIF}}$ for three multivariate distributions.

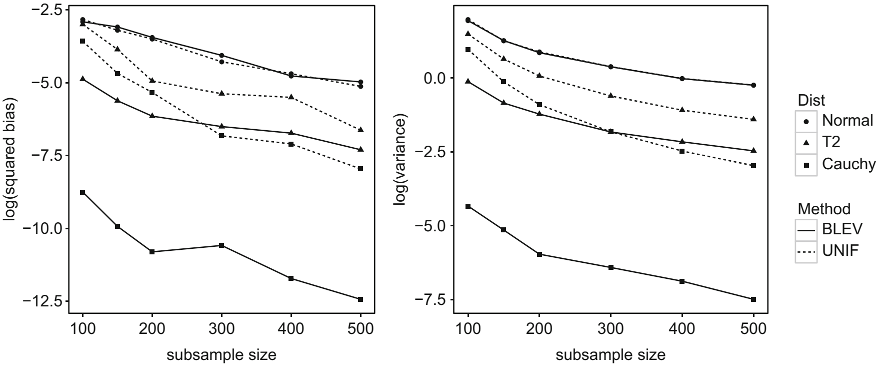

Fig. 3.4 Comparison of variances and squared biases of $\tilde{\boldsymbol{\beta}}_{\text{BLEV}}$ and $\tilde{\boldsymbol{\beta}}_{\text{UNIF}}$ in three distributions. In the graph, "Normal" stands for multivariate normal distribution, "T2" stands for multivariate t-distribution with degree of freedom 2, and "Cauchy" stands for the multivariate Cauchy distribution

Several features are worth noting about Fig. 3.4. First, in general the magnitude of bias is small compared to that of variance, corroborating our theoretical results on unbiasedness of estimators of leveraging methods in Sect. 1. Second, when the predictor vectors $\mathbf{x}_i$ were generated from normal distribution, the bias of BLEV and UNIF are close to each other, which is expected since we know from Fig. 3.2 that the leverage scores are very homogeneous. Same observation exists for the variance. In contrast, in case of T2 and Cauchy, both bias and variance of BLEV estimators are substantially smaller than bias and variance of UNIF estimators correspondingly.

3.4.2 *BLEV* and *LEVUNW*

Next, we turn to the comparison between BLEV and LEVUNW in Fig. 3.5. As we mentioned before, the difference between these two methods is from modeling approach. BLEV was computed using weighted least squares, whereas LEVUNW was computed by unweighted LS. Moreover, both $\tilde{\boldsymbol{\beta}}_{\mathrm{BLEV}}$ and $\tilde{\boldsymbol{\beta}}_{\mathrm{LEVUNW}}$ are unbiased estimator for the unknown coefficient $\boldsymbol{\beta}_0$. As shown in Fig. 3.5, the biases are in general small for both estimators; but when predictors were generated from T2 and Cauchy, LEVUNW consistently outperforms BLEV at all subsample sizes.

3.4.3 *BLEV* and *SLEV*

In Fig. 3.6, we compare the performance of BLEV and SLEV. In SLEV, the subsampling and weighting steps are performed with respect to a combination of the subsampling probability distribution of BLEV and UNIF. As shown, the SLEV(0.9) performs uniformly better than BLEV and SLEV(0.1); and BLEV is

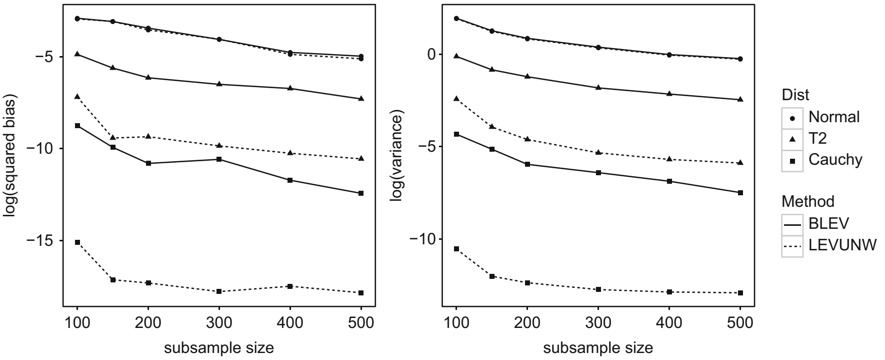

Fig. 3.5 Comparison of squared biases and variances of $\tilde{\boldsymbol{\beta}}_{\mathrm{BLEV}}$ and $\tilde{\boldsymbol{\beta}}_{\mathrm{LEVUNW}}$ in three distributions

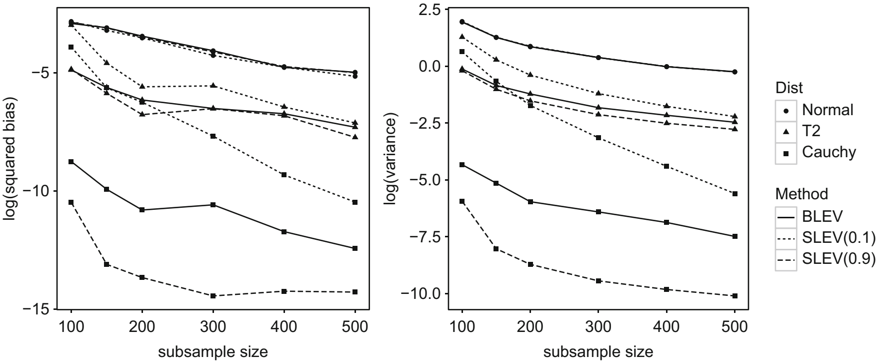

Fig. 3.6 Comparison of squared biases and variances of $\tilde{\boldsymbol{\beta}}_{\text{BLEV}}$ and $\tilde{\boldsymbol{\beta}}_{\text{SLEV}}$ in three distributions. In the graph, SLEV(0.1) corresponds to choosing $\pi_i = \lambda \frac{h_{ii}}{p} + (1-\lambda)\frac{1}{n}$, where $\lambda = 0.1$, and SLEV(0.9) corresponds to $\lambda = 0.9$

better than SLEV(0.1) in terms of both squared bias and variance. By construction, it is easy to understand that SLEV(0.9) and SLEV(0.1) enjoy unbiasedness, in the same way that UNIF and BLEV do. In Fig. 3.6, the squared biases are uniformly smaller than the variances for all estimators at all subsample sizes. Note that for SLEV $\pi_i \geq (1-\lambda)/n$, and the equality holds when $h_{ii} = 0$. Thus, the introduction of λ with uniform distribution in $\{\pi\}_{i=1}^n$ helps bring up extremely small probabilities and suppress extremely large probabilities correspondingly. Thus SLEV avoids the potential disadvantage of BLEV, i.e. extremely large variance due to extremely small probabilities and the unnecessary oversampling in BLEV due to extremely large probabilities. As shown in the graph, also suggested by Ma et al. (2013), as a rule of thumb, choosing $\lambda = 0.9$ strikes a balance between needing more samples and avoiding variance inflation.

3.4.4 BLEV and *PL*

In Fig. 3.7, we consider comparing BLEV and PL. As shown, the squared bias and variance for PL and BLEV are very close to each other in Normal distribution. In T2 distribution, as subsample size increases, we notice some slight advantage of PL over BLEV in both squared bias and variance. The superiority of PL over BLEV is most appealing in Cauchy distribution and as subsample size increases, the advantage of PL in terms of both bias and variance gets more significant.

3.4.5 SLEV and *PL*

Lastly, in Fig. 3.8 we compare SLEV(0.9) and PL. The squared bias and variance of SLEV(0.9) are slightly smaller than those of PL at all subsample sizes in T2

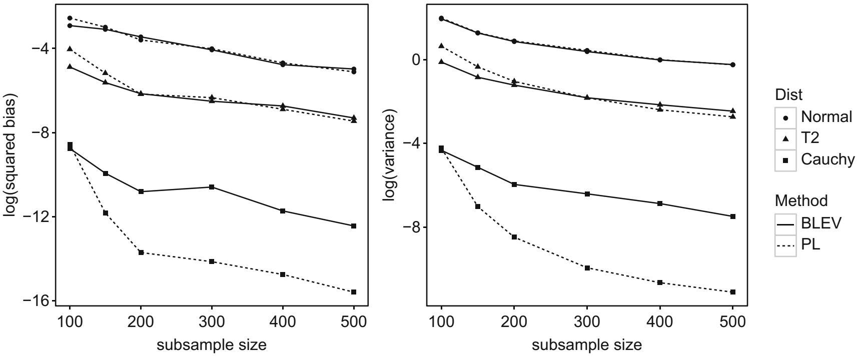

Fig. 3.7 Comparison of squared biases and variances of $\tilde{\boldsymbol{\beta}}_{\mathrm{PL}}$ and $\tilde{\boldsymbol{\beta}}_{\mathrm{BLEV}}$ in three distributions

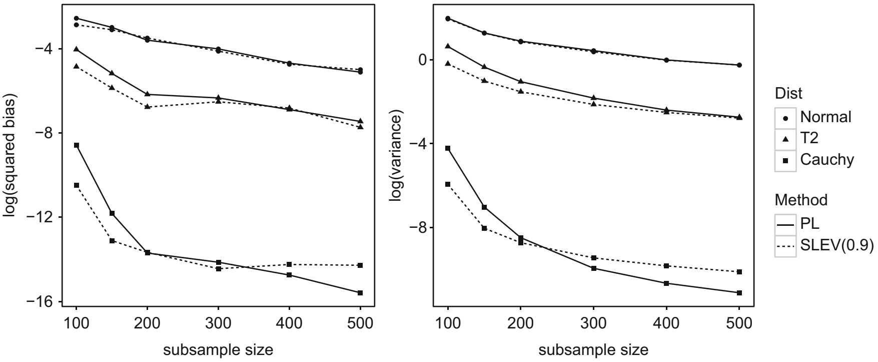

Fig. 3.8 Comparison of squared biases and variances of $\tilde{\boldsymbol{\beta}}_{\mathrm{SLEV(0.9)}}$ and $\tilde{\boldsymbol{\beta}}_{\mathrm{PL}}$ in three distributions

distribution. PL has smaller squared bias and variances compared to SLEV(0.9) at subsample sizes greater than 400. Considering that PL reduces computational time to $O(np)$, which makes PL especially attractive in extraordinarily large datasets.

Our simulation study shows that the statistical leveraging methods perform well in relatively large sample size data sets. Overall, compared to other methods, PL performs reasonably well in estimation in most cases being evaluated and it requires significantly less computing time than other estimators. In practice, we recommend starting analysis with constructing the PL subsampling probabilities, using the probability distribution to draw random samples from the full sample, and performing scatterplots on the sampled data to get a general idea of the dataset distribution. If the distribution of explanatory variables are more close to normal distribution, then we suggest also try BLEV, SLEV, LEVUNW using exact or approximating leverage scores; if the distribution of explanatory variables is close to t or Cauchy distribution, then we refer to the PL estimator.

3.5 Real Data Analysis

In this section, we analyze the "YearPredictionMSD" dataset (Lichman 2013), which is a subset of the Million Song Dataset (http://labrosa.ee.columbia.edu/millionsong/). In this dataset, 515,345 songs are included, and most of them are western, commercial tracks ranging from the year 1922 to 2011, peaking in the early 2000s. For each song, multiple segments are extracted and each segment is described by a 12-dimensional timbre feature vector. Then average, variance, and pairwise covariance are taken over all "segments" for each song.

In this analysis, the goal of analysis is to use 12 timbre feature averages and 78 timbre feature variance/covariances as predictors, totaling 90 predictors, to predict the year of release (response). We chose a linear model in (3.2) to accomplish this goal. Considering the large sample size, we opt to using the statistical leveraging methods.

First, we took random samples of size 100 using sampling probabilities of BLEV and PL to visualize the large dataset. Figure 3.9 shows the scatterplot for the subsample of one predictor using two different sampling probability distributions. In Fig. 3.9, we can see that there are several points with extra dark color standing out from the rest, indicating that the data distribution might be closer to *t* or Cauchy than to Normal distribution and that statistical leveraging methods will perform better than uniform sampling according to the simulation study. Also, these two sampling probability distributions are very similar to each other, suggesting that PL might be a good surrogate or approximation for BLEV. Since the computation of PL is more scalable, it is an ideal method for exploratory data analysis before other leveraging methods are applied to the data.

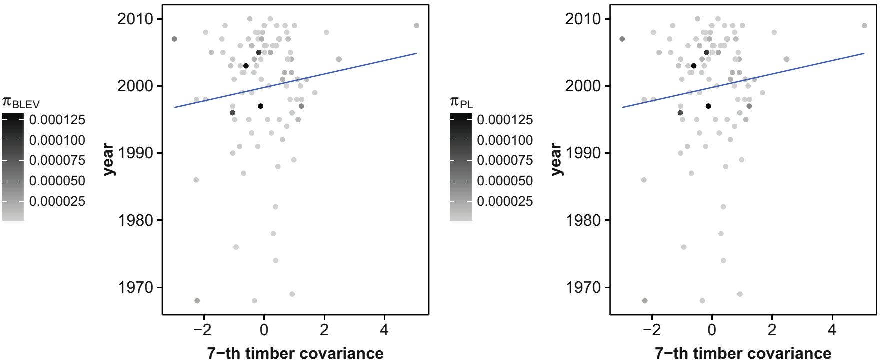

Fig. 3.9 Scatterplots of a random subsample of size 100 from "YearPredictionMSD" dataset (Lichman 2013). In this example, we scale each predictor before calculating the subsampling probabilities. The left panel displays the subsample drawn using subsampling probability distribution in BLEV in (3.4). The right panel displays the subsample drawn using subsampling probability distribution in PL in (3.7). Color of the points corresponds to the sampling probabilities: the darker the color, the higher the probability

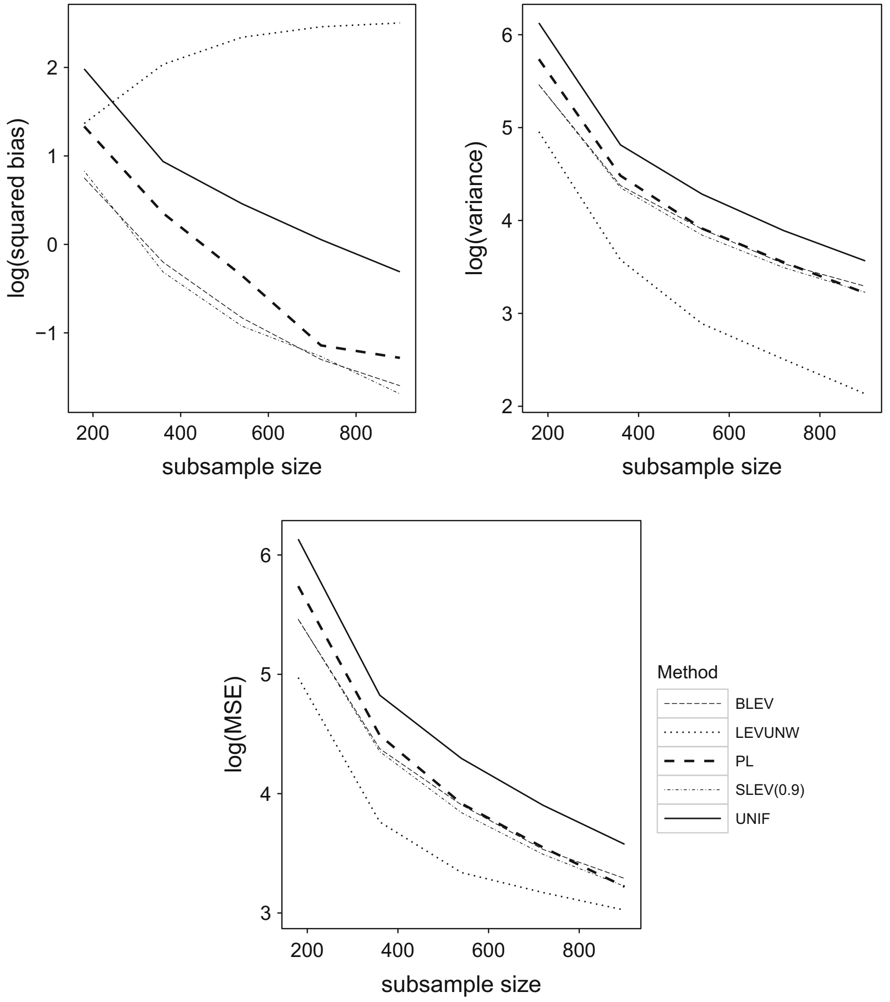

Fig. 3.10 The squared bias, variance and MSE (with respect to the LS estimator of whole dataset) of different statistical leveraging algorithms using "YearPredictionMSD" dataset

Next, we applied various reviewed statistical leveraging methods with different subsample sizes to the data. We scaled the predictors and centered the response prior to analysis. Each method is repeated 100 times and the bias, variance and mean squared error (MSE) to the full sample LS estimator are plotted in Fig. 3.10.

Consistent with results in the simulation study, the variance is larger compared to bias for all statistical leveraging estimators at all subsample sizes. As shown in Fig. 3.10, LEVUNW has the smallest MSE and variance, but the largest bias. As reviewed in Sect. 3.3 about the asymptotic properties of LEVUNW, we discerned that it is an unbiased estimator for the underlying true parameter $\boldsymbol{\beta}_0$ but a biased estimator for the LS estimator $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$. Thus in the figure, as subsample size is getting larger, the bias of the LEVUNW estimator becomes significantly larger than all other estimators. But since variance dominates bias, LEVUNW still outperforms other estimators in terms of MSE. The squared bias of BLEV and SLEV(0.9) are consistently smaller than that of PL at each subsample size; however, the variances and MSEs of BLEV and SLEV(0.9) are close to those of PL, especially at sample sizes larger than 400. This means that PL may be considered as a computationally practical surrogate for BLEV and SLEV(0.9), as suggested in Fig. 3.9.

3.6 Beyond Linear Regression

3.6.1 Logistic Regression

Wang et al. (2017) generalized the idea of statistical leveraging to logistic model defined as below:

$$\begin{aligned} y_i &\sim \mathrm{Binomial}(n_i, p_i) \\ \mathrm{logit}(p_i) &= \log\left(\frac{p_i}{1-p_i}\right) \\ &= \mathbf{x}_i^T \boldsymbol{\beta}_0. \end{aligned}$$

Logistic regression is one of the most widely used and typical examples of generalized linear models. The regression coefficients $\boldsymbol{\beta}_0$ are usually estimated by maximum likelihood estimation (MLE), i.e.

$$\hat{\boldsymbol{\beta}}_{\mathrm{MLE}} = \max_{\boldsymbol{\beta}} \sum_{i=1}^{n} [y_i \log p_i(\boldsymbol{\beta}) + (1-y_i)\log\{1-p_i(\boldsymbol{\beta})\}], \tag{3.16}$$

where $p_i(\boldsymbol{\beta}) = \exp(\mathbf{x}_i^T\boldsymbol{\beta})/\{1+\exp(\mathbf{x}_i^T\boldsymbol{\beta})\}$.

Unlike linear regression with normally distributed residuals as stated in (3.1), there exists no closed-form expression for (3.16). So an iterative procedure must be used instead; for example, Newton's method. As shown, Newton's method for numerically solving (3.16) corresponds to an iterative weighted least square (IWLS) problem (McCullagh and Nelder 1989). However, the generalization of BLEV is not trivial. The weights in each iterated WLS involve $\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$. Consequently, to construct the leverage scores one has to obtain $\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$ first. The mutual dependence of subsampling probabilities and $\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$ under GLM settings poses a dilemma for the intended generalization.

To tackle this problem, Wang et al. (2017) proposed a two-step procedure. First, we draw a subsample of size r_1 using uniform subsampling probabilities and obtain an estimate of coefficient values, denoted as $\tilde{\boldsymbol{\beta}}^{(1)}$. Second, with the $\tilde{\boldsymbol{\beta}}^{(1)}$, the weight matrix or variance-covariance matrix can be estimated and the subsampling probabilities for BLEV can be further constructed. Finally, we obtain another subsample of size r_2 to get the final estimator $\tilde{\boldsymbol{\beta}}$.

The computation time for $\tilde{\boldsymbol{\beta}}^{(1)}$ in the first step is $O(m_1 r_1 p^2)$ where m_1 is the number of iterations of IWLS in the first step; the computation time for construction of the subsampling probabilities can be as low as $O(np)$; the computation time for $\tilde{\boldsymbol{\beta}}$ in the second step is $O(m_2 r_2 p^2)$ where m_2 is the number of iterations of IWLS in the second step. So the overall time complexity is $O(np + m_1 r_1 p^2 + m r_2 p^2)$. Considering that p is fixed and m_1, m_2, r_1, and r_2 are all much smaller than n, the time complexity of whole procedure stays at the order of $O(np)$.

Another remark about the two-step procedure concerns the balance between r_1 and r_2. On one hand, in order to get a reliable estimate $\tilde{\boldsymbol{\beta}}^{(1)}$, r_1 should not be too small; on the other hand, the efficiency of the two-step algorithm decreases if r_1 grows larger compared to r_2. In practice, we find that the algorithm works well when the ratio $r_1/r_1 + r_2$ is between 0.2 and 0.4 and this is the rule of thumb recommended by Wang et al. (2017).

3.6.2 Time Series Analysis

Although we have reviewed so far on the setting of independent and identically distributed data, the natural extension of statistical leveraging methods can be made to handle the dependent data settings, e.g. time series data. Time series in big data framework are widely available in different areas, e.g. sensors data, which is the most widespread and is a new type of time series data. With storage costs coming down significantly, there are significant efforts on analyzing these big time series data (including instrument-generated data, climatic data, and other types of sensor data). However, analyzing the big time series data has new challenge due to the computational cost. Autoregressive and moving average (ARMA) model has been extensively used for modeling time series. But the traditional ARMA model is facing the limit from the computational perspective on analyzing big time series data. The leveraging theory and method thus have been proposed for fitting ARMA model (Xie et al. 2017). A distinguished feature of the novel leveraging method is that instead of sampling individual data points, we subsample blocks of time series so that the time dependence can be well estimated. Such leveraging subsampling approach has a significant challenge related to stationarity. In time series analysis, it is necessary to assume that at least some features of the underlying probability are sustained over a time period. This leads to the assumptions of different types of stationarity. However, a block of time series in stationarity does not necessarily imply the stationarity of the whole time series. Thus novel statistical methods are needed.

In the context of ARMA model, Xie et al. (2017) propose a novel sequential leveraging subsampling method for non-explosive AR(p) series. The sequential leveraging subsampling method can adapt to the availability of computing resources. When only single (or a few) computing processor but a large memory is available, we design a sequential leveraging method starting with one single data point. The idea is that we sample a single data point base on leverage-probability, and then expand the single data point to its neighborhood to form a block of time series sequentially. The key is that as long as the single block is long enough, all the features of the full sample are captured in the single block time series. When there are a large number of computing processors, each of which has moderate memory, we sample several points and perform the leveraging sequential sampling on each of them so that we have a snapshot of the whole time series.

3.7 Discussion and Conclusion

When analyzing big data with large sample size, one faces significant computational challenge, i.e., the high computational cost renders many conventional statistics methods inapplicable in big data. There is an extensive literature in the computer science community on efficient storage and computation of the big data, such as parallel computing algorithms using GPUs, etc. However, very few of them overcome the computational challenge from statistical perspective, e.g. the bias and variance of the big data estimation. In this chapter, we reviewed the statistical leveraging methods for analyzing big data. The idea of statistical leveraging is very simple, i.e., to take a random subsample, on which all subsequent computation steps are performed. Sampling is one of the most common tools in statisticians' toolkit and has a great potential to overcome the big data challenge. The key to success of statistical leveraging methods relies on the effective construction of the sampling probability distribution, based on which influential data points are sampled. Moreover, we also presented some preliminary ideas about extending the statistical leveraging to GLM and time series model. But obviously the power of the leveraging methods has not been fully exploited in this chapter. The performance of leveraging methods is waiting to be examined on more complicated problems, such as penalized regression.

Acknowledgements This work was funded in part by NSF DMS-1440037(1222718), NSF DMS-1438957(1055815), NSF DMS-1440038(1228288), NIH R01GM122080, NIH R01GM113242.

References

Agarwal A, Duchi JC (2011) Distributed delayed stochastic optimization. In: Advances in neural information processing systems, pp 873–881

Avron H, Maymounkov P, Toledo S (2010) Blendenpik: supercharging LAPACK's least-squares solver. SIAM J Sci Comput 32:1217–1236

Bhlmann P, van de Geer S (2011) Statistics for high-dimensional data: methods, theory and applications, 1st edn. Springer, Berlin

Chatterjee S, Hadi AS (1986) Influential observations, high leverage points, and outliers in linear regression. Stat Sci 1(3):379–393

Chen X, Xie M (2014) A split-and-conquer approach for analysis of extraordinarily large data. Stat Sin 24:1655–1684

Clarkson KL, Woodruff DP (2013) Low rank approximation and regression in input sparsity time. In: Proceedings of the forty-fifth annual ACM symposium on theory of computing. ACM, New York, pp 81–90

Clarkson KL, Drineas P, Magdon-Ismail M, Mahoney MW, Meng X, Woodruff DP (2013) The Fast Cauchy Transform and faster robust linear regression. In: Proceedings of the twenty-fourth annual ACM-SIAM symposium on discrete algorithms. Society for Industrial and Applied Mathematics, Philadelphia, pp 466–477

Coles S, Bawa J, Trenner L, Dorazio P (2001) An introduction to statistical modeling of extreme values, vol 208. Springer, Berlin

Drineas P, Mahoney MW, Muthukrishnan S (2006) Sampling algorithms for ℓ_2 regression and applications. In: Proceedings of the 17th annual ACM-SIAM symposium on discrete algorithms, pp 1127–1136

Drineas P, Mahoney MW, Muthukrishnan S, Sarlós T (2010) Faster least squares approximation. Numer Math 117(2):219–249

Drineas P, Magdon-Ismail M, Mahoney MW, Woodruff DP (2012) Fast approximation of matrix coherence and statistical leverage. J Mach Learn Res 13:3475–3506

Duchi JC, Agarwal A, Wainwright MJ (2012) Dual averaging for distributed optimization: convergence analysis and network scaling. IEEE Trans Autom Control 57(3):592–606

Friedman J, Hastie T, Tibshirani R (2001) The elements of statistical learning, vol 1. Springer series in statistics. Springer, Berlin

Golub GH, Van Loan CF (1996) Matrix computations. Johns Hopkins University Press, Baltimore

Hesterberg T (1995) Weighted average importance sampling and defensive mixture distributions. Technometrics 37(2):185–194

Hoaglin DC, Welsch RE (1978) The hat matrix in regression and ANOVA. Am Stat 32(1):17–22

Lichman M (2013) UCI machine learning repository

Ma P, Sun X (2015) Leveraging for big data regression. Wiley Interdiscip Rev Comput Stat 7(1):70–76

Ma P, Mahoney MW, Yu B (2014) A statistical perspective on algorithmic leveraging. In: Proceedings of the 31st international conference on machine learning (ICML-14), pp 91–99

Ma P, Mahoney MW, Yu B (2015) A statistical perspective on algorithmic leveraging. J Mach Learn Res 16:861–911

Ma P, Zhang X, Ma J, Mahoney MW, Yu B, Xing X (2016) Optimal subsampling methods for large sample linear regression. Technical report, Department of Statistics, University of Georgia

Mahoney MW (2011) Randomized algorithms for matrices and data. Foundations and trends in machine learning. NOW Publishers, Boston. Also available at: arXiv:1104.5557

Mahoney MW, Drineas P (2009) CUR matrix decompositions for improved data analysis. Proc Natl Acad Sci 106(3):697–702

McCullagh P, Nelder JA (1989) Generalized linear models, vol 37. CRC, Boca Raton

Meng X, Mahoney MW (2013) Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In: Proceedings of the forty-fifth annual ACM symposium on theory of computing. ACM, New York, pp 91–100

Meng X, Saunders MA, Mahoney MW (2014) LSRN: a parallel iterative solver for strongly over-or underdetermined systems. SIAM J Sci Comput 36(2):C95–C118

Raskutti G, Mahoney MW (2016) A statistical perspective on randomized sketching for ordinary least-squares. J Mach Learn Res 17(214):1–31

Velleman PF, Welsch ER (1981) Efficient computing of regression diagnostics. Am Stat 35(4): 234–242

Wang H, Zhu R, Ma P (2017) Optimal subsampling for large sample logistic regression. J Am Stat Assoc (in press)

Xie R, Sriram TN, Ma P (2017) Sequential leveraging sampling method for streaming time series data. Technical report, Department of Statistics University of Georgia

Zhang Y, Duchi JC, Wainwright MJ (2013) Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates. CoRR. abs/1305.5029