Outline: Strategies for Large, Unbalanced Samples
Subsampling, Partitioning and Rebalancing

(F. Chiaromonte)

- A key reference: R. Zhu, P. Ma, M. Mahoney, B. Yu (2015) Optimal Subsampling Approaches for Large Sample Linear Regression. arXiv 1509.05111v1.
- Additional references: throughout the slides.

Subsampling and Partitioning Approaches

A SHORT REMINDER ON THE NON-PARAMETRIC BOOTSTRAP (resampling with replacement)

The bootstrap algorithm:

- Generate B bootstrap samples of size n drawing with replacement from the data

  $x^*_{(b)} \quad b = 1 \dots B$

- On each, compute the statistic of interest – producing B bootstrap values; "copies" that mimic its sampling variability

  $\hat{\theta}^*_{(b)} = g(x^*_{(b)}) \quad b = 1 \dots B$

  - Estimate the standard error as the standard deviation of the bootstrap values

$$\widehat{se}_{BT} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}^*_{(b)} - \hat{\theta}^*_{(\cdot)})^2} \qquad \hat{\theta}^*_{(\cdot)} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^*_{(b)}$$

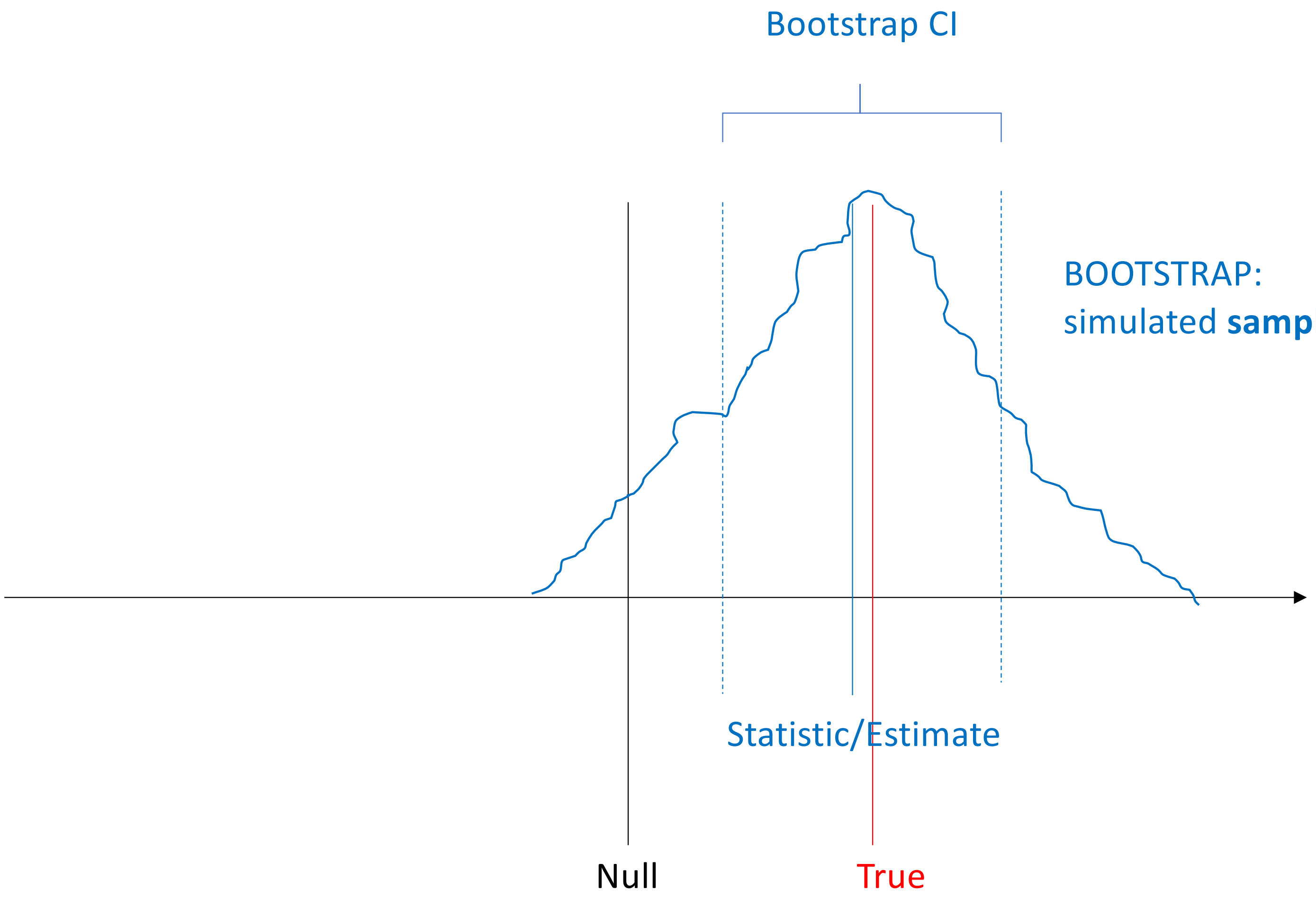
Already rather good with **B=200**.

  - Build the corresponding empirical cdf $\hat{G}$, a simulated version of the sampling distribution

  - Use its percentiles to define a percentile-based (1-α) coverage confidence interval as

$$LOW = G^{-1}\left(\frac{\alpha}{2}\right) \quad UP = G^{-1}\left(1 - \frac{\alpha}{2}\right)$$

Not the ultimate answer (can be further improved bias and potential non-constant variance corrections), but good with **B=1000, 2000**.



Bootstrap CI

Statistic/Estimate

Null

True

BOOTSTRAP:
simulated **sampling distribution** of the estimator

Rationale 1: since we cannot take more samples of size n from F, we do the next best thing, i.e., generate them from the empirical distribution $\hat{F}_n$. This is our best estimate of F based on the available sample of n observations (in fact, its non-parametric MLE).

Rationale 2: the bootstrap lets us gauge the variability of $\hat{\theta}$ creating perturbations of the original data by resampling (with replacement). These are less local of the perturbations created with the Jackknife (create n samples of size n-1 deleting one observation at a time).

BACK TO ULTRA-HIGH SAMPLE SIZE

Stochastic mechanism:

$$Y, \varepsilon \;\; r.vbls \; in \; R^1 \;\; X \; r.vct \; in \; R^p$$

$$E(X) = 0 \quad Cov(X) = \Sigma_X$$

$$\varepsilon \; indep \; X \quad \varepsilon \sim N(0, \sigma^2)$$

Estimation (fitting)

$$\hat{\beta}_{LS} = (X'X)^{-1}X'Y$$

When n is very large, computationally prohibitive even for relatively small p; O(np^2)

Independent sampling:

$$Y_{nx1} = X_{nxp}\beta_{px1} + \varepsilon_{nx1}$$

An intuitive solution: **SUBSAMPLING**

(A) Subsample with replacement from the data (empirical distribution; weight of each data point $w_{i,ID}$=1/n)

$$Y^{*}_{rx1};\ X^{*}_{rxp} \quad r \ll n$$

UNIF

UNIF refers to the fact that in subsampling each original data point has the same weight.

(B) Use the subsample as a surrogate and compute

$$\hat{\beta}^{*}_{LS}(w_{ID}) = (X^{*\prime} X^{*})^{-1} X^{*\prime} Y^{*}$$

- Approximating the full LS vector: (Conditionally on the data) $E(\hat{\beta}^{*}_{LS}(w_{ID})|F_n) = \hat{\beta}_{LS}$
- Estimating the true effects: (Unconditionally) $E(\hat{\beta}^{*}_{LS}(w_{ID})) = \beta$

Problem: **accuracy of the estimator can be pretty bad for moderate r!**

To improve accuracy, we need to **weigh the data points non-uniformly** when subsampling.

How? People have used **leverage scores** – exact or approximate:

$H = X(X'X)^{-1}X'$ Orthogonal projection operator of the LS (Hat-matrix)

$$h_{ii} = x_i(X'X)^{-1}x_i = ||x_i||^2_{(X'X)^{-1}}$$

$w_i = h_{ii}/\mathrm{p}$ **BLEV**

Basic Leverage Scores; length (squared) of x_i in the S_X^{-1} inner product.
Rationale: **represent the periphery of the data** when subsampling.

Estimating the leverage scores on the full data is prohibitive (as much as computing the full LS!) so people proposed approximate calculations **ALEV**

Some people also "shrink" the leverage weighting to the uniform weighting **SLEV**

Non-uniform weights induce a bias in approximating the full LS, which cannot be controlled with r (does not vanish as the subsample size grows)… but it reduces variance in approximating the full LS. In all, accuracy improves.

Note: *they do NOT induce bias in estimating the true effects*… more later on variance/accuracy.

Important results in the paper:

To correct the bias in approximating the full LS, one needs to ***weigh the LS*** on the subsample according to the weights used in subsampling!

$$\hat{\beta}^{*}_{WLS}(w) = (X^{*\prime} W X^{*})^{-1} X^{*\prime} W^{-1/2} Y^{*} \qquad W = Diag(w)$$

Under regularity conditions (bounding high order moments of the X's on the full data with weights), for large enough r

- **Thm 1**. Approximating the full LS vector: (Conditionally on the data) the weighted estimator is approximately Gaussian around the full LS with covariance matrix V(w)

$$V^{-1/2}(w)(\hat{\beta}^{*}_{WLS}(w) - \hat{\beta}_{LS}) \xrightarrow[r\to\infty]{D} \mathrm{N}(0, \mathrm{I})$$

- **Thm 2.** Estimating the true effects: (Unconditionally) the weighted estimator is approximately Gaussian around the true β with covariance matrix V_o (w) > V(w)

$$V_o^{-1/2}(w)(\hat{\beta}^{*}_{WLS}(w) - \beta) \xrightarrow[r\to\infty]{D} \mathrm{N}(0, \mathrm{I})$$

- **Thm 3.** Provided r grows linearly with n (e.g. a given fraction), weighing the LS actually does NOT improve accuracy for estimating the true effects

$$V_O(\hat{\beta}^{*}_{LS}(w)) \leq V_O(\hat{\beta}^{*}_{WLS}(w))$$

… FOR ESTIMATING THE TRUE β IT IS BETTER TO WEIGH THE SUBSAMPLING, BUT NOT TO WEIGH THE LS!

Also, the authors demonstrate that **(B,A,S) LEV weighting is NOT optimal in increasing accuracy** for approximating the full LS. The **Optimal weights** have a different form

$$w_i = \frac{\sqrt{(1-h_{ii})}||x_i||}{\sum\sqrt{(1-h_{jj})}||x_j||} = \frac{\sqrt{(1-||x_i||^2_{(X'X)^{-1}})||x_i||^2}}{\sum\sqrt{(1-||x_j||^2_{(X'X)^{-1}})||x_j||^2}}$$

OPT

Assuming the data is not too dispersed (compact bulk, not important how we account for the periphery) so that the leverages are not too heterogeneous (all ~p/n) we can simply use what they call **Predictor Length weights**

$$w_i = \frac{||x_i||}{\sum||x_j||}$$

PL

Similar to the LEV weights – but using the standard inner product!

Much cheaper to compute since they do not involve S^{-1} ; O(np).

… FOR APPROXIMATING THE FULL LS, SUBSAMPLE WITH PL WEIGHTS AND USE THE PL-WEIGHTED ESTIMATOR.

… FOR ESTIMATING THE TRUE β, SUBSAMPLE WITH PL WEIGHTS AND USE THE UNWEIGHTED ESTIMATOR.

SIMULATIONS TO EVALUATE PERFORMANCE ON “FINITE SUBSAMPLES”

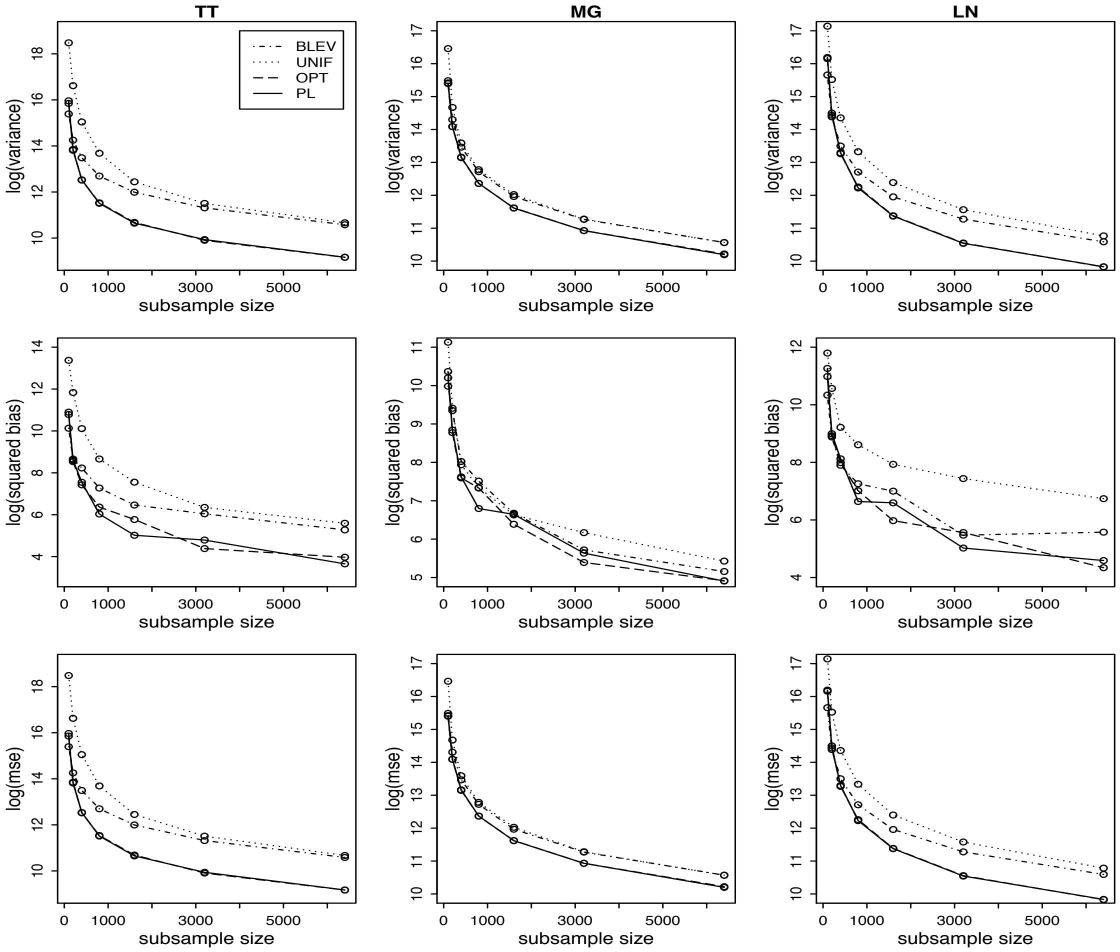
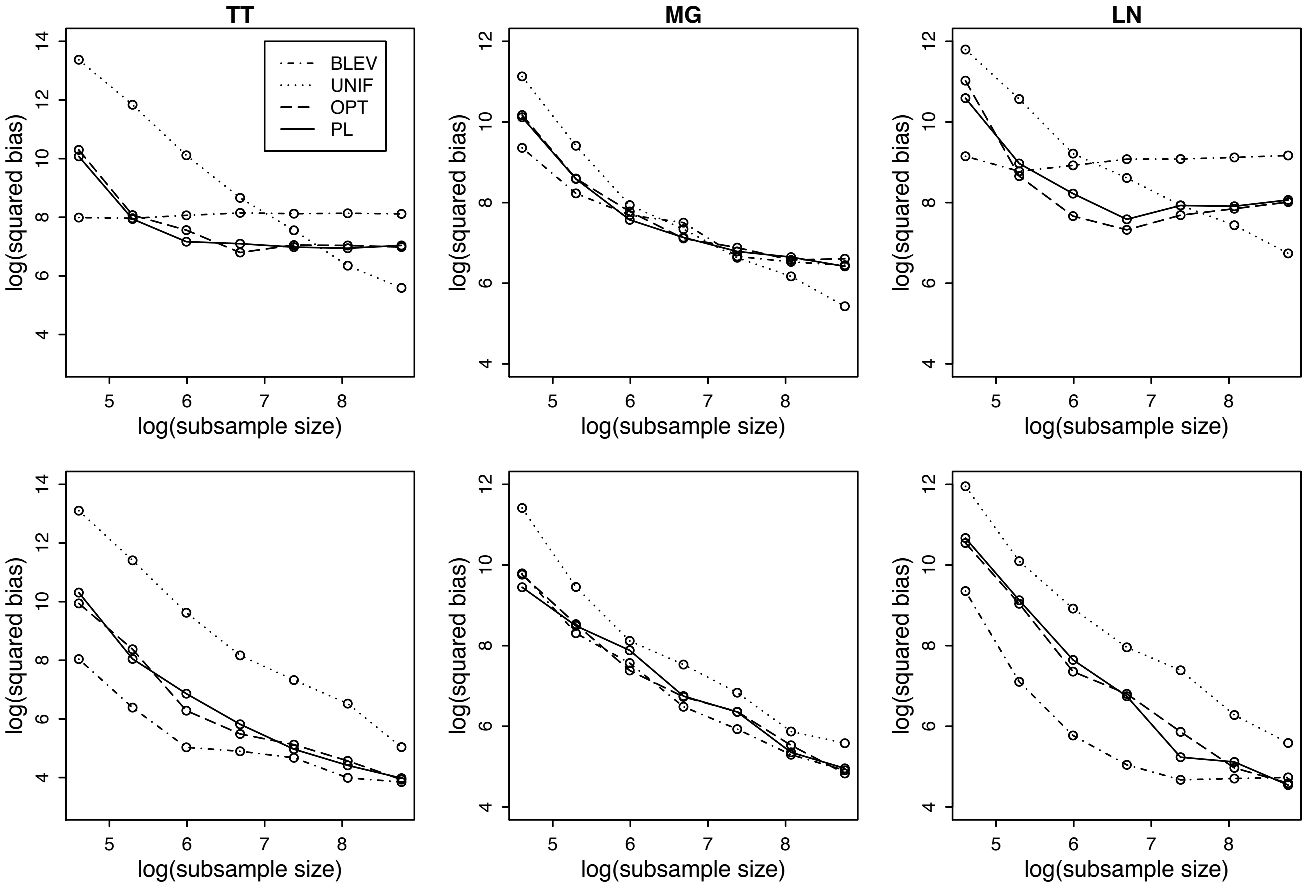


Figure 4: Empirical variances, squared biases and mean-squared error of $\mathbf{X}\tilde{\boldsymbol{\beta}}$ for predicting $\mathbf{X}\hat{\boldsymbol{\beta}}_{ols}$. Upper panels are the logarithm of variances, middle panels the logarithm of squared bias, and lower panels the logarithm of mean-squared error. From left to right: TT, MG and LN data, respectively.

NOTE: here empirical variances, squared biases and MSEs from simulations are on response predictions, not β estimates.

SIMULATIONS TO EVALUATE PERFORMANCE ON "FINITE SUBSAMPLES"



NOTE: here empirical variances, squared biases and MSEs from simulations are on response predictions, not β estimates.

Figure 5: Empirical squared biases of $\mathbf{X}\tilde{\boldsymbol{\beta}}^u$ for predicting $\mathbf{X}\hat{\boldsymbol{\beta}}_{ols}$ and $\mathbf{X}\boldsymbol{\beta}$, respectively. From top to bottom: upper panels are results for predicting $\mathbf{X}\hat{\boldsymbol{\beta}}_{ols}$, and lower panels results for predicting $\mathbf{X}\boldsymbol{\beta}$. From left to right: TT, MG and LN data, respectively.

ADDITIONAL REFERENCE:

A similar analysis is provided for **GLM**, logistic regression with ultra-high sample sizes:

H. Wang, R. Zhu, P. Ma (2018) Optimal Subsampling for Large Sample Logistic Regression. JASA, 113(522): 829–844.

Abstract: *For massive data, the family of subsampling algorithms is popular to downsize the data volume and reduce computational burden. Existing studies focus on approximating the ordinary least squares estimate in linear regression, where statistical leverage scores are often used to define subsampling probabilities. In this paper, we propose fast subsampling algorithms to efficiently approximate the maximum likelihood estimate in logistic regression. We first establish consistency and asymptotic normality of the estimator from a general subsampling algorithm, and then derive optimal subsampling probabilities that minimize the asymptotic mean squared error of the resultant estimator. An alternative minimization criterion is also proposed to further reduce the computational cost. The optimal subsampling probabilities depend on the full data estimate, so we develop a two-step algorithm to approximate the optimal subsampling procedure. This algorithm is computationally efficient and has a significant reduction in computing time compared to the full data approach. Consistency and asymptotic normality of the estimator from a two-step algorithm are also established. Synthetic and real data sets are used to evaluate the practical performance of the proposed method.*

An alternative to subsampling: **SPLIT-AND-CONQUER, PARTITIONING STRATEGIES**.

Note: sometimes the data cannot be (physically) merged in one “location” to perform a computation.

For a given procedure one needs to

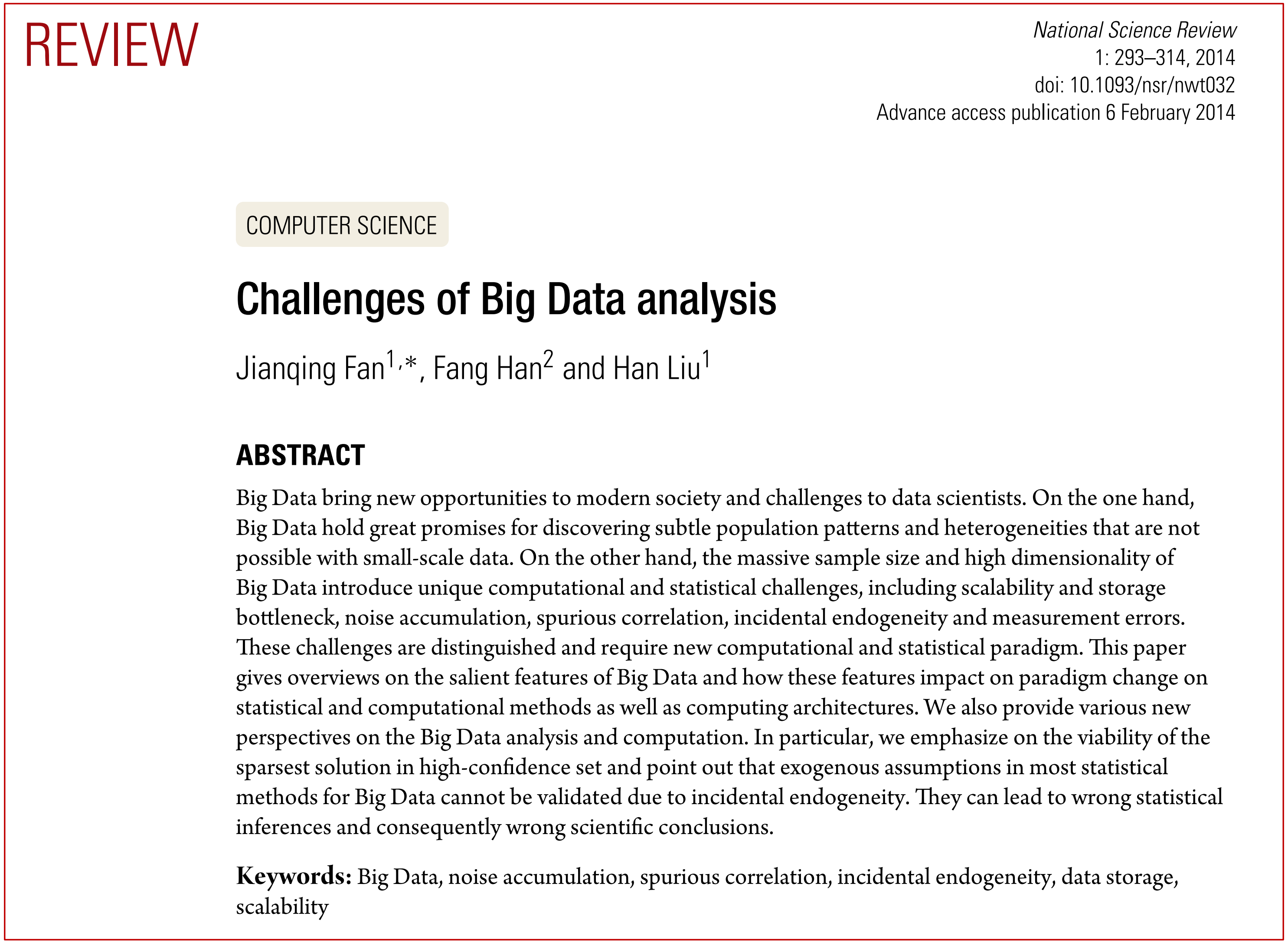
- split
- perform peripheral computation
- and then re-merge in a statistically appropriate way.

Some references:

- J. Fan, F. Han, H. Liu (2014) Challenges of big data analysis. National Science Review.
- A. Kleiner, A. Talwalkar, P. Sarkar, M. Jordan (2014) A scalable bootstrap for massive data. JRSS-B.
- X. Chen, M. Xie (2014) A split-and-conquer approach for analysis of extraordinarily large data. Statistica Sinica.
- Y. Zhang, J. Duchi, M. Weinwright (2015) Divide and conquer kernel Ridge regression: a distributed algorithm with minimax optimal rates. Journal of Machine Learning Research.

adding large p, perhaps in some of the parts…

- H. Battey, J. Fan, H. Liu, J. Lu, Z. Zhu (2018) Distributed testing and estimation under sparse high dimensional models. AoS.
- L. Tang, L. Zho, P. Song (2020) Distributed Simultaneous Inference in Generalized Linear Models via Confidence Distribution. Journal of Multivariate Analysis.



REVIEW

National Science Review
1: 293–314, 2014
doi: 10.1093/nsr/nwt032
Advance access publication 6 February 2014

COMPUTER SCIENCE

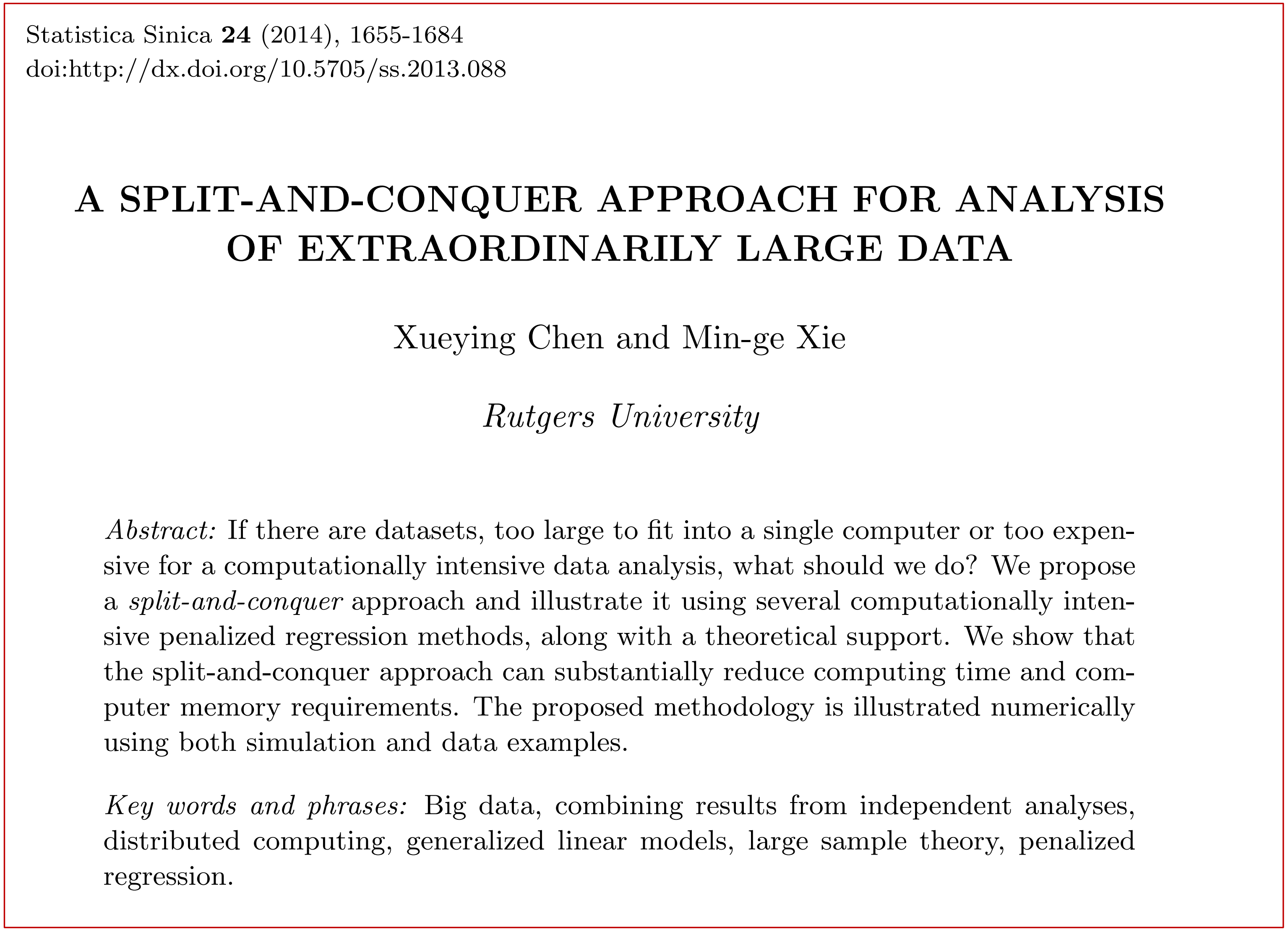
Challenges of Big Data analysis

Jianqing Fan[1,*], Fang Han[2] and Han Liu[1]

ABSTRACT

Big Data bring new opportunities to modern society and challenges to data scientists. On the one hand, Big Data hold great promises for discovering subtle population patterns and heterogeneities that are not possible with small-scale data. On the other hand, the massive sample size and high dimensionality of Big Data introduce unique computational and statistical challenges, including scalability and storage bottleneck, noise accumulation, spurious correlation, incidental endogeneity and measurement errors. These challenges are distinguished and require new computational and statistical paradigm. This paper gives overviews on the salient features of Big Data and how these features impact on paradigm change on statistical and computational methods as well as computing architectures. We also provide various new perspectives on the Big Data analysis and computation. In particular, we emphasize on the viability of the sparsest solution in high-confidence set and point out that exogenous assumptions in most statistical methods for Big Data cannot be validated due to incidental endogeneity. They can lead to wrong statistical inferences and consequently wrong scientific conclusions.

Keywords: Big Data, noise accumulation, spurious correlation, incidental endogeneity, data storage, scalability



Statistica Sinica **24** (2014), 1655-1684
doi:http://dx.doi.org/10.5705/ss.2013.088

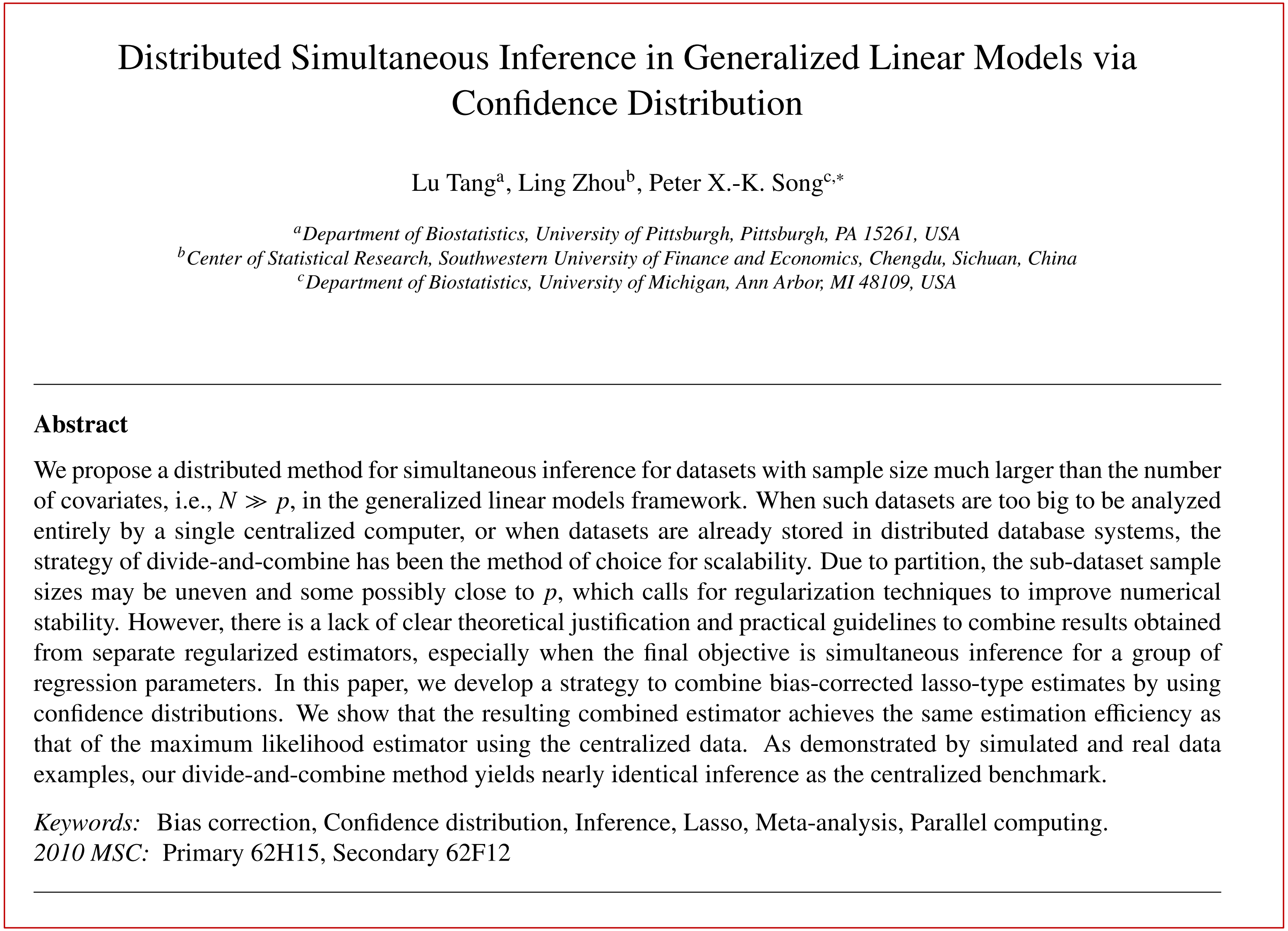
A SPLIT-AND-CONQUER APPROACH FOR ANALYSIS OF EXTRAORDINARILY LARGE DATA

Xueying Chen and Min-ge Xie

Rutgers University

Abstract: If there are datasets, too large to fit into a single computer or too expensive for a computationally intensive data analysis, what should we do? We propose a *split-and-conquer* approach and illustrate it using several computationally intensive penalized regression methods, along with a theoretical support. We show that the split-and-conquer approach can substantially reduce computing time and computer memory requirements. The proposed methodology is illustrated numerically using both simulation and data examples.

Key words and phrases: Big data, combining results from independent analyses, distributed computing, generalized linear models, large sample theory, penalized regression.



Distributed Simultaneous Inference in Generalized Linear Models via Confidence Distribution

Lu Tang[a], Ling Zhou[b], Peter X.-K. Song[c,*]

[a]*Department of Biostatistics, University of Pittsburgh, Pittsburgh, PA 15261, USA*
[b]*Center of Statistical Research, Southwestern University of Finance and Economics, Chengdu, Sichuan, China*
[c]*Department of Biostatistics, University of Michigan, Ann Arbor, MI 48109, USA*

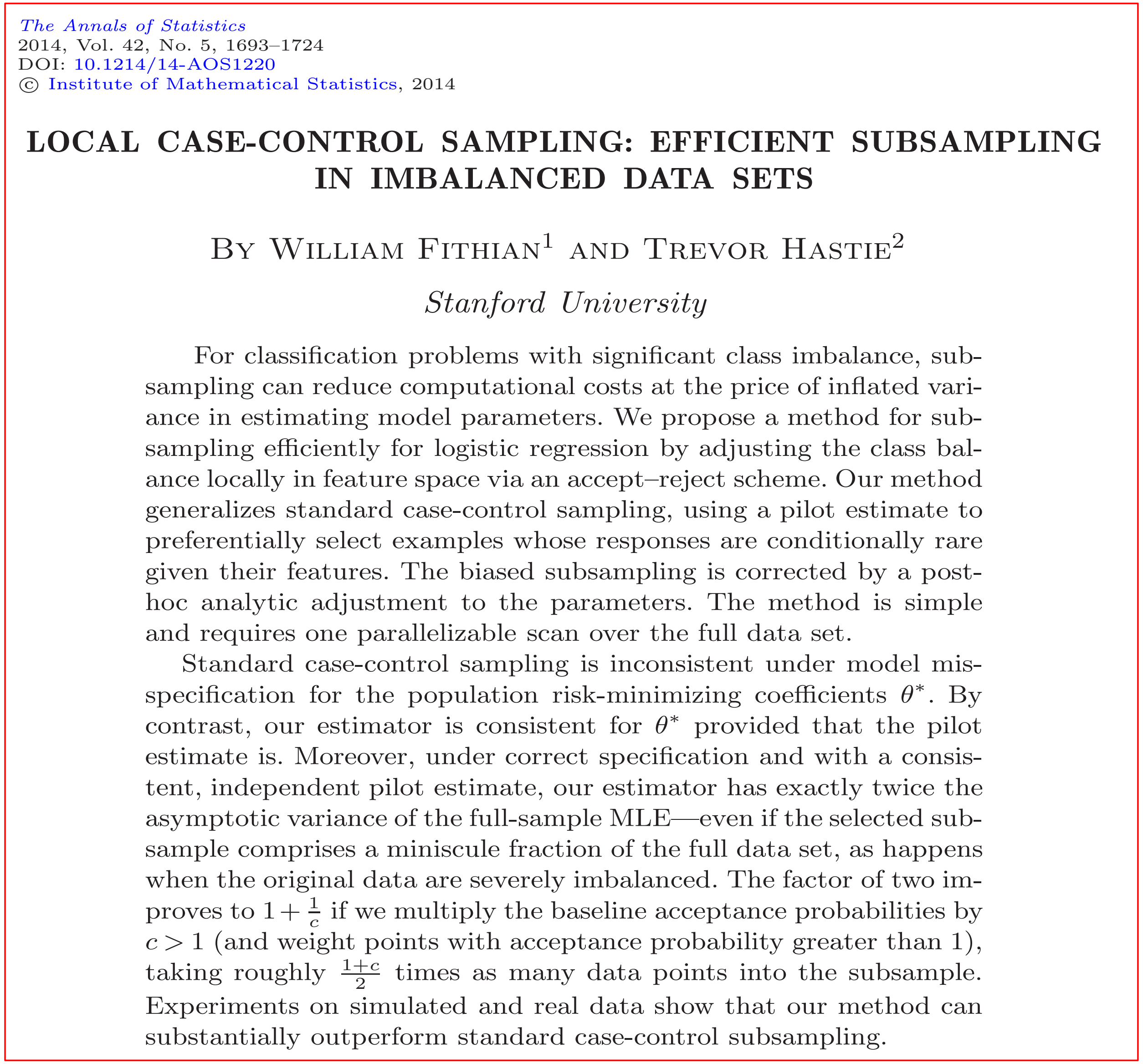
Abstract

We propose a distributed method for simultaneous inference for datasets with sample size much larger than the number of covariates, i.e., $N \gg p$, in the generalized linear models framework. When such datasets are too big to be analyzed entirely by a single centralized computer, or when datasets are already stored in distributed database systems, the strategy of divide-and-combine has been the method of choice for scalability. Due to partition, the sub-dataset sample sizes may be uneven and some possibly close to p, which calls for regularization techniques to improve numerical stability. However, there is a lack of clear theoretical justification and practical guidelines to combine results obtained from separate regularized estimators, especially when the final objective is simultaneous inference for a group of regression parameters. In this paper, we develop a strategy to combine bias-corrected lasso-type estimates by using confidence distributions. We show that the resulting combined estimator achieves the same estimation efficiency as that of the maximum likelihood estimator using the centralized data. As demonstrated by simulated and real data examples, our divide-and-combine method yields nearly identical inference as the centralized benchmark.

Keywords: Bias correction, Confidence distribution, Inference, Lasso, Meta-analysis, Parallel computing.
2010 MSC: Primary 62H15, Secondary 62F12

Approaches for Unbalanced Classification Problems

A reference article on subsampling in unbalanced classification problems:



The Annals of Statistics
2014, Vol. 42, No. 5, 1693–1724
DOI: 10.1214/14-AOS1220

LOCAL CASE-CONTROL SAMPLING: EFFICIENT SUBSAMPLING IN IMBALANCED DATA SETS

By William Fithian[1] and Trevor Hastie[2]

Stanford University

For classification problems with significant class imbalance, subsampling can reduce computational costs at the price of inflated variance in estimating model parameters. We propose a method for subsampling efficiently for logistic regression by adjusting the class balance locally in feature space via an accept–reject scheme. Our method generalizes standard case-control sampling, using a pilot estimate to preferentially select examples whose responses are conditionally rare given their features. The biased subsampling is corrected by a post-hoc analytic adjustment to the parameters. The method is simple and requires one parallelizable scan over the full data set.

Standard case-control sampling is inconsistent under model misspecification for the population risk-minimizing coefficients θ^*. By contrast, our estimator is consistent for θ^* provided that the pilot estimate is. Moreover, under correct specification and with a consistent, independent pilot estimate, our estimator has exactly twice the asymptotic variance of the full-sample MLE—even if the selected subsample comprises a miniscule fraction of the full data set, as happens when the original data are severely imbalanced. The factor of two improves to $1+\frac{1}{c}$ if we multiply the baseline acceptance probabilities by $c>1$ (and weight points with acceptance probability greater than 1), taking roughly $\frac{1+c}{2}$ times as many data points into the subsample. Experiments on simulated and real data show that our method can substantially outperform standard case-control subsampling.

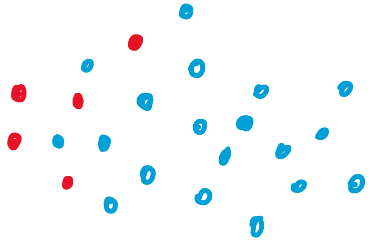
Some more references

Dubey, Rashmi et al. “Analysis of sampling techniques for imbalanced data: An n = 648 ADNI study.” *NeuroImage* vol. 87 (2014): 220-41. doi:10.1016/j.neuroimage.2013.10.005

Menardi, Torrelli (2010) “Training and assessing classification rules with unbalanced data”. Working Paper 2-2010. Dipartimento B. De Finetti, Universita’ di Trieste.

Imbalanced Learning tools, MIT (includes SMOTE)
https://imbalanced-learn.org/stable/index.html

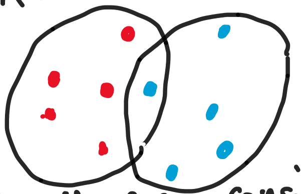
The data in feature space



"scarce" class n_R

"abundent" class n_B

① Reduce the abundant class



Keep these n_R fixed

consider only n_R of these at random

... can "repeat"

Form one, or several, datasets of size $2n_R < n_R + n_B$.

The red points are always the same.

(1B) A variant

Bootstrap the red points (resampling with replacement)

Sub-bootstrap the blue points (Select n_R blue points at random with replacement)

... can "repeat" both

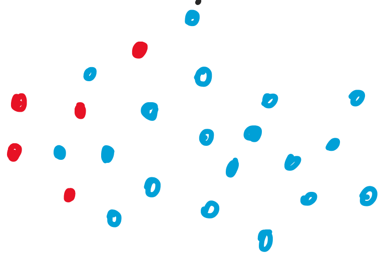
Form one, or several, datasets of size $2n_R < n_R + n_B$

The red points are bootstrapped too.

... MAKES MORE SENSE STATISTICALLY

we simulate sampling from the two populations.

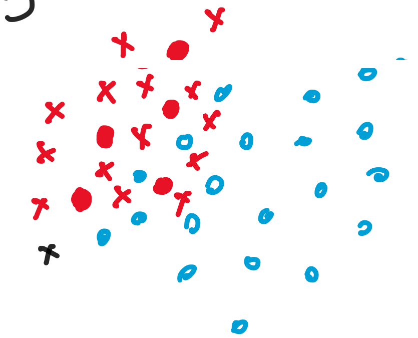
The data in feature space



"scarce" class n_R

"abundent" class n_B

② Augment the scarce class



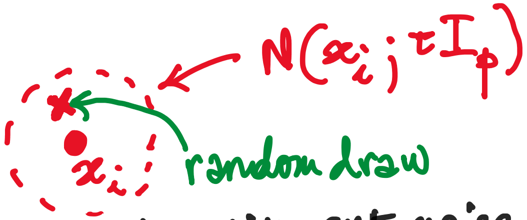
● = actual red points

× = artificial red points

Form a dataset of size $2n_B > n_R + n_B$

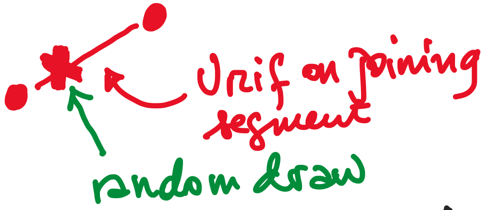
How do we create the artificial points? some options

(i) Over-bootstrap the red points (select n_B red points at random with replacement) ADDING NOISE to each draw



(can specify different noise models)

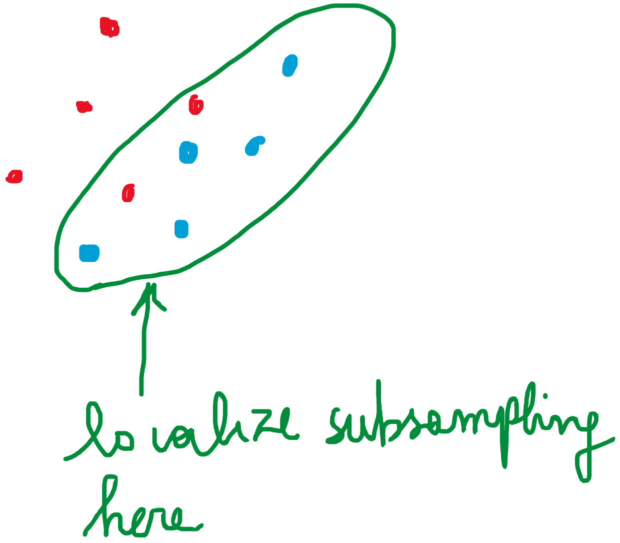
(ii) n_B times over: select at random two red points and a point between them



(can "localize" the selection of the red points, e.g., first at random, second at random among its closest red neighbors)

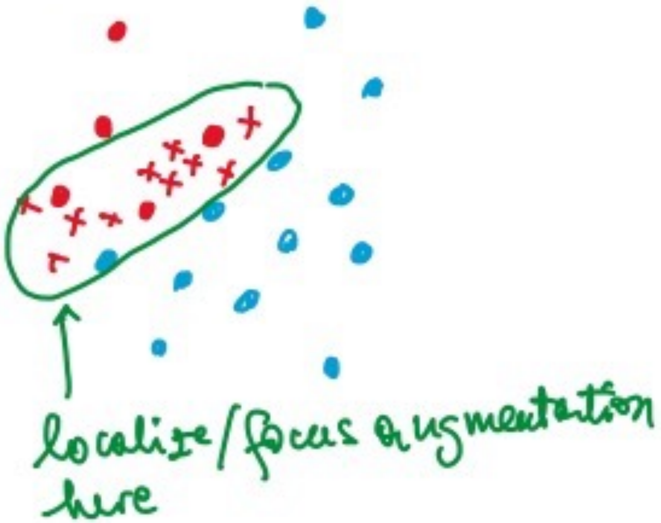
Localizing the reduction or the augmentation

preserve the abundant class where it is harder to discriminate, i.e., close to the red points



Focus the augmentation of the scarce class where it is harder to discriminate, i.e., close to the blue points.

... MAKES MORE SENSE STATISTICALLY


localize/focus augmentation here