A Split-and-Conquer Approach for Analysis of Extremely Large Data

Min-ge Xie

Department of Statistics and Biostatistics, Rutgers University

Joint work with **Xueying Chen**

Research supported in part by grants from NSF and DHS

June 7, 2013

Extremely large data

"The modern era of electronic data capture has given rise to extremely large data in almost every major field of human and machine interaction. Examples can be seen in the engineering and technology fields, retail consumer levels, drug development, genomics, and from various (including medical) insurance claims. Processing of such data frequently requires high performance computing and novel statistical approaches that collectively emphasize analysis, summary and visualization of big data. "

From *2013 NJ ASA Spring Symposium Announcement*

Difficulties in analysis of extremely large data

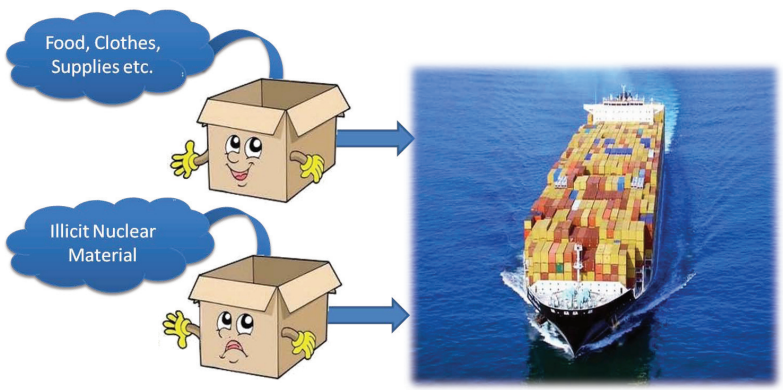
- Memory/Storage issue: too large to fit into a single computer.
- Computing time: too expensive to perform a computationally intensive analysis
  - Example: Penalized regression method

$$\hat{\boldsymbol{\beta}}^{(a)} = \text{argmax}_{\boldsymbol{\beta}} \left\{ \ell(\boldsymbol{\beta}; \boldsymbol{y}, \boldsymbol{X})/n - \rho(\boldsymbol{\beta}; \lambda_a) \right\},$$

    where ℓ is log-likelihood function, $\boldsymbol{y}_{n\times 1}$ are the responses, $\boldsymbol{X}_{n\times p}$ are the predictors, $\boldsymbol{\beta}$ are the coefficients, and ρ is a penalty function with tuning parameter λ_a.
  - The optimization is computationally intensive; e.g. LARS algorithm for LASSO regression is $O(n^2 p)$ when $p \geq n$.

Motivating example: Analysis of manifest data

- Port-of-entry inspection project (DHS/DIMACS)
  - Background: Standardized shipping containers transport 95% of the U.S. imports by tonnage; they are highly vulnerable vehicles for smuggling nuclear/radiological weapons and other illegal materials.
  - Inspection goal: Find shipments of high risk and identify important influencers (variables).
  - Approach: Analyze historical data and search for patterns
  - Problem: A huge amount of data



Manifest data

- Manifest data: A variety of cargo information from custom forms.
- One week's (02/20/2009 - 02/26/2009) manifest data of all shipments to the U.S.
  - 164721 shipments in all
  - 7 categorical variables with more than 200 dummy variables
  - Text fields can add even more variables
- Ultimate goal: potential approach for a scale-up analysis of manifest data in the entire data base

Dictionary of variables

Variables	Number of Categories	Definition
$\boldsymbol{X}_1$	9+	Vessel Country Code
$\boldsymbol{X}_2$	69+	Voyage Number
$\boldsymbol{X}_3$	9+	dp of Unlading
$\boldsymbol{X}_4$	14+	Foreign Port Lading
$\boldsymbol{X}_5$	68+	Foreign Port
$\boldsymbol{X}_6$	35+	Inbond Entry Type
$\boldsymbol{X}_7$	17+	Container Cotents

— Plus 3+ text fields of comments and free-styled texts

Penalized regression

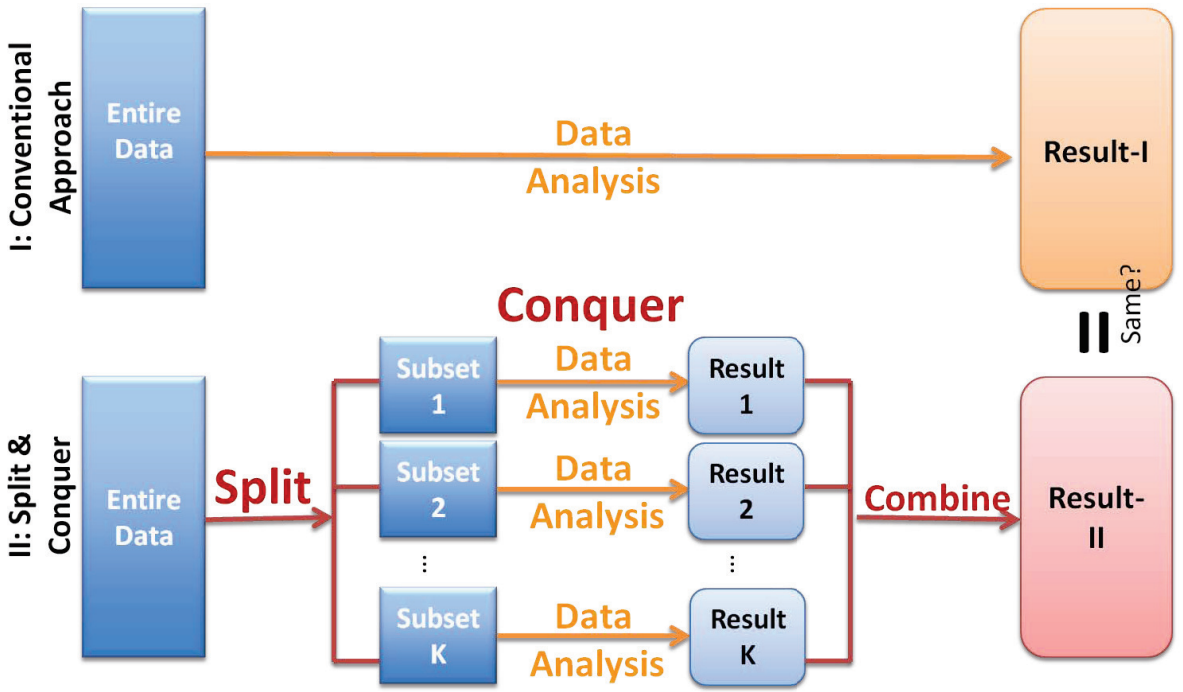
- Tasks:
  - Assign a risk score and flag for possible inspections of each shipment based on the information in its manifest forms
  - Identify variables that are relevant and ignore those that contain no information
- Complications:
  - Redundant information
  - Variables with high correlations
- A well-studied statistical method – penalized regression (variable selection/classification)

Research question

- **Question**: Is there any general scale-up solution for such problems without discarding any data?
  - Facts:
  - Only be able to analyze part of the data at one time.
  - Use existing statistical analysis procedures/algorithm
- **Our proposal**: A split – conquer – combine approach

Split–conquer – combine

Fig.1. A diagram of the Split and Conquer Approach and Research Objective



Research Question: Whether or under what conditions **Result-I = Result-II** ?

- Similar idea to parallel computing; but no systemic and theoretic study from statistical prospects

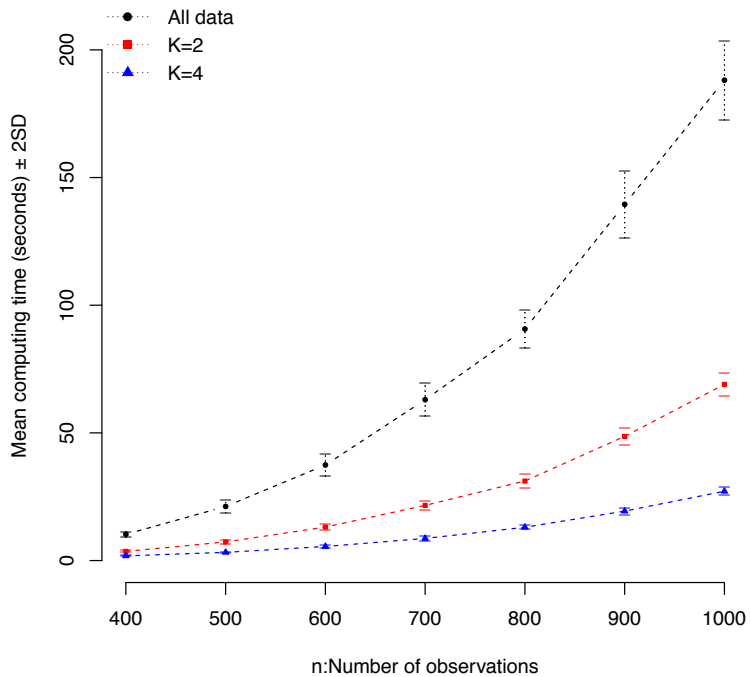
Desirable properties

- The result is **the same as/asymptotically equivalent to** the result of analyzing the entire data all at once
  - Makes it **feasible** to analyze extremely large data without the need of a super computer

Desirable properties

- The result is **the same as/asymptotically equivalent to** the result of analyzing the entire data all at once
  - Makes it **feasible** to analyze extremely large data without the need of a super computer
- The method can **substantially reduce computing time** and computer memory requirement, if a computationally intensive algorithm is used
  - For example, the computing order is $O(n^a)$, for an $a > 1$
- As a byproduct, the method is **more resistant to false model selections** caused by spurious correlations
  - Involves in combining the results from subsets of random splitting

Can the method substantially reduce computing time and computer memory requirement?



- Computing times (mean $\pm$ 2 sd) of K splits using LARS algorithm on 100 replications, for $p = 2n$ with $n = 400; 500; 600; 700; 800; 900; 1000$.

Can the combined result be the same as/asymptotically equivalent to the one using the entire dataset all together?

A simple case: regular Gaussian linear regression

- Ordinary least square estimator using the entire dataset

$$\hat{\beta}^{(a)} = \underline{(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}}.$$

- A combination of K least squared estimators $\hat{\beta}_k$, $k = 1, 2,$ ..., K, from K subsets:

$$\begin{aligned}\hat{\beta}^{(c)} &= (\sum_{k=1}^{K}\boldsymbol{X}_k^T\boldsymbol{X}_k)^{-1}\sum_{k=1}^{K}\{(\boldsymbol{X}_k^T\boldsymbol{X}_k)\hat{\beta}_k\} \\ &= (\sum_{k=1}^{K}\boldsymbol{X}_k^T\boldsymbol{X}_k)^{-1}\sum_{k=1}^{K}\boldsymbol{X}_k^T\boldsymbol{y}_k = \underline{(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}}.\end{aligned}$$

Can the combined result be the same as/asymptotically equivalent to the one using the entire dataset all together?

A simple case: regular Gaussian linear regression

- Ordinary least square estimator using the entire dataset

$$\hat{\beta}^{(a)} = \underline{(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}}.$$

- A combination of K least squared estimators $\hat{\beta}_k$, $k = 1, 2,$ ..., K, from K subsets:

$$\begin{aligned}\hat{\beta}^{(c)} &= (\sum_{k=1}^{K}\boldsymbol{X}_k^T\boldsymbol{X}_k)^{-1}\sum_{k=1}^{K}\{(\boldsymbol{X}_k^T\boldsymbol{X}_k)\hat{\beta}_k\} \\ &= (\sum_{k=1}^{K}\boldsymbol{X}_k^T\boldsymbol{X}_k)^{-1}\sum_{k=1}^{K}\boldsymbol{X}_k^T\boldsymbol{y}_k = \underline{(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}}.\end{aligned}$$

Our question: Do we have the same/similar result when more complicated methods are involved?

Illustration with penalized regression methods: GLM setting

- Consider a generalized linear model:

$$E(y_i) = g(\boldsymbol{x}_i^T \beta), i = 1, \ldots, n$$

where y_i is a response variable; $\boldsymbol{x}_i^{p\times 1}$ is the vector of explanatory variables; $\beta^{p\times 1}$ is the vector of unknown parameters; g is a link function; both n and p can be potentially very large.

- Given $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$, the conditional distribution of $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ follows:

$$f(\boldsymbol{y}; \boldsymbol{X}, \beta) = \prod_{i=1}^{n} f_0(y_i; \theta_i) = \prod_{i=1}^{n} \left\{ c(y_i) \exp\left[\frac{y_i \theta_i - b(\theta_i)}{\phi} \right] \right\}, \quad (1)$$

where $\theta_i = \boldsymbol{x}_i^T \beta, i = 1, \ldots, n$ and ϕ is a nuisance dispersion parameter.

Illustration with penalized regression methods: split & conquer

- Divide the whole dataset into K subsets and the k subset has n_k observations: $(\boldsymbol{x}_{k,i}, y_{k,i})$, $i = 1, \ldots, n_k$.
- Perform a penalized regression on each subset

$$\hat{\boldsymbol{\beta}}_k = \text{argmax}_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}; \boldsymbol{y}_k, \boldsymbol{X}_k)/n_k - \rho(\boldsymbol{\beta}; \lambda_k)$$

where $\ell(\boldsymbol{\beta}; \boldsymbol{y}_k, \boldsymbol{X}_k) = \log f(\boldsymbol{y}_k; \boldsymbol{X}_k, \boldsymbol{\beta})$ is the log-likelihood function, $\boldsymbol{y}_k^{n_k \times 1}$ is the response vector and $\boldsymbol{X}_k^{n_k \times p}$ is the design matrix of the k subset, and ρ is a penalty function with tuning parameter λ_k

Illustration with penalized regression methods: combine

- Majority voting for model selection $\Longrightarrow$ Keep sparsity

$$v_j = \begin{cases} 1 & \sum_{k=1}^{K} \mathbf{I}(\hat{\beta}_{k,j} \neq 0) > w \\ 0 & \text{otherwise} \end{cases}, \quad \text{for } j = 1, 2, ..., p$$

where w is a pre-fixed threshold.

- Weighted linear combination of penalized estimators from the subsets $\Longrightarrow$ Asymptotically unbiased + Normal

$$\hat{\boldsymbol{\beta}}^{(c)} \stackrel{\mathrm{d}}{=} \boldsymbol{A}\Big(\sum_{k=1}^{K} \boldsymbol{A}^T \boldsymbol{S}_k \boldsymbol{A}\Big)^{-1}\Big(\sum_{k=1}^{K} \boldsymbol{A}^T \boldsymbol{S}_k \boldsymbol{A} \hat{\boldsymbol{\beta}}_{k,\hat{\mathcal{A}}}\Big)$$

where $\boldsymbol{S}_k = \boldsymbol{X}_k^T b''(\boldsymbol{X}_k \hat{\boldsymbol{\beta}}_k) \boldsymbol{X}_k$ and $\boldsymbol{A} = \boldsymbol{E}_{\hat{\mathcal{A}}}$ with $\boldsymbol{E} = \mathrm{diag}(\boldsymbol{v}(w))$, $\hat{\mathcal{A}} = \{j : v_j \neq 0\}$ and $\boldsymbol{v}(w) = (v_1, ...v_p)$.

Model selection consistency

Theorem 1: Sparsity, L_∞-norm and L_2-norm consistency

Denote $\mathcal{A} = \{j : \beta_j^0 \neq 0\}$ and $\mathcal{B} = \bar{\mathcal{A}}$. Under some regularity conditions* and with probability approaching 1, as $n \to \infty$, we have

1 $\hat{\boldsymbol{\beta}}_{\mathcal{B}}^{(c)} = 0$;

2a $\|\hat{\boldsymbol{\beta}}_{\mathcal{A}}^{(c)} - \boldsymbol{\beta}_{\mathcal{A}}^0\|_\infty \leq b_*/2$, for some $\gamma \in [0, 1/2)$;

or

2b $\|\hat{\boldsymbol{\beta}}_{\mathcal{A}}^{(c)} - \boldsymbol{\beta}_{\mathcal{A}}^0\|_2 = O_p(\sqrt{s/n})$;

where, $b_* = \min\{|\beta_j^0| : j \in \mathcal{A}\}$ and $s =$ No. of elements in set $\mathcal{A}$.

Take home message: Behaves the same as the estimators $\hat{\boldsymbol{\beta}}^{(a)}$,

- The combined estimators of noninformative variables are 0 (sparsity)
- The combined estimators can sign-consistently/consistently estimate the non-zero coefficients (sign-consistency/consistency)

Oracle property

Theorem 2: Oracle property

Let $\boldsymbol{D}$ be a $q \times s$ matrix such that $\boldsymbol{D}\boldsymbol{D}' \to \boldsymbol{G}$, $\boldsymbol{G}$ is a $q \times q$ symmetric positive definite matrix. Under some regularity conditions*, we have

$$\boldsymbol{D}[\boldsymbol{X}_{\mathcal{A}}\Sigma(\theta^0)\boldsymbol{X}_{\mathcal{A}}]^{1/2}(\hat{\beta}_{\mathcal{A}}^{(c)} - \beta_{\mathcal{A}}^0) \xrightarrow{\mathrm{D}} N(\mathbf{0}, \phi\boldsymbol{G})$$

Take home message:

- The combined estimator $\hat{\beta}_{\mathcal{A}}^{(c)}$ has the same limiting distribution as $\hat{\beta}_{\mathcal{A}}^{(a)}$
- The combined estimators $\hat{\beta}^{(c)}$ and $\hat{\beta}^{(a)}$ are asymptotically equivalent (by the results of both Theorems 1 & 2)

Error bounds

Theorem 3: Model selection error bounds

Denote by $s^* = \max \bar{s}_k$ and $s_* = \min \bar{s}_k$, where $\bar{s}_k = E(|\hat{\mathcal{A}}_k|)$. Under some regularity conditions,

(i) the expected number of false selected variables has an upper bound: $E(|\mathcal{B} \cap \hat{\mathcal{A}}^{(c)}|) \leq |\mathcal{B}|\{1 - F(w|K, s^*/p)\}$,

(ii) the expected number of truly selected variables has a lower bound: $E(|\mathcal{A} \cap \hat{\mathcal{A}}^{(c)}|) \geq |\mathcal{A}|\{1 - F(w|K, s_*/p)\}$,

where $F(\cdot|m, q)$ is the cumulative distribution function of binomial distribution with m trials and success probability q.

Take home message:

- We have explicit formulas to bound model selection errors.
- A side benefit (automatic) is: more resistant to false model selection error caused by spurious correlation

Computing savings

Theorem 4: Computing savings

Assume a statistical procedure requires $O(n^a p)$ computing steps. Suppose the dataset is split into K subsets with sample size $n_k = O(n/K)$, for all $1 \leq k \leq K$, and the computing effort of the combination is ignorable. Then, the split–conquer–combine approach only needs $O(n^a p/K^{a-1})$ steps and the saving is $(1 - K^{-(a-1)})100\%$.

Take home message:

- When $a > 1$, the use of the split–conquer–combine approach can substantially reducing computing time.
  - For example: $a = 2$ and $K = 3 \Longrightarrow (1 - 3^{-1})100\% = 66.67\%$

But there is no free lunch ...

A further examination of the regularity conditions* in Theorems 1 & 2:

- The requirements of the signal strength and the sparsity assumption depends on the number of splits K.
  - When $K = O(1)$, we do not need any additional requirements
  - When $K \to \infty$, we require stronger conditions: $b_* = O(\sqrt{Ks/n})$ and $s = O(n^{1/3}/K^{2/3})$.
- Take home message: There is a price to pay when $K \to \infty$
  - Loosely speaking, we can only assure that the method can detect signals allowed by each subset of size $n_k = O(n/K)$.

A rule of thumb: choice of K

- K should be relatively large so that each subset is small enough to be analyzed by available computing resources.
  - Allow $K \to \infty$ as $n \to \infty$.
- K cannot be too large as each subset needs to contain enough data to provide a meaningful estimator.
  - $K \to \infty$ can not be too fast.

A rule of thumb: choice of K

- K should be relatively large so that each subset is small enough to be analyzed by available computing resources.
  - Allow $K \to \infty$ as $n \to \infty$.
- K cannot be too large as each subset needs to contain enough data to provide a meaningful estimator.
  - $K \to \infty$ can not be too fast.
- In our numerical studies:
  - Tried $K = 2, 4, 6$ and also $K = 10, 20, 100$ for large n ($n \geq 100,000$)

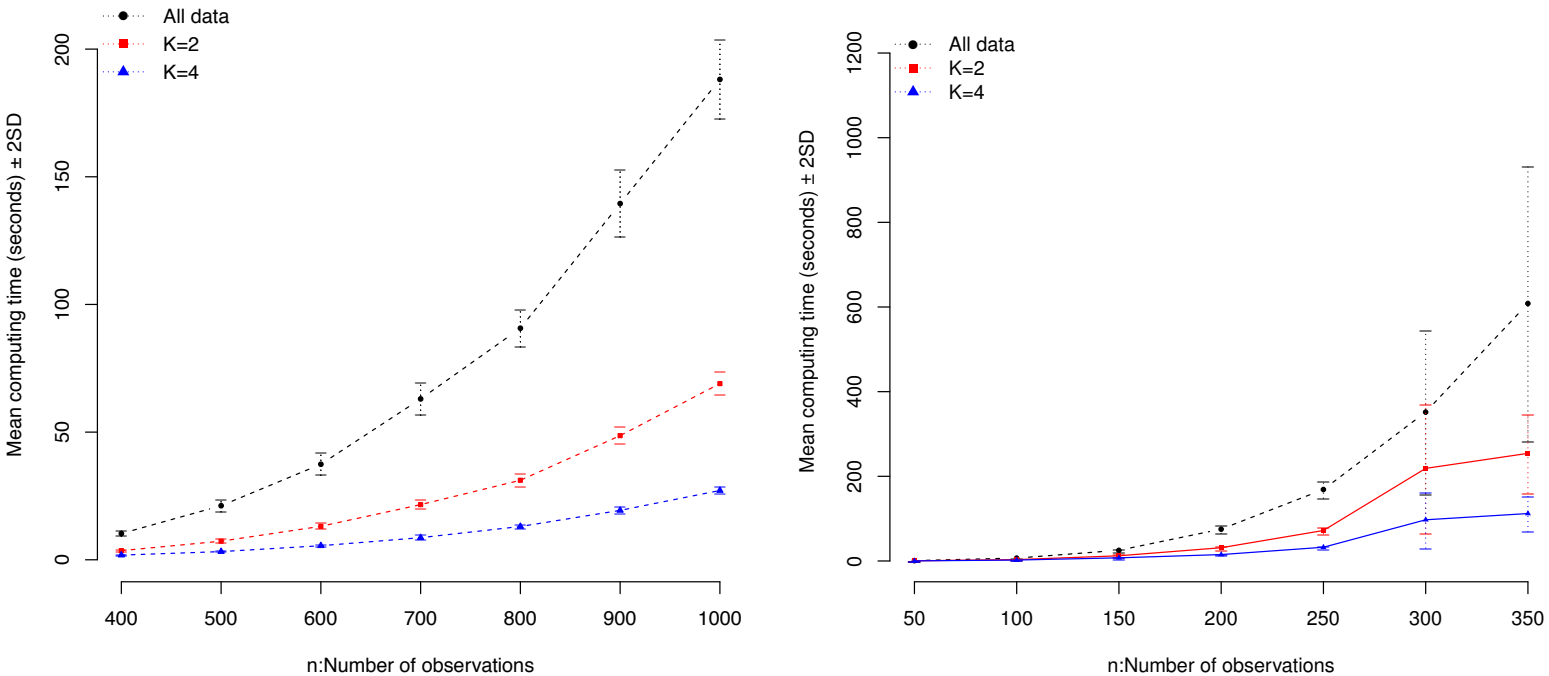
A rule of thumb: choice of K

- K should be relatively large so that each subset is small enough to be analyzed by available computing resources.
  - Allow $K \to \infty$ as $n \to \infty$.
- K cannot be too large as each subset needs to contain enough data to provide a meaningful estimator.
  - $K \to \infty$ can not be too fast.
- In our numerical studies:
  - Tried $K = 2, 4, 6$ and also $K = 10, 20, 100$ for large n ($n \geq 100,000$)
- Open question:
  - Do we have an "optimal" selection of K? Or, can we develop a criterion with a sound justification?

Simulation I: Computing savings

- Linear regression with L_1 penalty (LASSO)
- LARS package: computing order is $O(n^2p + n^3)$ when $p \geq n$
- Settings:
  - **(a)** $p = 2n$; $s = [\sqrt{p}]$; $n = 400, 500, 600, 700, 800, 900, 1000$
  - **(b)** $p = 100n$; $s = [\sqrt{p}]$; $n = 50, 100, 150, 200, 250, 300, 350$
- Repeat the analysis 100 times

Simulation I: Computing savings



- Computing times (mean $\pm$ 2 sd) versus n for $K = 1, 2, 4$ splits, using the LARS algorithm under the two settings (a) [left] and (b) [right].

  Replications = 100 at each point.

Simulation II: Asymptotic equivalence

1. Two settings of regression models
   - Linear regression: $\boldsymbol{y} = \boldsymbol{X}\beta + \varepsilon$
   - Logistic regression: $\boldsymbol{y} \in \{0, 1\}$ is generated with success probability $p(\boldsymbol{X}\beta) = e^{\boldsymbol{X}\beta}/(1 + e^{\boldsymbol{X}\beta})$
2. Two types of generated predictors $\boldsymbol{X}$
   - Independent variables: p independent variables are generated from an $N(0, \boldsymbol{I})$ distribution
   - Correlated variables: p correlated variables are generated from an $N(0, \Sigma)$ distribution, where $\Sigma(i, j) = 0.6^{|i-j|}$.
3. Two types of penalty functions
   - SCAD penalty
   - MCP
4. Repeat the analysis 100 times

Simulation II: Asymptotic equivalence

- Numerical results reported are:
  - *Computing time*
  - *Model selection accuracy*
    - Model selection **sensitivity**: # {truly selected variables}/the true model size
    - Model selection **specificity**: #{ truly removed variables}/ # noise variables
  - *Prediction accuracy*
    - Linear regression: **mean squared error**
    - Logistic regression: **misclassification rate**
- The side-by-side boxplots of $\hat{\beta}_{\mathcal{A}}^{(c)}$ versus $\hat{\beta}_{\mathcal{A}}^{(a)}$ are also presented, when both are available.

Simulation II: Asymptotic equivalence

Table : Linear regression with SCAD penalty

Simulation setting				Computing time (in second)	Model selection			MSE
Design matrix	n	p	K		# selected variables	sensitivity (in %)	specificity (in %)	
Independent	10000	1000	1	815.27 (77.98)	34.58 (9.81)	100 (0)	99.53 (1.01)	1.00 (0.01)
			10	104.96 (9.55)	30 (0)	100 (0)	100 (0)	1.00 (0.01)
Correlated	10000	1000	1	755.4 (157.56)	34.00 (12.22)	96.00 (19.79)	99.46 (1.02)	0.96 (0.20)
			10	289.17 (61.03)	28.72 (6.13)	95.87 (19.78)	100 (0)	1.00 (0.01)
Independent	100000	1000	1	- (-)	- (-)	- (-)	- (-)	- (-)
			100	1136.70 (74.65)	30 (0)	100 (0)	100 (0)	1.00 (0.01)
Correlated	100000	1000	1	- (-)	- (-)	- (-)	- (-)	- (-)
			100	3074.53 (25.01)	30 (0)	100 (0)	100 (0)	1.06 (0.01)

Note: Linear regression with MCP performs similarly.

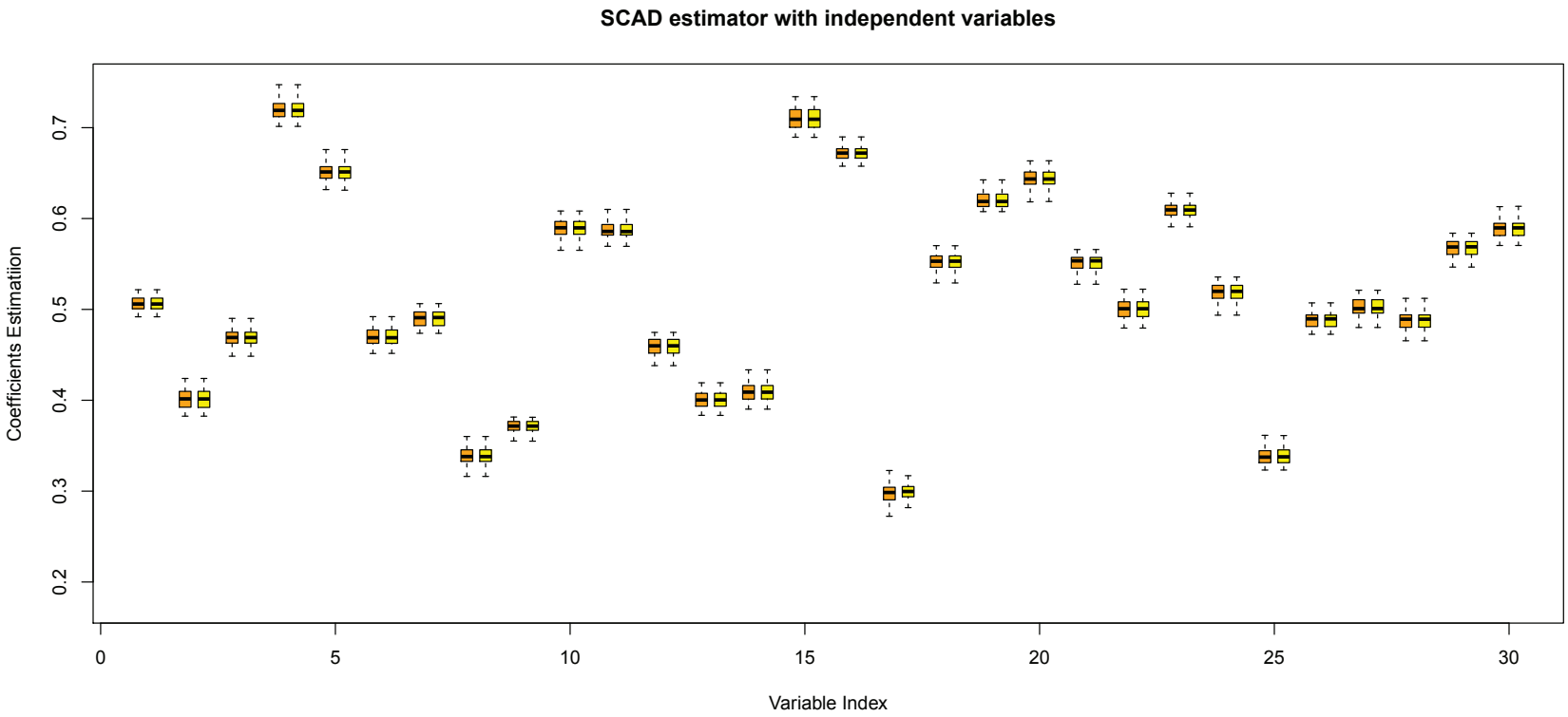
Simulation II: Asymptotic equivalence

Table : Logistic regression with SCAD penalty

Simulation setting					Model selection			
Design matrix	n	p	K	Computing time (in second)	# selected variables	sensitivity (in %)	specificity (in %)	Misclassificaton rate (in %)
Independent	10000	200	1	198.85 (5.88)	35.54 (5.71)	100 (0)	96.74 (3.36)	17.32 (0.40)
			5	116.49 (2.78)	31.70 (1.33)	100 (0)	99.00 (0.78)	17.40 (0.38)
Correlated	10000	200	1	463.61 (20.16)	38.18 (5.58)	99.33 (1.35)	95.02 (3.15)	9.90 (0.29)
			5	359.29 (7.94)	32.38 (2.42)	96.07 (2.75)	97.84 (1.27)	10.10 (0.26)
Independent	100000	200	1	- (-)	- (-)	- (-)	- (-)	- (-)
			20	1352.14 (76.2)	30 (0)	100 (0)	100 (0)	17.38 (0.12)
Correlated	100000	200	1	- (-)	- (-)	- (-)	- (-)	- (-)
			20	4014.48 (284.69)	29.97 (0.2)	99.87 (0.67)	100 (0)	9.96 (0.09)

Note: Logistic regression with MCP performs similarly.

Estimation



Side-by-side (pair-wise) boxplots: split-conquer-combine estimates (orange) vs. conventional estimates (yellow) of $s = 30$ nonzero coefficients

A DHS/POE project: Manifest data analysis

- One week's (02/20/2009 - 02/26/2009) manifest data (a variety of cargo information from custom forms) of all shipments to the U.S. from Customs and Border Protection (CBP) & DIMACS, Rutgers University
  - 164721 shipments in all
  - 7 categorical variables with more than 200 dummy variables
  - Text fields can add even more variables
- Tasks:
  - Assign a risk score and flag for possible inspections of each shipment based on the information in its manifest forms
  - Identify variables that are relevant and ignore those that contain no information

Manifest data analysis

Table : Comparison of the combined weekly estimator and daily estimators (standard deviation in the parenthesis)

	Model selection			
	# of selected variables	Sensitivity (in %)	Specificity (in %)	Misclassificaton rate (in %)
Week (Combined)	21.06 (0.38)	95.25 (0.09)	99.95 (0.14)	3.97 (0.05)
Mon	32.66 (4.00)	92.53 (0.36)	94.2 (1.78)	3.99 (0.05)
Tues	29.18 (3.07)	95.4 (0.05)	96.14 (1.44)	3.98 (0.05)
Wed	9.22 (4.58)	23.13 (1.2)	98.05 (1.18)	3.99 (0.05)
Thur	10.86 (4.6)	27.73 (1.08)	97.76 (1.28)	3.98 (0.05)
Fri	25.6 (2.09)	95.45 (0)	97.83 (0.98)	4.00 (0.05)
Sat	29.76 (3.47)	95 (0.14)	95.82 (1.61)	3.98 (0.05)
Sun	30.6 (3.31)	95.1 (0.12)	95.44 (1.57)	3.99 (0.05)

Remark: Due to security concerns, the observed risk indicators (responses) are not available and need to be simulated.

Estimation

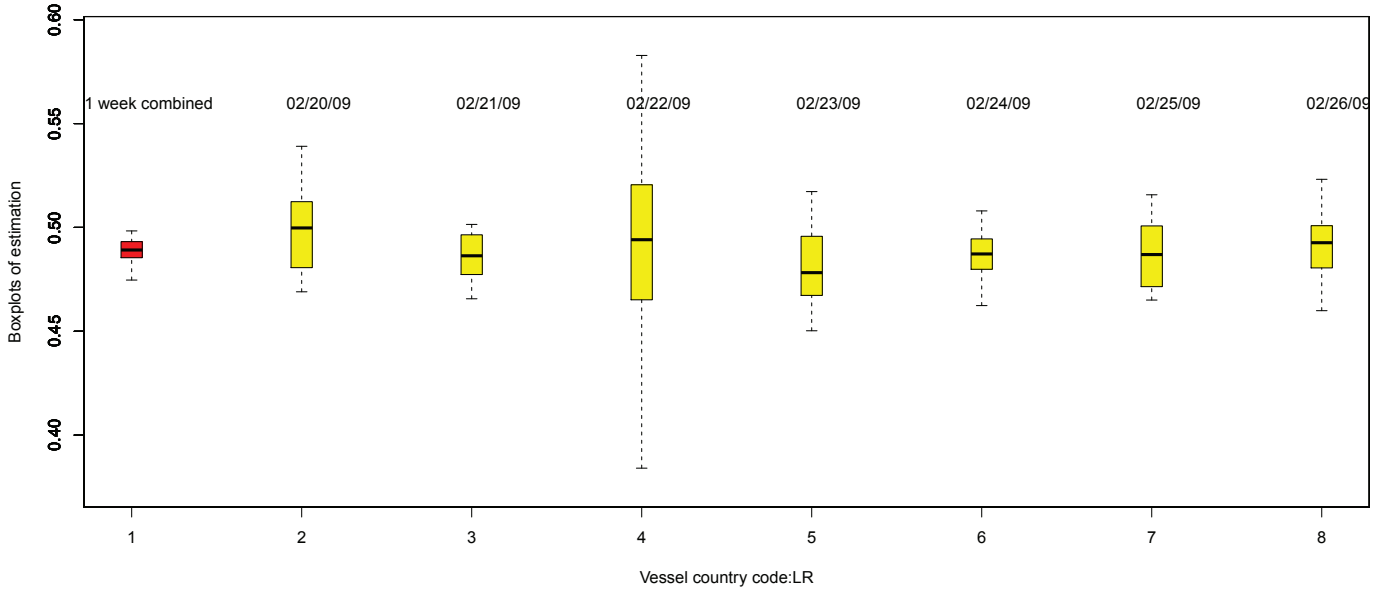


Figure : Weekly data vs. Daily data

- Similar plots for other coefficient estimators

Summary/Discussion

- We have proposed a split–conquer–combine approach for analysis of extremely large data
  - Do not cause any bias or efficiency loss when we perform penalized regressions
  - Can substantially save computing time when a computationally intensive algorithm is used
  - Improve LASSO regression performance

Summary/Discussion

- We have proposed a split–conquer–combine approach for analysis of extremely large data
  - Do not cause any bias or efficiency loss when we perform penalized regressions
  - Can substantially save computing time when a computationally intensive algorithm is used
  - Improve LASSO regression performance
- The split-and-conquer approach is **very general** and can have a lot of applications.

– Our understanding:

  - It works for any linear and MLE-type estimators
  - Combination is a key step – especially if we step outside linear estimators/MLEs

Further remarks on combination methodology

Combination methodology

I. Combination of **point estimators**/single values
   - Different versions of weighted sum of point estimators
   - Variety forms of combining *p*-values (e.g. Fisher 1932, Stouffer et al 1949, etc)

II. Combination of **interval estimators**
   - Tian's approach of combining intervals (risk difference) (Tian et al. 2009)

III. Combination of "**distribution estimators**"/**functions**
   - Multiplication of likelihood functions (frequentist)
   - Bayes formula (Bayesian)
   - Combination of "distribution estimators" (*confidence distributions*)

Parameter estimation/confidence distribution

Statistical inference:

- Point estimate
- Interval estimate
- Distribution estimate (e.g., confidence distribution)

Parameter estimation/confidence distribution

Statistical inference:

- Point estimate
- Interval estimate
- Distribution estimate (e.g., confidence distribution)

Example: $X_1, \ldots, X_n$ i.i.d. follows $N(\mu, 1)$

- Point estimate: $\bar{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i$
- Interval estimate: $(\bar{x}_n - 1.96/\sqrt{n}, \bar{x}_n + 1.96/\sqrt{n})$
- Distribution estimate: $N(\bar{x}_n, \frac{1}{n})$

The idea of the CD approach is to use **a sample-dependent distribution (or density) function** to estimate the parameter of interest.

Parameter estimation/confidence distribution

Statistical inference:

- Point estimate
- Interval estimate
- Distribution estimate (e.g., confidence distribution)

Example: $X_1, \ldots, X_n$ i.i.d. follows $N(\mu, 1)$

- Point estimate: $\bar{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i$
- Interval estimate: $(\bar{x}_n - 1.96/\sqrt{n}, \bar{x}_n + 1.96/\sqrt{n})$
- Distribution estimate: $N(\bar{x}_n, \frac{1}{n})$

The idea of the CD approach is to use **a sample-dependent distribution (or density) function** to estimate the parameter of interest.

- Wide range of examples: bootstrap distribution, (normalized) likelihood function, p-value functions, fiducial distributions, some informative priors and Bayesian posteriors, among others

Parameter estimation/confidence distribution (continue ...)

A confidence distribution (CD) can be loosely referred to as a **sample-dependent distribution function** that can represent confidence intervals of **all levels** for a parameter of interest.

- Cox (2013): The CD approach is “to provide simple and interpretable summaries of what can reasonably be learned from data (and an assumed model).”
- Efron (2013): The CD development is “a grounding process” to solve “perhaps the most important unresolved problem in statistical inference” on “the use of Bayes theorem in the absence of prior information.”

Parameter estimation/confidence distribution (continue ...)

A confidence distribution (CD) can be loosely referred to as a **sample-dependent distribution function** that can represent confidence intervals of **all levels** for a parameter of interest.

- Cox (2013): The CD approach is "to provide simple and interpretable summaries of what can reasonably be learned from data (and an assumed model)."
- Efron (2013): The CD development is "a grounding process" to solve "perhaps the most important unresolved problem in statistical inference" on "the use of Bayes theorem in the absence of prior information."

Our understanding: *Any approach, regardless of being frequentist, fiducial or Bayesian, can potentially be unified under the concept of confidence distributions, as long as it can be used to build confidence intervals of all levels, exactly or asymptotically.*

Unifying framework for combining information

Meta-analysis/information combining approaches unified under CD framework.

Classical approaches of combining p-values (from Marden, 1991)	Fisher method Stouffer (normal) method Tippett (min) method Max method Sum method
Model-based meta-analysis approaches (from Normand, 1999, Table IV)	Fixed-effects model: MLE method Fixed-effects model: Bayesian method Random-effects model: Method of moment Random-effects model: REML method Random-effects model: Bayesian method (normal prior on θ and fixed τ)
	(Xie et al., 2011)
Others (2×2 tables)	Tian et al. (2009, *Biostat.*) approach of combining intervals (risk difference) Mantel-Haenszel (MH) method (odds ratio) Peto method (odds-ratio)
	(Yang et al., 2012; Yang, 2013)
Combining functions	Multiplication of likelihood functions Bayesian formula
	(Singh, Strawderman & Xie, 2005; Xie & Singh, 2013)

Combining information and data mining

A nice feature of the CD combination framework

- Provides a theoretical framework to combine estimators from different statistical paradigms — frequentist, Bayesian, fiducial...

Combining information and data mining

A nice feature of the CD combination framework

- Provides a theoretical framework to combine estimators from different statistical paradigms — frequentist, Bayesian, fiducial...

Finally —

Combining information and data mining

A nice feature of the CD combination framework

- Provides a theoretical framework to combine estimators from different statistical paradigms — frequentist, Bayesian, fiducial...

Finally —

Data mining + combining methodology

- Marriage of two fields?

Thank You!