Efficient Leverage Score Sampling for the Analysis of Big Time Series Data

Ali Eshragh

School of Mathematical and Physical Sciences
The University of Newcastle

(Joint work with Fred Roosta, Asef Nazari, and Michael Mahoney)

1. Time Series Forecasting
   - Introduction
   - Autoregressive Model

2. Randomized Numerical Linear Algebra
   - Introduction
   - Background

3. Big Time Series Data and RandNLA
   - Theoretical Results
   - Empirical Results
   - Future Work

Time Series

Definition (Time Series)

A **time series** is a collection of random variables indexed according to the order they are obtained in **time**.

Objective

The **primary objective** of time series analysis is to develop **statistical models** to forecast the **future** behavior of the system.

Time Series

Definition (Time Series)

A **time series** is a collection of random variables indexed according to the order they are obtained in **time**.

Objective

The **primary objective** of time series analysis is to develop **statistical models** to forecast the **future** behavior of the system.

Box-Jenkins Model

- In 1976, Box and Jenkins introduced their celebrated **Autoregressive Moving Average** (**ARMA**) model for analyzing stationary time series.
- A special case of an ARMA model is **Autoregressive** (**AR**), which merely includes the autoregressive component.
- Despite their **simplicity**, AR models have a **wide** range of **applications** spanning from genetics and medical sciences to finance and engineering.

Box-Jenkins Model

- In 1976, Box and Jenkins introduced their celebrated **Autoregressive Moving Average** (**ARMA**) model for analyzing stationary time series.
- A special case of an ARMA model is **Autoregressive** (**AR**), which merely includes the autoregressive component.
- Despite their **simplicity**, AR models have a **wide** range of **applications** spanning from genetics and medical sciences to finance and engineering.

Box-Jenkins Model

- In 1976, Box and Jenkins introduced their celebrated **Autoregressive Moving Average** (**ARMA**) model for analyzing stationary time series.
- A special case of an ARMA model is **Autoregressive** (**AR**), which merely includes the autoregressive component.
- Despite their **simplicity**, AR models have a **wide** range of **applications** spanning from genetics and medical sciences to finance and engineering.

Autoregressive Model

- An **AR** model with the **order** p, denoted by $\texttt{AR}(p)$, is

$$Y_t = \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + W_t \,,$$

  where W_t is a **Gaussian white noise** with the mean function $\mathbb{E}[W_t] = 0$ and variance $Var(W_t) = \sigma_W^2$.

- **Partial Autocorrelation Function** (**PACF**) for an $\texttt{AR}(10)$ model:

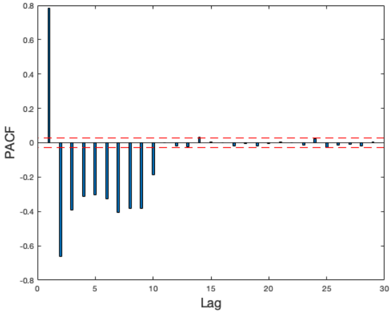
Autoregressive Model

- An **AR** model with the **order** p, denoted by $\mathtt{AR}(p)$, is

$$Y_t = \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + W_t\,,$$

where W_t is a **Gaussian white noise** with the mean function $\mathbb{E}[W_t] = 0$ and variance $Var(W_t) = \sigma_W^2$.

- **Partial Autocorrelation Function** (**PACF**) for an $\mathtt{AR}(10)$ model:



Fitting an AR Model in Big Data Regime

- In problems involving **big time series data**, fitting an **appropriate** AR model amounts to the solutions of **many** potentially large scale **Ordinary Least Squares** (**OLS**) problems.

Question

Can a **randomized sub-sampling** algorithm be designed to greatly **speed-up** such model fitting for **big** time series data?

Fitting an AR Model in Big Data Regime

- In problems involving **big time series data**, fitting an **appropriate** AR model amounts to the solutions of **many** potentially large scale **Ordinary Least Squares** (**OLS**) problems.

Question

Can a **randomized sub-sampling** algorithm be designed to greatly **speed-up** such model fitting for **big** time series data?

1. Time Series Forecasting
   - Introduction
   - Autoregressive Model
2. Randomized Numerical Linear Algebra
   - Introduction
   - Background
3. Big Time Series Data and RandNLA
   - Theoretical Results
   - Empirical Results
   - Future Work

Large OLS Problems

- In several **statistical models**, solving an over-determined **OLS** problem

$$\min_{\boldsymbol{\phi}} \|\boldsymbol{X}\boldsymbol{\phi} - \boldsymbol{y}\|^2 ,$$

involving an $n \times p$ **data matrix** $\boldsymbol{X}$ and an $n \times 1$ **observation vector** $\boldsymbol{y}$ is of interest.

- In **big data** regimes where $n \gg p$, naïvely solving an OLS problem which takes $\mathcal{O}(np^2)$ can be **costly**.

- **Randomized Numerical Linear Algebra** (**RandNLA**) has successfully employed various **random sub-sampling** strategies to **compress** the underlying data matrix into a smaller one, while approximately **retaining** many of its original properties.

Large OLS Problems

- In several **statistical models**, solving an over-determined **OLS** problem

$$\min_{\boldsymbol{\phi}} \left|\left|\boldsymbol{X}\boldsymbol{\phi} - \boldsymbol{y}\right|\right|^2,$$

involving an $n \times p$ **data matrix** $\boldsymbol{X}$ and an $n \times 1$ **observation vector** $\boldsymbol{y}$ is of interest.

- In **big data** regimes where $n \gg p$, naïvely solving an OLS problem which takes $\mathcal{O}(np^2)$ can be **costly**.

- **Randomized Numerical Linear Algebra** (**RandNLA**) has successfully employed various **random sub-sampling** strategies to **compress** the underlying data matrix into a smaller one, while approximately **retaining** many of its original properties.

Large OLS Problems

- In several **statistical models**, solving an over-determined **OLS** problem

$$\min_{\boldsymbol{\phi}} \|\boldsymbol{X}\boldsymbol{\phi} - \boldsymbol{y}\|^2 ,$$

involving an $n \times p$ **data matrix** $\boldsymbol{X}$ and an $n \times 1$ **observation vector** $\boldsymbol{y}$ is of interest.

- In **big data** regimes where $n \gg p$, naïvely solving an OLS problem which takes $\mathcal{O}(np^2)$ can be **costly**.

- **Randomized Numerical Linear Algebra** (**RandNLA**) has successfully employed various **random sub-sampling** strategies to **compress** the underlying data matrix into a smaller one, while approximately **retaining** many of its original properties.

RandNLA

- RandNLA subroutines involve **construction** of appropriate **sub-sampling matrix**, $\boldsymbol{S} \in \mathbb{R}^{s \times n}$ for $p \leq s \ll n$, and compressing the data matrix into a **smaller** version $\boldsymbol{SX} \in \mathbb{R}^{s \times p}$.
- In the classical **OLS** problem, RandNLA can readily be **applied** to the smaller scale problem

$$\min_{\phi_s} ||\boldsymbol{SX}\phi_s - \boldsymbol{Sy}||^2 ,$$

at much **lower** costs.

Question

If $\phi^{\star}$ and $\phi_s^{\star}$ are the **solutions** of the **original** OLS problem and the **smaller** scale problem, respectively, how they would **relate** to each other?

RandNLA

- RandNLA subroutines involve **construction** of appropriate **sub-sampling matrix**, $\boldsymbol{S} \in \mathbb{R}^{s \times n}$ for $p \leq s \ll n$, and compressing the data matrix into a **smaller** version $\boldsymbol{S}\boldsymbol{X} \in \mathbb{R}^{s \times p}$.

- In the classical **OLS** problem, RandNLA can readily be **applied** to the smaller scale problem

$$\min_{\boldsymbol{\phi}_s} ||\boldsymbol{S}\boldsymbol{X}\boldsymbol{\phi}_s - \boldsymbol{S}\boldsymbol{y}||^2 ,$$

at much **lower** costs.

Question

If $\phi^{\star}$ and $\phi_s^{\star}$ are the **solutions** of the **original** OLS problem and the **smaller** scale problem, respectively, how they would **relate** to each other?

RandNLA

- RandNLA subroutines involve **construction** of appropriate **sub-sampling matrix**, $\boldsymbol{S} \in \mathbb{R}^{s \times n}$ for $p \leq s \ll n$, and compressing the data matrix into a **smaller** version $\boldsymbol{SX} \in \mathbb{R}^{s \times p}$.

- In the classical **OLS** problem, RandNLA can readily be **applied** to the smaller scale problem

$$\min_{\boldsymbol{\phi}_s} ||\boldsymbol{SX}\boldsymbol{\phi}_s - \boldsymbol{Sy}||^2 ,$$

at much **lower** costs.

Question

If $\phi^\star$ and $\phi_s^\star$ are the **solutions** of the **original** OLS problem and the **smaller** scale problem, respectively, how they would **relate** to each other?

Error Bounds

Theorem (Drineas, Mahoney, Muthukrishnan and Sarlós)

*If s is **large** enough, for an **appropriate** sub-sampling matrix $\boldsymbol{S}$, with **high probability**, we have*

$$||\boldsymbol{X}\boldsymbol{\phi}^{\star} - \boldsymbol{y}||^2 \leq ||\boldsymbol{X}\boldsymbol{\phi}_s^{\star} - \boldsymbol{y}||^2 \leq (1 + \mathcal{O}(\epsilon))||\boldsymbol{X}\boldsymbol{\phi}^{\star} - \boldsymbol{y}||^2 .$$

Question

How an **appropriate** sub-sampling matrix $\boldsymbol{S}$ could be **constructed**?

Error Bounds

Theorem (Drineas, Mahoney, Muthukrishnan and Sarlós)

*If s is **large** enough, for an **appropriate** sub-sampling matrix $\boldsymbol{S}$, with **high probability**, we have*

$$||\boldsymbol{X}\boldsymbol{\phi}^{\star} - \boldsymbol{y}||^2 \leq ||\boldsymbol{X}\boldsymbol{\phi}_s^{\star} - \boldsymbol{y}||^2 \leq (1 + \mathcal{O}(\epsilon))||\boldsymbol{X}\boldsymbol{\phi}^{\star} - \boldsymbol{y}||^2 .$$

Question

How an **appropriate** sub-sampling matrix $\boldsymbol{S}$ could be **constructed**?

Leverage Score Sampling

Sampling Scheme

Among many different strategies, those schemes based on **statistical leverage scores** have not only shown to improve worst-case **theoretical guarantees** of matrix algorithms, but also they are amenable to high-quality **numerical implementations**.

Definition

Give the $n \times p$ data matrix $\boldsymbol{X}$, the **leverage scores** are denoted by $\ell_{n,p}(i)$ for $i = 1, \ldots, n$ and defined as the i^{th} **diagonal** element of the **hat** matrix $\boldsymbol{H}$ given by $\boldsymbol{H} := \boldsymbol{X}(\boldsymbol{X}^\intercal \boldsymbol{X})^{-1} \boldsymbol{X}^\intercal$.

- It can be **shown** that $\ell_{n,p}(i) \geq 0$ for $i = 1, \ldots, n$ and $\sum_{i=1}^{n} \ell_{n,p}(i) = p$, implying that $\{\pi_{n,p}(i) := \ell_{n,p}(i)/p\}_{i=1}^{n}$ defines a **sampling distribution** over the rows of $\boldsymbol{X}$.

Leverage Score Sampling

Sampling Scheme

Among many different strategies, those schemes based on **statistical leverage scores** have not only shown to improve worst-case **theoretical guarantees** of matrix algorithms, but also they are amenable to high-quality **numerical implementations**.

Definition

Give the $n \times p$ data matrix $\boldsymbol{X}$, the **leverage scores** are denoted by $\ell_{n,p}(i)$ for $i = 1, \ldots, n$ and defined as the i^{th} **diagonal** element of the **hat** matrix $\boldsymbol{H}$ given by $\boldsymbol{H} := \boldsymbol{X}(\boldsymbol{X}^\intercal \boldsymbol{X})^{-1}\boldsymbol{X}^\intercal$.

- It can be **shown** that $\ell_{n,p}(i) \geq 0$ for $i = 1, \ldots, n$ and $\sum_{i=1}^{n} \ell_{n,p}(i) = p$, implying that $\{\pi_{n,p}(i) := \ell_{n,p}(i)/p\}_{i=1}^{n}$ defines a **sampling distribution** over the rows of $\boldsymbol{X}$.

Leverage Score Sampling

Sampling Scheme

Among many different strategies, those schemes based on **statistical leverage scores** have not only shown to improve worst-case **theoretical guarantees** of matrix algorithms, but also they are amenable to high-quality **numerical implementations**.

Definition

Give the $n \times p$ data matrix $\boldsymbol{X}$, the **leverage scores** are denoted by $\ell_{n,p}(i)$ for $i = 1, \ldots, n$ and defined as the i^{th} **diagonal** element of the **hat** matrix $\boldsymbol{H}$ given by $\boldsymbol{H} := \boldsymbol{X}(\boldsymbol{X}^\intercal \boldsymbol{X})^{-1}\boldsymbol{X}^\intercal$.

- It can be **shown** that $\ell_{n,p}(i) \geq 0$ for $i = 1, \ldots, n$ and $\sum_{i=1}^{n} \ell_{n,p}(i) = p$, implying that $\{\pi_{n,p}(i) := \ell_{n,p}(i)/p\}_{i=1}^{n}$ defines a **sampling distribution** over the rows of $\boldsymbol{X}$.

Computational Complexity

- Clearly, **obtaining** the leverage scores is almost **as costly as** solving the original **OLS** problem, that is $\mathcal{O}(np^2)$.
- However, some **randomized approximation** algorithms have been developed, which **efficiently** estimate the leverage scores in $\mathcal{O}(np \log n + p^3)$.

Question

Due to the **special structure** of the data matrix in **AR** models, can we develop a **more** efficient algorithm to **approximate** the leverage scores?

Computational Complexity

- Clearly, **obtaining** the leverage scores is almost **as costly as** solving the original **OLS** problem, that is $\mathcal{O}(np^2)$.
- However, some **randomized approximation** algorithms have been developed, which **efficiently** estimate the leverage scores in $\mathcal{O}(np \log n + p^3)$.

Question

Due to the **special structure** of the data matrix in **AR** models, can we develop a **more** efficient algorithm to **approximate** the leverage scores?

Computational Complexity

- Clearly, **obtaining** the leverage scores is almost **as costly as** solving the original **OLS** problem, that is $\mathcal{O}(np^2)$.
- However, some **randomized approximation** algorithms have been developed, which **efficiently** estimate the leverage scores in $\mathcal{O}(np\log n + p^3)$.

Question

Due to the **special structure** of the data matrix in **AR** models, can we develop a **more** efficient algorithm to **approximate** the leverage scores?

1. Time Series Forecasting
   - Introduction
   - Autoregressive Model

2. Randomized Numerical Linear Algebra
   - Introduction
   - Background

3. Big Time Series Data and RandNLA
   - Theoretical Results
   - Empirical Results
   - Future Work

Notation

- Let $y_1, \ldots, y_n$ be a **time series** realization of the $\mathtt{AR}(p)$ model

$$Y_t = \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + W_t \,.$$

- The **data matrix** is given by

$$\boldsymbol{X}_{n,p} := \begin{pmatrix} y_1 & y_2 & \cdots & y_p \\ y_2 & y_3 & \cdots & y_{p+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n-p} & y_{n-p+1} & \cdots & y_{n-1} \end{pmatrix},$$

and the **observation vector** is

$$\boldsymbol{y}_{n,p} = \begin{bmatrix} y_{p+1} & y_{p+2} & \cdots & y_n \end{bmatrix}^{\intercal}.$$

Notation

- Let $y_1, \ldots, y_n$ be a **time series** realization of the $\mathtt{AR}(p)$ model

$$Y_t = \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + W_t \,.$$

- The **data matrix** is given by

$$\boldsymbol{X}_{n,p} := \begin{pmatrix} y_1 & y_2 & \cdots & y_p \\ y_2 & y_3 & \cdots & y_{p+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n-p} & y_{n-p+1} & \cdots & y_{n-1} \end{pmatrix},$$

and the **observation vector** is

$$\boldsymbol{y}_{n,p} = \begin{bmatrix} y_{p+1} & y_{p+2} & \cdots & y_n \end{bmatrix}^{\intercal}.$$

Estimate

- The **least square estimate** of the parameters is given by

$$\boldsymbol{\phi}_{n,p} := (\boldsymbol{X}_{n,p}^{\intercal}\boldsymbol{X}_{n,p})^{-1}\boldsymbol{X}_{n,p}^{\intercal}\boldsymbol{y}_{n,p}\,.$$

- **Sum square of residuals** is:

$$\|\boldsymbol{r}_{n,p}\|^2 := \|\boldsymbol{y}_{n,p} - \boldsymbol{X}_{n,p}\boldsymbol{\phi}_{n,p}\|^2 = \sum_{i=1}^{n-p} \boldsymbol{r}_{n,p}^2(i)\,,$$

where

$$\boldsymbol{r}_{n,p}(i) := y_{p+i} - \langle \boldsymbol{X}_{n,p}(i,:), \boldsymbol{\phi}_{n,p}\rangle \quad \text{for } i = 1, \ldots, n-p\,.$$

Estimate

- The **least square estimate** of the parameters is given by

$$\boldsymbol{\phi}_{n,p} := (\boldsymbol{X}_{n,p}^{\intercal}\boldsymbol{X}_{n,p})^{-1}\boldsymbol{X}_{n,p}^{\intercal}\boldsymbol{y}_{n,p} \,.$$

- **Sum square of residuals** is:

$$||\boldsymbol{r}_{n,p}||^2 := ||\boldsymbol{y}_{n,p} - \boldsymbol{X}_{n,p}\boldsymbol{\phi}_{n,p}||^2 = \sum_{i=1}^{n-p} \boldsymbol{r}_{n,p}^2(i) \,,$$

where

$$\boldsymbol{r}_{n,p}(i) := y_{p+i} - \langle \boldsymbol{X}_{n,p}(i,:), \boldsymbol{\phi}_{n,p} \rangle \quad \text{for } i = 1, \ldots, n-p \,.$$

Calculating Exact Leverage Scores

Theorem (E., Roosta, Nazari and Mahoney)

Let $y_1, \ldots, y_n$ be a time series data. The ***leverage scores*** *of an* $\texttt{AR}(1)$ *model is given by*

$$\ell_{n,1}(i) = \frac{y_i^2}{\sum_{t=1}^{n-1} y_t^2} \quad \text{for } i = 1, \ldots, n-1 \, .$$

For an $\texttt{AR}(\texttt{p})$ *model with* $p \geq 2$*, the* ***leverage scores*** *are obtained by the following* ***recursion****:*

$$\ell_{n,p}(i) = \ell_{n-1,p-1}(i) + \frac{\boldsymbol{r}_{n-1,p-1}^2(i)}{||\boldsymbol{r}_{n-1,p-1}||^2} \, .$$

Finding Approximate Leverage Scores

Definition (E., Roosta, Nazari and Mahoney)

Motivated from the exact recursive equation for the leverage scores, we define an **approximate** leverage score through the following **recursion**:

$$\hat{\ell}_{n,p}(i) = \hat{\ell}_{n-1,p-1}(i) + \frac{\hat{\boldsymbol{r}}^2_{n-1,p-1}(i)}{||\hat{\boldsymbol{r}}_{n-1,p-1}||^2} \quad \text{for } p \geq 2\,,\ i = 1, \ldots, n-p\,,$$

where $\hat{\boldsymbol{r}}_{n,p}$ is the **residual** vector, when the parameters are **estimated** by a compressed data matrix **sub-sampled** based on the **leverage scores** sampling distribution.

Theoretical Error Bound

Theorem (E., Roosta, Nazari and Mahoney)

*If the **sub-sample** size s is **large** enough, with **high probability**, we have,*

$$\frac{|\ell_{n,p}(i) - \widehat{\ell}_{n,p}(i)|}{\ell_{n,p}(i)} \leq \eta_{n,p}(p-1)\sqrt{\varepsilon} \quad \text{for } i = 1, \ldots, n-p \,,$$

*where $\eta_{n,p}$ is a bounded **constant** calculated based on the data matrix $\boldsymbol{X}_{n,p}$.*

Corrolary

*The **time complexity** of this approximation for estimating the **leverage scores** is $\mathcal{O}(n)$.*

Theoretical Error Bound

Theorem (E., Roosta, Nazari and Mahoney)

If the ***sub-sample*** *size s is* ***large*** *enough, with* ***high probability****, we have,*

$$\frac{|\ell_{n,p}(i) - \widehat{\ell}_{n,p}(i)|}{\ell_{n,p}(i)} \leq \eta_{n,p}(p-1)\sqrt{\varepsilon} \quad \text{for } i = 1, \ldots, n-p\,,$$

where $\eta_{n,p}$ is a bounded ***constant*** *calculated based on the data matrix $\boldsymbol{X}_{n,p}$.*

Corrolary

The ***time complexity*** *of this approximation for estimating the* ***leverage scores*** *is $\mathcal{O}(n)$.*

LSAR: Leverage Score Sampling Algorithm for AR Models

1. **Set** $h = 1$ and $\bar{p}$ **large** enough;
2. Compute the **approximate** leverage scores $\hat{\ell}_{n,h}(i)$;
3. Construct the sampling **distribution** $\hat{\pi}_{n,h}(i) = \frac{\hat{\ell}_{n,h}(i)}{h}$;
4. Form the $s \times n$ **sampling matrix** $\boldsymbol{S}$ by randomly choosing s rows of the corresponding identity matrix according to the probability distribution found in Step 3, **with replacement**;
5. Construct the **sampled** data matrix $\hat{\boldsymbol{X}}_{n,h} = \boldsymbol{S}\boldsymbol{X}_{n,h}$ and response vector $\hat{\boldsymbol{y}}_{n,h} = \boldsymbol{S}\boldsymbol{y}_{n,h}$;
6. **Solve** the associated **reduced** OLS problem to estimate the parameters $\hat{\boldsymbol{\phi}}_{n,h}$, residuals $\hat{\boldsymbol{r}}_{n,h}$ and the estimated **PACF** in lag h;
7. if $h < \bar{p}$, **increment** $h = h + 1$ and go to Step 2, otherwise **Stop**.

LSAR: Leverage Score Sampling Algorithm for AR Models

1. **Set** $h = 1$ and $\bar{p}$ **large** enough;
2. Compute the **approximate** leverage scores $\hat{\ell}_{n,h}(i)$;
3. Construct the sampling **distribution** $\hat{\pi}_{n,h}(i) = \frac{\hat{\ell}_{n,h}(i)}{h}$;
4. Form the $s \times n$ **sampling matrix** $\boldsymbol{S}$ by randomly choosing s rows of the corresponding identity matrix according to the probability distribution found in Step 3, **with replacement**;
5. Construct the **sampled** data matrix $\hat{\boldsymbol{X}}_{n,h} = \boldsymbol{S}\boldsymbol{X}_{n,h}$ and response vector $\hat{\boldsymbol{y}}_{n,h} = \boldsymbol{S}\boldsymbol{y}_{n,h}$;
6. **Solve** the associated **reduced** OLS problem to estimate the parameters $\hat{\boldsymbol{\phi}}_{n,h}$, residuals $\hat{\boldsymbol{r}}_{n,h}$ and the estimated **PACF** in lag h;
7. if $h < \bar{p}$, **increment** $h = h + 1$ and go to Step 2, otherwise **Stop**.

LSAR: Leverage Score Sampling Algorithm for AR Models

1. **Set** $h = 1$ and $\bar{p}$ **large** enough;
2. Compute the **approximate** leverage scores $\hat{\ell}_{n,h}(i)$;
3. Construct the sampling **distribution** $\hat{\pi}_{n,h}(i) = \frac{\hat{\ell}_{n,h}(i)}{h}$;
4. Form the $s \times n$ **sampling matrix** $\boldsymbol{S}$ by randomly choosing s rows of the corresponding identity matrix according to the probability distribution found in Step 3, **with replacement**;
5. Construct the **sampled** data matrix $\hat{\boldsymbol{X}}_{n,h} = \boldsymbol{S}\boldsymbol{X}_{n,h}$ and response vector $\hat{\boldsymbol{y}}_{n,h} = \boldsymbol{S}\boldsymbol{y}_{n,h}$;
6. **Solve** the associated **reduced** OLS problem to estimate the parameters $\hat{\boldsymbol{\phi}}_{n,h}$, residuals $\hat{\boldsymbol{r}}_{n,h}$ and the estimated **PACF** in lag h;
7. if $h < \bar{p}$, **increment** $h = h + 1$ and go to Step 2, otherwise **Stop**.

LSAR: Leverage Score Sampling Algorithm for AR Models

1. **Set** $h = 1$ and $\bar{p}$ **large** enough;
2. Compute the **approximate** leverage scores $\hat{\ell}_{n,h}(i)$;
3. Construct the sampling **distribution** $\hat{\pi}_{n,h}(i) = \frac{\hat{\ell}_{n,h}(i)}{h}$;
4. Form the $s \times n$ **sampling matrix** $\boldsymbol{S}$ by randomly choosing s rows of the corresponding identity matrix according to the probability distribution found in Step 3, **with replacement**;
5. Construct the **sampled** data matrix $\hat{\boldsymbol{X}}_{n,h} = \boldsymbol{S}\boldsymbol{X}_{n,h}$ and response vector $\hat{\boldsymbol{y}}_{n,h} = \boldsymbol{S}\boldsymbol{y}_{n,h}$;
6. **Solve** the associated **reduced** OLS problem to estimate the parameters $\hat{\boldsymbol{\phi}}_{n,h}$, residuals $\hat{\boldsymbol{r}}_{n,h}$ and the estimated **PACF** in lag h;
7. if $h < \bar{p}$, **increment** $h = h + 1$ and go to Step 2, otherwise **Stop**.

LSAR: Leverage Score Sampling Algorithm for AR Models

1. **Set** $h = 1$ and $\bar{p}$ **large** enough;
2. Compute the **approximate** leverage scores $\hat{\ell}_{n,h}(i)$;
3. Construct the sampling **distribution** $\hat{\pi}_{n,h}(i) = \frac{\hat{\ell}_{n,h}(i)}{h}$;
4. Form the $s \times n$ **sampling matrix** $\boldsymbol{S}$ by randomly choosing s rows of the corresponding identity matrix according to the probability distribution found in Step 3, **with replacement**;
5. Construct the **sampled** data matrix $\hat{\boldsymbol{X}}_{n,h} = \boldsymbol{S}\boldsymbol{X}_{n,h}$ and response vector $\hat{\boldsymbol{y}}_{n,h} = \boldsymbol{S}\boldsymbol{y}_{n,h}$;
6. **Solve** the associated **reduced** OLS problem to estimate the parameters $\hat{\boldsymbol{\phi}}_{n,h}$, residuals $\hat{\boldsymbol{r}}_{n,h}$ and the estimated **PACF** in lag h;
7. if $h < \bar{p}$, **increment** $h = h + 1$ and go to Step 2, otherwise **Stop**.

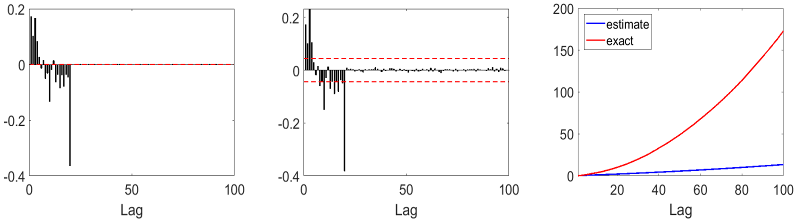
LSAR: Leverage Score Sampling Algorithm for AR Models

1. **Set** $h = 1$ and $\bar{p}$ **large** enough;
2. Compute the **approximate** leverage scores $\hat{\ell}_{n,h}(i)$;
3. Construct the sampling **distribution** $\hat{\pi}_{n,h}(i) = \frac{\hat{\ell}_{n,h}(i)}{h}$;
4. Form the $s \times n$ **sampling matrix** $\boldsymbol{S}$ by randomly choosing s rows of the corresponding identity matrix according to the probability distribution found in Step 3, **with replacement**;
5. Construct the **sampled** data matrix $\hat{\boldsymbol{X}}_{n,h} = \boldsymbol{S}\boldsymbol{X}_{n,h}$ and response vector $\hat{\boldsymbol{y}}_{n,h} = \boldsymbol{S}\boldsymbol{y}_{n,h}$;
6. **Solve** the associated **reduced** OLS problem to estimate the parameters $\hat{\boldsymbol{\phi}}_{n,h}$, residuals $\hat{\boldsymbol{r}}_{n,h}$ and the estimated **PACF** in lag h;
7. if $h < \bar{p}$, **increment** $h = h + 1$ and go to Step 2, otherwise **Stop**.

LSAR: Leverage Score Sampling Algorithm for AR Models

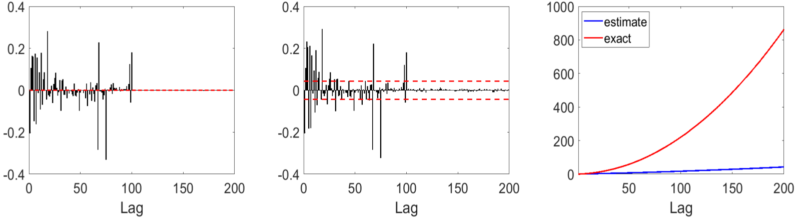
1. **Set** $h = 1$ and $\bar{p}$ **large** enough;
2. Compute the **approximate** leverage scores $\hat{\ell}_{n,h}(i)$;
3. Construct the sampling **distribution** $\hat{\pi}_{n,h}(i) = \frac{\hat{\ell}_{n,h}(i)}{h}$;
4. Form the $s \times n$ **sampling matrix** $\boldsymbol{S}$ by randomly choosing s rows of the corresponding identity matrix according to the probability distribution found in Step 3, **with replacement**;
5. Construct the **sampled** data matrix $\hat{\boldsymbol{X}}_{n,h} = \boldsymbol{S}\boldsymbol{X}_{n,h}$ and response vector $\hat{\boldsymbol{y}}_{n,h} = \boldsymbol{S}\boldsymbol{y}_{n,h}$;
6. **Solve** the associated **reduced** OLS problem to estimate the parameters $\hat{\boldsymbol{\phi}}_{n,h}$, residuals $\hat{\boldsymbol{r}}_{n,h}$ and the estimated **PACF** in lag h;
7. if $h < \bar{p}$, **increment** $h = h + 1$ and go to Step 2, otherwise **Stop**.

Synthetic Big Time Series Data: $\mathtt{AR}(20)$



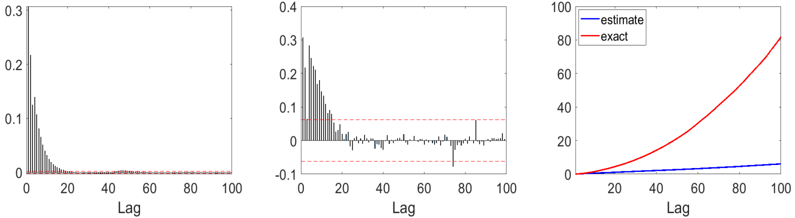
(a) Exact PACF (b) Estimated PACF (c) Time

Synthetic Big Time Series Data: $\mathtt{AR}(100)$



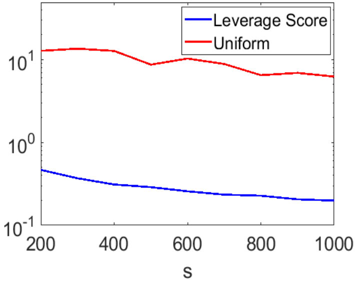
(a) Exact PACF (b) Estimated PACF (c) Time

Real-world Big Time Series Data: Gas Sensors Data



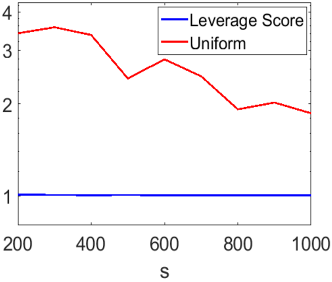
(a) Exact PACF (b) Estimated PACF (c) Time

Real-world Big Time Series Data: Gas Sensors Data



(a) Relative Error of Estimates:

$$\rightarrow \quad \frac{||\hat{\phi}_{n,p} - \phi_{n,p}||}{||\phi_{n,p}||}$$



(b) Ratio of Residual l_2-Norms:

$$\rightarrow \quad \frac{||\hat{\boldsymbol{r}}_{n,p}||}{||\boldsymbol{r}_{n,p}||}$$

Further Development

- **Studying** these theoretical results **extensively** on a wide range of **empirical** big time series data.
- **Developing** similar theoretical results for a more general **ARMA** model.
- **Developing** similar theoretical results for a **Multivariate** AR model.

Further Development

- **Studying** these theoretical results **extensively** on a wide range of **empirical** big time series data.
- **Developing** similar theoretical results for a more general **ARMA** model.
- **Developing** similar theoretical results for a **Multivariate** AR model.

Further Development

- **Studying** these theoretical results **extensively** on a wide range of **empirical** big time series data.
- **Developing** similar theoretical results for a more general **ARMA** model.
- **Developing** similar theoretical results for a **Multivariate** AR model.

Data Science Down Under Workshop

Data Science

8-12 December 2019, Newcastle, Australia

Down Under

A workshop in two parts:

- ***Boot Camp***
  8 December — 10 December
- ***Recent Advances***
  11 December — 12 December

This workshop will bring together Australian researchers and practitioners with key international academics in areas related to data science — including mathematics, statistics and computer science — to discuss recent work and to share ideas, and fostering new local and international collaborations. The inaugural theme of the **Boot Camp** will be 'Randomised Numerical Linear Algebra', while the **Recent Advances** will cover a diverse range of topics from machine learning and data analysis.

Abstract submission closes:
Friday 30th August

Registration closes:
Friday 6th November

Invited Speakers:

Kenneth Clarkson
IBM Research, USA

Michael Mahoney
University of California, Berkeley, USA

Kerrie Mengersen
Queensland University of Technology, Australia

Deanna Needell
UCLA, USA

Joshua Ross
University of Adelaide, Australia

Kate Smith-Miles
University of Melbourne, Australia

Peter Taylor
University of Melbourne, Australia

Matt Wand
University of Technology Sydney, Australia

David Woodruff
Carnegie Mellon University, USA

Peng Xu
Amazon AI Lab, USA

Sponsors:

ACEMS
AMSI Australian Mathematical Sciences Institute
CARMA
NSW Government
Priority Research Centre for Complex Dynamic Systems and Control, The University of Newcastle

Further information: carma.newcastle.edu.au/meetings/dsdu/ or **dsdu@newcastle.edu.au**
Venue: NewSpace, The University of Newcastle, 8-12 December 2019.

Organising committee: Ali Eshragh (Chair; UoN), Fred Roosta (Co-chair; UQ), Ricardo Campello (UoN), Elizabeth Stojanovski (UoN), Natalie Thamwattana (UoN)

End

Thank you $\cdots$ Questions?