Online Inference in High-dimensional Models

Yuanhang Luo[1]
Joint work with Ruijian Han[1], Lan Luo[2], Yuanyuan Lin[3], Jian Huang[1]

[1]Department of Applied Mathematics, The Hong Kong Polytechnic University
[2]Department of Biostatistics and Epidemiology, Rutgers University
[3]Department of Statistics, The Chinese University of Hong Kong

Feb 2024

Table of Contents

Table of Contents

Big Data

Big data is about

> *"datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze" [Manyika et al., 2011]*
>
> *McKinsey Global Institute*

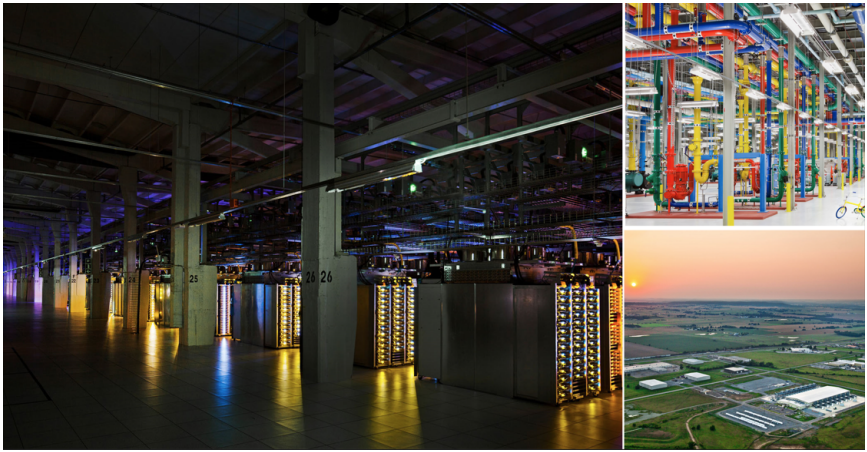

Figure: Google Data Center

Streaming Data

Many applications must process large streams of live data and provide results in near-real-time

- Algorithmic trading
- Real-time operations management
- Online fraud detection
- Proximity/location tracking
- Intrusion detection systems
- Traffic management
- Real time recommendations
- Social media/data analytics
- Precision agriculture
- Internet of things
- Gaming data feed
- ...

Some of the major challenges of handling streaming data.

- Retrieving historical data no longer possible. Needs to be processed one pass.
- Each data has large number of features (High-dimensional)
- Critical memory requirements.

Streaming Data

Proposed solution: Online learning

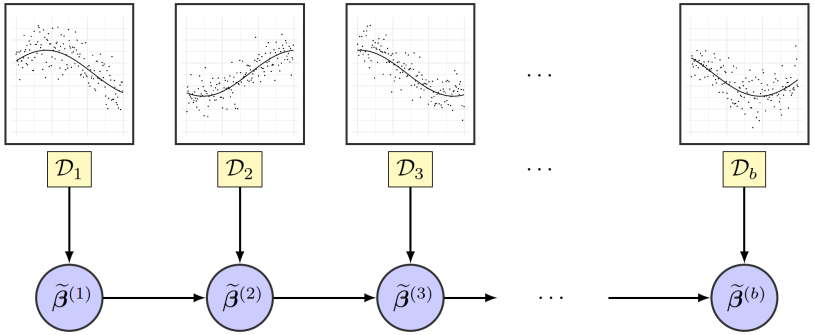

Figure: An online learning scheme

Objective: Develop an "online" estimator that

- can be updated sequentially using raw data in current batch only
- requires little time and space complexities
- still achieves desired statistical property

High-dimensional GLM

A sequence of n data points $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ are independently and identically sampled from a generalized linear model:

$$\mathbb{P}(y \mid \mathbf{x}; \boldsymbol{\beta}^*) \propto \exp\left\{ \frac{y\left(\mathbf{x}^\top \boldsymbol{\beta}^*\right) - \Phi\left(\mathbf{x}^\top \boldsymbol{\beta}^*\right)}{c(\sigma)} \right\}, \tag{1}$$

- $\mathbf{x} \in \mathbb{R}^p$. High-dimensional case: $n \ll p$.
- Sparsity: $\|\boldsymbol{\beta}^*\|_0 = s_0$.

The ground truth $\boldsymbol{\beta}^*$ is the minimum of the population cost function

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \mathbb{E}\left[F_n(\boldsymbol{\beta})\right], \tag{2}$$

where $F_n(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^n \left\{-y_i(\mathbf{x}_i^\top \boldsymbol{\beta}) + \Phi(\mathbf{x}_i^\top \boldsymbol{\beta})\right\}$, is the negative log-likelihood function of model $\mathbb{P}$.

Offline Approach

Usually, the estimate $\boldsymbol{\beta}^*$ using a finite set of i.i.d. observations $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ (offline approach):

$$\boldsymbol{\beta}^{(n)} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left\{ F_n(\boldsymbol{\beta}) + \lambda_n \|\boldsymbol{\beta}\|_1 \right\}. \tag{3}$$

To have a tractable limiting distribution, one-step Newton's correction is needed:

$$\boldsymbol{\beta}_{j,\mathrm{de}}^{(n)} = \boldsymbol{\beta}_j^{(n)} - \boldsymbol{\Theta}_j^\top \nabla F_n(\boldsymbol{\beta}^{(n)}), \tag{4}$$

where $\boldsymbol{\Theta}$ is the inverse of the information matrix. $\boldsymbol{\Theta}_j$ can be estimated by Low Dimensional Projection [Zhang and Zhang, 2014], Node-wise Lasso [van de Geer et al., 2014], Minimum Coherence [Javanmard and Montanari, 2014]...

Question

Can we bring the whole thing "online"?

Table of Contents

$$\begin{aligned}\boldsymbol{\beta}^{(n)} &= \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\left\{F_n(\boldsymbol{\beta}) + \lambda_n\|\boldsymbol{\beta}\|_1\right\},\\ \boldsymbol{\beta}^{(n)}_{j,\mathrm{de}} &= \boldsymbol{\beta}^{(n)}_j - \boldsymbol{\Theta}_j^\top \nabla F_n(\boldsymbol{\beta}^{(n)}),\end{aligned} \tag{5}$$

Question

Can we bring the whole thing "online"?

Sounds appealing, but we have some issues in online settings.

- **Q1**: How to update $\boldsymbol{\beta}_j^{(n)}$ with a single/batch of data?
- **Q2**: How to estimate $\boldsymbol{\Theta}_j$ incrementally?
- **Q3**: Computation of $\nabla F_n(\boldsymbol{\beta}^{(n)})$ is infeasible, we cannot retrieve the historical data $\{x_i\}_{i<n}$ after obtaining $\boldsymbol{\beta}^{(n)}$.

Q1

- **Q1**: How to update $\boldsymbol{\beta}_j^{(n)}$ with a single/batch of data?

We consider a variant of SGD:

Require : α, prox-function Ψ, $\boldsymbol{\beta}^{(0)}$, λ.
Initialize: $\mu^0 = 0$ and $\theta^0 = \boldsymbol{\beta}^{(0)}$.
for *Iteration* $t = 1, \ldots, n$ **do**

$$\mu^t = \mu^{t-1} + \partial \left\{ F(\theta_{t-1}) + \lambda \|\theta_{t-1}\|_1 \right\}.$$

Proximal step with $\alpha^t = \alpha/\sqrt{t}$,

$$\theta^t = \arg\min_{\theta} \left\{ \alpha^t \langle \mu^t, \theta \rangle + \Psi(\theta) \right\}.$$

end
Return $\widehat{\boldsymbol{\beta}}^{(n)} = \sum_{t=1}^{T} \theta^t / n$.

Algorithm 1: Dual Averaging [Nesterov, 2009]

The ℓ_1-convergence rate of the SGD is of order $s_0\{(\log p)/n\}^{-1/4}$:

$$\|\widehat{\boldsymbol{\beta}}^{(n)} - \boldsymbol{\beta}^\star\|_1 \lesssim s_0 \left(\frac{\log p}{n} \right)^{-1/4} \|\widehat{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}^\star\|_1$$

We cannot obtain the desired convergence rate unless starting from a good initial point.

Q1

Let's do a thought experiment: If we split n data into two parts equally:

- In the first half part,

$$\|\widetilde{\boldsymbol{\beta}}^{(n/2)} - \boldsymbol{\beta}^\star\|_1 \lesssim s_0 \left(\frac{\log p}{n/2}\right)^{-1/4} \|\widetilde{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}^\star\|_1$$

- In the second half part, we start from the result in the first step,

$$\begin{aligned}\|\widetilde{\boldsymbol{\beta}}^{(n)} - \boldsymbol{\beta}^\star\|_1 &\lesssim s_0 \left(\frac{\log p}{n/2}\right)^{-1/4} \|\widetilde{\boldsymbol{\beta}}^{(n/2)} - \boldsymbol{\beta}^\star\|_1 \\ &\lesssim \sqrt{2} s_0^2 \left(\frac{\log p}{n}\right)^{-1/2} \|\widetilde{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}^\star\|_1.\end{aligned}$$

Remark

We can apply this epoching trick.

Q1

- We consider a Multi-step SGD variant:

Require : Schedule $\{T_i\}_{i=1}^K$, initial radius R_0, α, prox-function Ψ, $\beta^{(0)}$, λ.
for *Epoch* $i = 1, 2, \ldots, K$ **do**
  Initialize: $\mu^0 = 0$ and $\theta^0 = \beta^{(0)}$.
  for *Iteration* $t = 1, \ldots, T_i$ **do**

$$\mu^t = \mu^{t-1} + \partial \left\{ F(\theta_{t-1}) + \lambda \|\theta_{t-1}\|_1 \right\}.$$

  Proximal step with $\alpha^t = \alpha/\sqrt{t}$,

$$\theta^t = \underset{\theta \in \Omega(R_i)}{\arg\min} \left\{ \alpha^t \left\langle \mu^t, \theta \right\rangle + \Psi(\theta) \right\}.$$

  end
  Dual averaging $\widehat{\beta}^i = \sum_{t=1}^{T_i} \theta^t / T_i$.
  Update $R_i^2 = R_{i-1}^2/2$.
end
Return $\widehat{\beta}^{(K)}$.

Algorithm 2: Regularization Annealed epoch Dual AveRaging [Agarwal et al., 2012]

We use RADAR, which can split the data in epochs $\{T_k\}_{k \in [K]}$, and achieve the 'optimal' rate.

- RADAR = SGD + Multi-step methods + Dual Averaging.

Online Approach

$$\begin{aligned}\boldsymbol{\beta}^{(n)} &= \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\left\{F_n(\boldsymbol{\beta}) + \lambda_n\|\boldsymbol{\beta}\|_1\right\},\\ \boldsymbol{\beta}^{(n)}_{j,\mathrm{de}} &= \boldsymbol{\beta}^{(n)}_j - \boldsymbol{\Theta}_j^\top \nabla F_n(\boldsymbol{\beta}^{(n)}),\end{aligned} \tag{6}$$

Question

Can we bring the whole thing "online"?

- **Q1**: How to update $\boldsymbol{\beta}_j^{(n)}$ with a single/batch of data? **Sol. 1**: RADAR
- **Q2**: How to estimate $\boldsymbol{\Theta}_j$ incrementally?
- **Q3**: Computation of $\nabla F_n(\boldsymbol{\beta}^{(n)})$ is infeasible, we cannot retrieve the historical data $\{x_i\}_{i<n}$ after obtaining $\boldsymbol{\beta}^{(n)}$.

Q2

- **Q2**: How to estimate $\mathbf{\Theta}_j$ incrementally? Online Node-wise Lasso.

  In an offline setting, [van de Geer et al., 2014] proposed the Node-wise lasso to estimate $\mathbf{\Theta}_j$ by solving the following problem

$$\min_{\gamma \in \mathbb{R}^n, \gamma_j = -1} \frac{1}{2n} \sum_{i=1}^{n} \left[\ddot{\Phi} \left(\mathbf{x}_i^\top \boldsymbol{\beta}^{(n)} \right) \left\{ (\mathbf{x}_i)^\top \boldsymbol{\gamma} \right\}^2 \right] + \lambda_n' \|\boldsymbol{\gamma}\|_1, \tag{7}$$

Q2

$$\min_{\gamma\in\mathbb{R}^n,\gamma_j=-1}\frac{1}{2n}\sum_{i=1}^{n}\left[\ddot{\Phi}\left(\mathbf{x}_i^\top\beta^{(n)}\right)\left\{(\mathbf{x}_i)^\top\gamma\right\}^2\right]+\lambda_n'\|\gamma\|_1, \tag{8}$$

where $\beta^{(n)}$ is the minimizer of $F_n(\beta;\lambda_n)$.

Note that (8) is just another composite minimization that can be solved by RADAR with stochastic gradient

$$(x_i)_{-j}\{(x_i)^\top\gamma\}\ddot{\Phi}(x_i^\top\beta^{(n)})+\lambda_k'\,\mathrm{sign}\left(\gamma_{-j}\right),\quad i\in[n].$$

However, $\beta^{(n)}$ is not available when $i<n$. Thus, the RADAR algorithm is not directly applicable, we replace $\beta^{(n)}$ with the most recent estimate of β^* instead:

$$(x_i)_{-j}\{(x_i)^\top\gamma\}\ddot{\Phi}(x_i^\top\widehat{\beta}^{(n_{k-1}')})+\lambda_k'\,\mathrm{sign}\left(\gamma_{-j}\right),\quad i\in[n_k']/[n_{k-1}'].$$

This procedure is more general than the original RADAR, we call it the Adaptive RADAR.

Online Approach

$$\begin{aligned}\boldsymbol{\beta}^{(n)} &= \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\left\{F_n(\boldsymbol{\beta}) + \lambda_n\|\boldsymbol{\beta}\|_1\right\},\\ \boldsymbol{\beta}^{(n)}_{j,\mathrm{de}} &= \boldsymbol{\beta}^{(n)}_j - \boldsymbol{\Theta}_j^\top \nabla F_n(\boldsymbol{\beta}^{(n)}),\end{aligned} \tag{9}$$

Question

Can we bring the whole thing "online"?

- **Q1**: How to update $\boldsymbol{\beta}_j^{(n)}$ with a single/batch of data? **Sol. 1**: RADAR
- **Q2**: How to estimate $\boldsymbol{\Theta}_j$ incrementally? **Sol. 2**: Adaptive RADAR
- **Q3**: Computation of $\nabla F_n(\boldsymbol{\beta}^{(n)})$ is infeasible, we cannot retrieve the historical data $\{x_i\}_{i<n}$ after obtaining $\boldsymbol{\beta}^{(n)}$.

Q3

- **Q3**: $\nabla F_n(\boldsymbol{\beta}^{(n)})$ is incomputable? Approximation via Taylor expansion.

$$\nabla F_n(\boldsymbol{\beta}^{(n)}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\{\dot{\Phi}(\mathbf{x}_i^\top \boldsymbol{\beta}^{(n)}) - y_i\}.$$

  - In linear case: $\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\dot{\Phi}(\mathbf{x}_i^\top \boldsymbol{\beta}^{(n)}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top \boldsymbol{\beta}^{(n)}$, and $\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top$ Can be updated recursively.
  - In logistic regression: $\dot{\Phi}(u) = 1/(1+\exp(-u))$, non-linear. $\dot{\Phi}(\mathbf{x}_i^\top \boldsymbol{\beta}^{(n)})$ depend on $\boldsymbol{\beta}^{(n)}$, but we cannot retrieve the historical data $\{x_i\}_{i<n}$ after obtaining $\boldsymbol{\beta}^{(n)}$.

Second-order approximation:

$$\dot{\Phi}(\boldsymbol{x_i}^\top \widehat{\boldsymbol{\beta}}^{(n)}) = \dot{\Phi}(\boldsymbol{x_i}^\top \widehat{\boldsymbol{\beta}}^{(i)}) + \ddot{\Phi}(\boldsymbol{x_i}^\top \widehat{\boldsymbol{\beta}}^{(i)})\boldsymbol{x_i}^\top(\widehat{\boldsymbol{\beta}}^{(n)} - \widehat{\boldsymbol{\beta}}^{(i)}) + \mathcal{O}(\|\widehat{\boldsymbol{\beta}}^{(n)} - \widehat{\boldsymbol{\beta}}^{(i)}\|_2^2), i \in [n]. \tag{10}$$

We construct the following summary statistics:

$$\boldsymbol{A}_1^{(k)} = \sum_{i=1}^{k} \boldsymbol{x}_i\{-y_i + \dot{\Phi}(\boldsymbol{x}_i^\top \widehat{\boldsymbol{\beta}}^{(i)})\}, \quad \boldsymbol{A}_2^{(k)} = \sum_{i=1}^{k} \boldsymbol{x}_i^\top \widehat{\boldsymbol{\gamma}}_{\cdot,j}^{(i)} \ddot{\Phi}(\boldsymbol{x}_i^\top \widehat{\boldsymbol{\beta}}^{(i)}) \boldsymbol{x}_i,$$

$$a_3^{(k)} = \sum_{i=1}^{k} \boldsymbol{x}_i^\top \widehat{\boldsymbol{\gamma}}_{\cdot,j}^{(i)} \ddot{\Phi}(\boldsymbol{x}_i^\top \widehat{\boldsymbol{\beta}}^{(i)}) \boldsymbol{x}_i^\top \widehat{\boldsymbol{\beta}}^{(i)}, \quad a_4^{(k)} = -\sum_{i=1}^{k} \boldsymbol{x}_i^\top \widehat{\boldsymbol{\gamma}}_{\cdot,j}^{(i)} (\boldsymbol{x}_i)_j \{\ddot{\Phi}(\boldsymbol{x}_i^\top \widehat{\boldsymbol{\beta}}^{(i)})\},$$

All are p-dimensional vectors and scalars. The approximated debiased lasso (ADL) estimator:

$$\widetilde{\boldsymbol{\beta}}_{j,\mathrm{de}}^{(k)} := \widehat{\boldsymbol{\beta}}_j^{(k)} - \frac{\{\boldsymbol{A}_1^{(k)}\}^\top \widehat{\boldsymbol{\gamma}}_{\cdot,j}^{(k)} + \{\boldsymbol{A}_2^{(k)}\}^\top \widehat{\boldsymbol{\beta}}^{(k)} - a_3^{(k)}}{a_4^{(k)}}.$$

Upon arrival of the k-th data, update

$$\boldsymbol{A}_1^{(k)} \longleftarrow \boldsymbol{A}_1^{(k-1)} + \boldsymbol{x}_k\{-y_k + \dot{\Phi}(\boldsymbol{x}_k^\top \widehat{\boldsymbol{\beta}}^{(k)})\}.$$

Q3

- Further, we consider the variance estimator. Construct

$$a_5^{(k)} = \sum_{i=1}^{k} \left(\boldsymbol{x}_i^\top \widehat{\gamma}_{\cdot,j}^{(i)} \right)^2 \left\{ \dot{\Phi} \left(\boldsymbol{x}_i^\top \widehat{\boldsymbol{\beta}}^{(i)} \right) - y_i \right\}^2,$$

then the standard deviation of the ADL estimate takes the form

$$\widetilde{\nu}_j^{(k)} = \sqrt{a_5^{(k)}} / a_4^{(k)}.$$

Online Approach

Question

Can we bring the whole thing "online"?

Sounds appealing, but we have some issues in online settings.

- **Q1**: How to update $\beta_j^{(n)}$ with a single/batch of data? **Sol. 1**: RADAR.
- **Q2**: How to estimate Θ_j incrementally? **Sol. 2**: Adaptive RADAR.
- **Q3**: Computation of $\nabla F_n(\beta^{(n)})$ is infeasible? **Sol. 3**: Taylor approximation.

Proposed Algorithm

Data: Streaming Data $\{x_k, y_k\}_{k=1}^n$, index j, significance level α, other hyperparameters.

Initialize $\{\widehat{\boldsymbol{\beta}}^{(0)}, \widehat{\boldsymbol{\gamma}}_{\cdot j}^{(0)}, \widetilde{\nu}_j^{(0)}, \boldsymbol{A}_1^{(0)}, \boldsymbol{A}_2^{(0)} a_3^{(0)}, a_4^{(0)}\}$

for $k = 1, \ldots, n$ **do**

Arrival of data $\{x_k, y_k\}$.

Update estimates $\widehat{\boldsymbol{\beta}}_j^{(k)}$ and $\widehat{\boldsymbol{\gamma}}_{\cdot j}^{(k)}$ via RADAR and Adaptive RADAR.

Update summary statistics $\widetilde{\nu}_j^{(k)}, \boldsymbol{A}_1^{(k)}, \boldsymbol{A}_2^{(k)} a_3^{(k)}, a_4^{(k)}$

Debias

$$\widetilde{\boldsymbol{\beta}}_{j,\mathrm{de}}^{(k)} := \widehat{\boldsymbol{\beta}}_j^{(k)} - \frac{\{\boldsymbol{A}_1^{(k)}\}^\top \widehat{\boldsymbol{\gamma}}_{\cdot j}^{(k)} + \{\boldsymbol{A}_2^{(k)}\}^\top \widehat{\boldsymbol{\beta}}^{(k)} - a_3^{(k)}}{a_4^{(k)}}.$$

Output the $(1-\alpha)$-confidence interval

$$\left(\widetilde{\boldsymbol{\beta}}_{j,\mathrm{de}}^{(k)} - z_{\alpha/2}\widetilde{\nu}_j^{(k)},\ \widetilde{\boldsymbol{\beta}}_{j,\mathrm{de}}^{(k)} + z_{\alpha/2}\widetilde{\nu}_j^{(k)}\right).$$

Recurse $\{\widehat{\boldsymbol{\beta}}^{(k)}, \widehat{\boldsymbol{\gamma}}_{\cdot j}^{(k)}, \widetilde{\nu}_j^{(k)}, \boldsymbol{A}_1^{(k)}, \boldsymbol{A}_2^{(k)} a_3^{(k)}, a_4^{(k)}\}$ and clear all other variables in memory.

end

Algorithm 3: Approximated debiased lasso (ADL)

Proposed Algorithm

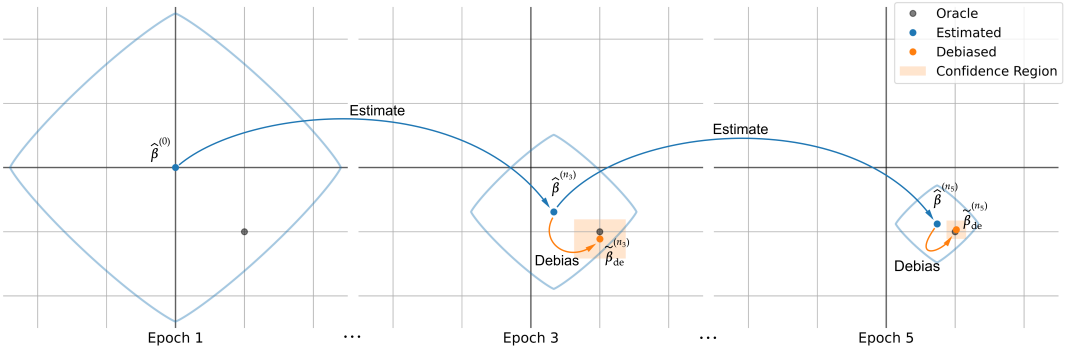

Figure: Visualization of the proposed algorithm.

Complexities

Table: A comparison in time and space complexities for constructing the interval estimator of one single parameter. LM stands for the linear model and GLM stands for the generalized linear model.

Works	Model	One-pass?	Space complexity
[Chen et al., 2020]	LM	Yes	$\mathcal{O}(np)$
[Deshpande et al., 2023]	LM	No	$\mathcal{O}\left(p^2\right)$
[Han et al., 2023]	LM	Yes	$\mathcal{O}(p)$
[Shi et al., 2021]	GLM	No	$\mathcal{O}(np)$
[Luo et al., 2023]	GLM	No	$\mathcal{O}\left(p^2\right)$
Proposed Method	GLM	Yes	$\mathcal{O}(p)$

Table of Contents

Theoretical Results (Sol. 1)

Assumption (1)

X is a zero-mean and sub-Gaussian random variable the sub-Gaussian norm κ_1. The corresponding covariance matrix Σ satisfies: $0 < M_1^{-1} \leq \Lambda_{\min}(\mathbf{\Sigma}) \leq \Lambda_{\max}(\mathbf{\Sigma}) \leq M_1 < \infty$ for a universal constant M_1.

Assumption (2)

Y is sub-Gaussian random variable with the sub-Gaussian norm κ_2. In addition, the link function Φ is Lipschitz and uniformly bounded, *i.e.*, $\max_{t \in \mathbb{R}} |\ddot{\Phi}(\iota)| \leq M_\Phi$.

Assumption (3)

$\|\boldsymbol{\beta}^*\|_0 = s_0$ and $\|\boldsymbol{\beta}^*\|_2 = m$ for a universal constant m.

Theoretical Results (Sol. 1)

Theorem (Convergence rate of RADAR)

Suppose that Assumptions 1-3 hold. With proper choice of hyperparameters, initial error $d_0 = \|\widehat{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}^*\|_1$ *the following events*

$$\|\widehat{\boldsymbol{\beta}}^{(i)} - \boldsymbol{\beta}^*\|_1 \leq C_1 s_0 d_0 \sqrt{\frac{(\log p)^3}{i}}, \quad \|\widehat{\boldsymbol{\beta}}^{(i)} - \boldsymbol{\beta}^*\|_2 \leq C_1 d_0 \sqrt{\frac{s_0 (\log p)^3}{i}} \tag{11}$$

for $i \geq n_1$ *hold uniformly for a universal constant* C_1 *wilh probability al least* $1 - 7(\log p)^{-6}$.

In offline approaches we usually have

$$\|\boldsymbol{\beta}^{(n)} - \boldsymbol{\beta}^\star\|_1 \lesssim s_0 \sqrt{\frac{\log p}{n}}.$$

What's different from ours?

- $d_0 = \|\widehat{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}^*\|_1$
- $(\log p)^{(3/2)}$

Theoretical Results (Sol. 2)

Let $\boldsymbol{\Theta}(\boldsymbol{\beta}) = [\mathbb{E}\{\boldsymbol{X}^\top \ddot{\Phi}\left(\boldsymbol{X}^\top \boldsymbol{\beta}\right) \boldsymbol{X}\}]^{-1}$, and $\overline{\boldsymbol{\Theta}}(\boldsymbol{\beta}) = [\mathrm{diag}\{\boldsymbol{\Theta}(\boldsymbol{\beta})\}]^{-1}\boldsymbol{\Theta}(\boldsymbol{\beta})$.

Assumption (4)

There exists a universal constant M_2 such that $0 < M_2^{-1} \leq \Lambda_{\min}\left(\overline{\Theta}\left(\beta^*\right)\right) \leq \Lambda_{\max}\left(\overline{\boldsymbol{\Theta}}\left(\boldsymbol{\beta}^*\right)\right) \leq M_2 < \infty$. The columns of $\overline{\boldsymbol{\Theta}}\left(\boldsymbol{\beta}^*\right)$ is s_1-sparse: $\max_{j\in[p]} \| \left\{\overline{\boldsymbol{\Theta}}\left(\boldsymbol{\beta}^*\right)\right\}_{\cdot j} \|_0 = s_1$.

Assumption (5)

For any $\boldsymbol{b}, \boldsymbol{\beta} \in \mathbb{R}^p$, $\|\overline{\boldsymbol{\Theta}}(\boldsymbol{\beta}) - \overline{\boldsymbol{\Theta}}(\boldsymbol{b})\|_1 \leq L\|\boldsymbol{\beta} - \boldsymbol{b}\|_1$, where L is a universal constant.

Theoretical Results (Sol. 2)

Theorem (Convergence rate of Adaptive RADAR)

Suppose Assumptions 1-5 holds, With proper choice of hyper-parameters, initial error $d_1 = \|\widehat{\gamma}_{\cdot,j}^{(0)} - \gamma_{\cdot,j}^*\|_1$. *Given any fixed* $j \in [p]$, *the following events*

$$\|\widehat{\gamma}_{\cdot,j}^{(i)} - \gamma_{\cdot,j}^*\|_1 \leq C_2 \left\{ s_1 d_1 + (s_1 + s_0)\, d_0 \right\} \sqrt{\frac{(\log p)^3}{i - n_1}}$$

for $i \geq n_1'$ *hold uniformly for a universal constant* C_2 *with probability at least* $1 - 14(\log p)^{-6}$

Theoretical Results (Sol. 3)

Theorem (Asymptotic Normality of ADL)

Suppose that Assumptions 1-5 hold and

$$\left(s_0 s_1 d_0 d_1 + s_0^2 d_0^2 + s_0 s_1 d_0^2\right) (\log p)^4 \log_2(n) = o_{\mathbb{P}}(\sqrt{n}).$$

With proper choice of hyper-parameters, given any fixed $j \in [p]$,

$$\frac{1}{\widetilde{\nu}_j^{(n)}} \left(\widetilde{\beta}_{j,de}^{(n)} - \beta_j^*\right) \to \mathcal{N}(0,1) \text{ as } n \to \infty \text{ in distribution.}$$

Remark

Existing inference methods in general rely on the bounded individual probability condition that

$$P\left(y_i = 1 \mid X_i\right) \in (c, 1-c)$$

for all $1 \le i \le n$ and some $c \in (0, 1/2)$. [Cai et al., 2023] and our's remove this condition.

Table of Contents

Settings of empirical studies

Simulations:
We generate a sequence of n i.i.d. copies $\{(x_i, y_i)\}_{i=1}^n$ realized from the logistic regression model (1), where $x_i \sim \mathcal{N}_p(0, \mathbf{\Sigma})$.
The random active set $\{D_1, D_2\}$ is divided into two halves, with total cardinality s_0.
True coefficients assigned as $\beta^*_{D_1} = 1, \beta^*_{D_2} = -1$ and others are set to 0.
The averaged performance of subsets of parameters is compared.

- Simulation 1. $n = 200, p = 500$ and $s_0 = 6$.
  - Case 1: $\mathbf{\Sigma} = 0.1 \times \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$
    We $\times 0.1$ to ensure the bounded individual probability condition
  - Case 2: $\mathbf{\Sigma} = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$
- Simulation 2. $n = 1000, p = 20000, s_0 = 20$, and $\mathbf{\Sigma} = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$.

Real Data Application:
Detection of malicious websites [Ma et al., 2010].

Simulation 1

Table: Case 1: $n = 200$, $p = 500$, $s_0 = 6$, $\mathbf{\Sigma} = 0.1 \times \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$. Simulation results are averaged over 200 replications. Other methods: deLasso [van de Geer et al., 2014], LSW [Cai et al., 2023], ODL [Luo et al., 2023].

	β_k^*	deLasso	LSW	ODL			ADL		
Sample size n		200	200	40	120	200	40	120	200
Coverage probability	0	0.953	0.988	0.971	0.963	0.958	0.978	0.966	0.959
	1	0.930	0.925	0.963	0.951	0.946	0.978	0.956	0.952
	-1	0.952	0.945	0.975	0.958	0.953	0.973	0.971	0.951
Absolute bias	0	0.410	0.368	0.066	0.043	0.038	0.738	0.465	0.366
	1	0.436	0.532	0.259	0.219	0.181	0.808	0.478	0.387
	-1	0.411	0.521	0.231	0.204	0.178	0.780	0.452	0.370
Coverage length	0	2.036	2.260	4.061	2.297	1.771	4.242	2.477	1.886
	1	2.036	2.265	4.059	2.293	1.766	4.237	2.473	1.879
	-1	2.045	2.284	4.09	2.304	1.776	4.269	2.486	1.891
Time (s)		7.214	20.22		0.539			0.324	

Simulation 1

Table: Case 2: $n = 200$, $p = 500$, $s_0 = 6$, $\mathbf{\Sigma} = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$. Simulation results are averaged over 200 replications. Other methods: deLasso [van de Geer et al., 2014], LSW [Cai et al., 2023], ODL [Luo et al., 2023].

	β_k^*	deLasso	LSW	ODL			ADL		
Sample size n		200	200	40	120	200	40	120	200
Coverage probability	0	0.983	0.996	1.000	0.999	0.998	0.951	0.951	0.961
	1	0.598	0.936	1.000	0.880	0.705	0.891	0.955	0.943
	-1	0.610	0.946	1.000	0.861	0.670	0.881	0.961	0.945
Absolute bias	0	0.114	0.113	0.022	0.019	0.017	0.901	0.425	0.309
	1	0.325	0.260	0.657	0.590	0.539	1.008	0.423	0.328
	-1	0.326	0.259	0.660	0.598	0.548	1.034	0.426	0.324
Coverage length	0	0.753	1.207	3.423	1.691	1.287	4.419	2.053	1.532
	1	0.769	1.261	3.434	1.689	1.286	4.555	2.086	1.536
	-1	0.767	1.271	3.416	1.648	1.282	4.431	2.078	1.541
Time (s)		6.550	20.04		0.628			0.365	

Simulation 2

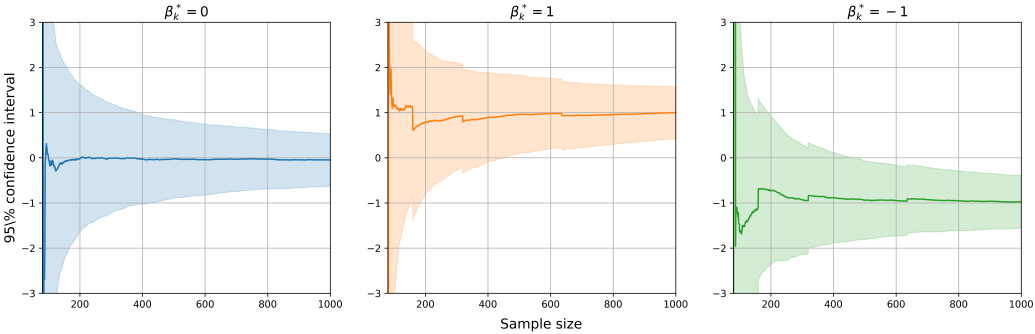

Figure: $n = 1000$, $p = 20000$, $s_0 = 20$, $\mathbf{\Sigma} = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$. Simulation results are averaged over 200 replications.

Simulation 2

Table: $n = 1000$, $p = 20000$, $s_0 = 20$, $\mathbf{\Sigma} = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$. Simulation results are averaged over 200 replications.

	β_k^*	ADL									
n		100	200	300	400	500	600	700	800	900	1000
Coverage probability	0	0.958	0.948	0.960	0.978	0.961	0.955	0.961	0.956	0.943	0.940
	1	0.953	0.928	0.950	0.933	0.930	0.945	0.933	0.948	0.946	0.946
	-1	0.953	0.920	0.938	0.945	0.933	0.943	0.925	0.941	0.938	0.926
Absolute bias	0	1.936	0.650	0.462	0.379	0.330	0.304	0.271	0.256	0.243	0.230
	1	2.005	0.728	0.518	0.436	0.376	0.331	0.299	0.280	0.265	0.245
	-1	2.133	0.709	0.481	0.414	0.353	0.321	0.303	0.276	0.267	0.250
Standard deviation	0	2.343	0.836	0.615	0.507	0.442	0.398	0.362	0.335	0.314	0.296
	1	2.426	0.838	0.616	0.509	0.443	0.397	0.362	0.336	0.314	0.296
	-1	2.693	0.838	0.619	0.511	0.446	0.400	0.365	0.337	0.315	0.296
Coverage length	0	9.186	3.277	2.412	1.988	1.735	1.563	1.422	1.315	1.232	1.161
	1	9.513	3.285	2.417	1.997	1.737	1.559	1.420	1.317	1.232	1.160
	-1	10.557	3.286	2.426	2.006	1.751	1.571	1.432	1.323	1.236	1.161

URL data example

Real data application : Detection of malicious websites [Ma et al., 2010].

- x: 3,231,961 features containing lexical, host-based information...
- y: Binary, is this URL a phishing site or not?
- Logistics regression.

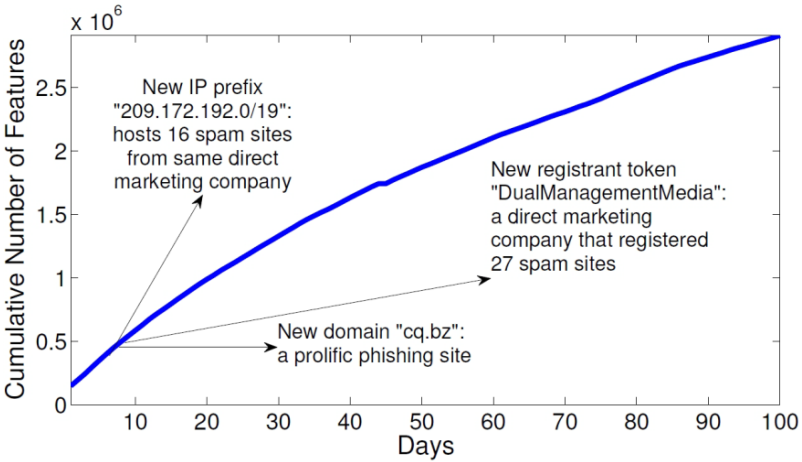

Figure: Trace plots of impacts of three features on the detection of malicious websites.

URL data example

Real data application on logistic regression:

- x: 3,231,961 features containing lexical, host-based information...
- y: Binary, is this URL a phishing site or not?

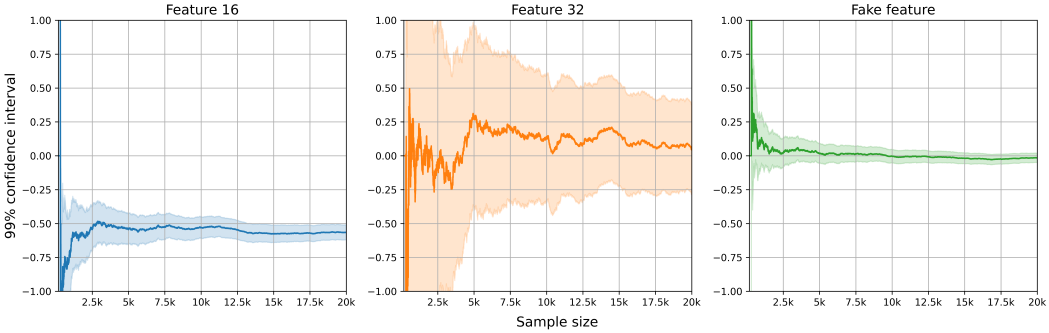

Figure: Trace plots of impacts of three features on detecting malicious websites.

- A incoming data needs around 2s to arrive on average.
- ADL consumes around 0.3s to process on an 11th Gen Intel Core i7-11370H.

Conclusion & Remarks

In this research, we develop online statistical inference approach in high-dimensional generalized linear models that

- sequentially updates both coefficient and variance without retrieving historical raw data .
- has a space complexity $\mathcal{O}(p)$, instead of $\mathcal{O}(p^2)$ nor $\mathcal{O}(np)$.
- achieve similar performance as the state-of-the-art offline methods in the literature.

Future works:

- non i.i.d. data
- non-convex objectives
- ...

Thanks

Thanks for watching!

$\mathcal{Q}\&\mathcal{A}$

Reference


Agarwal, A., Negahban, S., and Wainwright, M. J. (2012).
Stochastic optimization and sparse statistical recovery: Optimal algorithms for high dimensions.
Advances in Neural Information Processing Systems, 25.


Cai, T. T., Guo, Z., and Ma, R. (2023).
Statistical inference for high-dimensional generalized linear models with binary outcomes.
Journal of the American Statistical Association, 118(542):1319–1332.


Chen, X., Lee, J. D., Tong, X. T., and Zhang, Y. (2020).
Statistical inference for model parameters in stochastic gradient descent.
The Annals of Statistics, 48(1):251 – 273.


Deshpande, Y., Javanmard, A., and Mehrabi, M. (2023).
Online debiasing for adaptively collected high-dimensional data with applications to time series analysis.
Journal of the American Statistical Association, 118(542):1126–1139.


Han, R., Luo, L., Lin, Y., and Huang, J. (2023).
Online inference with debiased stochastic gradient descent.
Biometrika, 111(1):93–108.


Javanmard, A. and Montanari, A. (2014).
Confidence intervals and hypothesis testing for high-dimensional regression.
Journal of Machine Learning Research, 15(82):2869–2909.


Luo, L., Han, R., Lin, Y., and Huang, J. (2023).
Online inference in high-dimensional generalized linear models with streaming data.
Electronic Journal of Statistics, 17(2):3443 – 3471.


Ma, J., Kulesza, A., Dredze, M., Crammer, K., Saul, L., and Pereira, F. (2010).
Exploiting feature covariance in high-dimensional online learning.
In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 493–500.


Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., and Hung Byers, A. (2011).
Big data: The next frontier for innovation, competition, and productivity.


Nesterov, Y. (2009).
Primal-dual subgradient methods for convex problems.
Mathematical programming, 120(1):221–259.


Shi, C., Song, R., Lu, W., and Li, R. (2021).
Statistical inference for high-dimensional models via recursive online-score estimation.