개별 보고서

Online Debiased LASSO with Stochastic Gradient Descent algorithm

백승찬

December 17, 2023

Contents

1 Introduction

데이터 마이닝관련 방법론이 점차 발전해나가게 됨에 따라 실시간으로 수집된 데이터에 관한 분석관련 방법론 또한 연구되어왔다. 이러한 방법론들의 핵심적인 조건중 하나는 만족스러울만한 통계적 퍼포먼스를 발휘하면서 계산관련 요구사항을 축소시키는 것이었다. 그러나 대부분의 이러한 방법론들은 저차원 문제에 대해서만 적용이 가능했으며 고차원 문제에 대해서는 적용할 수 없었다.

오프라인 세팅에서의 경우, 즉 전체 데이터셋이 고정되어있고 사용가능한 경우에, 편향축소기법에 기반한 고차원 모형의 통계적 추론은 잘 확립이 되어있다.(Javanmard & Montanari,2014[3]; van de Geer et al.,2014[7]; Zhang&Zhang,2014[8])

최근 들어서는, LASSO(Least Absolute Shrinkage and Selection Operator)와 같은 방식으로 고차원 문제에서의 온라인 통계적 추론에 대해 Chen et al.(2020)[1], Han et al.(2021)[2] 등이 연구를 진행하였다. 그럼에도 불구하고, 이러한 방법들은 전체 데이터셋의 사용 가능성이 있거나, 모수의 차원이 매우 크거나 누적 표본 크기와 함께 빠르게 증가하는 특정 요약 통계량의 저장을 필요로 하는 단점이 있다.

본문에서 제시하는 방법의 경우, 고차원 선형 모형에서의 실시간 통계적 추론 접근방법을 제시하는 바이다. 이는 곧 새로운 데이터가 들어옴에 따라 순차적으로 회귀계수 추정량들을 업데이트하는 방식인데, 이전의 데이터셋에 대한 정보나 요약통계량에 대한 저장이 필요하지 않기에 데이터가 누적되어감에 따라 늘어나는 공간복잡도를 크게 줄여줄 수 있다는 장점을 가진다. 좀 더 구체적으로 말하자면, 예를 들어서 n개의 데이터셋 $\{(x_i, y_i)\}_{i=1}^n$이 다음과 같은 고차원 회귀모형으로부터 iid하게(independently and identically) 추출되었다고 하자:

$$y = x^\top \beta^* + \epsilon, \tag{1}$$

where $x \in \mathbb{R}^p$ is a predictor vector, the error term ϵ has zero mean and finite, but unknown, variance σ^2, and β^* is a s_0-sparse vector of coefficients, i.e., $||\beta^*||_0 = s_0$.

저차원 문제의 경우, 특히 p가 고정되어있을때 모형(1)의 온라인추론은 재귀적 최소 제곱법으로 쉽게 수행될 수 있음이 알려져있다.(Luo & Song, 2020)[4] 고차원 문제의 경우, 즉 $n \leq p$일때, β^*에 관한 신뢰구간 추정과 가설검정을 포함하는 온라인 통계적 추론의 경우 시간 복잡도와 공간 복잡도를 낮추는 것은 상당히 어렵다.

본문은, 위와 같은 고차원 선형 모형에 대한 온라인 통계적 추론 방법을 제시하는데 이는 곧 최소의 시간 복잡도와 공간 복잡도를 보장하는 방법이다. 특히 이에 대하여 새로운 편향축소기법을 제시하며 이를 활용함으로써 신뢰구간을 만들 수 있는 두 개의 온라인 알고리듬을 제안한다 : Debiased Stochastic Gradient Descent and Debiased Regularization Annealed Epoch Dual Averaging. 전자를 Online1, 후자를 Online2라고 하는데 Online2의 경우 Online1에 비해 월등한 성능을 딱히 보이지는 않다고 판단하여 보고서에 포함시키지는 않았다. 보고서는 이러한 Online1에 대한 알고리듬 작동 방식과 다른 알고리듬과의 성능을 비교하고 마무리하고자 한다.

2 Methods

2.1 Debiased Stochastic Gradient Descent

(Online) Debiased stochastic gradient descent에 대해 소개하고자 한다. 앞서 언급한 모형(1)에서 추출된 n개의 데이터셋 $\{(x_i, y_i)\}_{i=1}^n$에 대하여, β^*는 고차원의 sparse한 벡터이기에, 다음과 같은 ℓ_1-벌점 최소제곱항을 최소화함으로써 β^*에 관한 추정량을 얻을 수 있다:

$$\underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i^\top\beta)^2 + \lambda_n||\beta||_1 \right\}, \tag{2}$$

where λ_n is the regularization parameter.

온라인 세팅에서는 데이터가 시간이 지남에 따라 순서대로 입력되므로, (2)을 해결하는 간단한 방법은 바로 확률적 경사하강법 알고리듬을 soft-thresholding operator와 함께 사용하여 푸는 것이다. $\mathcal{S}(\beta;\lambda) = \text{sgn}(\beta)\odot\max\{|\beta|-\lambda, 0\}$라고 하자.(이때 $\odot$는 elementwise product를, $\text{sgn}(\cdot)$는 sign함수를 의미한다.) 학습률$\{\eta_i\}_{i=1}^n$과 이전과정에서의 LASSO추정량이 $\hat{\beta}^{(i-1)}$으로 주어졌을때, 확률적경사하강법 알고리듬을 통해 다음과 같은 결과를 얻을 수 있다:

$$\hat{b}^{(i)} = \hat{\beta}^{(i-1)} + \eta_i x_i(y_i - x_i^\top\hat{\beta}^{(i-1)}) \quad (i=1,\ldots,n), \tag{3}$$

with initial value being $\hat{\beta}^{(0)} = 0$.

이에 대하여 soft-thresholding operator를 적용하여 구한 $\hat{\beta}^{(i)}$는 다음과 같다:

$$\hat{\beta}^{(i)} = \mathcal{S}(\hat{b}^{(i)};\eta_i\lambda_i) \quad (i=1,\ldots,n) \tag{4}$$

위와 같이 $\hat{\beta}^{(i)}$를 구했을때, 다음과 같은 Polyak-Ruppert 평균화 반복법(Ruppert, 1988[6]; Polyak & Juditsky, 1992[5])을 통해 $\tilde{\beta}^{(i)}$를 구할 수 있다:

$$\tilde{\beta}^{(i)} = \frac{1}{i}\left\{(i-1)\tilde{\beta}^{(i-1)} + \hat{\beta}^{(i)}\right\} \quad (i=1,\ldots,n) \tag{5}$$

with the initial $\tilde{\beta}^{(0)} = 0$.

저차원 문제에서는 (5)과 같은 방식으로 구한 $\tilde{\beta}^{(n)}$은 β^*에 대한 만족스러운 추정량이 되며 이는 점근정규성 또한 잘 만족하게 된다. 그러나 고차원 문제에서의 경우 (2)에서 구한 $\hat{\beta}^{(i-1)}$이 root-n consistent하지 않으며 그 극한분포를 구할 수 없기에 (4)에서 구한 $\hat{\beta}^{(n)}$이나 (5)에서 구한 $\tilde{\beta}^{(n)}$ 모두 root-n consistent하지 않게 안된다. 그렇기에 $\tilde{\beta}^{(n)}$의 점근분포를 알 수 없어 유효한 신뢰구간 또한 구축이 불가능하게 된다. 이러한 문제를 해결하기위해, 본문의 저자들은 $\tilde{\beta}^{(n)}$의 편향을 줄이는 편향축소기법을 제안한다.

목표는 β_k (β의 k번째 원소, $k=1,\ldots,p.$)의 유효한 신뢰구간을 구축하는 것이다. 기존의 잘 알려진 편향축소방법들[8] 중에 핵심적인 단계의 경우 다음과 같은 목적 함수를 활용한 저차원 사영을

얻는 것이다:

$$\underset{\gamma\in\mathbb{R}^{p-1}}{\operatorname{argmin}}\left[\frac{1}{2n}\sum_{i=1}^{n}\{(x_i)_k-(x_i)_{-k}^{\top}\gamma\}^2+\lambda_n||\gamma||_1\right], \tag{6}$$

where $(x_i)_k$ is the kth element of x_i and $(x_i)_{-k}$ is a subvector of x_i that has had the kth element removed.

(6)의 최적화문제는 사실상 (2)문제와 거의 같기에, 앞서 (3)-(5)과 같은 방식으로 풀게되면, 다음과 같이 확률적경사하강법을 통해 $\hat{\gamma}^{(i)}$ 와 $\tilde{\gamma}^{(i)}$를 얻을 수 있게 된다:

$$\hat{r}^{(i)}=\hat{\gamma}^{(i-1)}+\eta_i(x_i)_{-k}\left\{(x_i)_k-(x_i)_{-k}^{\top}\hat{\gamma}^{(i-1)}\right\}\qquad(i=1,\ldots,n)$$

with soft-thresholding

$$\hat{\gamma}^{(i)}=\mathcal{S}(\hat{r}^{(i)};\eta_i\lambda_i)\qquad(i=1,\ldots,n) \tag{7}$$

and Polyak-Ruppert averaging

$$\tilde{\gamma}^{(i)}=\frac{1}{i}\left\{(i-1)\tilde{\gamma}^{(i-1)}+\hat{\gamma}^{(i)}\right\}\qquad(i=1,\ldots,n) \tag{8}$$

위와 같은 방식으로 구한$\tilde{\beta}^{(n)}$ 과 $\tilde{\gamma}^{(n)}$을 활용하여 다음과 같은 β_k 오프라인 편향축소 추정량 $\tilde{\beta}^{(n)}_{k,\text{off}}$ (van de Geer et al., 2014[7]; Zhang & Zhang, 2014[8])은 다음과 같이 정의할 수 있다:

$$\tilde{\beta}^{(n)}_{k,\text{off}}=\tilde{\beta}^{(n)}_{k}+\frac{\sum_{i=1}^{n}\{(x_i)_k-(x_i)_{-k}^{\top}\tilde{\gamma}^{(n)}\}(y_i-x_i^{\top}\tilde{\beta}^{(n)})}{\sum_{i=1}^{n}(x_i)_k\{(x_i)_k-(x_i)_{-k}^{\top}\tilde{\gamma}^{(n)}\}} \tag{9}$$

그러나, $\tilde{\beta}^{(n)}_{k,\text{off}}$을 온라인 세팅하에서 계산하는것은 결코 쉽지 않다. 좀 더 구체적으로 말하자면, (9) 계산의 경우 매 $i=1,\ldots,n-1$마다 $\{(x_i)_k-(x_i)_{-k}^{\top}\tilde{\gamma}^{(n)}\}$을 다시 계산해야 하는데, 이는 $n\to\infty$ 이게 되면 계산 불가능한 수준이 된다. 이러한 문제를 해결하기 위해, 저자들은 다음과 같은 방식을 제안한다:

$\tilde{\gamma}^{(n)}$을 사용하여 $\tilde{\beta}^{(n)}_k$의 편향을 수정하는 대신에, 제안하는 편향 축소접근 방식은 다음과 같다. 각 i번째 단계에서, i번째 표본이 도착할때, $\tilde{z}_{ik}=(x_i)_k-(x_i)_{-k}^{\top}\tilde{\gamma}^{(i)}\ \ (i=1,\ldots,n)$이라고 하자. 그러면 제안되는 편향축소추정량 $\tilde{\beta}^{(n)}_{k,\text{de}}$는 다음과 같이 정의된다:

$$\tilde{\beta}^{(n)}_{k,\text{de}}=\tilde{\beta}^{(n)}_{k}+\frac{\tilde{m}^{(n)}_{2k}-\tilde{m}^{(n)}_{3k}\tilde{\beta}^{(n)}}{\tilde{m}^{(n)}_{1k}}, \tag{10}$$

whih is $\tilde{\beta}^{(n)}_k$ plus a debiasing term, where

$$\tilde{m}^{(n)}_{1k}=\sum_{i=1}^{n}\tilde{z}_{ik}(x_i)_k,\quad \tilde{m}^{(n)}_{2k}=\sum_{i=1}^{n}\tilde{z}_{ik}y_i,\quad \tilde{m}^{(n)}_{3k}=\sum_{i=1}^{n}\tilde{z}_{ik}x_i^{\top} \tag{11}$$

그런데 (11)의 각 항들은 누적 합계로 쓸 수 있으므로, 다음과 같이 온라인 스타일로 업데이트가 능하게 표현할 수 있게 된다:

$$\tilde{m}^{(i)}_{1k}=\tilde{m}^{(i-1)}_{1k}+\tilde{z}_{ik}(x_i)_k,\quad \tilde{m}^{(i)}_{2k}=\tilde{m}^{(i-1)}_{1k}+\tilde{z}_{ik}y_i,$$

$$\tilde{m}_{3k}^{(i)} = \tilde{m}_{3k}^{(i-1)} + \tilde{z}_{ik} x_i^\top \quad (i = 1, \ldots, n)$$

추가적으로, $\tilde{\beta}_{k,\mathrm{de}}^{(n)}$의 표준편차의 자연 추정량은 다음과 같은 $\tilde{\sigma}^{(n)}\tilde{\tau}_k^{(n)}$가 된다:

$$\tilde{\sigma}^{(n)} = \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i^\top \tilde{\beta}^{(i)})^2 \right\}^{1/2}, \quad \tilde{\tau}_k^{(n)} = \frac{\left\{ \sum_{i=1}^{n} (\tilde{z}_{ik})^2 \right\}^{1/2}}{\tilde{m}_{1k}^{(n)}} \tag{12}$$

(10)과 (12)를 이용하게 되면, 다음과 같은β_k^*에 대한 $(1-\alpha)-$신뢰구간을 얻을 수 있게 된다:$(0 < \alpha < 1)$

$$(\tilde{\beta}_{k,\mathrm{de}}^{(n)} - z_{\alpha/2}\tilde{\sigma}^{(n)}\tilde{\tau}_k^{(n)}, \tilde{\beta}_{k,\mathrm{de}}^{(n)} + z_{\alpha/2}\tilde{\sigma}^{(n)}\tilde{\tau}_k^{(n)}) \tag{13}$$

where $z_{\alpha/2}$ is the z-score of $\alpha/2$.

추가적으로, 구체적인 실험을 들어가기에 앞서, learning rate $\{\eta_i\}_{i=1}^n$와 regularization parameter $\{\lambda_i\}_{i=1}^n$가 어떻게 정해지는건지 먼저 언급하고 넘어가고자 한다. 본문의 Supplementary material에 의하면 저자들은 $\eta_i = 0.0025/(i)^{0.51}$으로 설정했다고 전한다. 또한 λ_i의 경우 $c\{\log(p)/i\}^{1/2}$으로 정하였는데, 이때 c는 $c \in C = \{0.30, 0.35, 0.40, 0.45\}$으로, $S_i = \{(\log(p)/i)^{1/2}\}$이라고 했을때, λ_i는 다음과 같이 정하고 있음을 말하고 있다.

$$\lambda_i = \underset{\lambda \in CS_i}{\mathrm{argmin}} \{y_i - x_i^\top \hat{\beta}^{(i-1)}(\lambda)\}^2 \tag{14}$$

즉, 매 i번째 표본이 들어옴에 따라 λ_i는 바뀌게 되는데, 초기에 c값을 특정할 필요가 없다는 장점을 가진다. 이러한 일련의 과정을 Algorithm1에 정리하면 다음과 같다.

Algorithm 1 Debiased stochastic gradient descent

Input: learning rate $\{\eta_i\}_{i=1}^n$, regularization parameter $\{\lambda_i\}_{i=1}^n$, significance level α, index k

Initialize: $\hat{\beta}^{(0)}, \tilde{\beta}^{(0)}, \hat{\gamma}^{(0)}, \tilde{\gamma}^{(0)}, \tilde{m}_{1k}^{(0)}, \tilde{m}_{2k}^{(0)}, \tilde{m}_{3k}^{(0)}, \tilde{\sigma}^{(0)}, \tilde{\tau}_k^{(0)}$

For $i = 1, 2, \ldots, n$

collect data point (x_i, y_i)

compute $\hat{b}^{(i)} = \hat{\beta}^{(i-1)} + \eta_i x_i (y_i - x_i^\top \hat{\beta}^{(i-1)})$

compute $\hat{\beta}^{(i)} = \mathrm{sgn}(\hat{b}^{(i)}) \odot \max\{|\hat{b}^{(i)}| - \eta_i \lambda_i, 0\}$

compute $\tilde{\beta}^{(i)} = \left\{(i-1)\tilde{\beta}^{(i-1)} \hat{\beta}^{(i)}\right\} / i$

compute $\hat{r}^{(i)} = \hat{\gamma}^{(i-1)} + \eta_i (x_i)_{-k} \{(x_i)_k - (x_i)_{-k}^\top \hat{\gamma}^{(i-1)}\}$

compute $\hat{\gamma}^{(i)} = \mathrm{sgn}(\hat{r}^{(i)}) \odot \max\{|\hat{r}^{(i)}| - \eta_i \lambda_i, 0\}$

compute $\tilde{\gamma}^{(i)} = \frac{1}{i}\left\{(i-1)\tilde{\gamma}^{(i-1)} + \hat{\gamma}^{(i)}\right\}$

update $\tilde{m}_{1k}^{(i)} = \tilde{m}_{1k}^{(i-1)} + \tilde{z}_{ik}(x_i)_k$

update $\tilde{m}_{2k}^{(i)} = \tilde{m}_{1k}^{(i-1)} + \tilde{z}_{ik} y_i$

update $\tilde{m}_{3k}^{(i)} = \tilde{m}_{3k}^{(i-1)} + \tilde{z}_{ik} x_i^\top$

update $\tilde{\sigma}^{(i)} = \left\{\frac{1}{i}\sum_{j=1}^{i}(y_j - x_j^\top \tilde{\beta}^{(j)})^2\right\}^{1/2}$

update $\tilde{\tau}_k^{(i)} = \left\{\sum_{j=1}^{i}(\tilde{z}_{jk})^2\right\}^{1/2} / \tilde{m}_{1k}^{(i)}$

compute $\tilde{\beta}_{k,\mathrm{de}}^{(i)} = \tilde{\beta}_k^{(i)} + (\tilde{m}_{2k}^{(i)} - \tilde{m}_{3k}^{(i)} \tilde{\beta}^{(i)}) / \tilde{m}_{1k}^{(i)}$

obtain $(1-\alpha)$-confidence interval of β_k^*: $(\tilde{\beta}_{k,\mathrm{de}}^{(i)} - z_{\alpha/2}\tilde{\sigma}^{(i)}\tilde{\tau}_k^{(i)}, \tilde{\beta}_{k,\mathrm{de}}^{(i)} + z_{\alpha/2}\tilde{\sigma}^{(i)}\tilde{\tau}_k^{(i)})$

store $(\hat{\beta}^{(i)}, \tilde{\beta}^{(i)}, \hat{\gamma}^{(i)}, \tilde{\gamma}^{(i)}, \tilde{\beta}_{k,\mathrm{de}}^{(i)} \tilde{m}_{1k}^{(i)}, \tilde{m}_{2k}^{(i)}, \tilde{m}_{3k}^{(i)}, \tilde{\sigma}^{(i)}, \tilde{\tau}_k^{(i)})$ and clear other elements in memory

End

Return: $\tilde{\beta}_{k,\mathrm{de}}^{(n)}, \tilde{\tau}_k^{(n)}, (\tilde{\beta}_{k,\mathrm{de}}^{(n)} - z_{\alpha/2}\tilde{\sigma}^{(n)}\tilde{\tau}_k^{(n)}, \tilde{\beta}_{k,\mathrm{de}}^{(n)} + z_{\alpha/2}\tilde{\sigma}^{(n)}\tilde{\tau}_k^{(n)})$

3 Results

실험은 다음과 같은 세팅하에 진행하였다.

versioninfo()

1. Julia Version 1.9.3
2. Platform Info:
   - OS: Windows (x86_64-w64-mingw32)
   - CPU: 4 × Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz
   - WORD_SIZE: 64
   - LIBM: libopenlibm
   - LLVM: libLLVM-14.0.6 (ORCJIT, haswell)
3. Threads: 1 on 4 virtual cores / Environment: / JULIA_NUM_THREADS =

다만 다음과 같이 Online 1을 구현할때 코드에서 실수가 있었기에 이를 바로 잡아 $\tilde{\tau}$와 신뢰구간 계산을 올바르게 할 수 있었다.

코드 수정이전

```
tau_tilde_k = sqrt(tau_tilde_k_sum / i) / m_1k
```

코드 수정이후

```
tau_tilde_k = sqrt(tau_tilde_k_sum ) / m_1k
```

즉, 다음과 같은 수정이 이루어졌다:

$$\tilde{\tau}_k^{(i)} = \left\{ \frac{1}{i} \sum_{j=1}^{i} (\tilde{z}_{jk})^2 \right\}^{1/2} / \tilde{m}_{1k}^{(i)} \quad \Rightarrow \quad \tilde{\tau}_k^{(i)} = \left\{ \sum_{j=1}^{i} (\tilde{z}_{jk})^2 \right\}^{1/2} / \tilde{m}_{1k}^{(i)}$$

이러한 수정을 통해 $\tilde{\tau}_k^{(i)}$가 과소추정되는 문제가 해결되었고, 그 결과 신뢰구간 또한 올바르게 형성시킬 수 있었다. 그리고 실험의 경우 다음과 경우에 대하여 실험을 진행하게 되었다:

1. $n = 100, p = 400, s_0 = 6$ with two types of Σ, $\Sigma = I_p$ and $\Sigma = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$
2. $n = 400, p = 2000, s_0 = 10$ with two types of Σ, $\Sigma = I_p$ and $\Sigma = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$

또한 β^*는 $s_0 = 6$ 인 경우에 β^*의 첫 3개 성분은 1으로, 그 다음 3개의 성분은 0.1으로, 나머지는 0으로 하여 실험을 진행하였으며, $s_0 = 20$ 인 경우에 β^*의 첫 10개 성분은 1으로, 그 다음 10개의 성분은 0.1으로, 나머지는 0으로 하여 실험을 진행하였다. 그리고 이에 대한 절대표준편차와 표준편차, 신뢰구간길이는 각각 $\beta_k^* = 0, 0.1, 1$인 경우의 평균으로 구하였다.

마지막으로 $n = 400, p = 2000$인 경우에 $\beta_k^* = 0, 0.1, 1$에 관한 신뢰구간을 Online Debiased Stochastic Gradient Descent 방법을 통해 구해보았다. 실험결과는 다음과 같다:

3.1 $n = 100, p = 400, s_0 = 6$ with two types of Σ, $\Sigma = I_p$ and $\Sigma = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$

Table1 : Performances of three methods with $n = 100,\ p = 400,\ , s_0 = 6, \Sigma = I_p$

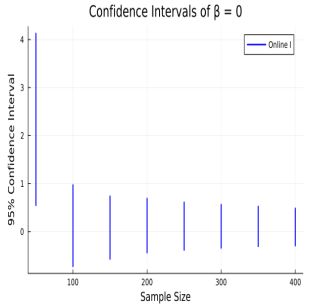
Sample size	β_k^*	Offline	Chen et al.			Online 1		
		100	40	70	100	40	70	100
Absolute bias	0	0.160	0.259	0.194	0.162	0.222	0.172	0.144
	0.1	0.165	0.255	0.196	0.167	0.199	0.176	0.149
	1	0.144	0.218	0.167	0.145	0.272	0.199	0.160
Standard Error	0	0.120	0.323	0.243	0.202	0.276	0.213	0.178
	0.1	0.103	0.279	0.218	0.182	0.235	0.209	0.186
	1	0.092	0.199	0.157	0.131	0.168	0.117	0.067
$\tilde{\tau}$	0	0.100	0.160	0.120	0.100	0.164	0.122	0.102
	0.1	0.100	0.160	0.120	0.100	0.161	0.122	0.102
	1	0.100	0.160	0.120	0.100	0.164	0.121	0.102
CI-Length	0	0.392	0.627	0.470	0.392	1.175	0.905	0.761
	0.1	0.392	0.627	0.470	0.392	1.155	0.907	0.764
	1	0.392	0.627	0.470	0.392	1.170	0.893	0.763
Mean Time(s)		0.182	1.745			1.294		

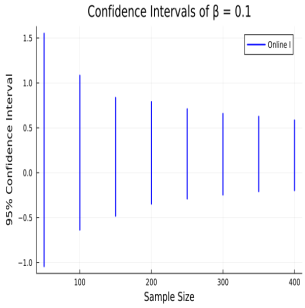
Table2 : Performances of three methods with $n = 100,\ p = 400,\ , s_0 = 6, \Sigma = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$

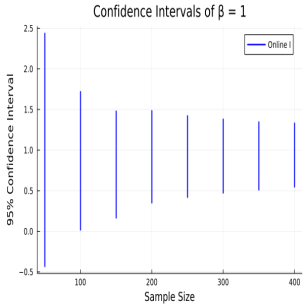
Sample size	β_k^*	Offline	Chen et al.			Online 1		
		100	40	70	100	40	70	100
Absolute bias	0	0.203	0.339	0.254	0.213	0.299	0.233	0.194
	0.1	0.575	0.641	0.612	0.602	0.566	0.540	0.526
	1	0.839	0.887	0.888	0.894	0.612	0.697	0.708
Standard Error	0	0.152	0.420	0.316	0.265	0.361	0.276	0.225
	0.1	0.334	0.415	0.382	0.364	0.372	0.362	0.310
	1	0.192	0.197	0.171	0.165	0.208	0.161	0.148
$\tilde{\tau}$	0	0.100	0.160	0.120	0.100	0.165	0.124	0.103
	0.1	0.100	0.160	0.120	0.100	0.166	0.124	0.103
	1	0.100	0.160	0.120	0.100	0.163	0.122	0.104
CI-Length	0	0.392	0.627	0.470	0.392	1.606	1.233	1.023
	0.1	0.392	0.627	0.470	0.392	1.607	1.232	1.020
	1	0.392	0.627	0.470	0.392	1.582	1.216	1.024
Mean Time(s)		0.188	1.732			1.263		

3.2 $n = 400, p = 2000, s_0 = 20$ with two types of Σ, $\Sigma = I_p$ and $\Sigma = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$

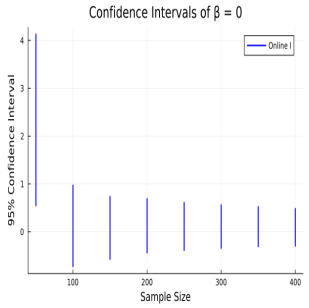
$$n = 400, p = 2000, s_0 = 20 \text{ with } \Sigma = I_p$$

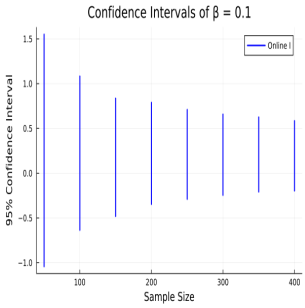


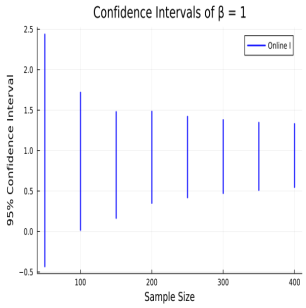




$$n = 400, p = 2000, s_0 = 20 \text{ with } \Sigma = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$$







4 Discussion

Online1(Debiased Stochastic Gradient Descent)을 줄리아 언어로 구현한 결과 실제로는 $n = 400, p = 5000$인 경우도 계산가능하였다. 그러나 Chen et al.의 경우 그렇지 못하여 공정한 성능비교를 위해 $n = 400, p = 2000$인 경우에 대하여 실험을 진행하게 되었다. 그리고 실험결과 공분산이 존재하지 않는 경우 ($\Sigma = I_p$) 회귀계수 추정오차(Absolute bias)가 본문의 실험결과처럼 만족할만한 수준으로 나왔으나 공분산이 존재하는 경우에는($\Sigma = \{0.5^{|i-j|}\}_{i,j=1,\ldots,p}$) 그렇지 못했다. 이외에, regularization parameter λ_i를 정할때, 실제로 $c = \{0.30, 0.35, 0.40, 0.45\}$의 4가지 경우에 대해 모두 계산한뒤 정하는 방식이기에, 이에 대하여 병렬처리를 해서 계산한다면 계산시간을 약 4분의 1로 줄일 수 있다고 생각한다. 즉 원래도 빠른 알고리즘이지만 이보다 더 빠르게 할 수 있다고 본다.

Online2(Debiased regularization annealed epoch dual averaging)의 경우 시간관계상 구현을 하지는 못했으나, 그 방식이 Online1과 매우 유사하며 다만 $\hat{\beta}^{(i)}$와 $\hat{\gamma}^{(i)}$를 매 반복 i 마다 계산하지 않고 데이터셋이 어느정도 누적될때까지 모아서 한꺼번에 계산하는, 이른바 annealed epoch dual averaging 방식을 취한다. 이는 딥러닝에서 배치(batch)에 대하여 미니배치(mini-batch)를 구하고

이러한 미니배치에 대해 그라디언트를 계산하고 누적시킨뒤 평균내는 방식과 사실상 같기에, 제대로만 구현한다면 본문에 나와있는 실험결과와 달리 각 미니배치마다 $\hat{\beta}^{(i)}$와 $\hat{\gamma}^{(i)}$를 업데이트할 수 있기 때문에 상당히 빠른 속도로(Online1보다 빠르게) 계산이 가능하며 메모리 사용량또한 적어진다고 본다.

Peer Review

- 황서화 : 발표슬라이드 작성, Offline, Chen et al. 구현 및 실험, 샘플링함수 구현

References

[1] Haoyu Chen, Wenbin Lu, and Rui Song. Statistical inference for online decision making via stochastic gradient descent. *Journal of the American Statistical Association*, 116(534):708–719, 2021.

[2] Ruijian Han, Lan Luo, Yuanyuan Lin, and Jian Huang. Online debiased lasso for streaming data. *Computer Science, Mathematics*, 2021.

[3] Adel Javanmard and Andrea Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *Journal of Machine Learning Research 15 (JMLR)*, 2014.

[4] Luo Lan and Peter X.-K. Song. Renewable estimation and incremental inference in generalized linear models with streaming data sets. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) (JRSS)*, 2020.

[5] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 1992.

[6] D. Ruppert. Efficient estimations from a slowly convergent robbins-monro process. *Cornell University Operations Research and Industrial Engineering*, 1988.

[7] S. van de Geer, Peter Bühlmann, Ya'acov Ritov, and Ruben Dezeure. On asymptotically optimal confidence regions and tests for high-dimensional models. *Annals of Statistics (AOS)*, 2014.

[8] Cun-Hui Zhang and Stephanie S. Zhang. Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) (JRSS)*, 2014.