具有组合结构的统计推断和在线算法

张志华
北京大学数学学院和统计科学中心
zhzhang@math.pku.edu.cn

MLA， 2022.11.05

- 具有组合结构问题
- 在线统计推断案例
- 一些潜在研究方向

Gaussian Location Models

Model:

$$\mathcal{P} = \{\mathcal{N}(\theta, I_p) : \theta \in \Theta\}$$

where I_p is the p-dimensional identity matrix and $\Theta \subset \mathbb{R}^p$. Equivalently,

$$X = \theta + Z \qquad Z \sim \mathcal{N}(0, I_p), \theta \in \Theta \subset \mathbb{R}^p.$$

- $p = 1$: scalar case
- $p > 1$: vector case
  We also encompass matrix case: By arranging a p^2-dimensional vector into a $p \times p$ matrix. In this case $\Theta \subset \mathbb{R}^{p \times p}$.

Objective: Examples of the objective include $T(\theta) = \theta$, $\|\theta\|_2$, $\theta_{max} = \max_{i \in [p]} \theta_i$, where $[p] = \{1, \cdots, p\}$.

Parameter space: Examples of the parameter space include the following:

a) $\Theta = \mathbb{R}^p$: unstructured.

b) $\Theta = \{\text{all } k\text{-sparse vectors}\} = \{\theta \in \mathbb{R}^p : \|\theta\|_0 \leq k\}$, where $\|\theta\|_0 \triangleq |\{i : \theta_i \neq 0\}|$ denotes the size of the support.

$\Theta = l_q$-norm balls, $0 \leq q \leq \infty$, where $\|\theta\|_q = (\sum |\theta_i|^q)^{\frac{1}{q}}$.

c) Matrix case: low-rank matrices: $\Theta = \{\theta : rank(\theta) \leq r\}$.

Note that by definition, more structure (smaller paramater space) always leads to smaller risk; but it need not simplify the computation issue.

Sparsity /Low Rank Structure

Testing: We have two scenarios and based on the observed data X, we want to determine which one is the true scenario.

$*$ Simple Hypothesis:

$$H_0 : \quad \theta = \theta_0$$
$$H_1 : \quad \theta = \theta_1$$

For instance θ_0 could be the all zero vector and θ_1 could be all one vector. Then this corresponds to sending a single bit repeatedly in Gaussian noise.

parameter space $= \Theta = \{\theta_0, \theta_1\} = \hat{\Theta} =$ decision space

$l(\theta, \hat{\theta}) = 1_{\{\theta \neq \hat{\theta}\}}$: This is Hamming loss (zero-one loss).

Hypothesis Testing

Stochastic Block Models

- The stochastic block model is a family of distributions over graphs, parametrized by integers n (the number of vertices) and k (the number of communities), and probabilities p_{in} (internal edge probability) and p_{out} (external edge probability).
- One samples $G \sim \mathcal{G}(n, k, p_{\text{in}}, p_{\text{out}})$ as follows:
  1. Identify the vertex set $V(G)$ with $[n]$.
  2. Sample a uniform partition of $[n]$ into k sets or "communities" of equal size. Let $X \in \{0,1\}^{n \times k}$ represent this partition, with $X_{i,c} = \mathbf{1}[\text{ vertex } i \in \text{ community } c]$.
  3. For each pair $i \neq j \in [n]$ independently, add edge (i, j) to $E(G)$ with probability p_{in} if i, j are in the same community, and with probability p_{out} otherwise.
- Our goal: Recovery: given $G \sim \mathcal{G}(n, k, p_{\text{in}}, p_{\text{out}})$, recover X (either exactly or approximately).

Online Adversary Learning

Two agents, a learner and an adversary, interact over a total of T rounds, for some T ∈ N.

The learner and adversary are given at the beginning a set X and a set F consisting of [0, 1]-valued functions on X , known as hypotheses.

The players perform the following for each round 1 ≤ t ≤ T: the learner chooses a hypothesis $f_t \in \mathcal{F}$ (which may be random), and the adversary picks $(x_t, y_t) \in \mathcal{X} \times [0,1]$ (which may be random) based on the history of moves $f_1, \ldots, f_{t-1}$ and $(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})$

The goal of the learner is to minimize its expected regret

$$\mathrm{Reg}_T := \mathbb{E}\left[\sum_{t=1}^{T} |f_t(x_t) - y_t| - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} |f(x_t) - y_t|\right]$$

- Proper: the learner picks $f_t \in \mathcal{F}$ Improper: the learner can pick arbitrary f
- Realizable: the adversary is constrained to choose the sequence $(x_t, y_t),\ 1 \le t \le T$ so that there is some $f^\star \in \mathcal{F}$ so that $f^\star(x_t) = y_t$ for all t.

Mixed Integer Linear Programming (MILP)

$$
\begin{aligned}
\text{minimize} \quad & c^\top x \\
\text{subject to} \quad & Ax \leq b \\
& l \leq x \leq u \\
& x_i \in \mathbb{Z}, \quad i \in \mathcal{I}
\end{aligned}
$$

$A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $l \in (\mathbb{R} \cup \{-\infty\})^n$, and $u \in (\mathbb{R} \cup \{\infty\})^n$

$\mathcal{I} \subseteq \{1, \ldots, n\}$ refers to the index set of integer variables.

Discovering faster matrix multiplication algorithms with reinforcement learning

Alhussein Fawzi[1,2]✉, Matej Balog[1,2], Aja Huang[1,2], Thomas Hubert[1,2], Bernardino Romera-Paredes[1,2], Mohammadamin Barekatain[1], Alexander Novikov[1], Francisco J. R. Ruiz[1], Julian Schrittwieser[1], Grzegorz Swirszcz[1], David Silver[1], Demis Hassabis[1] & Pushmeet Kohli[1]

Matrix multiplication

- Algorithms as tensor decomposition

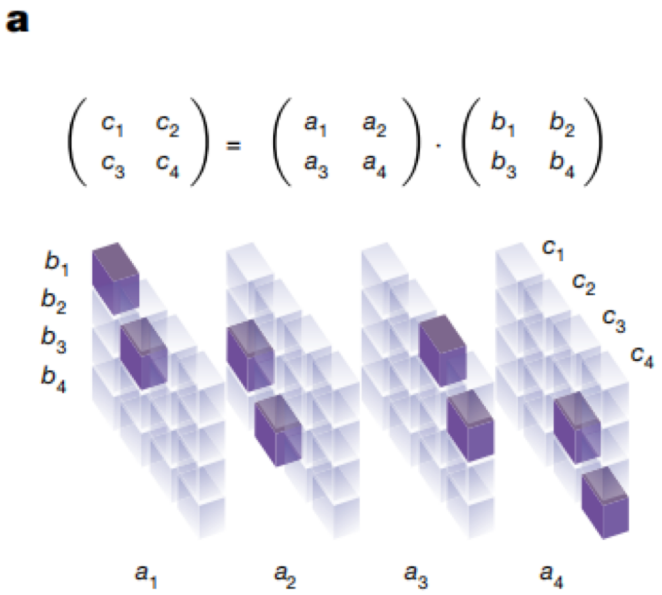
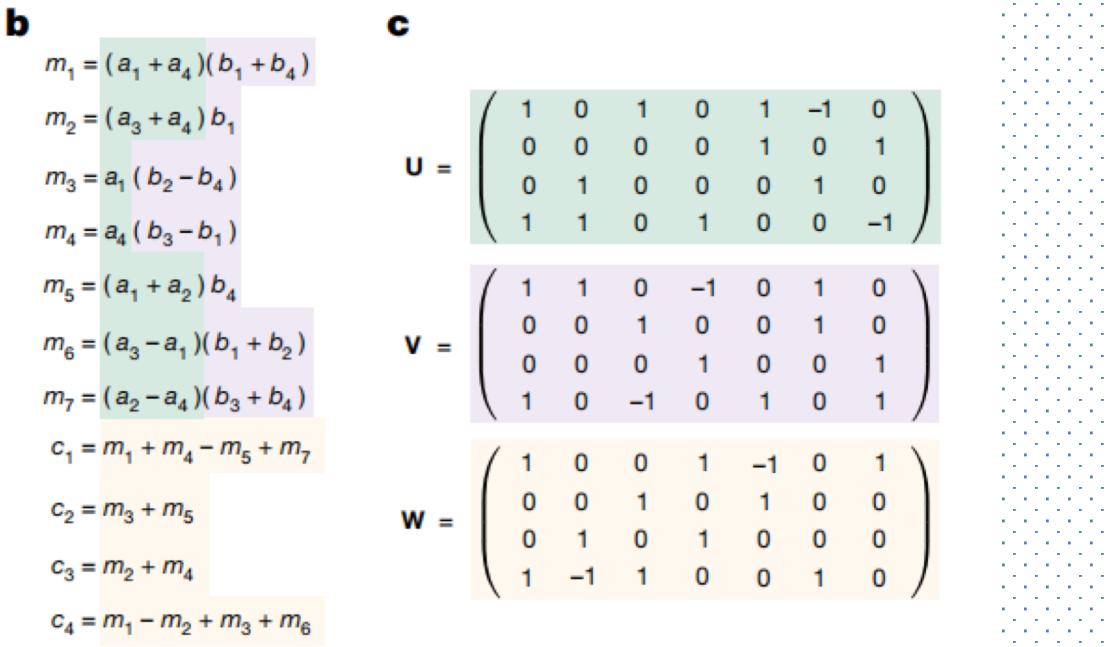


Fig. 1 | Matrix multiplication tensor and algorithms. a, Tensor $\mathcal{T}_2$ representing the multiplication of two 2×2 matrices. Tensor entries equal to 1 are depicted in purple, and 0 entries are semi-transparent. The tensor specifies which entries from the input matrices to read, and where to write the result. For example, as $c_1 = a_1 b_1 + a_2 b_3$, tensor entries located at (a_1, b_1, c_1) and (a_2, b_3, c_1) are set to 1.

Matrix multiplication

- Algorithms as tensor decomposition



b, Strassen's algorithm[2] for multiplying 2 × 2 matrices using 7 multiplications. **c**, Strassen's algorithm in tensor factor representation. The stacked factors **U**, **V** and **W** (green, purple and yellow, respectively) provide a rank-7 decomposition of $\mathcal{T}_2$ (equation (1)). The correspondence between arithmetic operations (**b**) and factors (**c**) is shown by using the aforementioned colours.

Matrix multiplication

- Algorithms as tensor decomposition

Algorithm 1

A meta-algorithm parameterized by $\{\mathbf{u}^{(r)}, \mathbf{v}^{(r)}, \mathbf{w}^{(r)}\}_{r=1}^{R}$ for computing the matrix product $\mathbf{C}=\mathbf{AB}$. It is noted that R controls the number of multiplications between input matrix entries.

Parameters: $\{\mathbf{u}^{(r)}, \mathbf{v}^{(r)}, \mathbf{w}^{(r)}\}_{r=1}^{R}$: length-$n^2$ vectors such that $\mathcal{T}_n = \sum_{r=1}^{R} \mathbf{u}^{(r)} \otimes \mathbf{v}^{(r)} \otimes \mathbf{w}^{(r)}$

Input: $\mathbf{A}, \mathbf{B}$: matrices of size $n \times n$

Output: $\mathbf{C}=\mathbf{AB}$

(1) **for** $r=1, \ldots, R$ **do**

(2) $\quad m_r \leftarrow (u_1^{(r)} a_1 + \cdots + u_{n^2}^{(r)} a_{n^2})(v_1^{(r)} b_1 + \cdots + v_{n^2}^{(r)} b_{n^2})$

(3) **for** $i=1, \ldots, n^2$ **do**

(4) $\quad c_i \leftarrow w_i^{(1)} m_1 + \cdots + w_i^{(R)} m_R$

return $\mathbf{C}$

Matrix multiplication

- Algorithms as tensor decomposition
  - They use $\mathcal{T}_{n,m,p}$ to describe the rectangular matrix multiplication operation of size $n \times m$ with $m \times p$ (note that $\mathcal{T}_n = \mathcal{T}_{n,n,n}$)
  - An algorithm is formulated as a decomposition of the tensor into ***R*** rank-one terms as below

$$\mathcal{T}_n = \sum_{r=1}^{R} \mathbf{u}^{(r)} \otimes \mathbf{v}^{(r)} \otimes \mathbf{w}^{(r)}$$

  - In particular, N $\times$ N matrices can be multiplied with asymptotic complexity $\mathcal{O}(N^{\log_n(R)})$
  - **Find minimal *R***

Q-Learning

The update rule of Q-learning is

$$Q_t = (1 - \eta_t) Q_{t-1} + \eta_t \hat{\mathcal{T}}_t(Q_{t-1})$$

$$\mathcal{T}(Q)(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \max_{a' \in \mathcal{A}} Q(s', a') \text{ for all } (s,a) \in \mathcal{S} \times \mathcal{A}$$

Local SGD (Federated Averaging)

A pool of K clients, in which the k-th client has a local dataset consisting of i.i.d samples from some unknown distribution

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \sum_{k=1}^{K} p_k f_k(\boldsymbol{x}) := \sum_{k=1}^{K} p_k \mathbb{E}_{\xi_k \sim \mathcal{D}_k} f_k(\boldsymbol{x}; \xi_k)$$

Algorithm 1 Local SGD

Input: functions $\{f_k\}_{k=1}^K$, initial point $\boldsymbol{x}_0$, step size η_0, communication set $\mathcal{I} = \{t_0, t_1, \cdots\}$.
Initialization: let $\boldsymbol{x}_0^k = \boldsymbol{x}_0$ for all k.
for round $m = 0$ **to** $T-1$ **do**
  for iteration $t = t_m + 1$ **to** t_{m+1} **do**
    for each device $k = 1$ **to** K **do**
      $\boldsymbol{x}_t^k = \boldsymbol{x}_{t-1}^k - \eta_m \nabla f_k(\boldsymbol{x}_{t-1}^k; \xi_{t-1}^k)$. # perform $E_m = t_{m+1} - t_m$ steps of local updates.
    end for
    The central server aggregates: $\bar{\boldsymbol{x}}_{t_{m+1}} = \sum_{k=1}^K p_k \boldsymbol{x}_{t_{m+1}}^k$.
    Synchronization: $\boldsymbol{x}_{t_{m+1}}^k \leftarrow \bar{\boldsymbol{x}}_{t_{m+1}}$ for all k.
  end for
end for
Return: $\widehat{\boldsymbol{x}} = \frac{1}{T} \sum_{m=1}^{T} \bar{\boldsymbol{x}}_{t_m}$.

Approximate Message Passing

- 近似讯息传输（Approximate Message Passing AMP）可看作是幂法的推广，**每步迭代在逐步降噪，在噪声足够低时，再做预测（先降噪，再预测）**，常常能达到Bayes估计的最优Rate

$$v^{k+1} = \frac{Av^k}{\|Av^k\|}. \quad \Rightarrow \quad \hat{v}^k = g_k(v^k), \qquad v^{k+1} = A\hat{v}^k - b_k\hat{v}^{k-1}$$

- 其中$g_k : \mathbb{R} \to \mathbb{R}$是某种降噪函数，取决于先验分布π，elementwise作用于向量，例如可以使用Lasso里的Soft-threshold来达到稀疏效果
- $b_k\hat{v}^{k-1}$ 被称为Onsager修正项，来保证每一步的迭代结果仍是渐近正态的

AMP for Estimating Rank 1 Matrix

- 问题：观测到 $A \equiv A(n) = \frac{\lambda}{n} v v^\top + W \in \mathbb{R}^{n \times n}$ 要估计 v
- 对称AMP迭代：Lipschitz的降噪函数 $(g_k)_{k=0}^{\infty}$

$$\hat{v}^{-1} \equiv \hat{v}^{-1}(n) := 0 \in \mathbb{R}^n \qquad \text{initialiser } v^0 \equiv v^0(n) \in \mathbb{R}^n$$

$$\hat{v}^k := g_k(v^k), \qquad b_k := \langle g_k'(v^k) \rangle_n = \frac{1}{n} \sum_{i=1}^{n} g_k'(u_i^k), \qquad v^{k+1} := A\hat{v}^k - b_k \hat{v}^{k-1}$$

- 其中，v^k 表示“Latent Space”的表示 $\hat{v}^k$ 是对 v 的估计
- 第一步是在降噪（同时估计），二三步在做Onsager修正，维持渐近正态性

数据驱动方法

- 数据驱动方法。数据变得更容易获取，且积累的规模越来越大，从数据中挖掘信息和推理结论的算法技术在不断改进。
- 利用和探索不确定性。概率描述不确定性的数学语言，统计是概率和数据的结合。
- 统计推断是实现数据驱动的一个最重要途径，这意味着我们要对数据及其潜在分布得出严谨、可证明的结论。

核心科学问题

- 然而由于我们的资源（包含数据、信息和计算）常常是有限的，因此我们需要设计能够尽可能高效地使用已有资源、信息和计算的算法；为实际问题提供建设性和明确的可行解决方案。
- 或者知道何时信息和计算有效的算法是不可能的。建立不可能性的结果则提供了算法使用的边界，可以理解为一种存在性理论分析 。

研究手段

- 统计学往往从样本有效推理角度来研究上述问题，而计算机科学关注计算有效算法。
- 两者的交汇产生了有趣和令人惊奇的效果。比如，计算机科学的多臂老虎机或在线算法为大规模统计推断提供了新的技术思路，平方求和证明技术为一些鲁棒统计估计问题提供了求解方法；统计假设检验成为理论计算机学家的研究工具或问题，最近发展起来的随机局部化技术则被理论计算学家用于分析凸离散优化问题。
- 因此，这种渐近和非渐近、随机和确定、连续和离散、概率和组合相结合的更现代观点为我们带来了新的视角和研究手段。

- 具有组合结构问题
- 在线统计推断案例
- 一些潜在研究方向

在线统计推断

- 在数据/信息层面，我们拥有的数据越多，我们拥有的信息就越多，我们就越能理解潜在分布。然而收集数据需要时间、经费和精力。
- 在计算层面，有了数据，我们将在其上运行一个算法来进行推理。同样我们可以用来处理数据的时间和内储也是有限的。计算资源需要代价，所以我们希望进行实时或在线推断和学习。

Stochastic Approximation

- Devote to finding a fixed point of a particular function.
  Was first proposed by Robbins and Monro [Robbins & Monro 51]

  $$Z_{n+1} = Z_n + \gamma_n[F(Z_n, \eta_{n+1})] \qquad \gamma_n \to 0, \ \sum \gamma_n = \infty$$

- Linear Regression, Stochastic Gradient Descent, Q-Learning are all adapted to this formulation.

$$Z_{n+1} = Z_n + \gamma_n[h(Z_n) + \eta_{n+1}]$$

- Another generalization of the classical version such as Simulate Annealing, Stochastic Gradient Langevin Dynamics

$$Z_{n+1} = Z_n + \gamma_n[h(Z_n) + \eta_{n+1}] + \sigma_n \xi_{n+1}$$

Asymptotic Analogy: Linear Setting

In order to find the root of the linear function $B(X-\theta)$ based on the observations

$Y_n = B(X_n - \theta) + \varepsilon_n$ consider the SA procedures of the form

$$X_{n+1} = X_n - \frac{A}{n} Y_n$$

The bias of the estimator should be negligible, which means $\lim n^{1/2} E(X_n - \theta) = 0$

This is true iff all eigenvalues of the matrix $C = I/2 - AB$ have negative real parts. When the matrix A satisfies the condition, the covariance of the asymptotic normal subjects to

$$C\sigma(A) + \sigma(A)C' = -A\Sigma A'$$

THEOREM 1. If $A \in \mathscr{D}$, then $\sigma(A) - \sigma\left(B^{-1}\right)$ is nonnegative definite. [Wei 87]

Obstructive Factor for Usage of Final Iteration

- For multidimensional problems, no matter what kind of procedures are taken for estimating the optimal choice matrix, the computation task involved is very intensive.
- The assignment of the step size is restrictive.
- As a consequence, the results in adaptive stochastic approximation are largely of theoretical nature and have not been used widely in various applications.

Acceleration by Simple Averaging

The averaging method, simultaneously introduced by [Polyak 90] and [Ruppert 91], is an asymptotic efficient algorithm.

$$x_t = x_{t-1} - \gamma_t y_t, \quad y_t = R(x_{t-1}) + \xi_t,$$
$$\bar{x}_t = \frac{1}{t}\sum_{i=0}^{t-1} x_i, \quad x_0 \in R^N$$

Assumption 3.1. There exists a function $V(x) : R^N \to R^1$ such that for some $\lambda > 0, \alpha > 0, \varepsilon > 0, L > 0$, and all $x, y \in R^N$, the conditions $V(x) \geqq \alpha|x|^2, |\nabla V(x) - \nabla V(y)| \leqq L|x - y|, \quad V(x^*) = 0, \nabla V(x - x^*)^T R(x) > 0$ for $x \neq x^*$ hold true. Moreover, $\nabla V(x - x^*)^T R(x) \geqq \lambda V(x)$ for all $|x - x^*| \leqq \varepsilon$.

Assumption 3.2. There exists a matrix $G \in R^{N \times N}$ and $K_1 < \infty, \varepsilon > 0, 0 < \lambda \leqq 1$ such that
(8) $|R(x) - G(x - x^*)| \leqq K_1|x - x^*|^{1+\lambda}$
for all $|x - x^*| \leqq \varepsilon$ and $\operatorname{Re}\lambda_i(G) > 0, i = \overline{1, N}$.

Acceleration by Simple Averaging

The averaging method, simultaneously introduced by [Polyak 90] and [Ruppert 91], is an asymptotic efficient algorithm.

$$x_t = x_{t-1} - \gamma_t y_t, \quad y_t = R(x_{t-1}) + \xi_t,$$
$$\bar{x}_t = \frac{1}{t}\sum_{i=0}^{t-1} x_i, \quad x_0 \in R^N$$

Assumption 3.3. $(\xi_t)_{t\geqq 1}$ is a martingale-difference process, defined on a probability space $(\Omega, \mathfrak{F}, \mathfrak{F}_t, P)$, i.e., $E(\xi_t | \mathfrak{F}_{t-1}) = 0$ almost surely, and for some K_2

$$E(|\xi_t|^2 | \mathfrak{F}_{t-1}) + |R(x_{t-1})|^2 \leqq K_2(1+|x_{t-1}|^2) \quad \text{a.s.}$$

Assumption 3.4. It holds that $(\gamma_t - \gamma_{t+1})/\gamma_t = o(\gamma_t)$, $\gamma_t > 0$ for all t;

(10) $$\sum_{t=1}^{\infty} (1+\lambda)/\gamma_t^2 t^{-1/2} < \infty.$$

Acceleration by Simple Averaging

THEOREM 2. If Assumptions 3.1-3.4 are satisfied, then $\bar{x}_t \to x^*$ almost surely, and

$$\sqrt{t}(\bar{x}_t - x^*) \xrightarrow{D} N(0, V)$$

Here

$$V = G^{-1} S (G^{-1})^T$$

[Polyak & Juditsky 92]

- Without complex computation, the average sequence can be updated recursively.
- Step size can be chosen as

$$n^{-\alpha} \text{ where } \alpha \in (\tfrac{1}{2}, 1)$$

Theorem 3 (Averaging, strong convexity) Assume (H1, H2', H3, H4, H6, H7). Then, for $\bar{\theta}_n = \frac{1}{n}\sum_{k=0}^{n-1}\theta_k$ and $\alpha \in (0,1)$, we have:

$$\begin{aligned}\left(\mathbb{E}\|\bar{\theta}_n - \theta^*\|^2\right)^{1/2} \leqslant & \frac{\left[\operatorname{tr} f''(\theta^*)^{-1}\Sigma f''(\theta^*)^{-1}\right]^{1/2}}{\sqrt{n}} + \frac{6\sigma}{\mu C^{1/2}}\frac{1}{n^{1-\alpha/2}} + \frac{MC\tau^2}{2\mu^{3/2}}\left(1 + (\mu C)^{1/2}\right)\frac{\varphi_{1-\alpha}(n)}{n} \\ & + \frac{4LC^{1/2}}{\mu}\frac{\varphi_{1-\alpha}(n)^{1/2}}{n} + \frac{8A}{n\mu^{1/2}}\left(\frac{1}{C} + L\right)\left(\delta_0 + \frac{\sigma^2}{L^2}\right)^{1/2} \\ & + \frac{5MC^{1/2}\tau}{2n\mu} A \exp\left(24L^4C^4\right)\left(\delta_0 + \frac{\mu\mathbb{E}\left[\|\theta_0 - \theta^*\|^4\right]}{20C\tau^2} + 2\tau^2C^3\mu + 8\tau^2C^2\right)^{1/2}\end{aligned}$$

where A is a constant that depends only on μ, C, L and α.

[Moulines & Bach 13]

Inference

Under the knowledge of the asymptotic efficiency of Polyak-Ruppert Averaging, it is natural to propose some practical inference approaches for these estimators.
One critical task is to give the variance a consistent estimate.

$$\sqrt{n}(\bar{x}_n - x^*) \Rightarrow \mathcal{N}\left(0, A^{-1}SA^{-1}\right)$$

$$A = \nabla^2 F(x^*) \quad S = \mathbb{E}\Big([\nabla f(x^*, \zeta)][\nabla f(x^*, \zeta)]^\top\Big)$$

- As the most intuitive method, the Plug-in estimator evaluate S and A separately and combine them to get the final evaluation of the asymptotic variance.

$$A_n := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f(x_{i-1}, \zeta_i), \quad S_n := \frac{1}{n}\sum_{i=1}^{n} \nabla f(x_{i-1}, \zeta_i) \nabla f(x_{i-1}, \zeta_i)^\top$$

- Computation of the Hessian matrix of the loss function and its inverse. Unavoidable burden.
- Can not be updated recursive.

Inference

- [Chen et al. 16] brought the batch-means estimator from Monte Carlo [Glynn & Whitt 91] to stochastic approximation procedures.
- Note the SA sequence is endowed some mixing property, which means the correlation between Xi and Xj decays rapidly when j-i going to infinity.

$$\underbrace{\{x_{s_0}, \ldots, x_{e_0}\}}_{\text{0-th batch}}, \underbrace{\{x_{s_1}, \ldots x_{e_1}\}}_{\text{1-st batch}}, \ldots, \underbrace{\{x_{s_M}, \ldots, x_{e_M}\}}_{M\text{-th batch}}.$$

- Once the batch sizes are predetermined, we can rewrite the batch-means estimator as

$$\frac{1}{M}\sum_{k=1}^{M} n_k (\bar{X}_{n_k} - \bar{X}_M)(\bar{X}_{n_k} - \bar{X}_M)^\top$$

$$\frac{1}{M}\sum_{k=1}^{M} n_k \bar{X}_{n_k}\bar{X}_{n_k}^\top + \frac{n}{M}\bar{X}_M\bar{X}_M^\top - 2\left(\frac{1}{M}\sum_{k=1}^{M} n_k \bar{X}_{n_k}\right)\bar{X}_M^\top$$

$$\bar{X}_{n_k} := \frac{1}{n_k}\sum_{i=s_k}^{e_k} x_i \qquad \bar{X}_M := \frac{1}{e_M - e_0}\sum_{i=s_1}^{e_M} x_i$$

Batch-mean has slower approximate rate than Plug-in

Inference

The following method is proposed based on the functional central limit theorem conclusion. [Lee et al. 21] [Li et al. 22 a]

$$\frac{1}{\sqrt{n}}\sum_{t=1}^{[nr]}(\beta_t - \beta^*) \Rightarrow \Upsilon^{1/2}W(r), \quad r \in [0,1]$$

When we take a hypothesis testing task $H_0 : R\beta^* = c$

Theorem 1. Suppose $\mathbf{rank}(R) = \ell$. Under Assumption 1 and H_0,

$$n(R\bar{\beta}_n - c)'\left(R\hat{V}_n R'\right)^{-1}(R\bar{\beta}_n - c)$$
$$\xrightarrow{d} W(1)'\left(\int_0^1 \bar{W}(r)\bar{W}(r)'dr\right)^{-1} W(1)$$

where W is an ℓ-dimensional vector of the standard Wiener processes and $\bar{W}(r) := W(r) - rW(1)$.

$$\hat{V}_n := \frac{1}{n}\sum_{s=1}^{n}\left\{\frac{1}{\sqrt{n}}\sum_{t=1}^{s}(\beta_t - \bar{\beta}_n)\right\}\left\{\frac{1}{\sqrt{n}}\sum_{t=1}^{s}(\beta_t - \bar{\beta}_n)\right\}'$$

Inference

- Compute the random scaling pivotal recursively in practice
- Similar to the batch-mean estimator's calculation, we reformulate the pivotal such that every term can be computed in an online fashion.

$$
\begin{aligned}
t^2\widehat{V}_t &= \sum_{s=1}^{t}\left(\sum_{j=1}^{s}\beta_j - s\bar{\beta}_t\right)\left(\sum_{j=1}^{s}\beta_j - s\bar{\beta}_t\right)' \\
&= \sum_{s=1}^{t}\sum_{j=1}^{s}\beta_j\sum_{j=1}^{s}\beta_j' - \bar{\beta}_t\sum_{s=1}^{t}s\sum_{j=1}^{s}\beta_j' \\
&\quad - \sum_{s=1}^{t}s\sum_{j=1}^{s}\beta_j\bar{\beta}_t' + \bar{\beta}_t\bar{\beta}_t'\sum_{s=1}^{t}s^2
\end{aligned}
$$

$$
\sum_{s=1}^{t}\sum_{j=1}^{s}\beta_j\sum_{j=1}^{s}\beta_j' = \sum_{s=1}^{t-1}\sum_{j=1}^{s}\beta_j\sum_{j=1}^{s}\beta_j' + t^2\bar{\beta}_t\bar{\beta}_t'
$$

$$
\sum_{s=1}^{t}s\sum_{j=1}^{s}\beta_j = \sum_{s=1}^{t-1}s\sum_{j=1}^{s}\beta_j + t^2\bar{\beta}_t.
$$

Stochastic Approximation with Constant Step Size

- Iterative algorithms of the form $\theta_{t+1} = \theta_t - \eta_t Z_{t+1}(\theta_t)$ to solve $z(\theta^*) = 0$
- $Z_{t+1}(\theta)$ is an unbiased random version of $z(\theta)$
- Focus on constant step size: $\eta_t = \eta$
- Examples:
  - Solving Linear Equations $A\theta = b$:

$$\theta_{t+1} = \theta_t - \eta(A_{t+1}\theta_t - b_{t+1}), \quad \mathbb{E}[A_{t+1}] = A, \ \mathbb{E}[b_{t+1}] = b$$

  - Stochastic Gradient Descent (SGD):

$$\theta_{t+1} = \theta_t - \eta \nabla f_{t+1}(\theta_t), \quad \mathbb{E}[\nabla f_{t+1}(\theta_t)] = \nabla F(\theta_t)$$

  - TD-Learning, Q-Learning, ...

Online Inference with Random Scaling

- Estimation: usually estimate θ^* with θ_T or $\bar{\theta}_T = \frac{1}{T}\sum_{t=1}^{T}\theta_t$
- **Inference**: provide a confidence interval (CI) for θ^*
- **Online**: such CI should be updated easily in each iteration
- Three steps:
  - Establishing FCLT
  - Constructing a CI via Random Scaling
  - Bounding Problem-Dependent Bias

[Xie and Z, NeurIPS 2022]

Functional Central Limit Theorem (FCLT)

- We leverage tools in Markov chains literature
- Under certain conditions, the iterates $\{\theta_t\}_{t=1}^{\infty}$ (after scaling) converges weakly to a Brownian path:

$$\frac{1}{\sqrt{n}} \sum_{s=1}^{\lfloor nt \rfloor} (\theta_s - \mathbb{E}_\pi \theta) \Rightarrow \Sigma^{1/2} B(t), \quad t \in [0,1],$$

where $\Sigma = \lim_{n\to\infty} \frac{1}{n} \mathrm{cov}_\pi(S_n)$, $S_n = \sum_{t=1}^{n} \theta_t$, and π is the limiting stationary distribution of $\{\theta_t\}_{t=1}^{\infty}$

Random Scaling

- The FCLT enables us to perform inference on $\mathbb{E}_\pi \theta$
- Let $\widehat{V}_n = \frac{1}{n} \sum_{j=1}^{n} \left\{ \frac{1}{\sqrt{n}} \sum_{i=1}^{j} (\theta_i - \bar{\theta}_n) \right\} \left\{ \frac{1}{\sqrt{n}} \sum_{i=1}^{j} (\theta_i - \bar{\theta}_n) \right\}^\top$
- For any matrix $R \in \mathbb{R}^{l \times d}$ of full row rank, we have

$$
\begin{aligned}
& n(R(\bar{\theta}_n - \mathbb{E}_\pi \theta))^\top (R \widehat{V}_n R)^{-1} (R(\bar{\theta}_n - \mathbb{E}_\pi \theta)) \\
& \xrightarrow{d} B_l(1)^\top \left(\int_0^1 (B_l(r) - r B_l(1))(B_l(r) - r B_l(1))^\top \mathrm{d}r \right)^{-1} B_l(1)
\end{aligned}
$$

- A valid CI for $\mathbb{E}_\pi \theta$ can be constructed via

$$
\mathbb{P}\left((\bar{\theta}_n)_j - q_{\frac{\alpha}{2}} \sqrt{\widehat{V}_{n,jj}/n} \leq (\mathbb{E}_\pi \theta)_j \leq (\bar{\theta}_n)_j + q_{\frac{\alpha}{2}} \sqrt{\widehat{V}_{n,jj}/n} \right) \to 1 - \alpha
$$

Problem-Dependent Bias

- For inference of θ^*, still need to bound Bias $:= \|\mathbb{E}_\pi \theta - \theta^*\|$
  - For linear stochastic approximation (LSA), Bias $= 0$
  - For nonlinear problems, usually Bias $= O(\sqrt{\eta})$ (problem-dependent)

Local SGD (Federated Averaging)

A pool of K clients, in which the k-th client has a local dataset consisting of i.i.d samples from some unknown distribution

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \sum_{k=1}^{K} p_k f_k(\boldsymbol{x}) := \sum_{k=1}^{K} p_k \mathbb{E}_{\xi_k \sim \mathcal{D}_k} f_k(\boldsymbol{x}; \xi_k)$$

Algorithm 1 Local SGD

Input: functions $\{f_k\}_{k=1}^K$, initial point $\boldsymbol{x}_0$, step size η_0, communication set $\mathcal{I} = \{t_0, t_1, \cdots\}$.
Initialization: let $\boldsymbol{x}_0^k = \boldsymbol{x}_0$ for all k.
for round $m = 0$ **to** $T-1$ **do**
  for iteration $t = t_m + 1$ **to** t_{m+1} **do**
    for each device $k = 1$ **to** K **do**
      $\boldsymbol{x}_t^k = \boldsymbol{x}_{t-1}^k - \eta_m \nabla f_k(\boldsymbol{x}_{t-1}^k; \xi_{t-1}^k)$. # perform $E_m = t_{m+1} - t_m$ steps of local updates.
    end for
    The central server aggregates: $\bar{\boldsymbol{x}}_{t_{m+1}} = \sum_{k=1}^K p_k \boldsymbol{x}_{t_{m+1}}^k$.
    Synchronization: $\boldsymbol{x}_{t_{m+1}}^k \leftarrow \bar{\boldsymbol{x}}_{t_{m+1}}$ for all k.
  end for
end for
Return: $\widehat{\boldsymbol{x}} = \frac{1}{T} \sum_{m=1}^{T} \bar{\boldsymbol{x}}_{t_m}$.

Local SGD (Federated Averaging)

Assumption 3.1 (Regularity of the objective) *For each $k \in [K]$, we assume the objective function $f_k(\cdot)$ is differentiable and strongly convex with parameter $\mu > 0$, i.e., for any $\boldsymbol{x}, \mathbf{y}$,*

$$f_k(\boldsymbol{x}) \geq f_k(\mathbf{y}) + \langle \nabla f_k(\mathbf{y}), \boldsymbol{x} - \mathbf{y} \rangle + \frac{\mu}{2} \|\boldsymbol{x} - \mathbf{y}\|^2.$$

In addition, each $f_k(\cdot)$ is L-average smooth, i.e.,

$$\sqrt{\mathbb{E}_{\xi_k} \|\nabla f_k(\boldsymbol{x}; \xi_k) - \nabla f_k(\mathbf{y}; \xi_k)\|^2} \leq L \|\boldsymbol{x} - \mathbf{y}\| \tag{3}$$

for some $L > 0$. Finally, the Hessian matrix of the global $f(\cdot)$ exists and is Lipschitz continuous in a neighborhood of the global optimal $\boldsymbol{x}^*$, i.e., there exist some $\delta_1 > 0$ and $L' > 0$ such that

$$\|\nabla^2 f(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{x}^*)\| \leq L' \|\boldsymbol{x} - \boldsymbol{x}^*\| \quad \textit{whenever} \quad \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \delta_1.$$

[Li, Liang, Chang and Z, COLT 2022]

Local SGD (Federated Averaging)

Assumption 3.2 (Regularized gradient noise) *We assume the ξ_k on different devices are independent, though they likely have different distributions. There exists some $C > 0$ such that for each $k \in [K]$,*

$$\left\| \mathbb{E}_{\xi_k}(\varepsilon_k(\boldsymbol{x})\varepsilon_k(\boldsymbol{x})^\top) - \boldsymbol{S}_k \right\| \leq C \left[\|\boldsymbol{x} - \boldsymbol{x}^*\| + \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 \right]. \tag{4}$$

Moreover, we assume there exists a constant $\delta_2 > 0$ such that $\sup_{\boldsymbol{x}} \mathbb{E}\|\varepsilon(\boldsymbol{x})\|^{2+\delta_2} < \infty$.

Assumption 3.3 (Slowly decaying effective step sizes) *Define $\gamma_m = E_m \eta_m$ as the effective step size, and assume it is non-increasing in m and satisfies* (i) $\sum_{m=1}^{\infty} \gamma_m^2 < \infty$; (ii) $\sum_{m=1}^{\infty} \gamma_m = \infty$; *and* (iii) $\frac{\gamma_m - \gamma_{m+1}}{\gamma_m} = o(\gamma_m)$.

[Li, Liang, Chang and Z, COLT 2022]

Local SGD (Federated Averaging)

Assumption 3.4 (Slowly increasing communication intervals). The sequence $\{E_m\}$ satisfies

(i) $\{E_m\}$ is either uniformly bounded or non-decreasing;

(ii) There exists some $\delta_3 > 0$ such that $\limsup_{T\to\infty} \frac{1}{T^2}\left(\sum_{m=0}^{T-1} E_m^{1+\delta_3}\right)\left(\sum_{m=0}^{T-1} E_m^{-(1+\delta_3)}\right) < \infty$;

(iii)

$$\lim_{T\to\infty} \frac{1}{T^2}\left(\sum_{m=0}^{T-1} E_m\right)\left(\sum_{m=0}^{T-1} E_m^{-1}\right) = \nu(\nu \geq 1);$$

(iv) $\lim_{T\to\infty} \frac{\sqrt{t_T}}{T} \cdot \left(\sum_{m=0}^{T} \gamma_m\right) = 0$ and $\lim_{T\to\infty} \frac{\sqrt{t_T}}{T} \frac{1}{\sqrt{\gamma_T}} = 0$ where $t_T = \sum_{m=0}^{T-1} E_m$

[Li, Liang, Chang and Z, COLT 2022]

Local SGD (Federated Averaging)

Theorem 4.2 (Functional CLT) *Let Assumptions 3.1, 3.2, 3.3 and 3.4 hold, and define*

$$h(r,T) = \max\left\{ n \in \mathbb{Z}, n > 0 \middle| r \sum_{m=1}^{T} \frac{1}{E_m} \geq \sum_{m=1}^{n} \frac{1}{E_m} \right\} \quad \textit{for} \quad r \in (0,1].$$

As $T \to \infty$, the following random function weakly converges to a scaled Brownian motion, i.e.,

$$\phi_T(r) := \frac{\sqrt{t_T}}{T} \sum_{m=1}^{h(r,T)} (\bar{\boldsymbol{x}}_{t_m} - \boldsymbol{x}^*) \Rightarrow \sqrt{\nu} \boldsymbol{G}^{-1} \boldsymbol{S}^{1/2} \mathbf{B}_d(r)$$

where $t_T = \sum_{m=0}^{T-1} E_m$, $\bar{\boldsymbol{x}}_{t_m} = \sum_{k=1}^{K} p_k \boldsymbol{x}_{t_m}^k$, and $\mathbf{B}_d(\cdot)$ is the d-dim standard Brownian motion.

[Li, Liang, Chang and Z, COLT 2022]

Local SGD (Federated Averaging)

Corollary 4.2 *Under the same assumptions of Theorem 4.2 and assuming* $g(r_m) \asymp \frac{m}{T}$ *for some continuous function* g *on* $[0,1]$ *, we have that*

$$\phi_T(1)^\top \Pi_T^{-1} \phi_T(1) \xrightarrow{d} \mathbf{B}_d(1)^\top \left[\int_0^1 (\mathbf{B}_d(r) - g(r)\mathbf{B}_d(1)) (\mathbf{B}_d(r) - g(r)\mathbf{B}_d(1))^\top dr \right]^{-1} \mathbf{B}_d(1).$$

This corollary follows immediately from Theorem 4.2 and the continuous mapping theorem. It implies $\phi_T(1)^\top \Pi_T^{-1} \phi_T(1)$ is asymptotically pivotal and thus can be used to construct valid asymptotic confidence intervals. Up to a constant factor, studentizing $\phi_T(1)$ via Π_T is equivalent to studentizing $\bar{\boldsymbol{y}}_T = \frac{1}{T}\sum_{m=1}^T \bar{\boldsymbol{x}}_{t_m}$ via $\widehat{\boldsymbol{V}}_T$ where

$$\widehat{\boldsymbol{V}}_T = \frac{1}{T^2 \sum_{m=1}^T \frac{1}{E_m}} \sum_{m=1}^T \frac{1}{E_m} \left(\sum_{n=1}^m \bar{\boldsymbol{x}}_{t_n} - m\bar{\boldsymbol{y}}_T \right) \left(\sum_{n=1}^m \bar{\boldsymbol{x}}_{t_n} - m\bar{\boldsymbol{y}}_T \right)^\top.$$

[Li, Liang, Chang and Z, COLT 2022]

Local SGD (Federated Averaging)

Corollary 4.3 *Under the same conditions of Corollary 4.2, we have that*

$$\mathbb{P}\left(\left[\bar{\boldsymbol{y}}_{T,j} - q_{\frac{\alpha}{2},g}\sqrt{\widehat{\boldsymbol{V}}_{T,jj}} \le \boldsymbol{x}_j^* \le \bar{\boldsymbol{y}}_{T,j} + q_{\frac{\alpha}{2},g}\sqrt{\widehat{\boldsymbol{V}}_{T,jj}}\right]\right) \to 1-\alpha,$$

where $q_{\frac{\alpha}{2},g}$ *is* $(1-\alpha/2)$*-quantile of the following random variable*

$$B_1(1) \Big/ \left(\int_0^1 (B_1(r) - g(r)B_1(1))^2 dr\right)^{1/2} \tag{5}$$

with $B_1(\cdot)$ *a one-dimensional standard Brownian motion.*

[Li, Liang, Chang and Z, COLT 2022]

Synchronous Q-Learning

Q-learning. The synchronous Q-learning maintains a Q-function vector, $\boldsymbol{Q}_t \in \mathbb{R}^D$, for all $t \geq 0$ and updates its entries via the following update rule:

$$\boldsymbol{Q}_t = (1 - \eta_t)\boldsymbol{Q}_{t-1} + \eta_t(\boldsymbol{r}_t + \widehat{\mathcal{T}}_t \boldsymbol{Q}_{t-1}) \tag{3}$$

where $\eta_t \in (0, 1]$ is the step size in the t-th iteration and $\widehat{\mathcal{T}}_t : \mathbb{R}^D \to \mathbb{R}^D$ is the empirical Bellman operator constructed by samples collected in the t-th iteration:

$$(\widehat{\mathcal{T}}_t \boldsymbol{Q})(s, a) = r_t(s, a) + \gamma \max_{a' \in \mathcal{A}} \boldsymbol{Q}(s_t, a'), \tag{4}$$

with $r_t(s, a) \sim R(s, a)$ and $s_t = s_t(s, a) \sim P(\cdot|s, a)$ for each state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$. In matrix form, $\widehat{\mathcal{T}}_t \boldsymbol{Q}_{t-1} = \boldsymbol{P}_t \boldsymbol{V}_{t-1}$ where $\boldsymbol{V}_{t-1}(s) = \max_a \boldsymbol{Q}_{t-1}(s, a)$ is

Synchronous Q-Learning

Assumption 3.1. *We assume* $\mathbb{E}|R(s,a)|^4 < \infty$ *for all* $(s,a) \in \mathcal{S} \times \mathcal{A}$.

Assumption 3.2. *There exists* $\pi^* \in \Pi^*$ *such that for any Q-function estimator* $\boldsymbol{Q} \in \mathbb{R}^D$, $\|(\boldsymbol{P}^{\pi_{\boldsymbol{Q}}} - \boldsymbol{P}^{\pi^*})(\boldsymbol{Q} - \boldsymbol{Q}^*)\|_\infty \leq L\|\boldsymbol{Q} - \boldsymbol{Q}^*\|_\infty^2$ *where* $\pi_Q(s) := \arg\max_{a \in \mathcal{A}} Q(s,a)$ *is the greedy policy w.r.t.* Q.

Assumption 3.3. *Assume* (i) $0 \leq \sup_t \eta_t \leq 1, \eta_t \downarrow 0$ *and* $t\eta_t \uparrow \infty$; (ii) $\frac{\eta_{t-1} - \eta_t}{\eta_{t-1}} = o(\eta_{t-1})$; (iii) $\frac{1}{\sqrt{T}} \sum_{t=0}^{T} \eta_t \to 0$ *for all* $t \geq 1$; (iv) $\frac{\sum_{t=0}^{T} \eta_t}{T\eta_T} \leq C$ *for all* $T \geq 1$.

We now present the FCLT for averaged Q-learning under the same conditions. Define the standardized partial-sum processes associated with $\{\boldsymbol{Q}_t\}_{t \geq 0}$ as follows:

$$\boldsymbol{\phi}_T(r) := \frac{1}{\sqrt{T}} \sum_{t=1}^{\lfloor Tr \rfloor} (\boldsymbol{Q}_t - \boldsymbol{Q}^*).$$

FCLT

Theorem 3.1. *Under Assumptions 3.1, 3.2 and 3.3, we have*

$$\boldsymbol{\phi}_T(\cdot) \xrightarrow{w} \mathrm{Var}_{\boldsymbol{Q}}^{1/2} \boldsymbol{B}_D(\cdot), \tag{8}$$

where $\mathrm{Var}_{\boldsymbol{Q}} \in \mathbb{R}^{D \times D}$ *is the asymptotic variance*

$$\mathrm{Var}_{\boldsymbol{Q}} = (\boldsymbol{I} - \gamma \boldsymbol{P}^{\pi^*})^{-1} \mathrm{Var}(\boldsymbol{Z}) (\boldsymbol{I} - \gamma \boldsymbol{P}^{\pi^*})^{-\top} \tag{9}$$

and $\boldsymbol{B}_D(\cdot) \in \mathbb{R}^D$ *is the standard Brownian motion on* $[0, 1]$.

[Li, Yang, Liang, Z and Jordan, 2022]

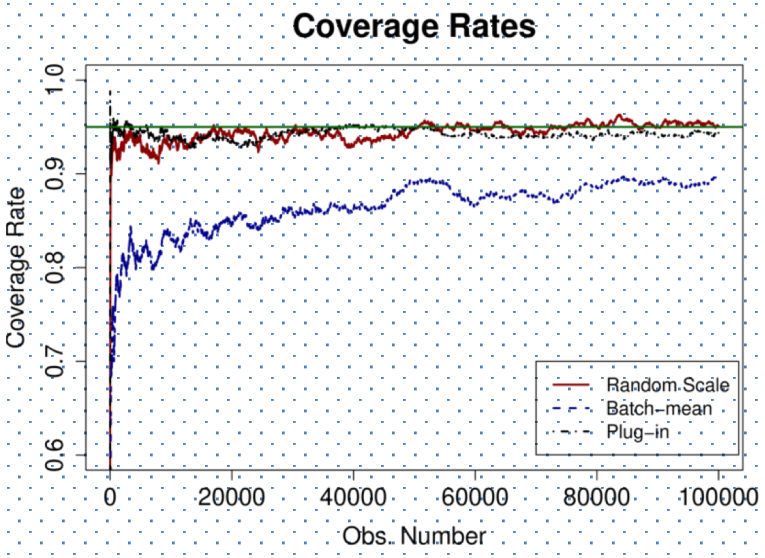
Online Statisticsl Inference

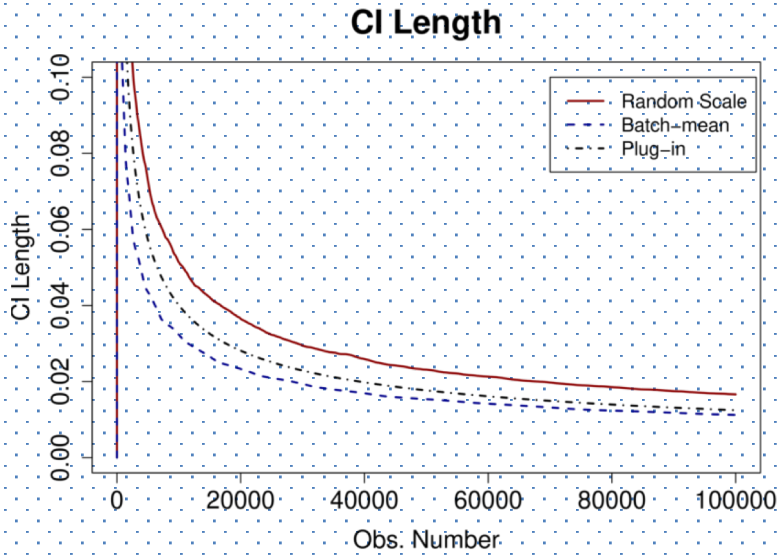
Proposition 3.1. *The continuous mapping theorem together with Theorem 3.1 yields that with probability approaching one,* $\int_0^1 \boldsymbol{\phi}_T(r)\boldsymbol{\phi}_T(r)^\top dr$ *is invertible and*

$$\begin{aligned} &\boldsymbol{\phi}_T(1)^\top \left(\int_0^1 \boldsymbol{\phi}_T(r)\boldsymbol{\phi}_T(r)^\top dr \right)^{-1} \boldsymbol{\phi}_T(1) \\ &\xrightarrow{d} \boldsymbol{B}_D(1)^\top \left(\int_0^1 \boldsymbol{B}_D(r)\boldsymbol{B}_D(r)^\top dr \right)^{-1} \boldsymbol{B}_D(1). \end{aligned} \tag{10}$$

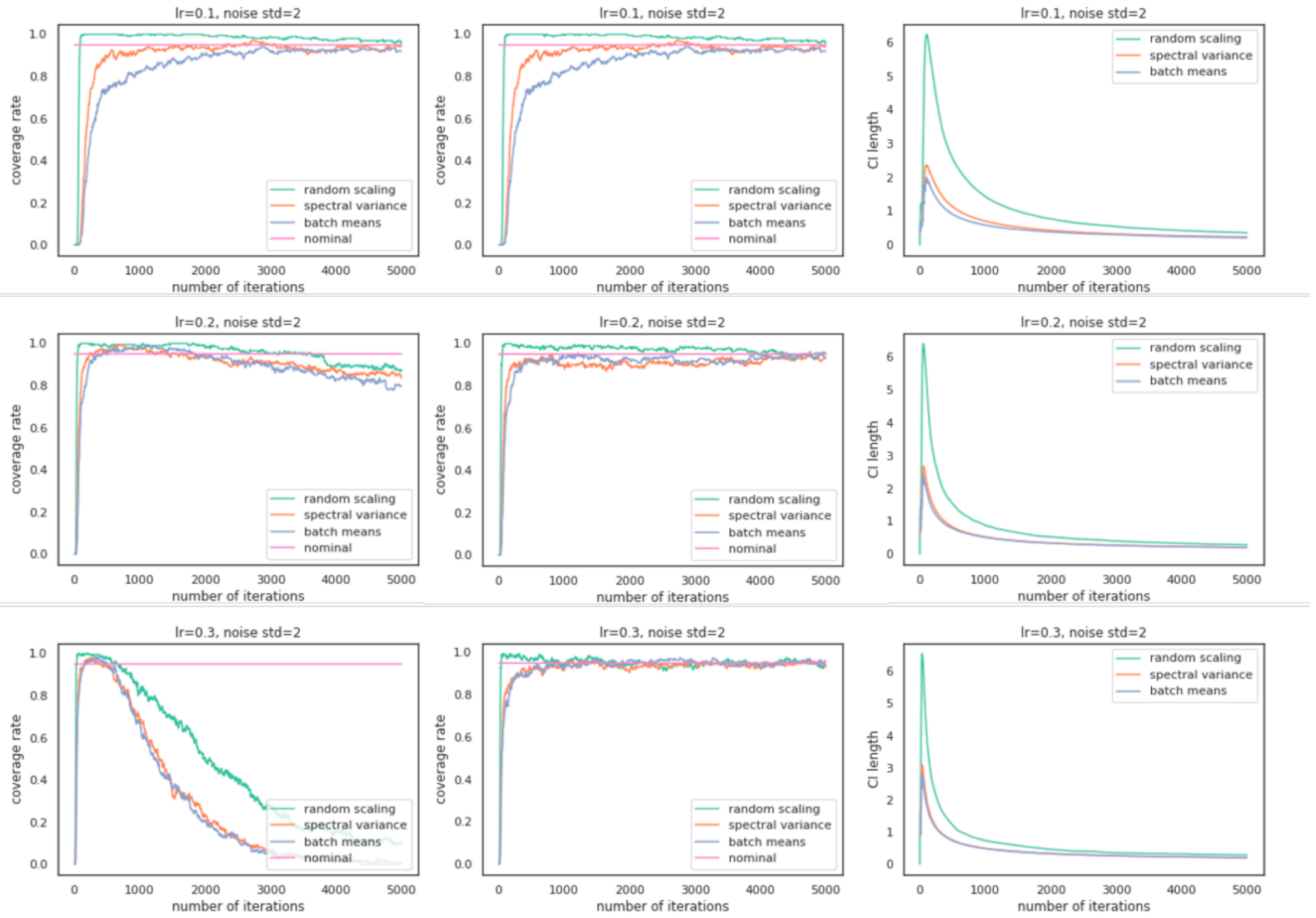
The left-hand side of (10) is a pivotal quantity involving samples and the unobservable parameter of interest $\boldsymbol{Q}^*$.

Local SGD


Coverage Rates
Coverage Rate
0.6
0.7
0.8
0.9
1.0
0
20000
40000
60000
80000
100000
Obs. Number
Random Scale
Batch−mean
Plug−in


CI Length
CI Length
0.00
0.02
0.04
0.06
0.08
0.10
0
20000
40000
60000
80000
100000
Obs. Number
Random Scale
Batch−mean
Plug−in

Q-Learning


lr=0.1, noise std=2
coverage rate
number of iterations
random scaling
spectral variance
batch means
nominal
lr=0.1, noise std=2
coverage rate
number of iterations
random scaling
spectral variance
batch means
nominal
lr=0.1, noise std=2
CI length
number of iterations
random scaling
spectral variance
batch means
lr=0.2, noise std=2
coverage rate
number of iterations
random scaling
spectral variance
batch means
nominal
lr=0.2, noise std=2
coverage rate
number of iterations
random scaling
spectral variance
batch means
nominal
lr=0.2, noise std=2
CI length
number of iterations
random scaling
spectral variance
batch means
lr=0.3, noise std=2
coverage rate
number of iterations
random scaling
spectral variance
batch means
nominal
lr=0.3, noise std=2
coverage rate
number of iterations
random scaling
spectral variance
batch means
nominal
lr=0.3, noise std=2
CI length
number of iterations
random scaling
spectral variance
batch means

Future Work

- More general environment
  - State dependent noise
  - Non-convex / Non-smooth
- Other statistical task
  - Semi-parameter / Non-parameter
  - Large deviation
- Bias / Variance trade-off (Non-asymptotic / Asymptotic trade-off)
- Unifying various stochastic optimization (approximation) algorithms

- 具有组合结构问题
- 在线统计推断案例
- 一些潜在研究方向

不确定性下的序贯决策

- 不确定性下的序贯决策是多个领域的基础问题，比如经济学和运筹学中的马尔可夫决策过程(MDP)、工程中的最优控制、计算机科学中的在线算法等。虽然MDP、最优控制和在线算法在各自的领域都取得了巨大的成功，但今天这三个领域正日益融合。这背后的一个核心原因是，随着数据变得更容易获得和技术不断改进，它允许针对不同设置对算法进行更大程度的定制。
- 因此，大家转向更灵活的“数据驱动决策过程”模型：即结合随机和对抗信息结构的混合模型，以及更好地适应信息、约束和目标结构的算法 。

多臂老虎机算法

- 多臂老虎机是经典的在线学习问题。传统的多臂老虎机问题主要分为随机情景、对抗情景两种模式。
- 在当前数据驱动的决策问题中，实际情况下可以建模成多臂老虎机的决策问题往往既不属于随机情景，也不属于对抗情景。而是属于部分数据对抗。
- 如何在不失鲁棒性的情况下保证多臂老虎机算法仍旧能够高效的确认最大奖励均值的臂，需要在算法设计上引入适当的改进。

强化学习

- 在强化学习场景中，策略梯度算法是最经典的算法框架，它利用价值函数的梯度估计和随机梯度算法来估计最优策略。由于强化学习问题中的目标函数对于策略参数并不具备凸性，策略梯度类方法的收敛性分析并不平凡，需要利用价值函数这一重要工具。由于策略梯度的计算需要依赖当前策略产生的轨道信息，该类强化学习方法面临着采样复杂度高、收敛速度慢的问题。
- 通过重要性采样和约束优化过程中相邻策略之间距离等技巧，经过改进的策略梯度方法能够复用非当前策略的轨道信息进行策略更新。与此同时，受传统优化中方差减小方法的启发，方差减小的策略梯度方法得以提出并取得了更好的理论收敛结果。

采样和优化

- 采样和优化在算法中是密切联系的两个部分。一方面我们可以用采样的观点来研究随机梯度下降算法的收敛性，依据朗之万方程在每一步引入合适的噪音来确保算法可以逃离局部鞍点,期待更好的泛化结果。
- 另一方面当我们研究采样算法时,也可以将朗之万蒙特卡洛方法看作是在概率空间中的梯度下降算法,可以利用优化中的一些加速方法来设计效率更高的采样算法。
- 凸离散优化中这种联系同样值得研究。最近凸离散优化中的随机局部化方法和重要结构拟阵的高效采样算法得到了充分的发展。基于采样算法设计更高效的离散优化算法,并将这些算法应用在组合优化,编码理论和经济学等领域同样值得期待。

谢谢大家！！！

感谢邀请！！！