SCIENCE CHINA
Information Sciences

• LETTER •

September 2020, Vol. 63 199203:1–199203:3
https://doi.org/10.1007/s11432-018-9731-1

Distributed gradient-based sampling algorithm for least-squares in switching multi-agent networks

Peng LIN[1,2] & Hongsheng QI[1,2*]

[1]*Key Laboratory of Systems and Control, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing* 100190*, China;*
[2]*School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing* 100049*, China*

Received 9 September 2018/Accepted 12 December 2018/Published online 26 March 2020

Citation Lin P, Qi H S. Distributed gradient-based sampling algorithm for least-squares in switching multi-agent networks. Sci China Inf Sci, 2020, 63(9): 199203, https://doi.org/10.1007/s11432-018-9731-1

Dear editor,

Recent years have witnessed a rapid growth of distributed design in multi-agent networks because of the scalability, robustness and low cost. Compared with the conventional centralized and parallel design, all agents in fully distributed design aim to achieve the global goal only based on the local measurement and information sharing with neighbors. Therefore, the distributed algorithms have been more and more popular in many areas, including economical systems, smart grids and machine learning [1–3].

Machine learning has attracted more and more research attention in recent years owing to various applications in data mining, pattern recognition, and knowledge discovery [4]. The most classical regression model, belonging to the supervised machine learning methods, is the least-squares, which aims to minimize the summation of the squared residuals to find the underlying rules between regression vectors and corresponding responses [5].

Note that many least-squares methods are basically centralized, and can process the whole dataset without active data selection. Unfortunately, the challenges with rapid growth of data and large scale networks appeal more effective algorithms to save the limited computational resources and process data over the networks [6]. To deal with the large size of data, random sampling is one of major methods, which only uses a small subset of data for the model fitting and inference. The gradient-based sampling algorithm for least-squares proposed in [7], different from uniform sampling and leverage-based sampling, takes both input vectors and response values into consideration. The sampling probability is proportional to the norm of gradient, which needs less computational resources. To deal with the circumstance that data are distributed over the networks, some fully distributed algorithms have been investigated [8, 9]. Unfortunately, if the local datasets are too large to process, these algorithms will fail to work, without considering the computational ability of each agents. Therefore, designing distributed algorithms with sampling becomes an urgent task for effective distributed large-size data processing.

Motivated by the above analysis, we study the distributed design with sampling for least-squares. By combining the idea of distributed subgradient method presented in [8] with the gradient-based sampling idea in [7], the distributed gradient-based sampling algorithm (DGSA) is proposed. As far as we know, DGSA is the first distributed algorithm that adapts the sampling idea to the circumstance that the computational ability of each agent is limited. Furthermore, the communication network in DGSA is assumed to be time-varying, which is more suitable to the complicated situations in reality. Both theoretical and empirical analyses are given to illustrate the effectiveness of DGSA.

Preliminaries. Consider a network with N

* Corresponding author (email: qihongsh@amss.ac.cn)

© Science China Press and Springer-Verlag GmbH Germany, part of Springer Nature 2020

Downloaded to IP: 10.159.164.174 On: 2020-04-15 08:44:59 http://engine.scichina.com/doi/10.1007/s11432-018-9731-1

agents, which is modeled by the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ represent the set of agents and edges, respectively. If agent i can receive information from agent j directly, then there exists a directed edge from j to i, which is denoted by $(j, i) \in \mathcal{E}$. A graph $\mathcal{G}$ is strongly connected if there is a directed path between any pair agents i, $j \in \mathcal{V}$. The time-varying network can be modeled by $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$. $A(t) = [a_{i,j}(t)] \in \mathbb{R}^{N\times N}$ is the adjacency matrix, where $a_{i,j}(t) > 0$ for any $(j, i) \in \mathcal{E}(t)$ and $a_{i,i}(t) > 0$. Otherwise, $a_{i,j}(t) = 0$.

Assumption 1. The graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ with its weighted adjacency matrix $A(t)$ is jointly connected; that means

- $A(t)$ is doubly stochastic;
- For all $i \in \mathcal{V}, a_{i,i}(t) \geqslant \epsilon$ and $a_{i,j}(t) \geqslant \epsilon$ if $(j, i) \in \mathcal{E}(t)$, where ϵ is a positive scalar;
- The graph $(\mathcal{V}, \mathcal{E}(t)\cup\mathcal{E}(t+1)\cup\cdots\cup\mathcal{E}(t+T-1))$ is strongly connected for all $t \geqslant 0$, where $T > 0$ is an integer.

Obviously, Assumption 1 guarantees that each agent can receive information from all its neighbors at least one time during each period of T.

Problem formulation. For every agent $i \in \mathcal{V}$, a training set $D_i = \{(x_i^k, y_i^k)\}_{k=1}^{m_i}$ is available, where $x_i^k \in \mathcal{X}$ is the input vector belonging to the input space $\mathcal{X} \subseteq \mathbb{R}^d$, $y_i^k \in \mathbb{R}$ is the response value, and m_i is the number of the data points. $\mathcal{D} = \{D_i\}_{i=1}^N$ is the whole dataset. The least-squares problem is to minimize the sample risk function of the parameters ω as follows:

$$\sum_{i=1}^{N} f_i(X_i, Y_i, \omega) = \sum_{i=1}^{N} \frac{1}{2} \sum_{k=1}^{m_i} (y_i^k - \omega^{\mathrm{T}} x_i^k)^2, \quad (1)$$

where $X_i = (x_i^1, x_i^2, \ldots, x_i^{m_i})^{\mathrm{T}} \in \mathbb{R}^{m_i\times d}$ and $Y_i = (y_i^1, y_i^2, \ldots, y_i^{m_i})^{\mathrm{T}} \in \mathbb{R}^{m_i}$ represent the local input matrix and response vector, respectively. The global input matrix and response vector are defined by $X = (X_1^{\mathrm{T}}, X_2^{\mathrm{T}}, \ldots, X_N^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{M\times d}$ and $Y = (Y_1^{\mathrm{T}}, Y_2^{\mathrm{T}} \ldots, Y_N^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^M$ separately, where $M = \sum_{i=1}^N m_i$ is the total number of data points. The solution of the problem (1) has the form of

$$\omega^* = (M^{-1}X^{\mathrm{T}}X)^{-1}(M^{-1}X^{\mathrm{T}}Y) = \Omega_M^{-1} b_M, \quad (2)$$

where $\Omega_M = M^{-1}X^{\mathrm{T}}X$ and $b_M = M^{-1}X^{\mathrm{T}}Y$.

When the size of input data is very large or distributed stored, Eq. (2) cannot be obtained in just one central agent in fact. Therefore, some distributed algorithms have been proposed [8]. Unfortunately, if the size of local data in every agent is also very large, which means $m_i \gg d$, the distributed algorithms cannot perform normally either.

To reduce the computational cost in each agent, sampling method is a good choice. The sampling probabilities in all agents are denoted by $\{\beta_i\}_{i=1}^N$, where $\beta_i = (\beta_i^1, \beta_i^2, \ldots, \beta_i^{m_i})^{\mathrm{T}}$, such that $\sum_{i=1}^N \sum_{k=1}^{m_i} \beta_i^k = 1$. According to the sampling probabilities, subsample set $\mathcal{D}_s = \{D_i^s\}_{i=1}^N$ are randomly chosen, and problem (1) is transformed to the following weighted least squares function:

$$\sum_{i=1}^{N} \tilde{f}_i(X_i^s, Y_i^s, \beta_i^s, \omega) = \sum_{i=1}^{N} \sum_{k\in D_i^s} \frac{1}{2\beta_i^k} (y_i^k - \omega^{\mathrm{T}} x_i^k)^2. \quad (3)$$

The solution $\tilde{\omega}$ of (3) has the following form: $\tilde{\omega} = (M^{-1}X_s^{\mathrm{T}}\Phi_s^{-1}X_s)^{-1}(M^{-1}X_s^{\mathrm{T}}\Phi_s^{-1}Y_s) = \Omega_s^{-1}b_s$, where Φ_s is the partition of $\Phi = \mathrm{diag}\{M_s\beta_i^k\}$, and $M_s = \sum_{i=1}^N m_i^s$ is the size of D_s.

The gradient-based sampling methods assume the data point that has large gradient is more important to detect the optimal solution. That means $\beta_i^k \propto ||g_i^k||$, where $g_i^k = x_i^k(y_i^k - \omega_0^{\mathrm{T}} x_i^k)$, and ω_0 is a pilot estimate for the ω. The pilot estimate ω_0 can be obtained by a good guess or from an initial subsample of size M_{s_0} by uniform sampling [7].

However, in fully distributed systems, the data are collected and stored spatially in different agents. Only based on the local data set $\mathcal{D}_i$, agents cannot obtain the sampling probability β_i^k, because they have no knowledge of other gradients in other agents. Therefore, it calls for a new distributed algorithm, which is based on the gradient sampling, to reduce the computational cost obviously.

Distributed gradient-based sampling algorithm. To deal with the problem (1) using the distributed sampling approach, our DGSA algorithm consists of two steps: the sampling step and the optimization step.

In the sampling step, the sampling probabilities for all data points should be determined. To make sure that $\sum_{i=1}^N \sum_{k=1}^{m_i} \beta_i^k = 1$, all agents need the total norm of gradients $\sum_{i=1}^N \sum_{k=1}^{m_i} ||g_i^k||$.

To help all agents to obtain this value, let $\{z_i(t)\}_{i=1}^N$ denote the local variables, and $z_i(0) = \sum_{k=1}^{m_i} ||g_i^k||$, which can be calculated by agent i itself. Then agent i communicates with its one-hop neighbor j according to the following equation until consensus:

$$z_i(t+1) = \sum_{j\in\mathcal{N}_i(t)} a_{i,j}(t) z_j(t). \quad (4)$$

Eventually, $z_i(t)$ converges to the average of the initial states $\bar{z}(0) = \frac{\sum_{i=1}^N \sum_{k=1}^{m_i} ||g_i^k||}{N}$. Therefore, β_i^k can be obtained according to $\beta_i^k = \frac{||g_i^k||}{N z_i(T)}$, where T is the stop time of (4). After the sampling probabilities are obtained, the poisson sampling is applied to determine $\{D_i^s\}_{i=1}^N$. The independent random variable $s_i^k \sim \mathrm{Bernoulli}(1, p_i^k)$ is generated,

Downloaded to IP: 10.159.164.174 On: 2020-04-15 08:44:59 http://engine.scichina.com/doi/10.1007/s11432-018-9731-1

where $p_i^k = M_s\beta_i^k$. The local sampling data set D_i^s consists of the data points according to the set $\{k : s_i^k = 1\}$, which means if $s_i^k = 1$, the k-th data point in agent i is chosen.

In the optimization step, because the sampling probabilities $\{\beta_i^k\}$ are fixed, the goal of all agents is to optimize the weighted loss function (3). To solve this problem, the distributed gradient-based algorithm is proposed as follows:

$$\omega_i(t+1) = P_\Gamma\left[\sum_{j=1}^N a_{i,j}(t)\omega_j(t) - \alpha(t)d_i(t)\right], \quad (5)$$

where $d_i(t)$ is the subgradient of $\tilde{f}_i(X_i^s, Y_i^s, \beta_i^s, \omega)$ at $\omega = \sum_{j=1}^N a_{i,j}(t)\omega_j(t)$ and $\Gamma = \bigcap_{i=1}^N \Gamma_i$ is the common domain of ω_i, which is always assumed to be non-empty and the optimal solution is contained in it. Γ_i is a convex and compact domain of ω_i, which is only known by agent i. The step-size $\alpha(t)$ satisfies the stochastic approximation conditions:

$$\begin{cases}\alpha(t) > 0, & \lim_{t\to\infty}\alpha(t) = 0,\\ \sum_{t=1}^\infty \alpha(t) = \infty, & \sum_{t=1}^\infty \alpha^2(t) < \infty.\end{cases} \quad (6)$$

Through this distributed algorithm, a good estimation $\hat{\omega}_i$ of $\tilde{\omega}$ can be obtained by agent i. Therefore, all agents can achieve a good estimation $\hat{\omega}$ of ω^* according to the above two steps. The whole procedure of our DGSA is summarized in Appendix A.

Here, we give a theorem to illustrate that the optimal solution of problem (3), which is determined by the sampling step, is actually a good estimate of the optimal solution of problem (1). For simplicity, we have the following notations:

$$\begin{cases}R_x = \max_{i,k}||x_i^k||^2,\\ R_b^2 = M^{-2}\sum_{i=1}^N\sum_{k=1}^{m_i}(\beta_i^k)^{-1}||\epsilon_i^k x_i^k||^2,\\ R_\Sigma = M^{-2}\sum_{i=1}^N\sum_{k=1}^{m_i}(\beta_i^k)^{-1}||x_i^k||^4,\\ \epsilon_i^k = y_i^k - \omega^{*\mathrm{T}}x_i^k.\end{cases}$$

Theorem 1. For any $\delta > \frac{2R_x\log d}{3M\lambda_{\min}(\Omega_M)}$, if M_s satisfies $M_s > \frac{72R_\Sigma^2M^2\log d}{(3M\delta\lambda_{\min}(\Omega_M)-2R_x\log d)^2}$, it is obtained that $\mathbb{P}(||\tilde{\omega}-\omega^*|| \leqslant C_M M_s^{-1/2}) \geqslant 1-\delta$, where $C_M = 3\lambda_{\min}(\Omega_M)\delta^{-1}R_b$.

The following theorem shows that all agents can achieve the optimal solution of problem (3) in the optimization step.

Theorem 2. In the optimization step in DGSA, if Assumption 1 and the step-size condition (6) hold, then $\lim_{t\to\infty}||\omega_i(t)-\tilde{\omega}|| = 0$ for all $i = 1, 2, \ldots, N$.

Therefore, through our DGSA algorithm, good estimates of the optimal solution of problem (1) can be obtained by all agents.

The proofs of Theorems 1 and 2 are included in Appendixes B and C, respectively. Simulations on both synthetic datasets and real datasets are given to evaluate the performance of our DGSA algorithm, which can be found in Appendix D.

Conclusion. In this study, we discussed a distributed design for the least-squares problem with sampling in a time-varying multi-agent network. To solve the problem and deal with the large local data size, a distributed gradient-based sampling algorithm was proposed. Moreover, theoretical analysis and simulations on synthetic and real datasets were presented to demonstrate the effectiveness of the proposed algorithm. There are still many challenging problems related to distributed optimization without considering the limited ability of every agent, which are still under our investigation.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2018YFA0703800) and National Natural Science Foundation of China (Grant Nos. 61873262, 61733018, 61333001).

Supporting information Appendixes A–D. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Nedic A, Ozdaglar A. Distributed subgradient methods for multi-agent optimization. IEEE Trans Autom Contr, 2009, 54: 48–61
2 Lou Y C, Hong Y G, Wang S Y. Distributed continuous-time approximate projection protocols for shortest distance optimization problems. Automatica, 2016, 69: 289–297
3 Yi P, Hong Y G, Liu F. Initialization-free distributed algorithms for optimal resource allocation with feasibility constraints and application to economic dispatch of power systems. Automatica, 2016, 74: 259–269
4 Bishop C M. Pattern Recognition and Machine Learning. New York: Springer, 2006
5 Wang Y H, Lin P, Hong Y G. Distributed regression estimation with incomplete data in multi-agent networks. Sci China Inf Sci, 2018, 61: 092202
6 Sagiroglu S, Sinanc D. Big data: a review. In: Proceedings of International Conference on Collaboration Technologies and Systems, California, 2013. 42–47
7 Zhu R. Gradient-based sampling: an adaptive importance sampling for least-squares. 2016. ArXiv: 1803.00841
8 Nedic A, Ozdaglar A, Parrilo P A. Constrained consensus and optimization in multi-agent networks. IEEE Trans Autom Contr, 2010, 55: 922–938
9 Gholami M R, Jansson M, Strom E G, et al. Diffusion estimation over cooperative multi-agent networks with missing data. IEEE Trans Signal Inf Process Netw, 2016, 2: 276–289

Downloaded to IP: 10.159.164.174 On: 2020-04-15 08:44:59 http://engine.scichina.com/doi/10.1007/s11432-018-9731-1