Math. Program., Ser. A (2016) 156:59–99
DOI 10.1007/s10107-015-0871-8

CrossMark

FULL LENGTH PAPER

Accelerated gradient methods for nonconvex nonlinear and stochastic programming

Saeed Ghadimi · Guanghui Lan

【优化】Accelerated Gradient

张楚珩
清华大学 交叉信息院博士在读

加速的梯度下降算法。

原文传送门

Ghadimi, Saeed, and Guanghui Lan. "Accelerated gradient methods for nonconvex nonlinear and stochastic programming." Mathematical Programming 156.1-2 (2016): 59-99.

Beck, Amir. First-order methods in optimization. Vol. 25. SIAM, 2017. 【Book】

特色

强化学习问题其实质就是一个优化问题；策略梯度方法就是一种梯度上升算法。这里罗列一些梯度上升/下降算法的结论。

正文

一、和强化学习的联系

首先，如果把强化学习问题看做优化问题的话，一般规定目标函数 $J(\pi_\theta) = V^{\pi_\theta}(\rho)$ 。那么该目标函数具有什么样的性质呢？即，该优化问题属于什么类型？

- 该目标函数是非凸的（nonconvex）
- 可以认为该目标函数是平滑的（smooth）
- 强化学习本身不带来局部极小值，但是策略的参数化方法可能带来局部极小值（single-mode，我瞎起的）
- 不考虑策略参数化方法的问题，一般梯度严格等于零时有全局最优（gradient dominance）
- 它还具有一些更为『好』的性质（other regularities）

Motivation

- $\max_\theta J(\pi_\theta)$ is a nonconvex problem
  - Obviously, $J(\pi) = (1 - \gamma P^\pi) r_\pi$ is not linear. If it is convex, let $r \to -r$, then we have another valid RL problem that is concave. Therefore, it is nonconvex.
- "Single-mode": exact zero gradient implies optimality
- More regularities, e.g., [1]

Motivation

- We can assume $J(\pi_\theta)$ is smooth [2]
  - Direct parametrization (tabular case):

$$\left\| \nabla_\pi V^\pi(s_0) - \nabla_\pi V^{\pi'}(s_0) \right\|_2 \leq \frac{2\gamma|\mathcal{A}|}{(1-\gamma)^3} \|\pi - \pi'\|_2$$

  - Softmax parametrization with relative entropy regularization:

$$L_\lambda(\theta) = V^{\pi_\theta}(\mu) + \frac{\lambda}{|\mathcal{S}|\,|\mathcal{A}|} \sum_{s,a} \log \pi_\theta(a|s) \qquad \|\nabla_\theta L_\lambda(\theta) - \nabla_\theta L_\lambda(\theta')\|_2 \leq \beta_\lambda \|\theta - \theta'\|_2 \qquad \beta_\lambda = \frac{8}{(1-\gamma)^3} + \frac{2\lambda}{|\mathcal{S}|}$$

  - Other Lipschitz and smooth parameterizations:

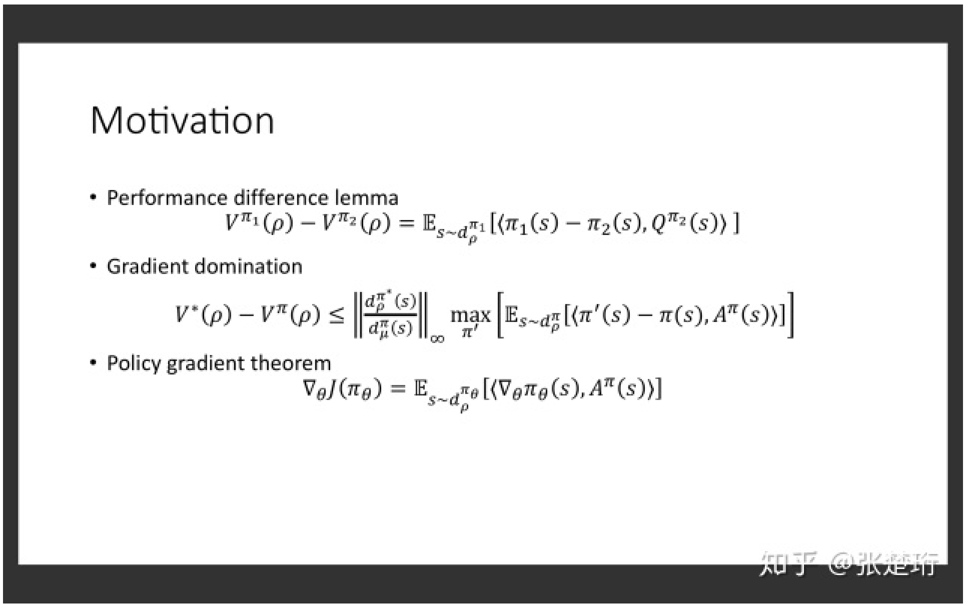
$$\begin{aligned} |\pi_\theta(a|s) - \pi_{\theta'}(a|s)| &\leq \beta_1 \|\theta - \theta'\|_2 && (\beta_1\text{-Lipschitz}) \\ \|\nabla_\theta \pi_\theta(a|s) - \nabla_\theta \pi_{\theta'}(a|s)\|_2 &\leq \beta_2 \|\theta - \theta'\|_2 && (\beta_2\text{-smooth}) \end{aligned} \qquad \|\nabla_\theta V^{\pi_\theta}(s_0) - \nabla_\theta V^{\pi_{\theta'}}(s_0)\|_2 \leq \beta \|\theta - \theta'\|_2 \qquad \beta = \frac{\beta_2|\mathcal{A}|}{(1-\gamma)^2} + \frac{2\gamma\beta_1^2|\mathcal{A}|^2}{(1-\gamma)^3}$$

Motivation

- Policy gradient methods = gradient ascent methods for smooth, nonconvex optimization problems
- Nonconvex (and possibly stochastic) gradient ascent ensures that the gradient norm decays well w.r.t. #iterations/#samples. [5]
- However, this does not imply fast performance improvement due to distribution mismatch. The gradient domination shows this, e.g., for direct parametrization (other parametrizations are more involved, but the idea is similar)

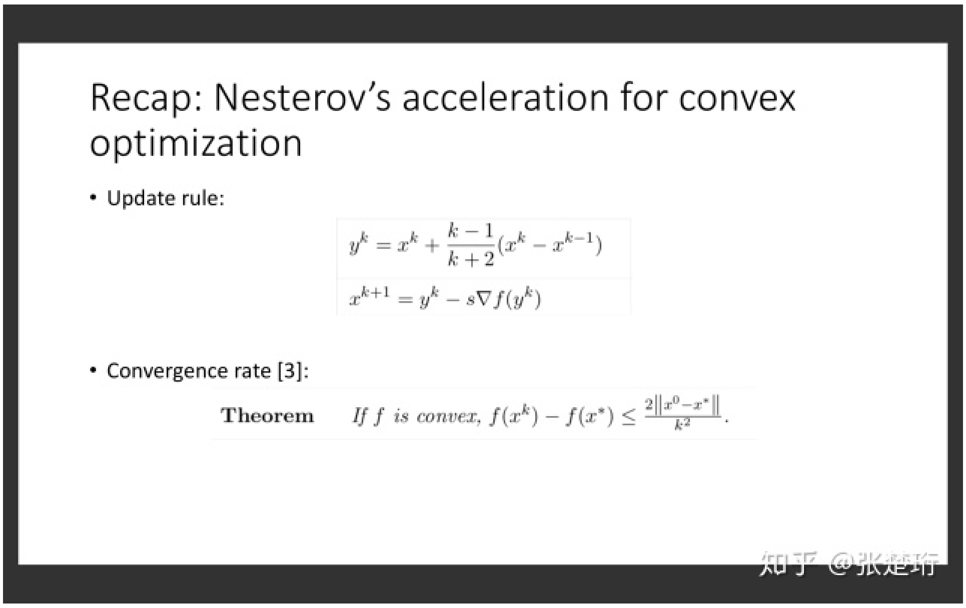
$$V^{*}(\rho) - V^{\pi}(\rho) \leq \left\| \frac{d_{\rho}^{\pi^{*}}}{d_{\mu}^{\pi}} \right\|_{\infty} \max_{\bar{\pi}} \, (\bar{\pi} - \pi)^{\top} \nabla_{\pi} V^{\pi}(\mu)$$

- This is exploration problem.



二、回顾：Nesterov's Accelerated Gradient

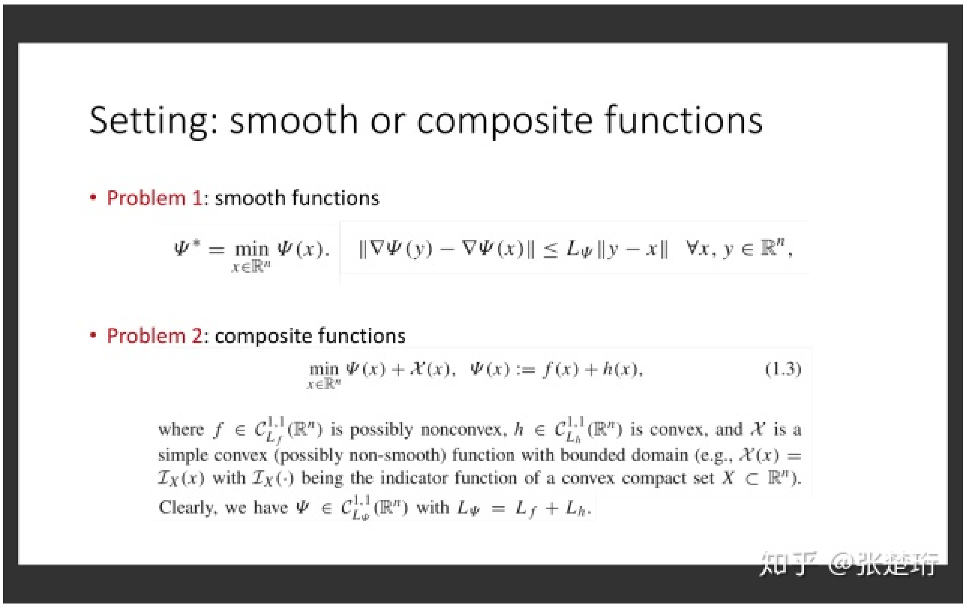
接下来，作为热身，回顾一下非常著名的 Nesterov 提出的加速梯度下降算法。该算法按照 $O(1/N^2)$ 收敛，其中 N 表示迭代次数（就是下面的 k）。但是该算法只针对凸优化问题，因此不使用强化学习问题。今天要重点读的文章就是针对非凸优化的加速算法。



三、问题设定

这里要研究的问题主要是两类：一类是普通的光滑非凸函数的优化；另一类这里写作 composite functions 的优化。后一类可能不太好理解：如果说前一类对应的为 gradient descent 算法的话，那么后一类优化问题对应的一种特殊情况是 projected gradient descent。因为强化学习里面还是会遇到这种要做 projection 的情形的（比如考虑一个 direct parameterization 或者说 tabular case），因此我也把相关结论抄了一下以备后用。

除此之外，还考虑两种情形：一种情形就是能够获取准确的梯度（exact gradient），这种情况下我们分析 iteration complexity；另一种情形就是需要对梯度进行估计（stochastic gradient），这种情况在强化学习的分析中更有用，因为强化学习关系的 sample complexity，能和 stochastic gradient 访问的次数相联系。



Settings for composite functions

- How to analyze projected gradient ascent? [4]

$$\mathbf{x}^{k+1} = P_C(\mathbf{x}^k - t_k \nabla f(\mathbf{x}^k)),$$

It can be written as $\mathbf{x}^{k+1} = \operatorname{argmin}_{\mathbf{x}\in C} \left\{ f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{1}{2t_k} \|\mathbf{x} - \mathbf{x}^k\|^2 \right\}.$

We can set $g(x) = \mathcal{I}_C(x)$ that equals 0 when $x \in C$, otherwise equals ∞.

$$\mathbf{x}^{k+1} = \operatorname{argmin}_{\mathbf{x}\in \mathbb{E}} \left\{ f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + g(\mathbf{x}) + \frac{1}{2t_k} \|\mathbf{x} - \mathbf{x}^k\|^2 \right\}.$$

In fact, this equals the definition of proximal operator

$$\operatorname{prox}_f(\mathbf{x}) = \operatorname{argmin}_{\mathbf{u}\in \mathbb{E}} \left\{ f(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{x}\|^2 \right\} \quad \textit{for any } \mathbf{x} \in \mathbb{E}.$$

Settings for composite functions

- Definition

$$\mathcal{P}(x, y, c) := \text{argmin}_{u \in \mathbb{R}^n} \left\{ \langle y, u \rangle + \frac{1}{2c}\|u - x\|^2 + \mathcal{X}(u) \right\}.$$

$$\mathcal{G}(x, y, c) := \frac{1}{c}[x - \mathcal{P}(x, y, c)].$$

- Example
  - If $\mathcal{X}(x) = \mathcal{I}_X(x)$, $\mathcal{P}(x, y, c) = P_X(x - cy)$.
  - If $y = \nabla\Psi(x)$, $\mathcal{G}(x, \nabla\Psi(x), c)$ is called gradient mapping.
  - If $\mathcal{X}(x) = 0$, $\mathcal{G}(x, \nabla\Psi(x), c) = \nabla\Psi(x)$.

Lemma 3 *Let $x \in \mathbb{R}^n$ be given and denote $g \equiv \nabla\Psi(x)$. If $\|\mathcal{G}(x, g, c)\| \le \epsilon$ for some $c > 0$, then*

$$-\nabla\Psi(\mathcal{P}(x, g, c)) \in \partial\mathcal{X}(\mathcal{P}(x, g, c)) + \mathcal{B}(\epsilon(cL_\Psi + 1)),$$

where $\partial\mathcal{X}(\cdot)$ denotes the subdifferential of $\mathcal{X}(\cdot)$ and $\mathcal{B}(r) := \{x \in \mathbb{R}^n : \|x\| \le r\}$.

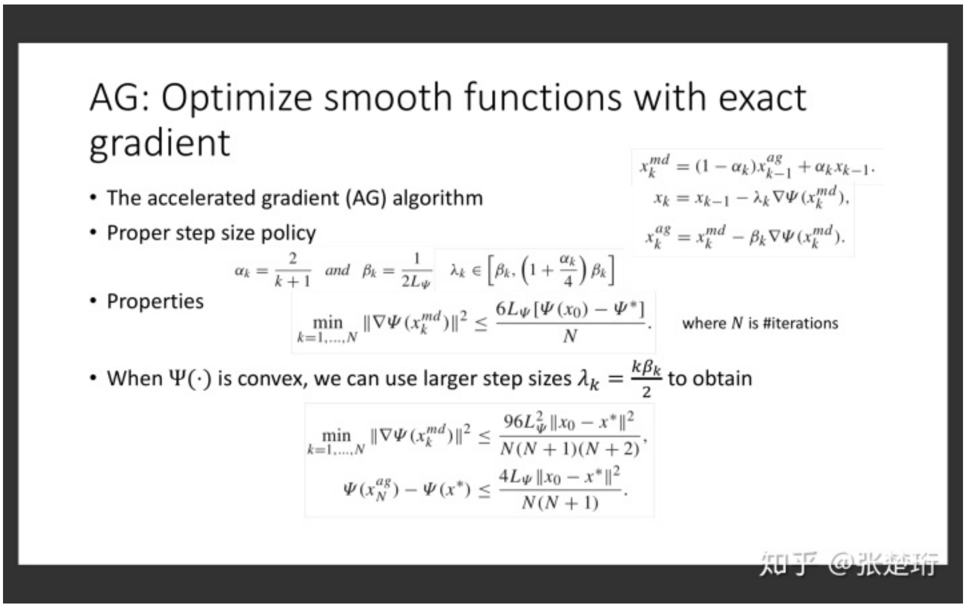
Settings for stochastic gradients

- Consider $\Psi \in \mathcal{C}^{1,1}_{L_\Psi}(\mathbb{E}^n)$ is bounded from below, and a stochastic oracle

a) $\mathbb{E}[G(x, \xi_k)] = \nabla\Psi(x)$,

b) $\mathbb{E}\left[\|G(x, \xi_k) - \nabla\Psi(x)\|^2\right] \le \sigma^2$.

四、算法和结论


AG: Optimize smooth functions with exact gradient
• The accelerated gradient (AG) algorithm
• Proper step size policy
• Properties
• When $\Psi(\cdot)$ is convex, we can use larger step sizes $\lambda_k = \frac{k\beta_k}{2}$ to obtain
$x_k^{md} = (1-\alpha_k)x_{k-1}^{ag} + \alpha_k x_{k-1}.$
$x_k = x_{k-1} - \lambda_k \nabla\Psi(x_k^{md}),$
$x_k^{ag} = x_k^{md} - \beta_k \nabla\Psi(x_k^{md}).$
$\alpha_k = \frac{2}{k+1}$ and $\beta_k = \frac{1}{2L_\Psi}$
$\lambda_k \in \left[\beta_k, \left(1+\frac{\alpha_k}{4}\right)\beta_k\right]$
$\min_{k=1,\ldots,N} \|\nabla\Psi(x_k^{md})\|^2 \le \frac{6L_\Psi[\Psi(x_0)-\Psi^*]}{N}.$
where N is #iterations
$\min_{k=1,\ldots,N} \|\nabla\Psi(x_k^{md})\|^2 \le \frac{96L_\Psi^2\|x_0-x^*\|^2}{N(N+1)(N+2)},$
$\Psi(x_N^{ag}) - \Psi(x^*) \le \frac{4L_\Psi\|x_0-x^*\|^2}{N(N+1)}.$
知乎 @张楚珩

AG: Optimize composite functions with exact gradient

$$x_k^{md} = (1-\alpha_k)x_{k-1}^{ag} + \alpha_k x_{k-1}.$$

$$x_k = \mathcal{P}(x_{k-1}, \nabla\Psi(x_k^{md}), \lambda_k),$$

$$x_k^{ag} = \mathcal{P}(x_k^{md}, \nabla\Psi(x_k^{md}), \beta_k).$$

- The accelerated gradient (AG) algorithm
- Proper step size policy

$$\alpha_k = \frac{2}{k+1} \quad and \quad \beta_k = \frac{1}{2L_\Psi}. \qquad \lambda_k = \frac{k\,\beta_k}{2}$$

- Properties

$$\min_{k=1,\ldots,N} \|\mathcal{G}(x_k^{md}, \nabla\Psi(x_k^{md}), \beta_k)\|^2 \le 24L_\Psi\left[\frac{4L_\Psi\|x_0 - x^*\|^2}{N(N+1)(N+2)} + \frac{L_f}{N}(\|x^*\|^2 + M^2)\right]$$

- Recall the setting

$$\min_{x\in\mathbb{R}^n} \Psi(x) + \mathcal{X}(x), \quad \Psi(x) := f(x) + h(x), \tag{1.3}$$

where $f \in \mathcal{C}_{L_f}^{1,1}(\mathbb{R}^n)$ is possibly nonconvex, $h \in \mathcal{C}_{L_h}^{1,1}(\mathbb{R}^n)$ is convex, and $\mathcal{X}$ is a simple convex (possibly non-smooth) function with bounded domain (e.g., $\mathcal{X}(x) = \mathcal{I}_X(x)$ with $\mathcal{I}_X(\cdot)$ being the indicator function of a convex compact set $X \subset \mathbb{R}^n$). Clearly, we have $\Psi \in \mathcal{C}_{L_\Psi}^{1,1}(\mathbb{R}^n)$ with $L_\Psi = L_f + L_h$.

RSAG: Optimize smooth functions with stochastic gradient

- Algorithm randomized stochastic AG (RSAG)
  - Update rule

$$x_k^{md} = (1-\alpha_k)x_{k-1}^{ag} + \alpha_k x_{k-1}.$$

$$x_k = x_{k-1} - \lambda_k G(x_k^{md}, \xi_k),$$

$$x_k^{ag} = x_k^{md} - \beta_k G(x_k^{md}, \xi_k).$$

  - Output x_R, where $R \sim Categorical(p_1, \cdots, p_N)$

RSAG: Optimize smooth functions with stochastic gradient

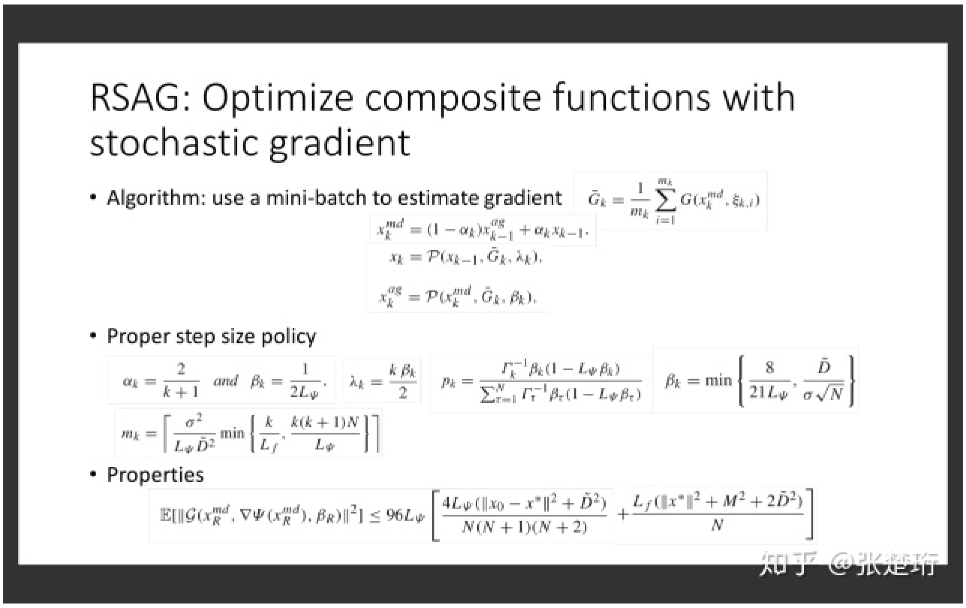
- Proper step size policy

$\alpha_k = \frac{2}{k+1}$ *and* $\beta_k = \frac{1}{2L_\Psi}$ $\lambda_k \in \left[\beta_k, \left(1 + \frac{\alpha_k}{4}\right)\beta_k\right]$ $p_k = \frac{\lambda_k C_k}{\sum_{\tau=1}^{N} \lambda_\tau C_\tau}$, $\beta_k = \min\left\{\frac{8}{21 L_\Psi}, \frac{\tilde{D}}{\sigma\sqrt{N}}\right\}$

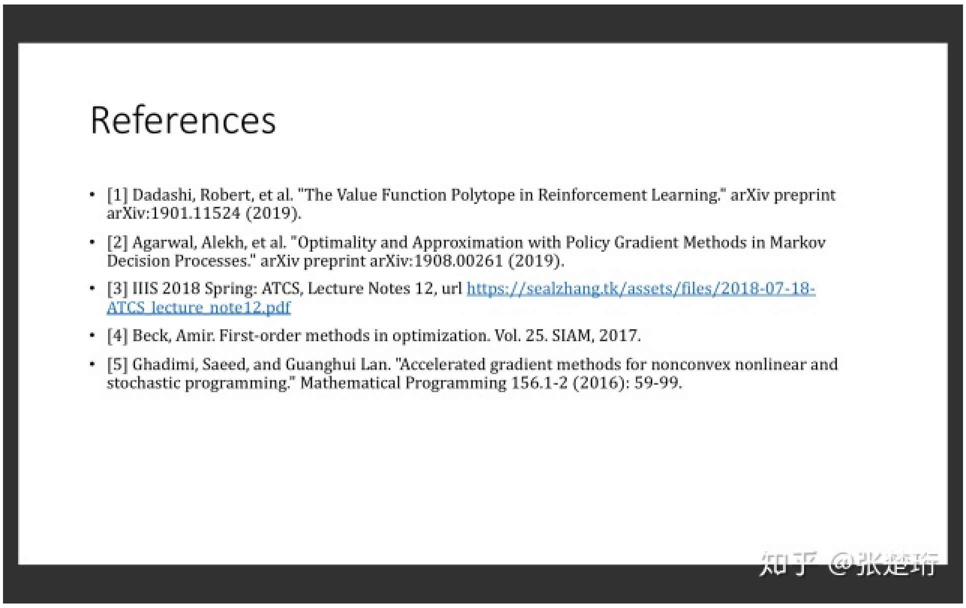
Where $\tilde{D} = \sqrt{[\Psi(x_0) - \Psi(x^*)]/L_\Psi}$

- Properties

$$\mathbb{E}[\|\nabla\Psi(x_R^{md})\|^2] \le \frac{21 L_\Psi[\Psi(x_0) - \Psi^*]}{4N} + \frac{4\sigma[L_\Psi(\Psi(x_0) - \Psi^*)]^{\frac{1}{2}}}{\sqrt{N}}$$


RSAG: Optimize composite functions with stochastic gradient
• Algorithm: use a mini-batch to estimate gradient $\bar{G}_k = \frac{1}{m_k}\sum_{i=1}^{m_k} G(x_k^{md}, \xi_{k,i})$
$x_k^{md} = (1-\alpha_k)x_{k-1}^{ag} + \alpha_k x_{k-1}$,
$x_k = \mathcal{P}(x_{k-1}, \bar{G}_k, \lambda_k)$,
$x_k^{ag} = \mathcal{P}(x_k^{md}, \bar{G}_k, \beta_k)$,
• Proper step size policy
$\alpha_k = \frac{2}{k+1}$ and $\beta_k = \frac{1}{2L_\Psi}$. $\lambda_k = \frac{k\beta_k}{2}$ $p_k = \frac{\Gamma_k^{-1}\beta_k(1-L_\Psi\beta_k)}{\sum_{\tau=1}^N \Gamma_\tau^{-1}\beta_\tau(1-L_\Psi\beta_\tau)}$ $\beta_k = \min\left\{\frac{8}{21L_\Psi}, \frac{\tilde{D}}{\sigma\sqrt{N}}\right\}$
$m_k = \left\lceil \frac{\sigma^2}{L_\Psi \tilde{D}^2}\min\left\{\frac{k}{L_f}, \frac{k(k+1)N}{L_\Psi}\right\}\right\rceil$
• Properties
$\mathbb{E}[\|\mathcal{G}(x_R^{md}, \nabla\Psi(x_R^{md}), \beta_R)\|^2] \le 96L_\Psi\left[\frac{4L_\Psi(\|x_0-x^*\|^2+\tilde{D}^2)}{N(N+1)(N+2)} + \frac{L_f(\|x^*\|^2+M^2+2\tilde{D}^2)}{N}\right]$
知乎 @张楚珩

参考文献


References
• [1] Dadashi, Robert, et al. "The Value Function Polytope in Reinforcement Learning." arXiv preprint arXiv:1901.11524 (2019).
• [2] Agarwal, Alekh, et al. "Optimality and Approximation with Policy Gradient Methods in Markov Decision Processes." arXiv preprint arXiv:1908.00261 (2019).
• [3] IIIS 2018 Spring: ATCS, Lecture Notes 12, url https://sealzhang.tk/assets/files/2018-07-18-ATCS_lecture_note12.pdf
• [4] Beck, Amir. First-order methods in optimization. Vol. 25. SIAM, 2017.
• [5] Ghadimi, Saeed, and Guanghui Lan. "Accelerated gradient methods for nonconvex nonlinear and stochastic programming." Mathematical Programming 156.1-2 (2016): 59-99.
知乎 @张楚珩

发布于 2019-11-02

最优化 机器学习 吴恩达（Andrew Ng）

▲ 赞同 28 ▼ 4 条评论 分享 喜欢 收藏 ···

文章被以下专栏收录

强化学习前沿

读呀读paper

进入专栏