Stochastic Approximate Gradient Descent via the (Underdamped) Langevin Algorithm

Carnegie Mellon University

Yixuan Qiu
Carnegie Mellon University

Xiao Wang
Purdue University

PURDUE UNIVERSITY

Introduction

- The stochastic gradient descent method (SGD) is one of the most popular and widely-used optimization techniques in large-scale machine learning problems.
- For a broad range of problems, the objective function is an expectation, $F(\theta) = \mathbb{E}[f(\theta; \xi)]$.
- A simple but commonly seen example is $F(\theta) = n^{-1}\sum_{i=1}^{n} f(\theta; X_i)$, where ξ follows the empirical distribution of the data.
- We consider the case that ξ follows a complicated continuous distribution $\pi(\xi)$, so that an unbiased stochastic gradient cannot be trivially obtained.
- We propose the stochastic *approximate* gradient descent (SAGD) as an extension to SGD for such cases, based on the underdamped Langevin sampling algorithm.
- Theoretical and empirical studies demonstrate the validity and usefulness of SAGD.

The Underdamped Langevin Algorithm

- We use a stochastic gradient $\tilde{g}(\theta) = K^{-1}\sum_{k=0}^{K-1} \nabla f(\theta; \xi_k)$ to approximate the true gradient $g(\theta) = \mathbb{E}[\nabla f(\theta; \xi)]$, where $\{\xi_k\}$ is generated by the underdamped Langevin algorithm.
- To approximately sample from an r-dimensional distribution with density $\pi(x) \propto \exp\{-V(x)\}$, run the following iterations:

$$\xi_{k+1} = \xi_k + \delta\rho_k,$$
$$\rho_{k+1} = (1-\gamma\delta)\rho_k - \delta\cdot\nabla V(\xi_k) + \sqrt{2\gamma\delta}\eta_k, \quad k \geq 0,$$

  where δ is the step size, $\{\eta_k\} \overset{iid}{\sim} N(0, I_r)$, and η_k is independent of $\{(\xi_k, \rho_k)\}_{i=0}^{k-1}$.
- γ, ξ_0, and ρ_0 are arbitrary constants.

Theorem

Under mild and verifiable conditions (see below), there exist constants $C_1 > 0$ and $C_2 > 0$ such that for any $\gamma > 0$ and $K > 0$, we have

$$|\mathbb{E}(\tilde{g}) - g| \leq C_1\left(\frac{1}{K\delta} + \delta\right), \qquad \mathbb{E}\left[(\tilde{g} - g)^2\right] \leq C_2\left(\frac{1}{K\delta} + \delta^2\right).$$

- Key insight: $\tilde{g}$ is a *biased* estimator for the true gradient g.
- But its bias can be well controlled and quantified.
- Compared with prior art, we provide realistic and **verifiable** conditions for $V(x)$.

Verifiable Conditions

- $V(x)$ is bounded from below: $V(x) \geq \nu_0$.
- The second derivative of V is bounded: $\|\nabla^2 V(x)\| \leq \nu$.
- $V(x)$ is infinitely differentiable, and $V(x)$ and its derivatives have polynomial growth.
- There exist constants $\alpha > 0$ and $0 < \beta < 1$ such that for all $x \in \mathbb{R}^r$,

$$\frac{1}{2}\langle\nabla V(x), x\rangle \geq \beta V(x) + \gamma^2 C_\beta\|x\|^2 - \alpha, \quad C_\beta = \frac{\beta(2-\beta)}{8(1-\beta)}.$$

- As an example, the following proposition justifies the use of Langevin algorithm to sample from popular generative models (e.g. VAE).

Example: Neural Networks

Consider a single-layer neural network $h(z) = \sigma(Wz + b)$, where $\sigma(x) = \log(1 + e^x)$. Let $p(z|x)$ denote the conditional density of Z given $X = x$, where $Z \sim N(0, I_r)$ and $X|\{Z = z\} \sim N(h(z), \tau^2 I)$. Then $V(z) = -\log p(z|x)$ satisfies the conditions above.

Stochastic Approximate Gradient Descent

- Outline of the SAGD algorithm.

SAGD for minimizing $F(\theta) = \mathbb{E}[f(\theta; \xi)]$

1. **For** $t = 0, 1, \ldots, T-1$ **Do**
2. $\quad\xi_{t,0} \leftarrow \xi_0$, $\rho_{t,0} \leftarrow \rho_0$
3. $\quad$**For** $k = 1, \ldots, K_t - 1$ **Do**
4. $\quad\quad\xi_{t,k+1} \leftarrow \xi_{t,k} + \delta_t\rho_{t,k}$
5. $\quad\quad\eta_{t,k} \sim N(0, I_r)$
6. $\quad\quad\rho_{t,k+1} \leftarrow (1-\gamma\delta_t)\rho_{t,k} - \delta_t\cdot\nabla V(\xi_{t,k}) + \sqrt{2\gamma\delta_t}\eta_{t,k}$
7. $\quad$**End For**
8. $\quad\tilde{g}_t(\theta) \leftarrow K_t^{-1}\sum_{k=0}^{K_t-1}\nabla f(\theta; \xi_{t,k})$
9. $\quad\theta_{t+1} \leftarrow \mathcal{P}_\Theta(\theta_t - \alpha_t\cdot\tilde{g}_t(\theta_t))$
10. **End For**
11. **Return** $\hat{\theta} = T^{-1}\sum_{t=1}^{T}\theta_t$

Convergence Properties

Theorem - Convex objective functions

Suppose that $F(\theta)$ is convex and L-Lipschitz continuous in $\theta \in \Theta$, and Θ is a closed convex set with diameter $D < \infty$. Choose $\delta_t = C_1/\sqrt{t}$, $K_t = C_2 t$, and $\alpha_t = \alpha_0/\sqrt{t}$, where $C_1, C_2, \alpha_0 > 0$ are constants. Then under regularity conditions, we have

$$\mathbb{E}[F(\hat{\theta})] - F^* \leq \mathcal{O}(1/\sqrt{T}).$$

Theorem - Non-convex objective functions

Suppose that $g(\theta)$ is G-Lipschitz continuous in θ, and let $\delta_t = C_1 t^{-c}$, $K_t = C_2 t^{2c}$, and $\alpha_t = \alpha_0/t$ for some constants $0 < \alpha_0 < 1/(2G)$ and $C_1, C_2, c > 0$. Then under regularity conditions, we have

$$\liminf_{t\to\infty}\ \mathbb{E}[\|g(\theta_t)\|^2] = 0.$$

Application I - Automated EM Algorithm

EM Algorithm

Target: To compute the maximum likelihood estimator (MLE) for θ given the complete log-likelihood $L(\theta; x, u) = \log[p(x, u; \theta)]$, where the additional random vector U represents the missing data.

- **E-step**: Define $Q(\theta; \theta_k) = \mathbb{E}_{U|X=x,\theta_k}[L(\theta; x, U)]$.
- **M-step**: Update θ: $\theta_{k+1} = \arg\max_\theta Q(\theta; \theta_k)$.

Automated EM Algorithm

- The $Q(\theta; \theta_k)$ function can be written as $Q(\theta; \theta_k) = \mathbb{E}[L(\theta; x, \xi)]$, where $\xi \sim p(u|x; \theta_k) \propto p(x, u; \theta_k)$.
- $Q(\theta; \theta_k)$ can be maximized by running SAGD with $V(\xi) = -L(\theta_k; x, \xi)$.

- **Experiment**: latent variables $Z_1, \ldots, Z_n \overset{iid}{\sim} N(0, 1)$ and data $X_i|\{Z_1 = z_1, \ldots, Z_n = z_n\} \sim \text{Gamma}(10\cdot\sigma(a + bz_i))$, $\sigma(x) = 1/(1+\exp(-x))$. True parameters $\theta = (a, b) = (2, 0.5)$.
- Use EM algorithm to estimate θ: $\theta_{k+1} = \arg\max_\theta Q(\theta; \theta_k) = \mathbb{E}_{Z|X=x,\theta_k}[L(\theta; x, Z)]$.
- In each M-step, we run SAGD and exact gradient descent for $T = 100$ iterations.

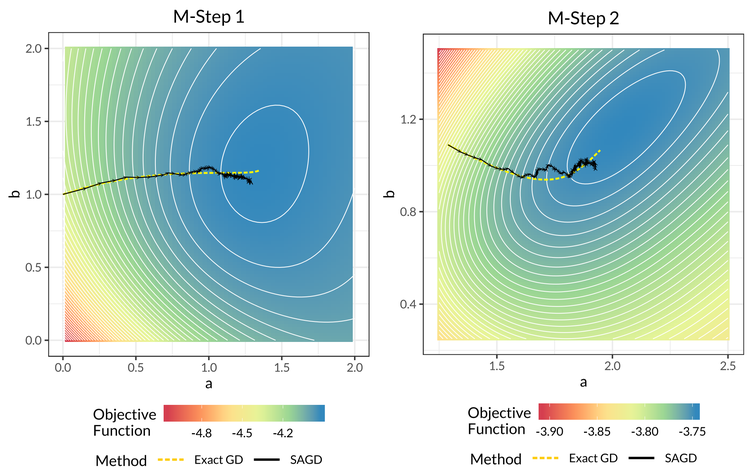

Figure 1: Comparison of exact GD and SAGD for two M-steps. The path of (a, b) values are overlaid on the contour plot of the $Q(\theta; \theta_k)$ function.

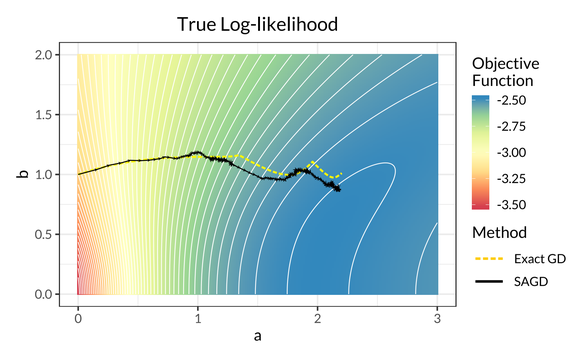

Figure 2: The paths of (a, b) on the surface of the true log-likelihood function.

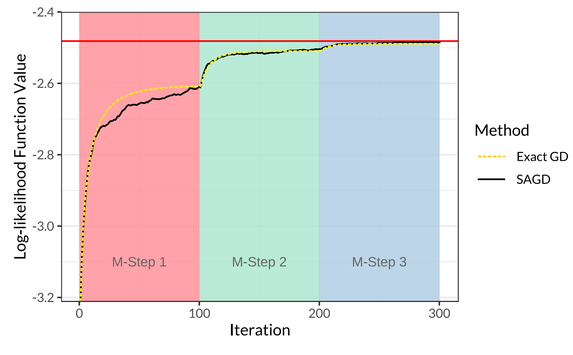

Figure 3: The log-likelihood function value versus the number of gradient updates.

Application II - Debiased VAE

Variational Autoencoder (VAE)

Target: To approximate the MLE for θ given the joint distribution $p(x, u; \theta)$, where U represents the latent variable for X.

- VAE maximizes

$$\tilde{\ell}(\theta; x) = \mathbb{E}_{u\sim q}[\log p(x|u;\theta)] - \text{KL}[q(u|x)\|p(u)]$$
$$= \ell(\theta; x) - \text{KL}[q(u|x)\|p(u|x;\theta)].$$

- For any $q(u|x)$, $\tilde{\ell}(\theta; x)$ is a lower bound of $\ell(\theta; x)$, where $\ell(\theta; x)$ is the marginal log-likelihood function.
- If $q(u|x) = p(u|x;\theta)$, then $\tilde{\ell}(\theta; x) = \ell(\theta; x)$.
- If $q(u|x) \neq p(u|x;\theta)$, then $\tilde{\ell}(\theta; x) < \ell(\theta; x)$, resulting bias.

Bias Correction for VAE

Similar to the automated EM algorithm:

- Define $Q(\theta; \theta_k) = \mathbb{E}_{U\sim p(u|x;\theta_k)}[\log p(x|u;\theta) + \log p(u)]$.
- Update θ by

$$\theta_{k+1} = \theta_k + \alpha_k\cdot[\partial Q(\theta; \theta_k)/\partial\theta]|_{\theta=\theta_k},$$

  with the true expectation replaced by the Langevin-based approximate gradient.

- **Experiment**: Given latent random variables $Z_i \overset{iid}{\sim} \pi(z)$, generate data points $X_i = Z_i + e_i$, $e_i \overset{iid}{\sim} N(0, 1)$.
- Fit a VAE model to recover the unknown $\pi(z)$.
- Consider three settings of $\pi(z)$: (a) $N(1, 0.5^2)$; (b) exponential distribution with mean 2; (c) $0.4\cdot N(0, 0.5^2) + 0.6\cdot N(3, 0.5^2)$.
- Compare VAE, SAGD, and other bias correction methods.

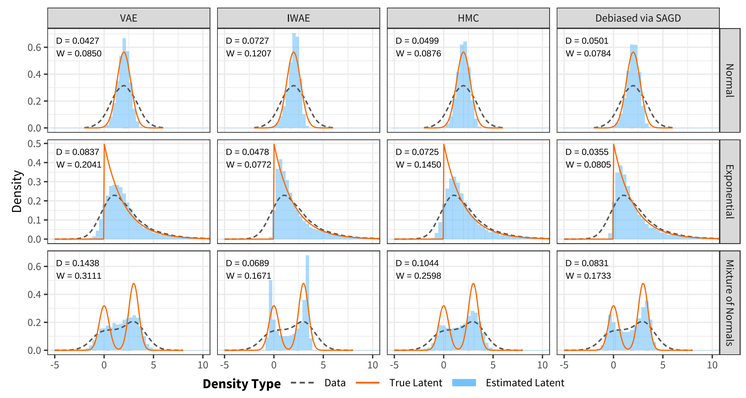

Figure 4: A demonstration of the marginal data distribution, true marginal latent density, and estimated marginal latent distributions. D and W stand for the K–S distance and 1-Wasserstein distance, respectively.

- **Experiment**: VAE for MNIST data.

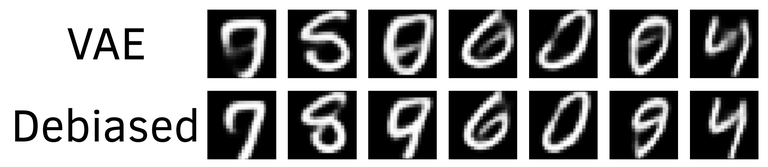

Figure 5: Representative examples from randomly generated digits that show significant improvement after the refining step using SAGD.

Conclusion

- SAGD is an extension to SGD for complicated statistical and machine learning problems.
- We prove the convergence of SAGD for both convex and non-convex objective functions.
- SAGD can be applied to important statistical and machine learning problems such as the EM algorithm and VAE.

Selected References

- Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407.
- Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *Siam Review*, 60(2), 223-311.
- Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th ICML*, 681–688.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological), 39(1), 1-22.
- Kingma, D. P. and Welling, M. (2014). Stochastic gradient VB and the variational auto-encoder. In *Proceedings of the 2nd ICLR*.

All correspondence goes to Yixuan Qiu <yixuanq@andrew.cmu.edu>