Bridging the Gap between Stochastic Gradient MCMC and Stochastic Optimization

Changyou Chen, David Carlson, Zhe Gan, Chunyuan Li,
Lawrence Carin

May 2, 2016

Outline

Introduction

- This paper is about how to better solve a complex, high-dimensional, nonlinear optimization problem in a big-data setting.
- Stochastic optimization:
  - computationally efficient, fast convergence, prone to local optimal
- Stochastic gradient MCMC:
  - computationally efficient, slower convergence, able to explore the parameter space
- Can we combine advantages from both?

Stochastic optimization

- Stochastic gradient descent (SGD)
  - basic stochastic optimization algorithm, without considering neither momentum and preconditioning
- SGD with momentum (SGD-M)
  - extending SGD with momentum
- RMSProp, Adadelta $\cdots$
  - extending SGD with preconditioner
- Adam
  - extending SGD with both momentum and preconditioner

Stochastic gradient MCMC

- Stochastic gradient Langevin dynamics (SGLD)
  - Bayesian analog of SGD, without considering neither momentum and preconditioning
- Stochastic gradient Hamiltonian Monte Carlo (SGHMC)
  - Bayesian analog of SGD-M, with momentum
- Preconditioned stochastic gradient Langevin dynamics (PSGLD)
  - Bayesian analog of RMSProp, with preconditioner
- Multivariate stochastic gradient thermostats (mSGNHT)
  - Bayesian sampling with adaptive momentum, does not have a stochastic optimization analog

Bridging the gap

- We propose a stochastic optimization algorithm, Santa, that starts from a preconditioned version of mSGNHT, whose temperature is then annealed to zero.
- It has the advantages of both adaptive preconditioner and adaptive momentum.

Table: SG-MCMC algorithms and their optimization counterparts.

Algorithms	SG-MCMC		Optimization
Basic	SGLD	$\Longleftrightarrow$	SGD
Precondition	pSGLD	$\Longleftrightarrow$	RMSprop
Momentum	SGHMC	$\Longleftrightarrow$	SGD-M
Thermostat	mSGNHT	$\Longleftrightarrow$	Santa

Outline

The Santa algorithm

Input: η_t (learning rate), σ, λ, $burnin$, $\beta = \{\beta_1, \beta_2, \cdots\} \to \infty$, $\{\zeta_t \in \mathbb{R}^p\} \sim N(\mathbf{0}, \mathbf{I}_p)$.

Initialize θ_0, $\boldsymbol{u}_0 = \sqrt{\eta} \times N(0, I)$, $\alpha_0 = \sqrt{\eta} C$, $\boldsymbol{v}_0 = 0$;

for $t = 1, 2, \ldots$ **do**

  Evaluate $\tilde{\boldsymbol{f}}_t \triangleq \nabla_\theta \tilde{U}(\theta_{t-1})$ on the t^{th} mini-batch;

  $\boldsymbol{v}_t = \sigma \boldsymbol{v}_{t-1} + \frac{1-\sigma}{N^2} \tilde{\boldsymbol{f}}_t \odot \tilde{\boldsymbol{f}}_t$;

  $\boldsymbol{g}_t = 1 \oslash \sqrt{\lambda + \sqrt{\boldsymbol{v}_t}}$;

  if $t < burnin$ **then**

    `/* exploration */`

    $\alpha_t = \alpha_{t-1} + (\boldsymbol{u}_{t-1} \odot \boldsymbol{u}_{t-1} - \eta / \beta_t)$;

    $\boldsymbol{u}_t = \frac{\eta}{\beta_t} (1 - \boldsymbol{g}_{t-1} \oslash \boldsymbol{g}_t) \oslash \boldsymbol{u}_{t-1} + \sqrt{\frac{2\eta}{\beta_t} \boldsymbol{g}_{t-1}} \odot \zeta_t$

  else

    `/* refinement */`

    $\alpha_t = \alpha_{t-1}$; $\boldsymbol{u}_t = \mathbf{0}$;

  end

  $\boldsymbol{u}_t = \boldsymbol{u}_t + (1 - \alpha_t) \odot \boldsymbol{u}_{t-1} - \eta \boldsymbol{g}_t \odot \tilde{\boldsymbol{f}}_t$; $\theta_t = \theta_{t-1} + \boldsymbol{g}_t \odot \boldsymbol{u}_t$;

end

Theory

- The Santa algorithm is based on the following stochastic differential equations, whose marginal distribution corresponds to the true posterior distribution of interest, with temperature $\frac{1}{\beta}$.

$$\begin{cases} \mathrm{d}\theta &= G_1(\theta)\boldsymbol{p}\mathrm{d}t \\ \mathrm{d}\boldsymbol{p} &= \Big(-G_1(\theta)\nabla_\theta U(\theta) - \Xi\boldsymbol{p} + \frac{1}{\beta}\nabla_\theta G_1(\theta) \\ & \quad + G_1(\theta)(\Xi - G_2(\theta))\nabla_\theta G_2(\theta)\Big)\mathrm{d}t + (\frac{2}{\beta}G_2(\theta))^{\frac{1}{2}}\mathrm{d}w \\ \mathrm{d}\Xi &= \Big(\boldsymbol{Q} - \frac{1}{\beta}I\Big)\mathrm{d}t\ , \end{cases} \tag{1}$$

  where $\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{p}\odot\boldsymbol{p})$, w is standard Brownian motion, $\boldsymbol{G}_1(\theta)$ and $\boldsymbol{G}_2(\theta)$ are some preconditioners.
- Santa algorithm is derived by solving (1) numerically with an increasing sequence of β.

Convergence properties

- The goal of Santa is to obtain θ^* such that

$$\theta^* = \operatorname*{argmin}_{\theta} U(\theta)$$

- $\{\theta_1, \cdots, \theta_L\}$: parameters collected from the algorithm.
- Sample average: $\hat{U} \triangleq \frac{1}{L} \sum_{t=1}^{L} U(\theta_t)$.
- Global optima: $\bar{U} \triangleq U(\theta^*)$.
- We study the convergence of the bias: $\left|\mathbb{E}\hat{U} - \bar{U}\right|$, and mean square error (MSE): $\mathbb{E}\left(\hat{U} - \bar{U}\right)^2$.

Convergence properties

Theorem

Under certain assumptions, the bias and MSE converge, for some constant C and D, and stepsize h, as

$$\text{Bias} \leq Ce^{-U(\theta^*)}\left(\frac{1}{L}\sum_{t=1}^{L}\int e^{-\beta_t \Delta U(\theta)}\mathrm{d}\theta\right) + D\left(\frac{1}{Lh} + h^2\right) .$$

$$\text{MSE} \leq C^2 e^{-2U(\theta^*)}\left(\frac{1}{L}\sum_{t=1}^{L}\int e^{-\beta_t \Delta U(\theta)}\mathrm{d}\theta\right)^2 + D^2\left(\frac{1}{Lh} + h^4\right) .$$

- The first part characterizes the distance between the global optima and the annealing distributions $e^{-\beta_t U(\theta)}$; the second part characterizes the distance between the sample average and the annealing posterior average. Both decrease with increasing L.

Convergence properties

Theorem

Under certain assumptions, the bias and MSE converge, for some constant C and D, and stepsize h, as

$$\text{Bias} \leq Ce^{-U(\theta^*)}\left(\frac{1}{L}\sum_{t=1}^{L}\int e^{-\beta_t \Delta U(\theta)}\mathrm{d}\theta\right) + D\left(\frac{1}{Lh} + h^2\right).$$

$$\text{MSE} \leq C^2 e^{-2U(\theta^*)}\left(\frac{1}{L}\sum_{t=1}^{L}\int e^{-\beta_t \Delta U(\theta)}\mathrm{d}\theta\right)^2 + D^2\left(\frac{1}{Lh} + h^4\right).$$

- The theorem indicates Santa converges in expectation closed to the global optima.

Outline

Illustration

- Optimizing the double-well potential:

$$U(\theta) = (\theta + 4)(\theta + 1)(\theta - 1)(\theta - 3)/14 + 0.5 \ .$$

- Start close to a local mode.
- RMSProp gets stuck, while Santa is able to jump out of the local mode.

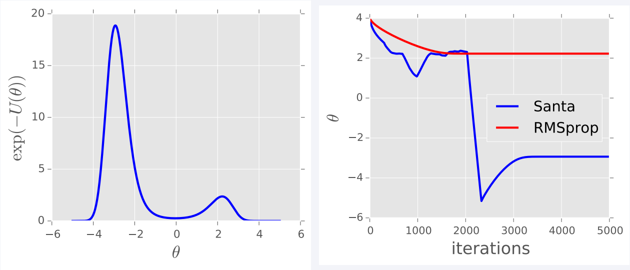

Figure: (Left) Double-well potential. (Right) The evolution of θ using Santa and RMSprop algorithms.

Feedforward neural networks and convolutional neural networks

- Detailed parameter setting is given in the paper.
- Santa outperforms other algorithms in most cases.

Table: Test error on MNIST classification using FNN and CNN.

Algorithms	FNN-400	FNN-800	CNN
Santa	**1.21%**	**1.16%**	**0.47%**
Adam	1.53%	1.47%	0.59%
RMSprop	1.59%	1.43%	0.64%
SGD-M	1.66%	1.72%	0.77%
SGD	1.72%	1.47%	0.81%
SGLD	1.64%	1.41%	0.71%
BPB$^\diamond$	1.32%	1.34%	–
SGD, Dropout$^\diamond$	1.51%	1.33%	–
Stoc. Pooling$^\triangleright$	–	–	0.47%
NIN, Dropout$^\circ$	–	–	0.47%
Maxout, Dropout*	–	–	0.45%

Recurrent neural networks (RNN)

- Language modeling with vanilla RNN.
- Test on four publicly available datasets.

Table: Test negative log-likelihood on 4 datasets.

Algorithms	Piano.	Nott.	Muse.	JSB.
Santa	**7.60**	**3.39**	**7.20**	**8.46**
Adam	8.00	3.70	7.56	8.51
RMSprop	7.70	3.48	7.22	8.52
SGD-M	8.32	3.60	7.69	8.59
SGD	11.13	5.26	10.08	10.81
HF$^\diamond$	7.66	3.89	**7.19**	8.58
SGD-M$^\diamond$	8.37	4.46	8.13	8.71

GoogleNet for ImageNet classification

- These are preliminary results, did not report in the main text (included in the supplement).
- Use ILSVRC 2011 for training (ILSVRC 2012 has similar performance).
- Compared with SGD with momentum, other algorithms did not seem to work.
- Did not tune the parameters, use the default setting for GoogleNet provided in the Caffe package.
- Santa converges much faster than SGD-M.

GoogleNet for ImageNet classification

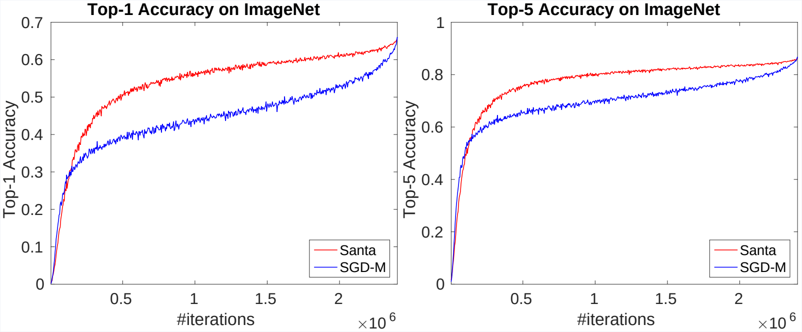

Figure: Santa vs. SGD with momentum on ImageNet.

Code

- Code provided at https://github.com/cchangyou/Santa.
- Also provide a Caffe implementation.
- Welcome for feedbacks.

Thanks for your attention