Variational Inference via Stochastic Backpropagation

Kai Fan

February 27, 2016

Outline

Bayesian inference on latent variable model

- $\mathbf{y}$ observed data
  $\mathbf{x}$ latent variable
  $p_\theta(\mathbf{x}, \mathbf{y})$ probabilistic model

Bayesian inference on latent variable model

- $\mathbf{y}$ observed data
  $\mathbf{x}$ latent variable
  $p_\theta(\mathbf{x}, \mathbf{y})$ probabilistic model
- Purpose: we are (very) interested in inferring a posterior distribution $p_\theta(\mathbf{x}|\mathbf{y})$
  - Enables learning parameters in latent variable models
  - Deep learning

Bayesian inference on latent variable model

- $\mathbf{y}$ observed data
  $\mathbf{x}$ latent variable
  $p_\theta(\mathbf{x}, \mathbf{y})$ probabilistic model
- Purpose: we are (very) interested in inferring a posterior distribution $p_\theta(\mathbf{x}|\mathbf{y})$
  - Enables learning parameters in latent variable models
  - Deep learning
- Difficulty: $p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{y})}$ is most often intractable.

Non-variational approx. inference methods

- Point estimate of $p_\theta(\mathbf{x}|\mathbf{y})$ (MAP)
  - Fast
  - Overfitting
- Markov Chain Monte Carlo (MCMC)
  - Asymptotically unbiased
  - Expensive, slow to assess convergence

Variational Inference

- Introduce variational distribution $q_\phi(\mathbf{x})$ or $q_\phi(\mathbf{x}|\mathbf{y})$ of true posterior.
  ϕ variational parameters

- Objective: minimize w.r.t. the KL-divergence

$$D_{KL}(q_\phi(\mathbf{x}|\mathbf{y})||p_\theta(\mathbf{x}|\mathbf{y}))$$

- $q_\phi(\mathbf{x}|\mathbf{y}) = p_\theta(\mathbf{x}|\mathbf{y})$ achieves 0 KL divergence.

Lower Bound

- From marginal log-likelihood to lower bound,

$$\begin{aligned}\log p_\theta(\mathbf{y}) &= \mathbb{E}_q\left[\log \frac{p_\theta(\mathbf{y},\mathbf{x})}{q_\phi(\mathbf{x}|\mathbf{y})}\right] + D_{KL}(q_\phi(\mathbf{x}|\mathbf{y})||p_\theta(\mathbf{x}|\mathbf{y})) \\ &\geq \mathbb{E}_q[\log p_\theta(\mathbf{y},\mathbf{x}) - \log q_\phi(\mathbf{x}|\mathbf{y})] \\ &\triangleq \mathcal{L}\end{aligned}$$

- Objective: maximize w.r.t the Lower Bound

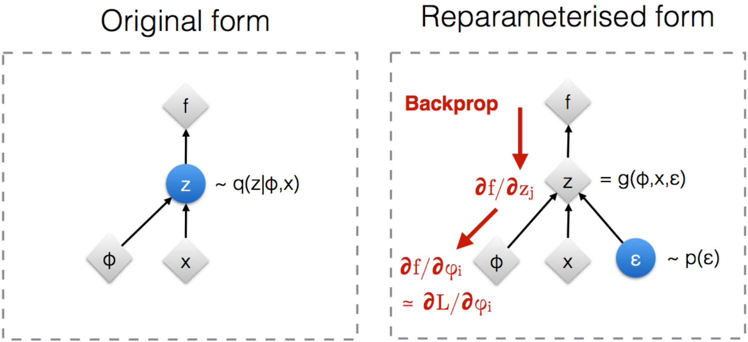
- Non-gradient-based optimization technique: Mean-Field VB with fixed-point equations
  - Efficiency
  - Intractable / not applicable in many cases

Outline

Reparameterized Gradient Estimator

- Consider a general form of lower bound $\mathcal{L} = \mathbb{E}_{q_\phi(\mathbf{x}|\mathbf{y})}[f(\mathbf{y}, \mathbf{x})]$
- Monte Carlo Gradient Approximation at Iteration t
  - sample ϵ^t from some base distribution $p(\epsilon)$
  - transformation $\mathbf{x}^t = g_\phi(\epsilon^l)$, s.t. $\mathbf{x}^t \sim q_\phi(\mathbf{x}|\mathbf{y})$
  - compute $\nabla_\phi f(\mathbf{y}, \mathbf{x}^t)$ to approximate $\nabla_\phi \mathcal{L}$
- Reparameterization has to exist. E.g. Gaussian, Laplace, Student t's, etc.

Reparameterization Trick



Gaussian Backpropagation

- $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$, we have following identities.

$$
\begin{aligned}
\nabla_{\mu_i} \mathbb{E}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\mathbf{C})}[f(\mathbf{x})] &= \mathbb{E}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\mathbf{C})}[\nabla_{z_i} f(\mathbf{x})] \\
\nabla_{C_{ij}} \mathbb{E}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\mathbf{C})}[f(\mathbf{x})] &= \frac{1}{2} \mathbb{E}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\mathbf{C})}[\nabla^2_{z_i,z_j} f(\mathbf{x})], \\
\nabla^2_{C_{i,j},C_{k,l}} \mathbb{E}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\mathbf{C})}[f(\mathbf{x})] &= \frac{1}{4} \mathbb{E}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\mathbf{C})}[\nabla^4_{z_i,z_j,z_k,z_l} f(\mathbf{x})], \\
\nabla^2_{\mu_i,C_{k,l}} \mathbb{E}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\mathbf{C})}[f(\mathbf{x})] &= \frac{1}{2} \mathbb{E}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\mathbf{C})} \left[\nabla^3_{z_i,z_k,z_l} f(\mathbf{x})\right].
\end{aligned}
$$

- Unbiased estimator of $\nabla^k \mathbb{E}[f]$, $k = 1, 2$
- Higher order derivatives need to calculated w.r.t. f

Reparameterized Gaussian Backpropagation

- $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{R}\mathbf{R}^\top)$, thus $\mathbf{x} = \boldsymbol{\mu} + \mathbf{R}\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- New identities

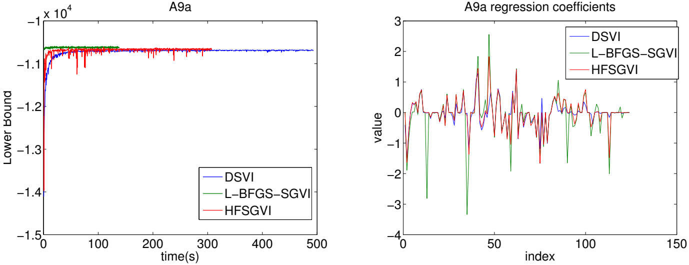
$$
\begin{aligned}
\nabla_{\mathbf{R}} \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}[f(\mathbf{x})] &= \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\mathbf{I}_{d_z})}[\boldsymbol{\epsilon}\mathbf{g}^\top] \\
\nabla^2_{\boldsymbol{\mu},\mathbf{R}} \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}[f(\mathbf{x})] &= \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\mathbf{I}_{d_z})}[\boldsymbol{\epsilon}^\top \otimes \mathbf{H}] \\
\nabla^2_{\mathbf{R}} \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}[f(\mathbf{x})] &= \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\mathbf{I}_{d_z})}[(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) \otimes \mathbf{H}]
\end{aligned}
$$

  where $\otimes$ is Kronecker product, and gradient $\mathbf{g}$, Hessian $\mathbf{H}$ are evaluated at $\boldsymbol{\mu} + \mathbf{R}\boldsymbol{\epsilon}$ in terms of $f(\mathbf{x})$.

- Still easy to obtain unbiased estimator
- Hessian-vector multiplication due to the fact that $(A^\top \otimes B)vec(V) = vec(AVB)$

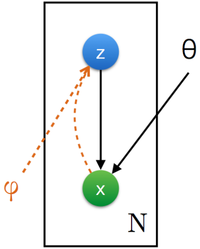
Bayesian Logreg

- Prior $\mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$ where $\mathbf{\Lambda}$ is diagonal
- Variational distribution $q(\beta|\boldsymbol{\mu}, \mathbf{D})$ where $\mathbf{D}$ is diagonal for simplicity.



Outline

Model Formulation

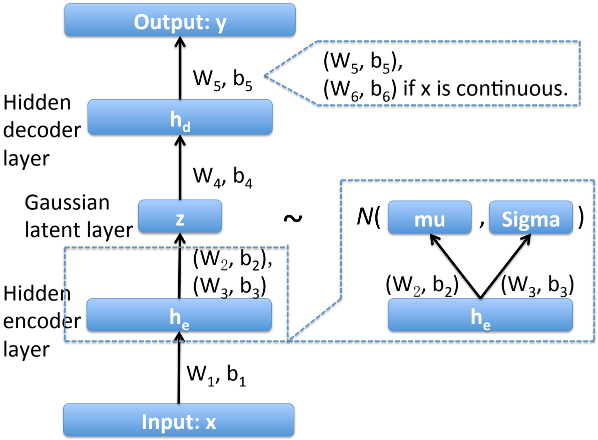


- Gaussian latent variable, prior $p(\mathbf{x}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$
- Generative model $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$, characterize a non-linear transformation, e.g. MLP
- Recognition model $q_{\phi}(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{D})$, where $\phi = [\boldsymbol{\mu}, \mathbf{D}] = \mathrm{MLP}(\mathbf{y}; W, b)$ and denote $\psi = (W, b)$

Objective Function: $L = \log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x}) - \log q(\mathbf{x}|\mathbf{y})$

- $\log p(\mathbf{y}|\mathbf{x})$ reconstruction error
- $\log p(\mathbf{x}) - \log q(\mathbf{x}|\mathbf{y})$ regularization
- Unlike VEM, $(\boldsymbol{\theta}, \psi)$ is optimized simultaneously, by gradient based algorithm.

Unrolled VAE



Back to Backpropagation

- Fast Gradient computation

$$\nabla_{\psi_l}\mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}[f(\mathbf{x})] = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\left[\mathbf{g}^\top \frac{\partial(\boldsymbol{\mu}+\mathbf{R}\boldsymbol{\epsilon})}{\partial\psi_l}\right]$$

$$\nabla^2_{\psi_{l_1}\psi_{l_2}}\mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}[f(\mathbf{x})] =$$

$$\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\left[\frac{\partial(\boldsymbol{\mu}+\mathbf{R}\boldsymbol{\epsilon})}{\partial\psi_{l_1}}^\top \mathbf{H}\frac{\partial(\boldsymbol{\mu}+\mathbf{R}\boldsymbol{\epsilon})}{\partial\psi_{l_2}} + \mathbf{g}^\top \frac{\partial^2(\boldsymbol{\mu}+\mathbf{R}\boldsymbol{\epsilon})}{\partial\psi_{l_1}\partial_{l_2}}\right]$$

- $\mathcal{O}(d_z^2)$ algorithmic complexity for both 1st and 2nd derivative.

Back to Backpropagation

- For any F, $\mathbf{H}_\psi \mathbf{v} = \lim_{\gamma \to 0} \frac{\nabla F(\psi + \gamma \mathbf{v}) - \nabla F(\psi)}{\gamma}$

$$
\begin{aligned}
\mathbf{H}_\psi \mathbf{v} &= \frac{\partial}{\partial \gamma} \nabla F(\psi + \gamma \mathbf{v}) \Big|_{\gamma=0} \\
&= \frac{\partial}{\partial \gamma} \mathbb{E}_{\mathcal{N}(0,\mathbf{I})} \left[\mathbf{g}^\top \frac{\partial \left(\boldsymbol{\mu}(\psi) + \mathbf{R}(\psi)\epsilon \right)}{\partial \psi} \Big|_{\psi \leftarrow \psi + \gamma \mathbf{v}} \right] \Bigg|_{\gamma=0} \\
&= \mathbb{E}_{\mathcal{N}(0,\mathbf{I})} \left[\frac{\partial}{\partial \gamma} \left(\mathbf{g}^\top \frac{\partial \left(\boldsymbol{\mu}(\psi) + \mathbf{R}(\psi)\epsilon \right)}{\partial \psi} \Big|_{\psi \leftarrow \psi + \gamma \mathbf{v}} \right) \right] \Bigg|_{\gamma=0}
\end{aligned}
$$

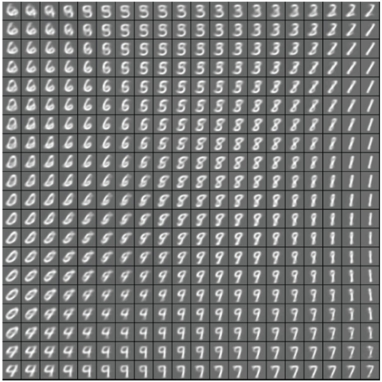
- PCG only requires $\mathbf{H}_\psi \mathbf{v}$ to solve linear system $Hp = -g$.
- For K iteration of PCG, relative tolerance $e < \exp(-2K/\sqrt{c})$, where c is matrix conditioner. Thus, c can be nearly as large as $O(K^2)$.
- Complexity for each iteration: $\mathcal{O}(Kdd_z^2)$ v.s. $\mathcal{O}(dd_z^2)$

Theoretical Perspective

- If f is an L-Lipschitz differentiable function and $\epsilon \sim \mathcal{N}(0, \mathbf{I}_{d_z})$, then $\mathbb{E}[(f(\epsilon) - \mathbb{E}[f(\epsilon)])^2] \leq \frac{L^2\pi^2}{4}$.
- $\mathbb{P}\left(\left|\frac{1}{M}\sum_{m=1}^{M} f(\epsilon_m) - \mathbb{E}[f(\epsilon)]\right| \geq t\right) \leq 2e^{-\frac{2Mt^2}{\pi^2 L^2}}$.
- In most application, $M = 1$ is used as MC integration.

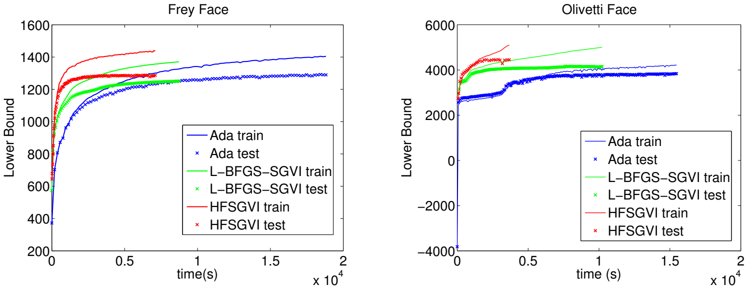
VAE Experiments

Manifold of Generative Model by setting $d_z = 2$



VAE Experiments

Lower Bound



Outline

Preliminaries

Stochastic Backpropagation

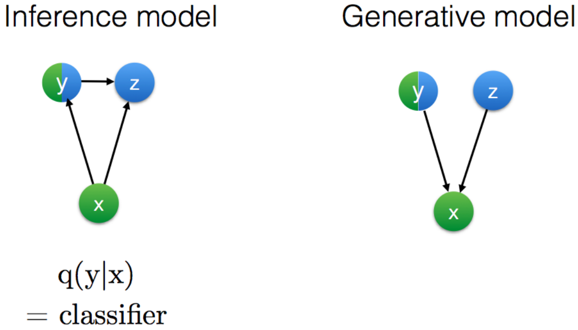
Variational Auto-Encoding

Related Work

Summary

Semi-supervised VAE (NIPS 2014)

Generative Model: $p(y) = Cat(y|\boldsymbol{\pi})$; $p(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$;
$p(\mathbf{y}|y, \mathbf{x}) = MLP$



Recognition Model: $q(\mathbf{x}|y, \mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_\phi(y, \mathbf{y}), \mathbf{D}_\phi(\mathbf{y}))$ and
$q(y|\mathbf{y}) = Cat(y|\boldsymbol{\pi}(\mathbf{y}))$, parameter function is also MLP.

Neural Variational Inference (ICML 2014)

Sigmoid Belief Networks

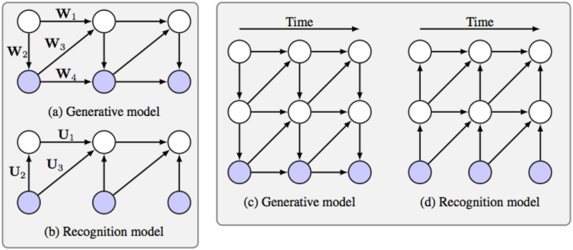
- Generative Model: $\mathbf{h}_L \to \mathbf{h}_{L-1} ... \to \mathbf{h}_1 \to \mathbf{y}$
- Recognition Model: reverse the arrow direction
- Learning signal or control variate for variance reduction borrowing idea form RL

$$\begin{aligned}\nabla_\phi L &= \mathbb{E}_q[(\log p_\theta(x,z) - \log q_\phi(z|x)) \times \nabla_\phi \log q_\phi(z|x)] \\ &= \mathbb{E}_q[(\log p_\theta(x,z) - \log q_\phi(z|x) - C_\xi(x)) \times \nabla_\phi \log q_\phi(z|x)]\end{aligned}$$

- (θ, ϕ, ξ) joint learning

Dynamic Modeling

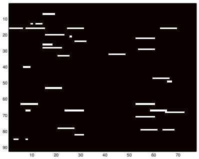
- DRAW (Dynamic VAE with LSTM, ICML 2015, reviewed before)
- DSBN (NIPS 2015), generative model is similar to HMM



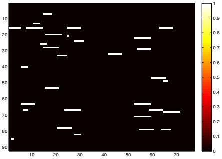
(a) Generative model
(b) Recognition model
(c) Generative model
(d) Recognition model

Dynamic Modeling, ctd

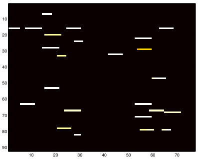
hGCHMM, my model



(g) NVI



(h) Doctor



(i) GibbsEM

Bayesian Dark Knowledge (NIPS 2015)

- Teacher Model: deep neural networks $T(y|x,\theta)$, prior $p(\theta|\lambda)$
- Student Model: deep neural networks $S(y|x,\omega)$, prior $p(\omega|\gamma)$
- Two step training (or distilled SGLD, term they used in paper)
  - Mini-bactch data (X, Y) with size B
  - SGLD update θ

$$\Delta\theta_{t+1} = \frac{\eta_t}{2}\left(\nabla_\theta \log p(\theta|\lambda) + \frac{N}{B}\sum_{x_i \in X} \nabla_\theta \log p(y_i|x_i,\theta)\right) + \mathcal{N}(0,\eta_t)$$

  - SGD update ω with noisy data $\tilde{X}$ only; $\tilde{y}_i$ is obtained by feeding $\tilde{x}_i$ to current teacher model

$$\Delta\omega_{t+1} = \rho_t\left(\frac{1}{B}\sum_{\tilde{x}_i \in \tilde{X}} \nabla_\omega \log p(\tilde{y}_i|\tilde{x}_i,\omega) + \gamma\omega_t\right)$$

Outline

Summary

- Minimize the difference between Generative model and recognition model
- Variational inference framework