Proximity Variational Inference

Jaan Altosaar [1], Rajesh Ranganath, David Blei
Princeton University, Columbia University

Rajesh Ranganath

David Blei

[1] Contact: altosaar@princeton.edu

Variational Inference

Given

- Data set $\mathbf{x}$
- Model $p(\mathbf{x}, \mathbf{z})$ with latent variables $\mathbf{z} \in \mathbb{R}^d$

Goal

- Infer posterior $p(\mathbf{z} \mid \mathbf{x})$

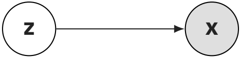
Recipe for Variational Inference

- Write down the model $p(\mathbf{x}, \mathbf{z})$
- Write down the approximate family $q(\mathbf{z}; \boldsymbol{\lambda})$
- Optimize the evidence lower bound objective:

$$\mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_{q(\mathbf{z};\lambda)}[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}; \boldsymbol{\lambda})]$$

- Result: approximate posterior $q(\mathbf{z}; \boldsymbol{\lambda}^*)$

Bernoulli factor model



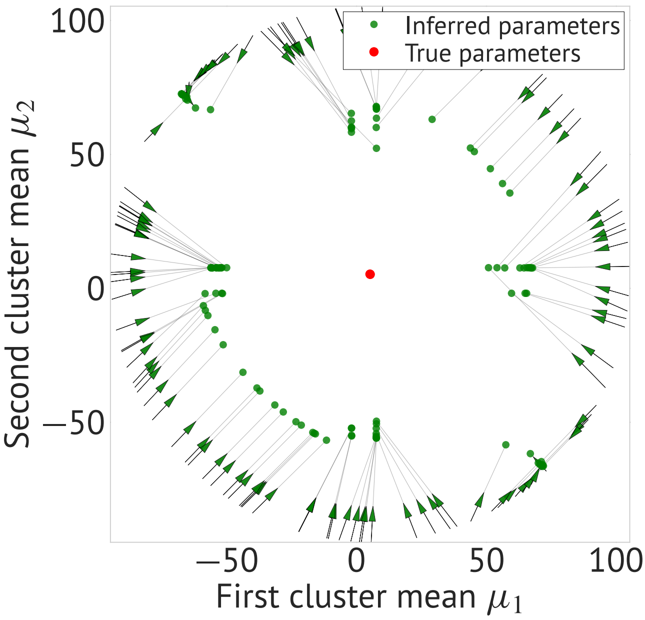
$$z_{ik} \sim \mathrm{Bernoulli}(\pi)$$
$$x_i \sim \mathrm{Normal}(z_i^\top \mu, \sigma^2 = 1)$$

Optimal update for approximate posterior:

$$q^*(z_{ik} = 1) \propto \exp(\mathbb{E}_{-z_{ik}}[-\frac{1}{2\sigma^2}(x_i - z_i^\top \mu_j)^2])$$

- The probability that z_{ik} is 1 goes to 0 when cluster means μ are initialized away from x

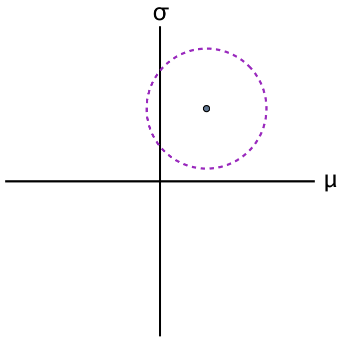
Bernoulli factor model in 2D:



Gradient ascent using proximity operators

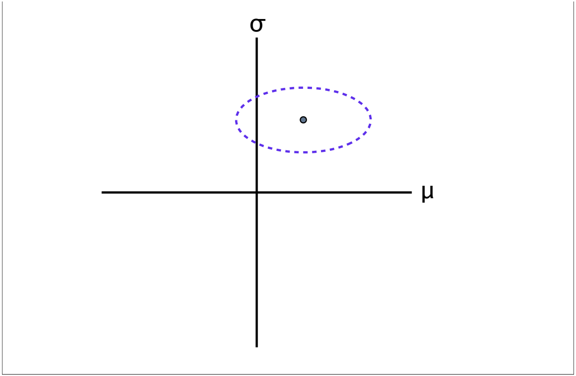
$$
\begin{aligned}
U(\boldsymbol{\lambda}_{t+1}) = & \mathcal{L}(\boldsymbol{\lambda}_t) + \nabla \mathcal{L}(\boldsymbol{\lambda}_t)^\top (\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t) \\
& - \frac{1}{2\rho} (\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t)^\top (\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t)
\end{aligned}
$$

$$
\Rightarrow \boldsymbol{\lambda}^*_{t+1} = \boldsymbol{\lambda}_t + \rho \nabla \mathcal{L}(\boldsymbol{\lambda}_t)
$$



Proximity operators for variational inference

$$U(\boldsymbol{\lambda}_{t+1}) = \mathcal{L}(\boldsymbol{\lambda}_t) + \nabla\mathcal{L}(\boldsymbol{\lambda}_t)^\top(\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t) - \frac{1}{2\rho}(\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t)^\top(\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t) - k \cdot d(f(\boldsymbol{\lambda}_t), f(\boldsymbol{\lambda}_{t+1}))$$



Examples of proximity statistics $f(\boldsymbol{\lambda})$

- Entropy $H(q(\mathbf{z};\boldsymbol{\lambda}))$
- Kullback-Leibler divergence $KL(q(\mathbf{z};\boldsymbol{\lambda})||p(\mathbf{z}))$
- Mean/variance $\mathbb{E}_q[z], \mathrm{Var}(\mathbf{z})$

Recipe for Proximity Variational Inference

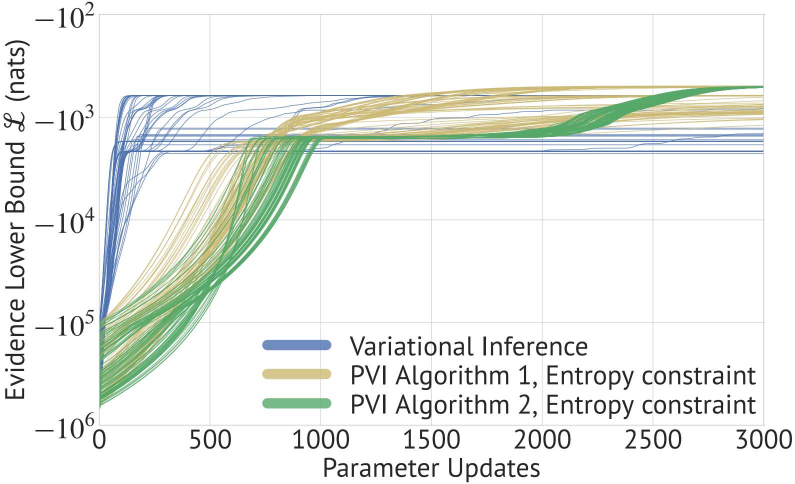
1. Design proximity statistic for variational parameters $f(\lambda)$
2. Choose distance function d
3. Optimize $\mathcal{L}_{\text{proximity}}$

$$\begin{aligned}\mathcal{L}_{\text{proximity}}(\boldsymbol{\lambda}_{t+1}) = & \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{z})] - \mathbb{E}_q[\log q(\boldsymbol{\lambda}_{t+1})] \\ & - k \cdot d(f(\boldsymbol{\lambda}_{t-m}), f(\boldsymbol{\lambda}_{t+1})).\end{aligned}$$

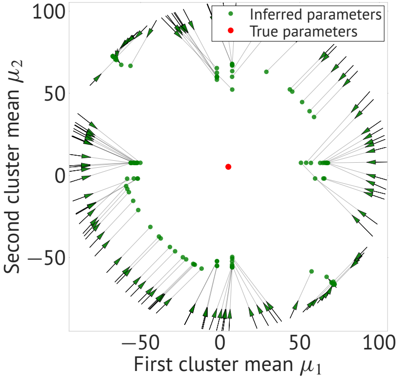
TensorFlow example

```
elbo = log_p_x_z - log_q_z
constraint = -k * tf.square(
    q_z.entropy() - q_z_lagged.entropy())
elbo_proximity = elbo + constraint
optim = tf.train.AdamOptimizer(0.001)
train_op = optim.minimize(-elbo_proximity)
```

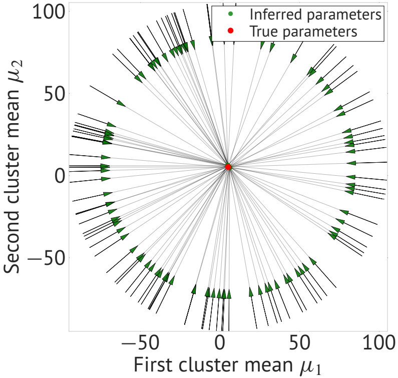
Robustness to local optima



Reduced sensitivity to initialization

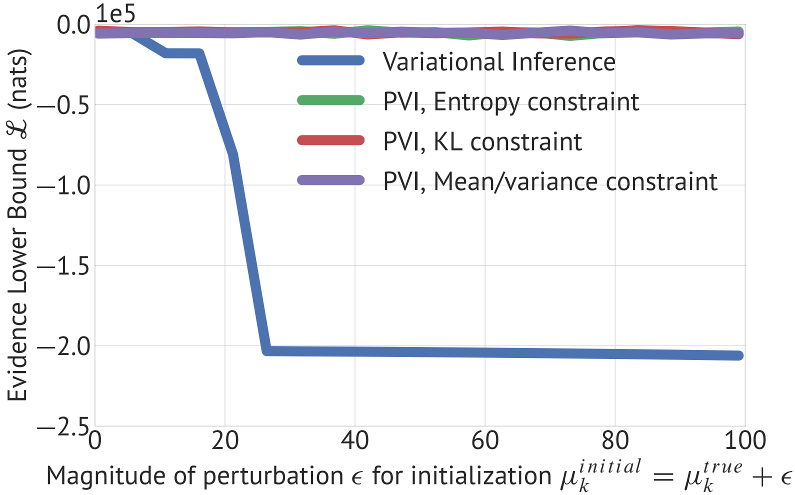


Variational Inference

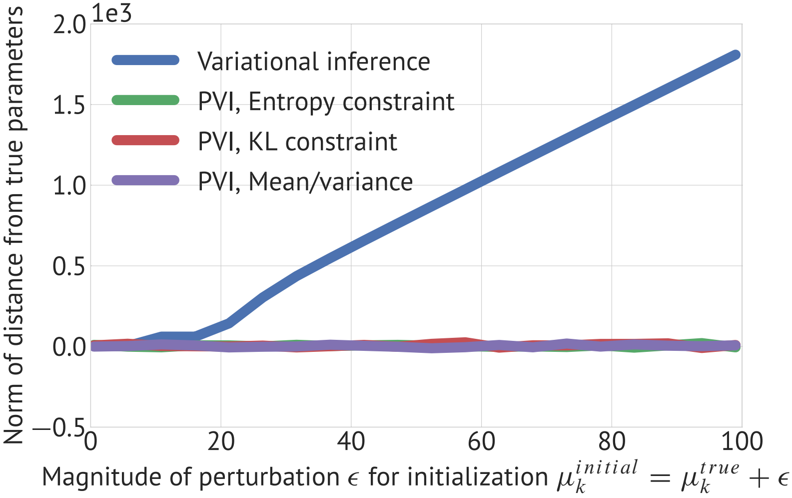


Proximity Variational Inference

Bernoulli factor model



Bernoulli factor model



Sigmoid Belief Network

- Sigmoid Belief Network; a neural net model with latent variables
- Binarized MNIST dataset
- One to Three stochastic layers of 200 dimensions
- Badly initialize weights to -100

1-Layer Sigmoid Belief Network

Inference Method	ELBO	Held-out Log-likelihood
Variational Inference	–226.9	–212.1
PVI, Entropy constraint	–165.7	–139.7
PVI, KL constraint	–190.6	–189.6
PVI, Mean/variance constraint	–153.2	–128.7

3-Layer Sigmoid Belief Network

Inference Method	ELBO	Held-out Log-likelihood
Variational Inference	–222.8	–208.3
PVI, Entropy constraint	–167.5	–139.1
PVI, KL constraint	–188.8	–173.8
PVI, Mean/variance constraint	–185.6	–149.7

1-Layer Sigmoid Belief Network

Data 

Variational Inference 

PVI, Entropy constraint 

PVI, KL constraint 

PVI, Mean/variance constraint 

3-Layer Sigmoid Belief Network

Data 

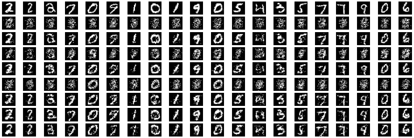
Variational Inference 

PVI, Entropy constraint 

PVI, KL constraint 

PVI, Mean/variance constraint 

Sigmoid Belief Network



Data	
Variational Inference	One layer
PVI, Entropy constraint	One layer
PVI, KL constraint	One layer
PVI, Mean/variance constraint	One layer
Variational Inference	Three layers
PVI, Entropy constraint	Three layers
PVI, KL constraint	Three layers
PVI, Mean/variance constraint	Three layers

Summary

- Easy to implement and test which proximity constraints can fix issues with variational inference
- Email me for TensorFlow code: altosaar@princeton.edu
- Preprint will be on arXiv soon