Inference Networks

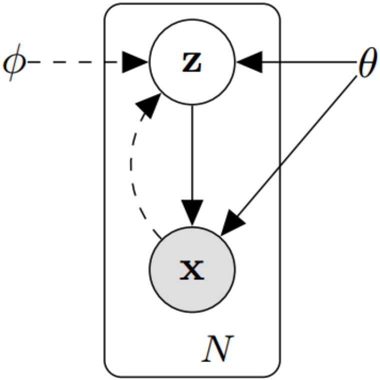
Semih Akbayrak

April 12, 2017

Outline

1. Latent Factor Models
2. Variational Inference
3. Variational EM
4. Variational Inference with Neural Nets

Latent Factor Models

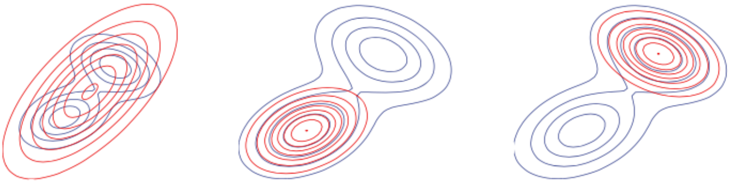


- Interpretable latent factors
- Breaking curse of dimensionality(use for different other tasks like classification)
- Visualization
- Complete missing x vals. restore defficient parts of x
- Explain the data distribution

Variational Inference

- Approximate $p(z|x)$ with $q_\phi(z)$ by tuning its parameters with optimization methods.
- Objective: $min\ KL(q_\phi(z)||p(z|x))$

Why Reverse KL?



(a) (b) (c)

Variational Inference

$$\min_{\phi} KL(q_\phi(z)||p(z|x)) = \int q_\phi(z) \log \frac{q_\phi(z)}{p(z|x)} dz$$

$$KL(q_\phi(z)||p(z|x)) = \int q_\phi(z) \log \frac{q_\phi(z)p(x)}{p(x,z)} dz$$

$$= \int q_\phi(z) \log \frac{q_\phi(z)}{p(x,z)} dz + \int q_\phi(z) \log p(x) dz$$

$$= \int q_\phi(z) \log \frac{q_\phi(z)}{p(x,z)} dz + \log p(x)$$

$\log p(x) = L(\phi) + KL(q_\phi(z)||p(z|x))$ where $L(\phi) = -\int q_\phi(z) \log \frac{q_\phi(z)}{p(x,z)} dz$

- $\log p(x)$ is constant w.r.t. ϕ so $\min_{\phi} KL(q_\phi(z)||p(z|x)) = \max_{\phi} L(\phi)$

Variational Inference

$$L(\phi) = -\int q_\phi(z) \log \frac{q_\phi(z)}{p(x,z)} dz$$
$$= E_q[-\log q_\phi(z) + \log p(x|z) + \log p(z)]$$
$$= E_q[\log p(x|z) + \log \frac{p(z)}{q_\phi(z)}]$$
$$= E_q[\log p(x|z)] + \int q_\phi(z) \log \frac{p(z)}{q_\phi(z)} dz$$
$$= E_q[\log p(x|z)] - KL(q_\phi(z)||p(z))$$

- we got rid of $p(z|x)$. Use an appropriate optimization method to maximize L w.r.t. variational parameters ϕ.

Variational EM

- What if model also has parameters?

$$\log p_\theta(x) = L(\theta, \phi) + KL(q_\phi(z)||p_\theta(z|x))$$

E-step: $\phi = \underset{\phi}{argmax} L(\theta, \phi)$

M-step: $\theta = \underset{\theta}{argmax} L(\theta, \phi)$

where $L(\theta, \phi) = E_{q_\phi}[\log p_\theta(x|z)] - KL(q_\phi(z)||p_\theta(z))$

Variational Inference with Neural Nets

- Can we approximate $p(z|x)$ with Neural Networks?
- Objective: $q_\phi(z|x) \leftarrow q_\phi(z)$

$$maxL(\theta, \phi) = E_{q_\phi}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p_\theta(z))$$
$$minLoss = KL(q_\phi(z|x)||p_\theta(z)) - E_{q_\phi}[\log p_\theta(x|z)]$$

- First term works as encoder and second term works as decoder.
- A typical autoencoder with additional constraints on hidden layer.

Variational Inference with Neural Nets

- We define a general model

$p_\theta(z) = N(z; 0, I), p_\theta(x|z) = N(x; \mu_\theta, \Sigma_\theta)$
$\mu_\theta, \Sigma_\theta$ are NNs which take z as input.
θ: model parameters, weights of NNs in this model.

$q_\phi(z|x) = N(z; \mu_\phi, \Sigma_\phi)$
μ_ϕ, Σ_ϕ are NNs which take x as input.
ϕ: variational parameters, weights of NNs in this model.

Variational Inference with Neural Nets

$$Loss = KL(q_\phi(z|x)||p_\theta(z)) - E_{q_\phi}[\log p_\theta(x|z)]$$

First term can be computed analytically.
$KL(q_\phi(z|x)||p_\theta(z)) = \frac{1}{2}(tr(\Sigma_\phi) + \mu_\phi^T \mu_\phi - dim(z) - \log\det(\Sigma_\phi))$
This is differentiable w.r.t. ϕ and does not contain θ

Second term can be computed with Monte Carlo integration
$E_{q_\phi}[\log p_\theta(x|z)] \approx \frac{1}{S}\sum_{s=1}^{S} \log p_\theta(x|z^{(s)})$ for $z^{(s)} \sim q_\phi(z|x)$
However derivative of this integration w.r.t. ϕ is 0 because it doesn't contain ϕ parameters, although z depends on ϕ.

Variational Inference with Neural Nets

- We will use reparametrization trick.

Instead of $\frac{1}{S}\sum_{s=1}^{S} \log p_\theta(x|z^{(s)})$ for $z^{(s)} \sim q_\phi(z|x)$

Use $\frac{1}{S}\sum_{s=1}^{S} \log p_\theta(x|z^{(s)})$ for $z^{(s)} = \mu_\phi + \Sigma_\phi^{\frac{1}{2}}.\epsilon^{(s)}$ and $\epsilon^{(s)} \sim N(0, I)$

- This is now differentiable w.r.t. ϕ. To compute gradient, we can compute it for every $z^{(s)}$ and use this value in summation.