HW4: wed

COMPSCI 688: Probabilistic Graphical Models

Lecture 19: Black-Box Stochastic Variational Inference

→ VAE → HW5

Dan Sheldon

Manning College of Information and Computer Sciences
University of Massachusetts Amherst

Partially based on materials by Benjamin M. Marlin (marlin@cs.umass.edu) and Justin Domke (domke@cs.umass.edu)

Review

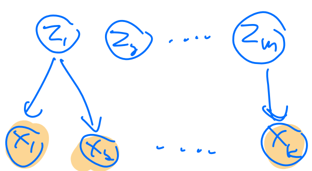
Variational Inference

1. **Input**: $p(z, x)$ and fixed x
2. Choose some approximating family $q_\phi(z)$
3. Maximize ELBO(ϕ) wrt ϕ — equivalent to minimizing KL($q_\phi(z) \,\|\, p(z|x)$)
4. Use $q_\phi(z)$ as a proxy for $p(z|x)$

$$\text{ELBO}(\phi) = \mathbb{E}_{q_\phi(Z)}\left[\log \frac{p(Z,x)}{q_\phi(Z)}\right] = \mathbb{E}_{q_\phi(Z)}\left[\log p(Z,x)\right] - \mathbb{E}_{q_\phi}\left[\log q_\phi(Z)\right]$$

$$\sum_z q_\phi(z) \log \frac{p(z,x)}{q_\phi(z)}$$

Variational Inference



Something we skipped: $p(z, x)$ discrete graphical model, $q(z) = \prod_j q_j(z_j)$ ("mean field") → "coordinate ascent VI, CAVI", message-passing

Today: z continuous, $p(z, x)$ black box, $q(z)$ TBD

BBVI ~ HMC

Applications: • stats model, PPL (stan)
• VAEs, interpretable generative models

Motivation: Continuous Latent Variable Models

Factor Analysis

$p(z) \approx q_\phi(z)$

Factor analysis is a classical statistical model. It posits an observed vector $\mathbf{x} \in \mathbb{R}^d$ is generated as a linear combination of basis vectors $\mathbf{w}_1, \dots, \mathbf{w}_m$ with weights $z_1, \dots, z_m$ plus noise:

$$\underbrace{\begin{bmatrix} x \end{bmatrix}}_{x} \Big\}d = \underbrace{\begin{bmatrix} w_1 & w_2 & \cdots & w_m \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix}}_{z}$$

"dictionary"

$x = Wz + \text{noise}$

"factor weight"

$$x = z_1 \cdot \vec{w}_1 + z_2 \vec{w}_2 + \dots + z_m \vec{w}_m + \text{noise}$$

$x = Wz + \text{noise}, \quad \text{noise} \sim \mathcal{N}(0, \Psi)$

$\Rightarrow x \sim \mathcal{N}(Wz, \Psi)$

Probabilistic factor analysis assumes the weights are drawn from a standard normal. The generative process is:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, I)$$
$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z}, \Psi)$$

(Typically Ψ is diagonal and the data is pre-processed so $\mathbf{x}$ has zero mean.)

Visualization: PCA Demo

Factor Analysis: Learning

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, I)$$
$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z}, \Psi)$$

$\theta = (W, \Psi)$

$z \to x$

Consider learning the parameters $\theta = (\mathbf{W}, \Psi)$ given data $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ assumed to be independently drawn from this model.

Since $\mathbf{z}$ is latent, the log-likelihood of a single datum $\mathbf{x}$ is $\log p(\mathbf{x})$, the "log-marginal likelihood".

In this model, the marginal likelihood is available in *closed form*:

$$p(\mathbf{x}) = \int \underbrace{\mathcal{N}(\mathbf{z}; 0, I)}_{p(z)} \underbrace{\mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z}, \Psi)}_{p(x|z)} d\mathbf{z} = \mathcal{N}(\mathbf{x}; 0, \mathbf{W}\mathbf{W}^\top + \Psi)$$

Therefore, we can learn by maximizing the log-marginal likelihood:

$$\mathcal{L}(\theta) = -\frac{N}{2}\log(|2\pi\Sigma|) - \frac{1}{2}\sum_{n=1}^{N} \mathbf{x}^{(n)\top}\Sigma^{-1}\mathbf{x}^{(n)}, \qquad \Sigma = \mathbf{W}\mathbf{W}^\top + \Psi$$

Alternately, there is an EM algorithm for this model where both E and M steps have simple forms.

Factor Analysis: Generalizations

This model is "easy", but factor analysis has many generalizations that make exact learning and inference intractable.

A variational autoencoder (VAE) uses a nonlinear-function f_θ instead of a linear transformation $\mathbf{W}$ to map from $\mathbf{z}$ to the mean of $\mathbf{x}$:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, I)$$
$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; f_\theta(\mathbf{z}), \Psi)$$

Wz

$z \xmapsto{Wz} \mu$

$z \xmapsto{f_\theta(z)} \mu$

A typical structure for f_θ is a multi-layer neural network, e,g.

$$f_\theta(\mathbf{z}) = h_2(\mathbf{b}_2 + \mathbf{W}_2 \cdot \overbrace{h_1(\underbrace{\mathbf{b}_1 + \mathbf{W}_1\mathbf{z}}))}^{z'}$$

where h_2, h_1 are element-wise nonlinear functions.

$\begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} \mu \end{bmatrix} + \text{noise}$ $\qquad$ $\overset{0/1}{\begin{bmatrix} x \end{bmatrix}} \sim \text{Bernoulli}\left(\begin{bmatrix} \mu \end{bmatrix}\right)$ $\qquad$ $x_j \sim \text{Bernoulli}(\mu_j)$

Another generalization changes the likelihood, e.g., to a Bernoulli distribution:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, I)$$
$$p(x_j|\mathbf{z}) = \text{Bernoulli}(x_j; (f_\theta(\mathbf{z}))_j), \qquad j = 1, \ldots, d$$

μ_j

$\mu_j = 0.7$

Inference and Learning in Generalized Models

Almost any change from the basic factor analysis model makes it so we can't compute the marginal likelihood $p(\mathbf{x})$ exactly, so inference and learning become hard.

The model is *only* tractable with linear transformations and a Gaussian likelihood.

We need additional inference tools for the generalizations.

Black-Box Stochastic Variational Inference

Black-Box Stochastic Variational Inference

z

A general inference approach that works well for models with continuous latent variables, including factor analysis, is *black-box stochastic variational inference*:

- **Black box**: only requires computing $\log p(z, x)$ and its gradients for different z
- **Stochastic**: optimizes the ELBO using Monte Carlo estimates

Stochastic Variational Inference

$$\max \ \mathrm{ELBO}(\phi) = \mathbb{E}_{q_\phi(z)}\left[\log \frac{p(z,x)}{q_\phi(z)}\right]$$

- **Input**: $p(z, x)$ and fixed x
- Start with some ϕ
- For $t = 1, 2, \ldots, T$
  - $g \leftarrow$ **unbiased estimate** of ∇_ϕELBO
  - Take a small step: $\phi \leftarrow \phi + \epsilon g$ (or Adam or other optimizer)
- **Return** ϕ

Main issue: how to get g?

$\log p_\theta(x)$ $p(z|x)$