5. Scalable Bayesian methods.

1) Probabalistic PCA.

~~Classical PCA:~~ Classical PCA:

$$p(X, Z | \Theta) = \prod_{i=1}^{n} \mathcal{N}(\underset{\mathbb{R}^D}{x_i} | V z_i, \sigma^2 I)\, \mathcal{N}(\underset{\mathbb{R}^d}{z_i} | 0, I), \quad D >> d.$$

$\Theta = [V, \sigma]$

Can use EM to find $\arg\max_{\Theta} p(X_{tr} | \Theta)$. Why?

- EM updates have compl. $O(nDd)$, analytical solution - $O(nD^2)$.
- Can process missing parts in X (or account some known z_j)
- Can determine d if $p(\Theta)$ is established.
- Can be extended to more general models.

Mixture of PCA:

$$p(X, Z, T | \Theta) = \prod_{i=1}^{n} p(x_i | t_i, z_i, \Theta)\, p(z_i | \Theta)\, p(t_i | \Theta) = \prod_{i=1}^{n} \mathcal{N}(x_i | V_{t_i} z_i, \sigma_{t_i}^2 I) \cdot \mathcal{N}(z_i | 0, I) \cdot \pi_{t_i}.$$

~~Can use to~~ Delivers non-linear dimension reduction.

$$\text{E-step: } q(Z, T) = \prod_{i=1}^{n} \frac{\mathcal{N}(x_i | V_{t_i} z_i, \sigma_{t_i}^2 I)\, \mathcal{N}(z_i | 0, I)\, \pi_{t_i}}{\int_{t, z} \text{—}\,//\,\text{—}}$$

$$\text{M-step: } \mathbb{E}_{Z, T} \log p(X, Z, T | \Theta) \longrightarrow \max_{\Theta}$$

2) Non-linear model and VAE.

How to account non-linear subspaces?

Use: $p(x, z | \Theta) = \underbrace{p(x | z, \Theta)}_{\substack{\text{generator} \\ \text{(decoder)}}} \underbrace{p(z | \Theta)}_{\text{prior}}$

Idea: use Neural Network.

$$p(X, Z | \Theta) = \prod_{i=1}^{n} p(x_i | z_i, \Theta)\, p(z_i) = \prod_{i=1}^{n} \left(\prod_{j=1}^{D} \mathcal{N}(x_{ij} | \mu_j(z_i), \sigma_j^2(z_i)) \right) \mathcal{N}(z_i | 0, I)$$

where μ_j and σ_j^2 are calculated by FFN.

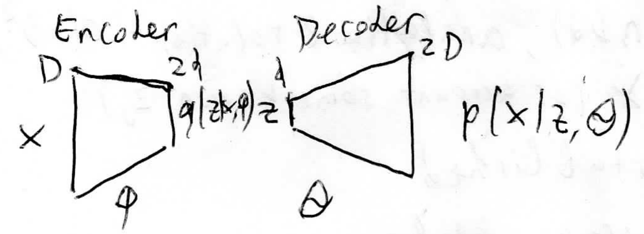
$\int p(x_i|z_i,\theta)\,p(z_i)\,dz_i$ is Intractable, so we couldn't compute MAP estimation ~~and~~ or simple E-step $\Rightarrow$ we use Variational-EM

$$q(z_i|x_i,\varphi) = \prod_{j=1}^{d} \mathcal{N}(z_{ij}\,|\,\mu_j(x_i), \sigma_j^2(x_i)),$$

$\approx p(z_i|x_i,\theta)$

where the mean and variance are given by another ~~NN~~ FFN parametrized by φ. (Encoder).

General structure.



ELBO: $\hat{\mathcal{L}}_V(\varphi,\theta) = \int q(z|x,\varphi)\log\frac{p(x|z,\theta)\,p(z)}{q(z|x,\varphi)}\,dz \to \max\limits_{\varphi,\theta}$

3) $\nabla_\theta \hat{\mathcal{L}}_V(\varphi,\theta) = \nabla_\theta \sum\limits_{i=1}^{\hat{}} \int q(z_i|x_i,\varphi)\log\frac{p(x_i|z_i,\theta)\,p(z_i)}{q(z_i|x_i,\varphi)}\,dz_i \approx$

mini-batching estim.

$\approx n\int q(z_i|x_i,\varphi)\,\nabla_\theta \log p(x_i|z_i,\theta)\,dz_i \approx, \quad i\sim \mathcal{U}\{1,\dots,n\}$

$\approx n\,\nabla_\theta \log p(x_i|z_i^*,\theta), \quad z_i^*\sim q(z_i|x_i,\varphi)$

Monte-Carlo estim.

4) $q(z_i|x_i,\varphi)$ depends on $\varphi \Rightarrow$ we have problems with $\nabla_\varphi \hat{\mathcal{L}}_V(\varphi,\theta)$ estimation.

~~Define Let $h(x,z) = \log\frac{p(x|z,\theta)p(z)}{q(z|x,\varphi)}$, then~~

$\cancel{\nabla_\varphi \hat{\mathcal{L}}_V(\varphi,\theta) = \sum_{i=1}^{n}}$

Consider differentiation of $\mathbb{E}_{y|x} h(x,y)$ in details:

$$\frac{\partial}{\partial x}\int p(y|x)\,h(x,y)\,dy = \int p(y|x)\frac{\partial}{\partial x}h(x,y)\,dy + \underbrace{\int h(x,y)\frac{\partial}{\partial x}p(y|x)\,dy}_{\text{not an expectation}}$$

To deal with second term we need <u>log-derivative trick</u>: $\frac{\partial}{\partial x}p(y|x) = p(y|x)\frac{\partial}{\partial x}\log p(y|x)$

which yields to:

$$\frac{\partial}{\partial x}\int p(y|x)\, h(x,y)\, dy = \int p(y|x) \frac{\partial}{\partial x} h(x,y)\, dy + \cancel{\ }$$

$$+ \int p(y|x)\, h(x,y) \frac{\partial}{\partial x} \log p(y|x)\, dy \approx \underbrace{\frac{\partial}{\partial x} h(x, y_0) + h(x,y_0) \frac{\partial}{\partial x} \log p(y_0|x)}_{\text{stochastic gradient of } \mathbb{E}_{y|x} h(x,y)},$$

$y_0 \sim p(y|x)$

Overall, we have now:

$$\hat{\mathcal{L}}_v(\varphi, \theta) \approx \int q(z|x,\varphi) \log p(x|z,\theta)\, dz - \underbrace{D_{KL}(q(z|x,\varphi)\,||\,p(z))}_{\text{can be computed and diff. in closed form}}$$

$\frac{\partial}{\partial \varphi}$

$$\underbrace{n \log p(x_i|z_i^*, \theta)}_{\text{too big at early steps}} \frac{\partial}{\partial \varphi} \log q(z_i^*|x_i, \varphi)$$ – Variance is too big

5) To deal with big variance:

1) REINFORCE – use baselines:

consider $b(z_i, \varphi)$ s.t. $\int q(z_i|\varphi)\, b(z_i, \varphi)\, dz_i = 0$,

e.g. $b(z_i, \varphi) = B(\varphi) \frac{\partial}{\partial \varphi} \log q(z_i|\varphi)$ – score function;

then: $\frac{\partial}{\partial \varphi} \int q(z_i|x_i,\varphi) \log p(x_i|z_i,\theta)\, dz_i =$

$$= \frac{\partial}{\partial \varphi} \int q(z_i|x_i,\varphi)(\log p(x_i|z_i,\theta) - b(z_i,\varphi))\, dz_i \approx$$

$$\approx (\log p(x_i|z_i^*, \theta) - B(\varphi)) \frac{\partial}{\partial \varphi} \log q(z_i^*|x_i, \varphi), \text{ where } z_i^* \sim q(z_i|x_i,\varphi)$$

we can make $B(\varphi) = B(\varphi, x_i)$

if $B(\varphi, x_i)$ is close to $\mathbb{E}_{z_i} \log p(x_i|z_i,\theta)$ then it may reduce the variance

practical only for small z

2] Reparameterization trick ('14):

~~"Express $y = g(x, \varepsilon)$ and~~

To ~~differe~~ get stochastic gradient for $\frac{\partial}{\partial x} \int p(y|x) h(x,y) dy$

express $y = g(x, \varepsilon)$, ε - random variable. Then:

$\int p(y|x) h(x,y) dx = \int r(\varepsilon) h(x, g(x,\varepsilon)) d\varepsilon$, and stochastic different. is simply:

$$\frac{\partial}{\partial x} \int p(y|x) h(x,y) dx = \frac{\partial}{\partial x} \int r(\varepsilon) h(x, g(x, \varepsilon)) d\varepsilon \approx$$

$$\approx \frac{d}{dx} h(x, g(x, \hat{\varepsilon})), \text{ where } \hat{\varepsilon} \sim r(\varepsilon)$$

Examples of reparametrization:

~~$\mathcal{N}(x|\mu, \Sigma)$~~ $p(x|y)$	$r(\varepsilon)$	$g(y, \varepsilon)$
$\mathcal{N}(x|\mu, \Sigma)$	$\mathcal{N}(\varepsilon|0, I)$	$x = A\varepsilon + \mu$, where $AA^T = \Sigma$
$\mathcal{E}(x|\lambda)$	$\mathcal{U}(\varepsilon|0,1)$	$x = -\frac{\log \varepsilon}{\lambda}$
$G(x|1,\beta)$	$G(\varepsilon|1,1)$	$x = \beta\varepsilon$

Discrete distrib. cannot be reparameterized. Finally, we have:

$$\frac{\partial}{\partial \varphi} \int q(z_i|x_i, \varphi) \log p(x_i|z_i, \theta) = \frac{\partial}{\partial \varphi} \int r(\varepsilon) \log p(x_i| g(\varepsilon, x_i, \varphi), \theta) d\varepsilon \approx$$

$$\approx \frac{\partial}{\partial \varphi} \log p(x_i| g(\hat{\varepsilon}, x_i, \varphi), \theta), \ \hat{\varepsilon} \sim r(\varepsilon)$$

5) VAE: final algorithm.

1. Input: Training data X, dimension of latent space d.
2. Pick random $i \sim \mathcal{U}\{1, \ldots, n\}$ and compute stochastic gradients of ELBO:
   - w.r.t. θ: $\nabla_\theta \hat{\mathcal{L}}_v(\varphi, \theta) \approx n \frac{\partial}{\partial \theta} \log p(x_i|z_i^*, \theta)$, $z_i^* \sim q(z_i|x_i, \varphi)$
   - w.r.t. φ: $\nabla_\varphi \hat{\mathcal{L}}_v(\varphi, \theta) \approx n[\frac{\partial}{\partial \varphi} \log p(x_i| g(\hat{\varepsilon}, x_i, \varphi), \theta) - \frac{\partial}{\partial \varphi} D_{KL}(q(z_i|x_i, \varphi) \| p(z_i))]$, $\hat{\varepsilon} \sim r(\varepsilon)$
3. Update θ and φ using stochastic gradients.

7) Automatic differentiation.

$\xleftarrow{\text{Backprop}}$

$$X \to \square \to \quad \square \to T$$

$$\Theta_{ML} = \arg\max_{\Theta} p(T_{tr}|X_{tr}, \Theta)$$

Stochastic computational graph

$$X \to \square \to \square \to \square \to T$$

Random Variable ξ

<u>RL</u>: states: $s_t \sim p(s_t|s_{t-1}, a_{t-1})$,

actions: $a_t \sim \pi(a_t|s_t)$

rewards: $r_t = r(s_t, a_t)$

$a_t \sim \pi(a_t|s_t)$ transition $p(s_{t+1}|s_t, a_t)$

$$\xrightarrow{s_t} \square \to \square \to \square \to s_{t+1}$$

Reward $r_t = r(s_t, a_t)$

<u>VAEs</u>:

random variable ξ random variable η

$$X \to \underset{\varphi}{\square} \to z \to \underset{\theta}{\square} \to X$$

<u>Dropout</u>:

Binary random ξ

$$X \to \square \to \square \to \square \to T$$

~~Parametri~~

<u>Attention</u>:

Attention mechanism   Attention map $\leftarrow$ Random sample ξ

$$\square \to \square$$

$$X \to \square \to \square \to \square \to T$$

Bayesian regularization:

$$p(T,\theta|X)=p(T|X,\theta)\,p(\theta)$$

$$X \to \underbrace{\square \to \square \to \square}_{\text{all steps are stochastic.}} \to T$$

With CG: use chain rule.

$$x \to \overset{g(\cdot)}{\bigcirc} \to \overset{p(\xi|g(x))}{\bigcirc} \to \overset{f(\cdot)}{\bigcirc} \to t$$

$$t=\mathbb{E}_{\xi} f(\xi)=\int p(\xi|g(x))\, f(\xi)\,d\xi$$

To propagate the (stochastic) gradients we need log-derivative or reparameterization tricks.

with log-derivative:

$$\frac{\partial t}{\partial x}=\int \frac{\partial p(\xi|g(x))}{\partial x} f(\xi)\,d\xi \approx \frac{\partial \log p(\hat{\xi}|g(x))}{\partial x} f(\hat{\xi}) =$$

$$=\frac{\partial \log p(\hat{\xi}|g(x))}{\partial g}\frac{\partial g(x)}{\partial x} f(\hat{\xi}),\quad \hat{\xi}\sim p(\xi|g(x)).$$

with reparameterization:

$$\frac{\partial t}{\partial x}=\frac{\partial}{\partial x}\int p(\xi|g(x))\, f(\xi)\,d\xi=\frac{\partial}{\partial x}\int p(\varepsilon)\, f(h(\varepsilon,g(x)))\,d\varepsilon=$$

$$=\int p(\varepsilon)\frac{\partial f(h(\varepsilon,g(x)))}{\partial x}\,d\varepsilon \approx \frac{\partial f(h(\hat{\varepsilon},g(x)))}{\partial x}=\frac{\partial f(h)}{\partial h}\frac{\partial h(g)}{\partial g}\frac{\partial g(x)}{\partial x},$$

$$h(g)=h(\hat{\varepsilon},g),\quad \hat{\varepsilon}\sim p(\varepsilon).$$