ETH zürich

EM Algorithm

- algorithm for finding maximum likelihood solutions for probabilistic models with latent variables

X - observed variables

Z - hidden variables $\quad p(X, Z|\theta)$ - joint distribution

<u>Goal:</u> Maximize $p(X|\theta) = \sum_Z p(X, Z|\theta)$

<u>Note:</u> If Z continuous, replace $\sum$ with $\int$ everywhere.

<u>Why?:</u> optimization of $p(X|\theta)$ is hard and optimization of $p(X,Z|\theta)$ easier

Introduce $q(Z)$ and observe that this holds

$$\ln p(X|\theta) = \mathcal{L}(q, \theta) + KL(q\|p), \text{ where}$$

$$\mathcal{L}(q, \theta) = \sum_Z q(Z) \ln \left\{ \frac{p(X, Z|\theta)}{q(Z)} \right\}$$

functional

$$KL(q\|p) = -\sum_Z q(Z) \ln \left\{ \frac{p(Z|X, \theta)}{q(Z)} \right\}$$

Proof: plug in $\ln p(X,Z|\theta) = \ln p(Z|X,\theta) + \ln p(X|\theta)$ here

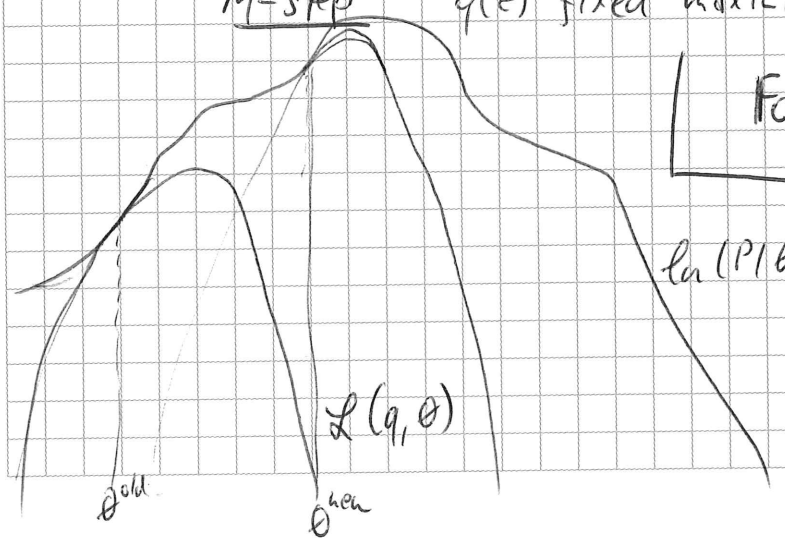
Thus, $\mathcal{L}(q, \theta) \leq \ln p(X|\theta)$

Two stage iterative algorithm:

<u>E-step</u> $\quad \mathcal{L}(q, \theta^{old})$ is maximized with respect to $q(Z)$, θ^{old} fixed

solution would be when $q(Z) = p(Z|X, \theta^{old})$, i.e. KL vanishes

<u>M-step</u> $\quad q(Z)$ fixed maximize $\mathcal{L}(q, \theta^{new})$ w.r.t. θ



For iid data ($p(X,Z) = \prod_n p(x_n, z_n)$)

$$p(Z|X,\theta) = \frac{p(X,Z|\theta)}{\sum_Z p(X,Z|\theta)} = \frac{\prod_n p(x_n, z_n|\theta)}{\sum_Z \prod_n p(x_n, z_n|\theta)}$$

$$= \prod_{n=1}^{N} p(z_n|x_n, \theta)$$

ETH zürich

Use EM to solve MAP (maximize posterior $p(\theta|X) = \frac{p(\theta, X)}{p(X)}$

$$\ln p(\theta|X) = \ln p(\theta, X) - \ln p(X)$$

$$\overset{\text{decomposition}}{=} \mathcal{L}(q, \theta) + KL(q\|p) + \ln p(\theta) - \ln p(X)$$

$$\geq \mathcal{L}(q, \theta) + \ln p(\theta) - \underbrace{\ln p(X)}_{\text{constant}}$$

E-step - the same ~~max q(z) w.r.t.~~ $\max \mathcal{L}(q, \theta)$ w.r.t. $q(z)$; θ fixed

M-step - modified to ~~optimize~~ include prior as well

→ Why does it converge:

E-step: $\ln(X|\theta)$ does not depend on $q(z)$. Therefore the largest value of $\mathcal{L}(q, \theta)$ is when KL divergence vanishes.

M-step: (TL;DR - optimizing lower bound)

Increasing $\mathcal{L}(q, \theta)$ will cause $\ln(X, \theta)$ to increase because $q(z)$ is fixed. (KL distribution ~~is the same~~ grows as less or equal as the likelihood

By plugging $q(z) = p(z|X, \theta^{old})$ into $\mathcal{L}(q, \theta)$, we get

$$\mathcal{L}(q, \theta) = \underbrace{\sum_z p(z|X, \theta^{old}) \ln p(X, Z|\theta)}_{\text{expectation of complete data log likelihood}} - \underbrace{\sum_z p(z|X, \theta^{old}) \ln p(z|X, \theta^{old})}_{\text{entropy independent of } \theta}$$

→ Examples Gaussian mixture models

$$p(X) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \qquad 0 \leq \pi_k \leq 1 \qquad \sum_k \pi_k = 1$$

Latent variable Z: $z_k = 1$ and all other are zero.

$$p(z_k = 1) = \pi_k \xrightarrow{\text{vector}} p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$$

$$p(X|z_k = 1) = \mathcal{N}(x|\mu_k, \Sigma_k) \xrightarrow{\text{vector}} p(x|z) = \prod_{k=1}^{K} \mathcal{N}(x|\mu_k, \Sigma_k)^{z_k}$$

$$p(X) = \sum_z p(z) p(x|z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

ETH zürich

E-step

$$\gamma(z_k) = p(z_k = 1 \mid x) \overset{\text{Bayes rule}}{=} \frac{p(x \mid z_k = 1)\, p(z_k = 1)}{\sum_{j=1}^{K} p(x \mid z_j = 1)\, p(z_j = 1)} = \frac{\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x \mid \mu_j, \Sigma_j)}$$

$\{x_1 \text{ — } x_N\} = X \in \mathbb{R}^{N \times D}$ → observed data (iid)

$\{z_1 \text{ — } z_N\} = Z \in \mathbb{R}^{N \times K}$ - latent variables

M-step

log likelihood: $\ln p(X \mid \pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k) \right\}$

setting derivative to zero: w.r.t. μ_k

$$0 = -\sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)}}_{\gamma(z_{nk})} \Sigma_k (x_n - \mu_k)$$

$$\Rightarrow \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

weighted mean

Similarly: $\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T$ ← weighted variance

Maximize $\ln p(X \mid \pi, \mu, \Sigma)$ w.r.t. π_k s.t. $\sum_k \pi_k = 1$

$$\ln p(X \mid \pi, \mu, \Sigma) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) \Rightarrow 0 = \sum_{n=1}^{N} \frac{\mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)} + \lambda$$

$$\Rightarrow \pi_k = \frac{N_k}{N}$$

ETH zürich

Variational inference

Fully Bayesian model, where all parameters are given prior distribution

$X = \{x_1 __ x_N\}$ iid data

$Z = \{z_1 __ z_N\}$ latent variables

Our model: $p(X, Z)$

Our goal: approximate $p(Z|X)$ and $p(X)$

$$\ln p(X) = \mathcal{L}(q) + KL(q\|p), \text{ where}$$

$$\mathcal{L}(q) = \int q(Z) \ln \left\{ \frac{p(X,Z)}{q(Z)} \right\} dZ$$

$$KL(q\|p) = -\int q \ln \left\{ \frac{p(Z|X)}{q(Z)} \right\} dZ$$

$\Rightarrow \mathcal{L}(q)$ lower bound, only equal when $q(Z) = P(Z|X)$

Suppose: $p(Z|X)$ is intractable.

$\Longrightarrow$ Optimize $q(Z)$ within restricted family so that KL divergence is minimized.

Idea: restrict (fix) family of distributions and parametrize $q(Z|\omega)$

Now, optimize $\mathcal{L}(q)$ w.r.t. ω

Mean-field (Factorized distributions)

Restrict $q(Z) = \prod_{i=1}^{M} q_i(Z_i)$, i.e. q factorized to disjoined partitions of Z

denoting $q(Z_i) = q_i$

$$\mathcal{L}(q) = \int \prod_i^M q_i \left\{ \ln p(X,Z) - \sum_i \ln q_i \right\} dZ$$

$$= \int q_j \left\{ \int \ln p(X,Z) \prod_{i \neq j} q_i \, dZ_i \right\} dZ_j - \int q_j \ln q_j \, dZ_j + \text{const}$$

$$= \int q_j \ln \hat{p}(X, Z_j) \, dZ_j - \int q_j \ln q_j \, dZ_j + \text{const}$$

where $\ln \hat{p}(X, Z_j) = \mathbb{E}_{i \neq j}[\ln p(X,Z)] + \text{const}$

ETH zürich

$$\mathbb{E}_{i\neq j}[\ln p(X,Z)] = \int \ln p(X,Z) \prod_{i\neq j} q_i \, dZ_i$$

Now, let's keep $\{q_{i\neq j}\}$ fixed and max $\mathcal{L}(q)$ w.r.t. $q_j(Z_j)$

Equivalent to minimizing $KL(q_j(Z_j) \| \tilde{p}(X, Z_j))$ and

minimum occurs when $q_j(Z_j) = \tilde{p}(X, Z_j)$

Coordinate ascent $\longrightarrow$ Optimal solution $\ln q_j^*(Z_j) = \underbrace{\mathbb{E}_{i\neq j}[\ln p(X,Z)]}_{\ln \tilde{p}(X, Z_j)} + \text{const.}$

~~Example: Mean field for Gaussian Mixture with Dirichlet Priors~~

~~EM with E identical M must include the prior~~

E-step: $\forall j$ evaluate $q^*(Z_j) = \frac{1}{Z}\exp \mathbb{E}_{i\neq j}(\log p(X,Z,\theta))$

M-step: Find $\theta_{t+1} = \arg\max_\theta \mathcal{L}_{BO}(q^{t+1}, \theta)$ set $q^{t+1} = \prod_i q_i^*$

Exponential Family distributions

~~$X = \{x_n\}_{n=1..N}$ iid data $Z = \{z_n\}$ corresponding latent variables~~

~~$$p(X,Z|\eta) = \prod_{n=1} h(x_n, z_n) g(\eta) \exp\{\eta^T u(x_n, z_n)\}$$~~

~~Conjugate prior $p(\eta | \nu_0, \chi_0) = f(\nu_0, \chi_0) g(\eta)^{\nu_0} \exp\{\nu_0 \eta^T \chi_0\}$~~

$$p(X|\eta) = h(X) \cdot g(\eta) \exp(\eta^T u(X))$$

η - natural parameters

$g(\eta)$ - normalization ensuring $g(\eta) \int h(X) \exp(\eta^T u(X)) = 1$

random message: taking gradient of normalization on both sides

rearranging and using normalization condition

$$-\nabla \log g(\eta) = \mathbb{E}[u(X)]$$

For $X = \{x_1 \ldots x_N\}$ iid data, we have likelihood

$$p(X|\eta) = \left(\prod_{n=1}^N h(x_n)\right) g(\eta)^N \exp\left(\eta^T \sum_{n=1}^N u(x_n)\right)$$

$$-\nabla \log g(\eta) = \frac{1}{N}\sum_{n=1}^N u(x_n)$$

ETH zürich

Mean Field for exponential family

$$q^*(z_j) \propto \exp \mathbb{E}_{-j}[\log p(z_j \mid Z_{-j}, x)]$$

Assume conditional in exponential family:

$$p(z_j \mid z_{-j}, x) = h(z_j) \exp(\eta(z_{-j}, x)^T t(z_j) - a(\eta(z_{-j}, x)))$$

Log of conditional: $p(z_j \mid z_{-j}, x) = \log(h(z_j)) + \eta(z_{-j}, x)^T t(z_j) - a(\eta(z_{-j}, x))$

expectation w.r.t. $q(z_{-j})$

$$\mathbb{E}[\log p(z_j \mid z_{-j}, x)] = \log(h(z_j)) + \mathbb{E}[\eta(z_{-j}, x)^T] t(z_j) - \mathbb{E}[a(\eta(z_{-j}, x))]$$

$$\Rightarrow q^*(z_j) \propto h(z_j) \exp(\mathbb{E}[\eta(z_{-j}, x)^T] t(z_j))$$

Note: for exponential family $q(z_j)$ is in the same family as conditional

Assuming variational parameter r

$$q(z_1 \text{ — } z_t \mid r) = \prod_{i=1}^{t} q(z_i \mid r_i)$$ i.e. q factorizes

r variational parameter

$$r_j^* = \mathbb{E}[\eta(z_{-j}, x)]$$

Conjugate priors for exp. family

$$p(x_i \mid \eta) = h_i(x) \exp(\eta^T t(x) - a_i(\eta))$$

$$p(\eta \mid \lambda) = h_c(x) \exp(\lambda_1^T \eta + \lambda_2^T(-a_i(\eta)) - a_c(\lambda))$$

$\lambda = (\lambda_1, \lambda_2)$ has dimension $\dim(\eta)+1$

Let β be vector of global latent variables (prior) and z vector of local latent variables

$$q(z, \beta) = q_\lambda(\beta) \prod_{i=1}^{t} q_{\varphi_i}(z_i)$$

while not ELBO converged

for each data point i

$\varphi_i \leftarrow \mathbb{E}_\lambda[\eta_i(\beta, x_i)]$ ← local update

Update $\lambda \leftarrow \mathbb{E}_\varphi[\eta_g(x, z)]$ ← global update

ETH zürich

Expectation Propagation

Instead of minimizing $KL(q\|p)$, we minimize $KL(p\|q)$

Assume $q(x|\eta) = h(x)g(\eta)\exp(\eta^T u(x))$

Then, $KL(p\|q)(\eta) = -\log g(\eta) - \eta^T \mathbb{E}_{p(z)}[u(z)] + \text{const}$

Set gradient to zero: $\cancel{\nabla g(\eta) = \mathbb{E}_{p(z)}[u(z)]}$

$\cancel{\mathbb{E}_{q(z)}[u(z)]} - \nabla \log g(\eta) = \mathbb{E}_{p(z)}[u(z)]$

$$\Rightarrow \mathbb{E}_{q(z)}[u(z)] = \mathbb{E}_{p(z)}[u(z)]$$

Optimal solution implies matching of expected sufficient statistics

Example: If $q(z)$ is Gaussian $\mathcal{N}(z|\mu, \Sigma)$ then we minimize the KL by setting μ equal to mean of $p(z)$ and Σ equal to covariance of p. (so called moment matching)

Factorized posteriors

Problem: $KL(q\|p)$ requires expectations wrt q while $KL(p\|q)$ requires expectations wrt p – intractable

Assume:
$$p(\mathcal{D}, \theta) = \prod_{i=1}^{N} f_i(\theta)$$
$$q(\theta) = \frac{1}{Z_{EP}} \prod_{i=1}^{N} \tilde{f}_i(\theta)$$

Goal: minimize
$$KL\left(\frac{1}{p(\mathcal{D})}\prod_{i=1}^{N} f_i(\theta) \,\Big\|\, \frac{1}{Z_{EP}}\prod_{i=1}^{N}\tilde{f}_i(\theta)\right)$$

intractable problem,

instead update factors iteratively by moment matching

ETH zürich

Alg:

1) Input: Initialized $\tilde{f}_i(\cdot)$
2) Until Convergence:
3) For each factor i:
4) delete factor i $q^{\setminus i} = \frac{q(\theta)}{\tilde{f}_i(\theta)} = \prod_{h \neq i} \tilde{f}_h(\theta)$
5) Projection $\tilde{f}_i^{new} = \arg\min_{\tilde{f}_i} KL(f_i(\theta) q^{\setminus i}(\theta) \| \tilde{f}_i(\theta) q^{\setminus i}(\theta))$
6) Update $q = \tilde{f}_i^{new}(\theta) q^{\setminus i}(\theta)$
7) Return $p(\mathcal{D}) \approx \int \prod_h \tilde{f}_h(\theta) d\theta$

<u>Note:</u> Main application GP
No guarantee of convergence

<u>α-Divergence</u>

$$D_\alpha(p\|q) = \frac{\int \alpha p(x) + (1-\alpha) q(x) - p(x)^\alpha q(x)^{1-\alpha}}{\alpha(1-\alpha)} dx$$

Props: convex, ~~non~~ nonnegative, $0 \Leftrightarrow p = q$

$\lim_{\alpha \to 0} D_\alpha(p\|q) = KL(q\|p)$

$\lim_{\alpha \to 1} D_\alpha(p\|q) = KL(p\|q)$

α large positive - q covers all modes (avoid false negatives)

explanation: α large negative - q covers mode with highest prob mass

For computation of marginals $\alpha = 1$ is best

$\alpha = -1$ zero forcing - avoiding ~~[illegible]~~ false positives
- small $p(x)$ forces q to be small

ETH zürich

MCMC

Sampling based inference

- Rejection sampling
- Metropolis-Hastings (Gibbs sampling)
  - more exact, slower (takes time to converge)
  - create a markov chain that is efficient to simulate and has stationary distribution

Given unnormalized distribution $Q(x)$

Build MC with stationary distribution $\pi(x) = \frac{1}{Z} Q(x)$

Detailed balance $Q(x)\, P(x'|x) = Q(x')\, P(x|x')$

Proposal distribution $R(x'|x)$

Acceptance distribution: $x_t = x$

x' - sample from $R(x'|x)$

with prob $\alpha = \min\left\{1, \frac{Q(x')\, R(x|x')}{Q(x)\, R(x'|x)}\right\}$

set $x_{t+1} = x'$

else $x_{t+1} = x$

Advantages:

easy to implement

very accurate when converged

Disadvantages:

take time to converge

ETH zürich Extentions to VI

Stochastic Variational Inference

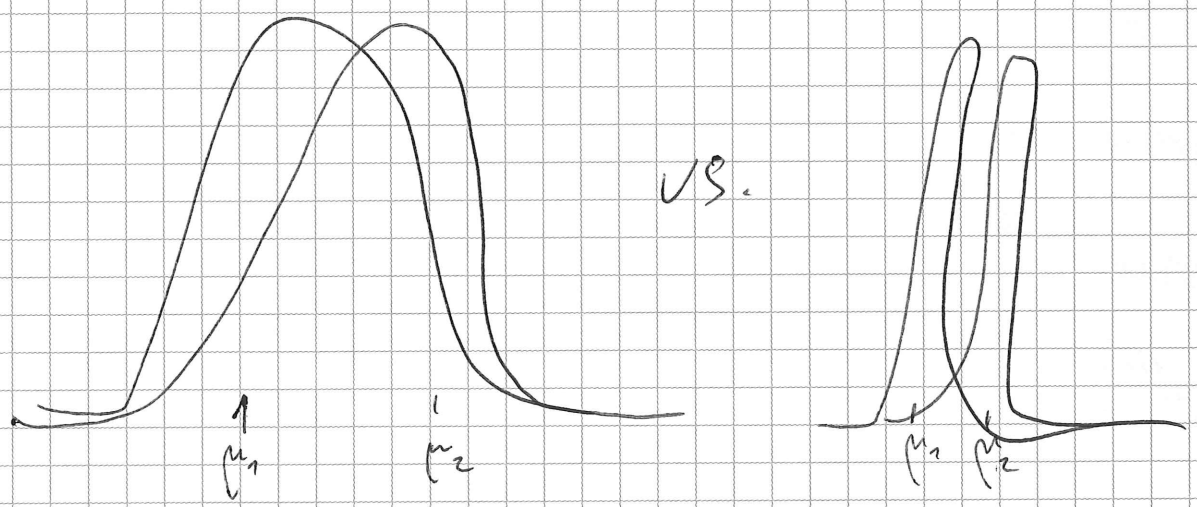
$$\lambda_{t+1} = \lambda_t + \delta \nabla_\lambda f(\lambda_t)$$

equally $\underset{d\lambda}{\operatorname{argmax}}\ f(\lambda + d\lambda)$ s.t. $\|d\lambda\|^2 \leq \varepsilon$

Euclidean distance is a problem

Natural Gradient $\underset{d\lambda}{\operatorname{argmax}}\ ELBO(\lambda + d\lambda)$ s.t. $D_{KL}^{sym}(q_\lambda, q_{\lambda+d\lambda}) \leq \varepsilon$

where $D_{KL}^{sym}(q,p) = KL(q\|p) + KL(p\|q)$



Riemannian metric $G(\lambda)$ such that $d\lambda^T G(\lambda) d\lambda = D_{KL}^{sym}(q_\lambda(\beta), q_{\lambda+d\lambda}(\beta))$

$$\hat{\nabla}_\lambda ELBO = G^{-1}(\lambda) \nabla_\lambda ELBO$$

where $G(\lambda) = \mathbb{E}[(\nabla_\lambda \log q_\lambda(\beta))(\nabla_\lambda \log_\lambda(\beta))^T]$

for exponential family $\hat{\nabla}_\lambda ELBO = \mathbb{E}[\eta(x,z)] - \lambda$

Gradient step $\lambda_t = \lambda_{t-1} + \delta_t \nabla_{\lambda_{t-1}} ELBO$, where δ_t is the step size

Thus $\lambda_t = (1-\delta_t)\lambda_{t-1} + \delta_t \mathbb{E}[\eta(x,z)]$

Mean Field with natural gradient

Input: model p, variational family $q_\varphi(z)$, $q_\lambda(\beta)$

while ELBO not converged

for each datapoint i

Update $\varphi_i \leftarrow \mathbb{E}_\lambda[\eta_i(\beta, x_i)]$

Update $\lambda \leftarrow \lambda + \delta \nabla_\lambda ELBO = (1-\delta_t)\lambda + \delta \mathbb{E}_{q(\varphi)}[\eta_g(x, z)]$

Stochastic Optimization

→ use noisy gradient

→ convergence guaranteed to local optimum when $\sum_{t=1}^{\infty} \delta_t = \infty$, $\sum_{t=1}^{\infty} \delta_t^2 < \infty$

Input: model p, variational family $q_\varphi(z)$, $q_\lambda(\beta)$

while stopping criteria not fulfilled

Sample index $j \sim Uniform(1 \dots n)$

Update $\varphi_i \leftarrow E_\lambda[\eta_i(\beta, x_i)]$

Compute global parameters estimate $\hat{\lambda} = \mathbb{E}_\varphi[\eta_g(x_j, z_j)]$

Optimize the global variational parameter $\lambda_{t+1} \leftarrow \lambda_t(1-\delta_t) + \delta_t \hat{\lambda}$

Check step size and update if needed

Black Box variational inference

$$ELBO = \mathbb{E}_{q_r}[\log p_\theta(z, x)] + \mathbb{E}_q[\log q_r(z)]$$

Aim: Maximize ELBO

Problem: Need unbiased estimates of $\nabla_{r,\theta}$ ELBO

Reparametrization trick:

$$\mathbb{E}_{q_r}[f_r(z)] = \mathbb{E}_q[f_r(g_r(\varepsilon))]$$

expectation does not depend on r

Then $\nabla_r \mathbb{E}_q[f_r(g_r(\varepsilon))] = \mathbb{E}_q[\nabla_r f_r(g_r(\varepsilon))]$

Obtain unbiased estimates by taking MC estimate of expectation on the right

ETH zürich

Score functions

$$\mathbb{E}_{q(z)}[\log p(z,x) - \log q(z)] = \mathbb{E}[g(z)] \qquad \nabla_\lambda \log q(z;\lambda) = \frac{\nabla_\lambda q(z;\lambda)}{q(z;\lambda)}$$

score function

$$\nabla_\lambda \text{ELBO} = \nabla_\lambda \mathbb{E}[g(z)] = \int \nabla_\lambda g(z)\, q(z)\, dz =$$

$$= \int \frac{q(z;\lambda)}{q(z;\lambda)} \nabla_\lambda q(z;\lambda)\, g(z)\, dz$$

$$= \int q(z;\lambda) \nabla_\lambda \log q(z;\lambda)\, g(z)\, dz = \mathbb{E}_{q(z)}[\nabla_\lambda g(z) \nabla_\lambda \log q(z;\theta)] \overset{\text{Approx B}}{=} \mathbb{E}_{q(z)}[g(z) \nabla_\lambda \log q(z;\theta)]$$

$\nabla_\lambda g(z)$ → $g(z)$ can be taken out

Alg: Black box variational inference

Input: data x, model $p(x,z)$, variational family $q_\varphi(z)$

while stopping criteria not fulfilled:

Draw L samples $z_\ell \sim q_\varphi(z)$

Update variational parameter using collected samples

$$\varphi \leftarrow \varphi + \delta_t \frac{1}{L} \sum_{\ell=1}^{L} \nabla \log q(z_\ell)(\log p(x,z_\ell) - \log q(z_\ell))$$

Check step size and update if needed

Note: active research on how to reduce the variance of noisy gradient

Disadvantage: takes long, noisy variant

Control Variates

use modified estimator ~~$\nabla_\theta \mathbb{E}_{p(z;\theta)}[f(z)] = \mathbb{E}_{p(z;\theta)}[(f(z)-2)\nabla_\theta \log p(z;$~~

$$\nabla_\lambda \mathbb{E}_{q(z;\lambda)}[g(z)] = \mathbb{E}_{q(z;\lambda)}[g(z) - \theta) \nabla_\theta \log q(z$$

(original BBVI) compare with $\nabla_\lambda \mathbb{E}_{q(z;\lambda)}[g(z)] = \mathbb{E}_{q(z;\lambda)}[g(z) \nabla_\theta \log q(z$

θ is control variate - any term that has zero mean

- they affect variance

~~$\lambda = \nabla_\lambda \log q$~~ $\theta = \nabla_\lambda \log q(z;\lambda)$ b/c $\mathbb{E}_q \theta = 0$

Variational Autoencoders

- explicit generative probabilistic model
  → specify explicit (parametrized) distribution + use MLE
  (most models in PGM(A))

- x observations, z hidden vars, α additional params

$$p(z|x,\alpha) = \frac{p(z,x|\alpha)}{\int p(z,x|\alpha)\,dx}$$

☺ Pick dist. family over latent variables with its own variational ~~family~~ parameter

$$q(z|\nu) = \dots$$

⇒ find ν s.t. $p(z|x)$ close to q

⇒ find lower bound [Choose q s.t. $\mathbb{E}$ easy to compute!]

$$\log p(x) = \log \int_z p(z,x) = \log \int_z p(z,x)\,\frac{q(z)}{q(z)}$$

$$= \log \mathbb{E}_q\left[\frac{p(z,x)}{q(z)}\right]$$

$$\overset{\text{Jensen}}{\geq} \mathbb{E}_q\left[\log \frac{p(z,x)}{q(z)}\right] = \underbrace{\mathbb{E}_q[\log p(z,x)] - \mathbb{E}_q[\log q]}_{\text{ELBO}}$$

$$KL[q(z)\|p(z|x)] = \mathbb{E}_q\left[\log \frac{q(z)}{p(z|x)}\right] = \dots$$

$$\boxed{\log p(x) = KL + \text{ELBO}} \quad -(\mathbb{E}_q[\log(z|x)] - \mathbb{E}_q[\log q]) + \log p(x)$$

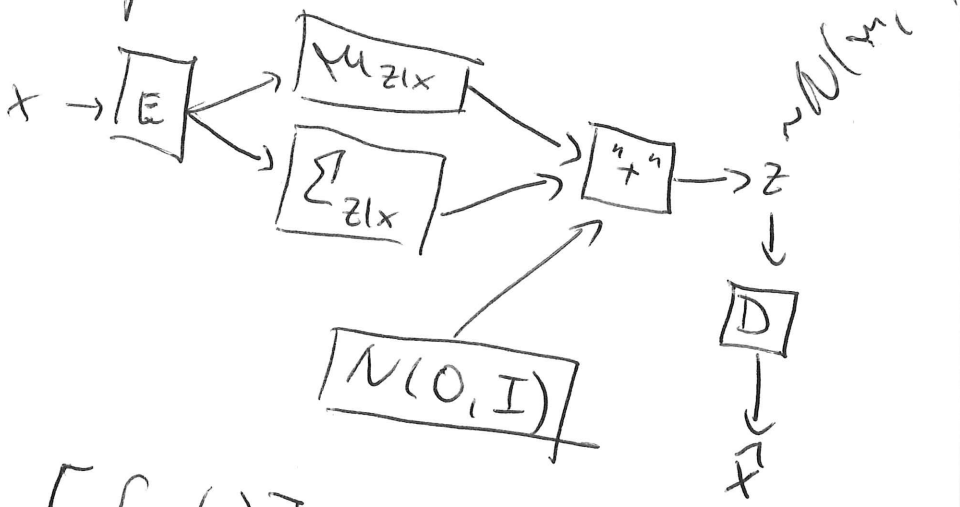
VAE: $\overset{+\text{reparam.}}{\text{encoder}}$ → lat. space → decoder → orig. space
"q" "p(x,z)"

Want q & p parametrized $q(z|x;\theta)$, $p(x|z;\varphi)$ and want them to be differentiable, e.g. NNets

$$\underset{\text{encoder}}{E_\theta(x)} = z \;,\; \underset{\text{decoder}}{D_\varphi(z)} = \hat{x}$$

, both diff. able

Want to be able to <u>backpropagate</u> through the stochastic operation (z-sampling) between E & D. ⇒ e.g. for Gaussians:



In general:

Want $\nabla_\nu \mathbb{E}_{q_\nu}[f_\nu(z)]$

Assume $\exists\, g, \varepsilon$ s.t. $\mathbb{E}_{q_\nu}[f_\nu(z)] = \underbrace{\mathbb{E}_q}_{\text{no } \nu \text{ dependence}}[f_\nu(g_\nu(\varepsilon))]$

$$\Rightarrow \nabla_\nu \mathbb{E}_{q_\nu}[f_\nu(z)] = \underline{\mathbb{E}_q[\nabla_\nu f_\nu(g_\nu(\varepsilon))]}$$

use MC to get unbiased est.

! Prior over latent vars (usually Gaussian) enables us to "walk" over the latent space.

⊕ Continuous changes, nicely parametrizable

⊖ Less sharp samples (prior too simplistic)

→ VAEs not asymptotically consistent to posterior!

⊕ If latent space factorizes ⇒ interpretability.

Loss fn of (β-) VAE .. (VAE for β=1)

$$\mathcal{F}(\theta, \phi, \beta; x, z) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$$

encoder decoder

↓ use SGD on this

$$-\beta D_{KL}(q_\phi(z|x) \,\|\, p(z))$$

encoder prior on lat. space,

e.g. $\phi = (\mu_{z|x}, \Sigma_{z|x})$ e.g. $p(z) = \mathcal{N}(z|0, I)$

GD $J(\theta)$

If loss diffable ⇒ for $t = 1 \ldots$

~~for $i = 1 \ldots N$~~

~~compute $\nabla J(\theta)$~~

$$\theta_{t+1} \leftarrow \theta_t - \alpha_t \underbrace{\nabla_\theta J(\theta; D)}_{\text{all samples!}}$$

SGD

for $t = 1 \ldots$

sample $x_t \sim D$ (possibly minibatch)

$$\theta_{t+1} \leftarrow \theta_t - \alpha_t \nabla_\theta J(\theta; \{x_t\})$$

Properties of SGD

- Asymptotically converges to a local min of $J(\theta)$ (just like GD)
- Uses a noisy estimate of the gradient (single sample / batch)
- Learning rate conv. conditions (Robbins-Monro)

$$\sum^{\infty} \alpha_t = \infty \quad \& \quad \sum^{\infty} \alpha_t^2 < \infty$$

$\boxed{\beta\text{-VAE}}$ → cost fn above.

$\beta \in [0, \infty)$ weights the KL term in $\mathcal{F}$.

Desired effects

$\mathcal{F}$: min VAE loss s.t. $KL < \delta$

- Low $\beta < 1$ ... "likelihood" preferred
  ⇒ less blurry, less coherent
- High $\beta > 1$ ... "enforce" prior more than reconstruction ⇒ more disentangl.
- Still simplistic prior
- Learns disentangled latent factors

VAE from BBVI

2 approaches to VI:
- score fn (BBVI) → ~~using~~ Estimator has high variance
- Pathwise (reparam, VAE)
  ↳ needs diffable p ⇒ p continuous (can't be discrete)
  → better variance

- BBVI has high variance for the gradient estimator → what can we do?
- Reparametrize
  → reformulate ELBO so it has closed form + reparametrize
  → encoder, decoder, all diffable

$$p(z) = \mathcal{N}(0, I)$$

$$\Rightarrow \varepsilon \sim p(z)$$

$$\tilde{z} = \mu_{z|x} + \sigma \cdot \varepsilon_{z|x}$$

4. Dimensionality reduction

4.1 Derive ~~factor models PCA~~ Factor models (FA)

Data $x^{(i)} \in \mathbb{R}^n$. What if $n \gg m$?
↳ # samples

⇒ Dimensionality reduction → data have underlying low-dim. data manifold
↳ continuous vars → FA, PCA, ICA
↳ discrete: mixture models (clustering)

Factor analysis (FA)

$z \in \mathbb{R}^k$ latent, x observed, $k < n$ (dim. red)

$$z \sim N(0,1)$$
$$x|z \sim N(\underbrace{\mu}_{\mathbb{R}^n} + \underbrace{\Lambda}_{\mathbb{R}^{n\times k}} z, \Psi)$$

↳ diagonal $\mathbb{R}^{n\times n}$

Equivalently

$$\left.\begin{array}{l} z \sim N(0,1) \\ \varepsilon \sim N(0,\Psi) \end{array}\right\} \text{indep.}$$
$$x = \mu + \Lambda z + \varepsilon$$

joint model: $\begin{bmatrix} z \\ x \end{bmatrix} \sim N(\mu_{zx}, \Sigma)$

Goal: identify μ_{zx}, Σ

$$\Rightarrow \begin{bmatrix} z \\ x \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ \mu \end{bmatrix}, \begin{bmatrix} 1 & \Lambda^T \\ \Lambda & \Lambda\Lambda^T + \Psi \end{bmatrix}\right)$$

Marginal: $x \sim N(\mu, \Lambda\Lambda^T + \Psi)$

Log-likelihood of params: $\ell(\mu, \Lambda, \Psi) = \log \prod^m N(x^{(i)} | \mu, \Lambda\Lambda^T + \Psi)$

EM for FA

MLE using EM:

$\theta = (\mu, \Lambda, \Psi)$

E $\quad q^{t+1} = p(z|x, \theta^t)$ // marginalize from joint

M

$$\theta^{t+1} = \arg\max_{\theta} \sum_n \int_z \underbrace{q^{t+1}(z|x)}_{E\text{ step}} \underbrace{\log p(x, z | \theta)}_{\text{joint}} dz$$

$\Rightarrow$ Results: E: $\quad q^{t+1} = \mathcal{N}(z | \overset{q \times 1}{m^{(i)}}, V^{(i)})$

$$V^{(i)} = (1 + \Lambda^T \Psi^{-1} \Lambda)^{-1}$$

$$m^{(i)} = V^{(i)} \Lambda^T \Psi^{-1} (x - \mu)$$

M $\quad \Lambda^{t+1} = \left(\sum_i x^{(i)} m^{(i)T} \right) \left(\sum_i V^{(i)} \right)^{-1}$

$$\Psi^{t+1} = \frac{1}{n} \operatorname{diag}\left[\sum_i x^{(i)} x^{(i)T} + \Lambda^{t+1} \sum_i m^{(i)} x^{(i)T} \right]$$

Generalization

$\Rightarrow$ State space models $\quad x_t = A x_{t-1} + G w_t$

where $w_t = \mathcal{N}(0, Q)$

4.2 Principal Component Analysis (PCA)

- Data model Gaussian, same as FA

$$\hat{=} \; L(\Theta, D) = \dots =: -\frac{N}{2}\log|V| - \frac{1}{2}\operatorname{trace}(V^{-1}S)$$

model cov.

sample cov.

- If we set FA $\Psi = \sigma^2 I \Rightarrow$ PPCA.
  - For $\sigma^2 \to 0$ we get PCA.
  - MLE params same, except for σ^2

$\Rightarrow$ inference easy $\quad \lim_{\sigma^2 \to 0} \Lambda^T(\Lambda\Lambda^T + \sigma^2 I)^{-1} = \underbrace{\Lambda^T(\Lambda\Lambda^T)^{-1}}_{\text{orth. projection}}$

$\Rightarrow \mu_{z|x} = \Lambda^+(x - \mu)$

pseudoinverse

Λ^+ ... right inverse

$\Lambda\Lambda^+ = I$

- PPCA can be used in mixtures, with priors, also in a Bayesian setting can determine subspace size automatically.
  ↳ also can be run generatively!

vs. FA

- PCA:
  - rotation (data) unimportant ↳ invariant
  - scale important! (normalize)
  - $\Rightarrow$ looks for ↗ of largest variance
- FA
  - rot ~~unimp~~ important! ↳ covariant
  - scale unimportant
  - looks for ↗ of largest correlation

Latent covariance

Now: $z \sim \mathcal{N}(0, P)$

$x|z \sim \mathcal{N}(\mu + \Lambda z, \Psi)$

Marginal: $x \sim \mathcal{N}(\mu, \Lambda P \Lambda^T + \Psi)$

$\Rightarrow P = E D E^T$ decomp.,

$\Lambda := \Lambda E D^{1/2}$???

$\Rightarrow P := I$ (FA)

or

$\Rightarrow$ Force col of Λ to be orthogonal. (PCA)

Linear autoencoder

$m < n$, $x_i \in \mathbb{R}^n$ $i = 1 \dots N$

find $C \in \mathbb{R}^{m \times n}$, $D \in \mathbb{R}^{n \times m}$

s.t. $C^*, D^* = \underset{C, D}{\operatorname{argmin}} \sum_i^N \| x_i - DCx_i \|^2$

$z = Cx_i$

$\hat{x} = Dz$

$\Rightarrow$ Sol: PCA

PCA and SVD + variance / reconstruction error

$X \in \mathbb{R}^{m \times n}$ $\quad$ n - #samples, m - dim.

assume $X = \tilde{X} - \bar{x}$ where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} \tilde{x}_i$

$\Sigma = \frac{1}{n-1} X X^T$, Σ is symmetric & real

$\overset{\text{Schur}}{\Longrightarrow} \Sigma = \underset{m\times m}{Q} \underset{m\times m}{\Lambda} \underset{m\times m}{Q^T}$ $\quad$ Q orthonormal, Λ diag.

Then PCA for $k < m$ dim:

$\underset{n\times k}{Q^{(k)}}$ $\quad$ Then $Z = \underset{k\times m}{Q^{(k)T}} \underset{m\times n}{X}$

$\Rightarrow \underbrace{\tilde{X}^{(k)} = Q^{(k)} Z}_{m\times k \quad k\times n}$

Long form (min. reconstruction err PCA)

Basis: $\{q_i\}_{i=1}^{m}$ (orthonormal)

$\Rightarrow x = \sum_{i=1}^{m} (x^T q_i) q_i$

can $\rightarrow$ $\quad$ cannot depend on x anymore!

$\overset{\text{WLOG}}{\Longrightarrow} \tilde{x} = \sum_{i=1}^{k} a_i q_i + \sum_{i=k+1}^{m} b_i q_i$

$\overset{\text{Data}}{\Longrightarrow} J = \frac{1}{n} \sum_{i=1}^{n} \|\tilde{x}_i - x_i\|_2^2$

$= \frac{1}{n} \sum_{i=1}^{n} \| \sum_{j=1}^{k} a_j q_j + \sum_{j=k+1}^{m} b_j q_j - x_i \|_2^2$

$$\frac{\partial J}{\partial a_{ij}} \overset{!}{=} 0 \quad \Rightarrow \quad a_{ij} = x_i^T q_j$$

$$\frac{\partial J}{\partial b_j} \Rightarrow b_j = \bar{x}^T q_j$$

$\Rightarrow$ PCA reconstr. error:

$$\Rightarrow x_i - \tilde{x}_i = \sum_{j=k+1}^{m} \left(x_i^T q_j\right) q_j - \sum_{j=k+1}^{m} \left(\bar{x}^T q_j\right) q_j$$

$$= \underbrace{\sum_{j=k+1}^{m} (x_i - \bar{x})^T q_j \; q_j}_{\text{in space 1 to principal subspace } \mathrm{Span}(\{q_j\}_{j=1}^{k})}$$

$$\Rightarrow J = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=k+1}^{m} \left(x_i^T q_j - \bar{x}^T q_j\right)^2$$

... b.c. $\boxed{q_j^T q_j = 1 \qquad q_i^T q_j = 0 \quad i \neq j}$

$$= \sum_{j=k+1}^{m} q_j^T S q_j$$

where $S = \frac{1}{n}(X - \bar{x})(X - \bar{x})^T$

data covariance

$$= \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T$$

$\Rightarrow$ hence if $\{q_j\}_{j=1}^{m}$ are eigenvectors of S

$$\Rightarrow J = \sum_{j=k+1}^{m} \lambda_j .$$

PCA & SVD part 2

$X \in \mathbb{R}^{m \times n} \Rightarrow X = \underset{m\times m}{U} \overset{\text{diag.}}{\underset{m\times n}{S}} \underset{n\times m}{V^T}$

$$\Rightarrow \underset{m\times m}{\Sigma} = \frac{1}{n-1} X X^T = \frac{1}{n-1} U S V^T V S^T U^T$$

$$= \underset{m\times m}{U} \underbrace{\frac{S S^T}{n-1}}_{m\times n} \underbrace{U^T}_{m\times n}$$

☺ Schur //

$$\Rightarrow \underline{\underline{U^T X}} = U^T U S V^T = \underline{\underline{S V^T}}$$

Z //

$$\Rightarrow Z^{(k)} = \underset{k\times m}{U^{(k)T}} \underset{m\times n}{X}$$

$$\tilde{X}^{(k)} = \underset{m\times k}{U^{(k)}} \underset{k\times m}{U^{(k)T}} \underset{m\times n}{X}$$

$$\hat{X}^{(k)} = U^{(k)} S^{(k)} V^{(k)T}$$

☺ Eckart-Young

$$\hat{X}^{(k)} = \underset{A}{\operatorname{argmin}} \|X - A\|_{F/2}$$

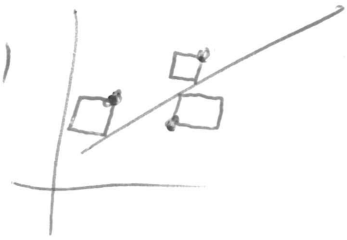
Proof $\|\cdot\|_2 \Rightarrow$ SVD of X & def $\|\cdot\|_2$

$Aw = 0$

$\Rightarrow \hat{X}^{(k)} = \tilde{X}^{(k)}$, bc. $\tilde{X}^{(k)}$ minimizes reconstr. error

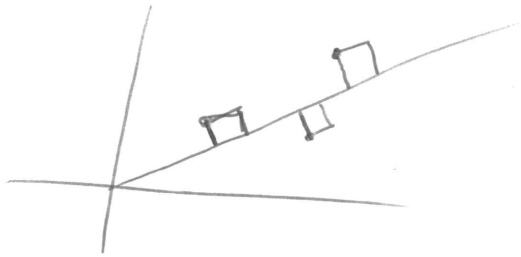
PCA vs. least squares (1D)

Least squares (OLS)



min complementary proj to PCA

PCA



orth. proj

<u>Note</u> $\|x\|^2 = \|Ix\|^2 = \underbrace{\|(I - QQ^T)x\|^2}_{\text{complement (OLS)}} + \underbrace{\|QQ^Tx\|^2}_{\text{PCA}}$

$\Rightarrow \max_Q \|QQ^Tx\|^2 = \min_Q \|(I - QQ^T)x\|^2$

if Q fixed
$\Rightarrow$

$\Rightarrow \|(I - QQ^T)x_i\|^2 = \min_{w_i} \|x_i - Qw_i\|$

where $w_i^* = Q^Tx_i$

$\Rightarrow$ for a Q, PCA reproj. provides the best linear approx for X.

Eigenfaces

Idea: efficient representation + remove noise, lighting, etc.

→ use PCA to form basis of $K \ll N$ images

eigenface ≡ eigenvector of cov. fn.

⇒ use lower-dim. repr. for classification, ...

Alg. Downsample img, flatten, run PCA (w/ SVD).

Problems:
- Images are large (M big) ⇒ SVD hard
- Eigenfaces do not describe parts/features of faces, but rather dense ~~cov.~~ directions
  ↳ not sparse
- Reconstruction blurry, converges to "mean face" → removes glasses, changes hairstyles, ...
- doesn't deal well with varying face positions, or scaling

Kernel trick w/ Eigenfaces

- Will learn to represent non-linear features (i.e. not brightness, but "glasses")
- Disadvantages → even more expensive to compute
  → choosing the kernel + hyperparams
- Adv → task may become linearly sep. in the latent space

0#6

Kernel PCA

→ can be derived even if mean of $\phi(x)$ is not 0

→ e.g. in SVM $y(x') = \sum_{i=1}^{n} w_i y_i k(x_i, x')$

Kernel trick: instead of scalar products, use an implicit sc. prod. function that satisfies sc. product axioms.

e.g. Linear kernel $K(x, x') = x^T x'$

Poly $K(x, x') = (x^T x')^d + 1$

Is equivalent to a feature map + standard scalar product. Easier to represent than feature maps!

$\varphi \leftrightarrow K$ works, but possibly in $\mathbb{R}^\infty$

$\rightsquigarrow$: $K(x, x') = \varphi(x)^T \varphi(x')$

Mercer's thm

$K(\cdot, \cdot) \to \mathbb{R}$ sym., ~~Non-neg. defi~~ PSD

(~~$\iint K(x,y) c$~~

$\sum_i \sum_j K(x_i, x_j) c_i c_j \geq 0$

$\forall x_1 \dots x_n \in [a, b]$

$\forall c_1 \dots c_n \in \mathbb{R}$

$\Rightarrow$ it is a kernel.

(aka, has a feature map in ∞ dim space).

Kernel PCA

~~The~~ data $\{x_n\}_{n=1}^N$ $x_n \in \mathbb{R}^D$

PCA: eigenvectors of Σ. $\Sigma u_i = \lambda_i u_i$ $i = 1 \dots D$

Want: express PCA s.t. x only appears in $x^T x'$.

Consider: $\Phi : \mathbb{R}^D \to \mathbb{R}^m$, $m >> D$.

$\to$ Can do PCA with $\Phi(x_n) \Rightarrow$ but $\sim \sqrt{0}$, possibly $m = \infty$

$$\Rightarrow \quad \Sigma = \frac{1}{N} \sum_{n=1}^{N} \phi(x_n)\, \phi(x_n)^T$$

$$\Rightarrow \quad \Sigma v_i = \lambda_i v_i \qquad i = 1 \dots m$$

$$\Rightarrow \quad \frac{1}{N} \sum_{n=1}^{N} \phi(x_n)\, \phi(x_n)^T v_i = \lambda_i v_i$$

$$\text{if } \lambda_i > 0 \Rightarrow v_i = \sum_{n=1}^{N} a_{in}\, \phi(x_n)$$

Back substitute

$$\frac{1}{N} \sum_{n=1}^{N} \phi(x_n)\, \phi(x_n)^T \sum_{m=1}^{N} a_{im}\, \phi(x_m) = \lambda_i \sum_{n=1}^{N} a_{in}\, \phi(x_n)$$

$$/ \cdot \phi(x_l)^T \qquad \dots + k(x, x') = \phi(x)^T \phi(x')$$

$$\Rightarrow \frac{1}{N} \sum_{n=1}^{N} k(x_l, x_n) \sum_{m=1}^{N} a_{im}\, k(x_n, x_m) = \lambda_i \sum_{n=1}^{N} a_{in}\, k(x_l, x_n)$$

$$\Rightarrow K^2 a_i = \lambda_i N K a_i$$

$$a_i = \begin{pmatrix} a_{1i} \\ \vdots \\ a_{ni} \end{pmatrix}$$

$\Rightarrow$ Can solve eig val problem

$$K \underline{a_i} = \lambda_i N \underline{a_i}$$

+ normalization conditions: $1 = v_i^T v_i = a_i^T K a_i = \lambda_i N a_i^T a_i$

$$\Rightarrow \quad z_i(x) = \phi(x)^T v_i = \sum_{n=1}^{N} a_{in}\, k(x, x_n) \qquad (i\text{-th coord})$$

Kernel PCA (cont.)

Disadvantage $K \in \mathbb{R}^{n \times n}$

- ⊖ Have to find eigenvectors of $N \times N$ matrix instead of $D \times D$ in PCA.
- ⊖ Cannot do low rank approx. in ϕ
  $\Rightarrow$ because pre-image of the low-rank might not exist. (can do approx.)

Adv.

- ⊕ Can model non-linear features.

Sequential Data

Explain HMM & KF (inference & learning)

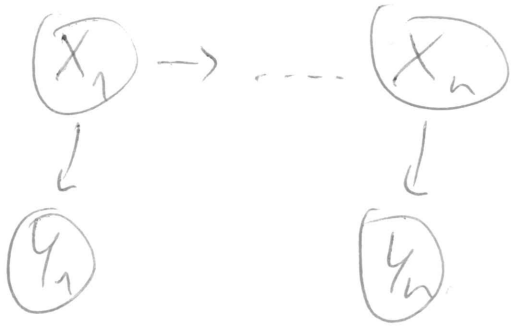
Dynamic systems - state space models

↳ HMM: hidden states X_i: discrete categorical
observed Y_i: categorical or arbitrary

↳ KF: X_i, Y_i } Gaussian dists

Markov assumption:

$$X_{1:t-1} \perp X_{t+1:T} \mid X_t \quad \forall t > 1$$



Inference: Bayesian Filtering! (DP)

a) Filtering $P(X_t \mid Y_{1:t})$

b) Smoothing $P(X_t \mid Y_{1:T})$ $1 \le t < T$.

FW-BW (sum-product)

c) Prediction $P(X_{t+\tau} \mid Y_{1:t})$ $\tau \ge 1$

d) Most probable explanation

$$\underset{X_{1:T}}{\operatorname{argmax}}\, P(x_{1:T} \mid y_{1:T})$$

↳ Viterbi (max product)

Need
- Prior $P(X_1)$
- Transition probs. $P(X_t \mid X_{t-1})$ → motion model
- Emission prob. $P(Y_t \mid X_t)$ → sensor model

Learning HMMs (using EM)

- $\log P(X_{1:T}, Y_{1:T}) = \log P(X_1) + \sum_{t=1}^{T} \log P(Y_t|X_t) + \sum_{t=2}^{T} \log P(X_t|X_{t-1})$

  X_t ... one-hot $\{1 \dots K\}$

  $$\Rightarrow P(X_t|X_{t-1}) = \prod_{i=1}^{K} \prod_{j=1}^{K} A_{ij}^{X_{t,i} \cdot X_{t-1,j}}$$ ← matrix multiplication!

  s.t. $A_{ij} \in [0,1]$ $\forall i,j$, $\sum_j A_{ij} = 1 \quad \forall i$

  $$\Rightarrow \log P(X_t|X_{t-1}) = \sum_i^K \sum_j^K X_{t,i} \cdot X_{t-1,j} \cdot \log A_{ij}$$

  $$= \underline{X_t^T (\log A) X_{t-1}}$$

- prior $P(X_1)$ ... $\pi_i = P(X_{1,i} = 1) \Rightarrow \pi \in \mathbb{R}^K$

  $$\Rightarrow P(X_1|\pi) = \prod_{i=1}^{K} \pi_i^{X_{1,i}} \Rightarrow \log P(X_1) = \sum_{i=1}^{K} X_{1,i} \log \pi_i$$

  $$= X_1^T (\log \pi)$$

- Y_t discrete $\{1 \dots D\}$

  $$\log P(Y_t|X_t) = Y_t^T (\log B) X_t$$

  $B \in \mathbb{R}^{D \times K}$ $\quad B_{ij} \in [0,1] \ \forall i,j$

  $\sum_j B_{ij} = 1 \quad \forall i$

- $\theta = (A, B, \pi)$

  Goal: $\arg\max_\theta \log P(Y)$

<u>M-step</u>

$$A_{ij} \propto \sum_{t=2}^{T} \mathbb{E}[X_{t,i} \cdot X_{t-1,j}]$$

$$Z = \sum_{t=2}^{T} \mathbb{E}[X_{t-1,j}]$$

$$\pi_i \leftarrow \mathbb{E}[X_{1,i}]$$

$$B_{di} \leftarrow \sum_{t=1}^{T} Y_{td} \, \mathbb{E}[X_{t,i}] \Big/ \sum_{t=1}^{T} \mathbb{E}[X_{t,i}]$$

E-step

Forward alg. $\alpha_t = \cancel{P(Y_{t+1:T} | X_t)}\ P(X_t, Y_{1:t})$

$$\cancel{= \left[\sum_{X_{t-1}} P(X_{t-1}, Y_{1:t-1}) P(X_t | X_{t-1})\right] P(Y_t | X_t)}$$

$$= P(Y_{1:t} | X_t) P(X_t) = P(Y_t | X_t) P(Y_{1:t-1} | X_t) P(X_t)$$

$$= P(Y_t | X_t) P(Y_{1:t-1}, X_t) = P(Y_t | X_t) \sum_{X_{t-1}} P(Y_{1:t-1}, X_{t-1}, X_t)$$

$$= P(Y_t | X_t) \sum_{X_{t-1}} P(Y_{1:t-1}, X_t | X_{t-1}) P(X_{t-1})$$

$$= P(Y_t | X_t) \sum_{X_{t-1}} \underbrace{P(Y_{1:t-1} | X_{t-1}) P(X_t | X_{t-1}) P(X_{t-1})}_{\alpha_{t-1}}$$

$$= P(Y_t | X_t) \sum_{X_{t-1}} \alpha_{t-1} P(X_t | X_{t-1})$$

Backward alg.

$$\beta_t = P(Y_{t+1:T} | X_t).$$

$$= \cancel{P(X_{t+1}} \sum_{X_{t+1}} P(Y_{t+1:T}, X_{t+1} | X_t)$$

cond + Markov

$$= \sum_{X_{t+1}} P(Y_{t+1:T} | X_{t+1}) P(X_{t+1} | X_t)$$

$$= \sum_{X_{t+1}} \underbrace{P(Y_{t+2:T} | X_{t+1})}_{\beta_{t+1}} P(Y_{t+1} | X_{t+1}) P(X_{t+1} | X_t)$$

$$\Rightarrow \mathbb{E}[X_{t,i}] = \gamma_{ti} = \frac{\alpha_{t,i} \beta_{t,i}}{\sum_j \alpha_{t,j} \beta_{t,j}}$$

$$\mathbb{E}[X_{t,i} \cdot X_{t-1,j}] = \xi_{tij} = \frac{\alpha_{t-1,j} \cdot A_{ij} \cdot P(Y_t | X_{t,i}) \beta_{t,i}}{\sum_{k,l} \alpha_{t-1,k} A_{kl} P(Y_t | X_{t,l}) \beta_{t,l}}$$

MPE (Viterbi)

$$V_{1,k} = P(y_1 | k) \cdot \pi_k \qquad V_{t,k} = \max_{x \in \{1,\dots,K\}} P(y_t | k) A_{xk} V_{t-1,x}$$

+ set pointers

$$\Rightarrow x_T = \operatorname{argmax}_{x \in \{1,\dots,K\}} V_{T,x}$$

$$x_t = P_{tr}(x_{t,t+1}) \quad \text{for } t = T-1, \dots, 1$$

Linear Gaussian State Space Models (KF, RTSS)

KS ↳ Kalman Smoother

Assumption: $x_1 \sim N(\mu_1, Q_1)$

Linear dynamics $\Rightarrow P(x_{t+1} | x_t) = N(A x_t, Q)$

$$P(y_t | x_t) = N(C x_t, R)$$

$\Rightarrow$ likelihood $P(X_{1:T}, Y_{1:T}) = P(X_1) \prod_{t=2}^{T} P(X_t | X_{t-1}) \prod_{t=1}^{T} P(Y_t | X_t)$

Gaussian!

$\Rightarrow$ can do closed-form EM.

x appears in M step $\Rightarrow$ use expectations! (KS)

- Sum-product (Forward) $\Rightarrow$ Kalman Filter
- Backward (+forward) $\Rightarrow$ Kalman Smoother / Rauch-Tung-Striebel Smoother

↳ TODO: Explain difference ?! Same as HMM smoother vs. filter, but equations are bleah.

2.8 Deep Generative Models

2.8.1 Explain GANs + compare to VAEs.

Gen. models: $\theta^* = \arg\max_\theta \mathbb{E}_{x \sim p_{data}} \log p_{model}(x \mid \theta)$
(MLE)

GANs → Implicit density fn (direct)
→ Asymptotically consistent (not VAE!)
→ "Sharper samples than VAE"
→ less variance (mode collapse!)

Generator: $x = G(z; \theta^{(G)}) \approx p_{model}$
differentiable!

Discriminator: $D: (x; \theta^{(D)}) \longmapsto [0, 1]$

Cost fn: $J^{(D)}(\theta^{(D)}, \theta^{(G)}) = -\frac{1}{2} \mathbb{E}_{x \sim p_{data}} \log D(x)$
D, G different
$- \frac{1}{2} \mathbb{E}_z \log (1 - D(G(z)))$

⇒ standard cross-entropy

D: $\min_{\theta^{(D)}} J^{(D)}(\theta^{(D)}, \theta^{(G)})$

(0-sum game): $\Rightarrow J^{(G)} = -J^{(D)}$
$=: V(\theta^{(G)}, \theta^{(D)})$

G: $\theta^{(G)*} = \arg\min_{\theta^{(G)}} \max_{\theta^{(D)}} V(\theta^{(G)}, \theta^{(D)})$

Solving this game ~ $\min D_{JS}(p_{data}, p_{model})$
(only for convex G, D)

! Heuristically less saturating $J^{(G)} = -\frac{1}{2} \mathbb{E}_z \log D(G(z))$

MLE with GANs

$$J^{(G)} = -\frac{1}{2}\mathbb{E}_z \exp\left(\sigma^{-1}(D(G(z)))\right)$$

Divergence choice → Google lecture: doesn't matter (much)

But: a) $D_{KL}(p_{data} \| p_{model})$ b) $D_{KL}(p_{model} \| p_{data})$

MLE "JS"-like

OPT discriminator: $D(x) \quad \frac{p_{data}(x)}{p_{data}(x) + p_{model}(x)}$

Disadv

- Only continuous outputs (... WassGAN)
- Divergence → oscillation → between modes
  → mode collapse: all $z \mapsto$ one mode
  - order of min max matters!
- Evaluation still hard, no bounds

Adv
- Empirically better samples
- Faster (parallel) sampling compared to fully visible belief nets
  pixel rnn/cnn

---

vs VAEs

↳ explicit (approximate) density fn → "interpretable" $P(x)$

→ given x find z is easy, impossible for GAN

VAEGAN

$x \xrightarrow{enc} z \xrightarrow{dec} \hat{x} \rightarrow$ discriminator

$z \sim \mathcal{N}(m, \Sigma)$, $x' \rightarrow$ discriminator

$z_0 \sim \mathcal{N}(0, I)$

sometimes latent space too simple

$q_\theta(z|x)$

Evaluating Generative Models & their latent repr.

- $p(x)$ and $\hat{p}(x)$ have a lot of implicit or explicit params => very hard to ~~re~~evaluate all modes / "cases"
  - Hard to compare different "kinds" of gen models!
  - ~~It's usually~~
- For tractable likelihood models:
  - Evaluate likelihood on test split
  - Proxy to —"— might exist: ELBO (VAE)
- For intractable...
  - visual evaluation (subjective!) (qualitative!)
  - "metrics" → Inception Scores, Frechet InceptionDist, Kernel I. Dist.
    → img. heuristics (sharpness, diversity, ...)
  - downstream supervised task!
    - unclear what the exact influence is (too many unknowns in experiment)

Evaluating latent representations

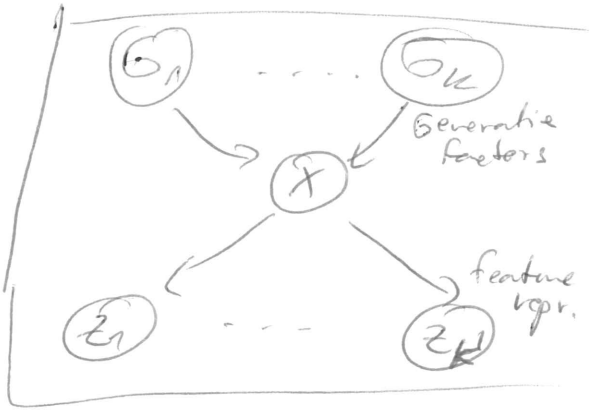
- clustering (verify with labels, e.g. MNIST) → Problem: How to eval clustering?!
- compression
- <u>disentanglement</u>
  ↳ splitting sources of variation

Robustness of lat. representations

Ground truth G

$\Rightarrow$ demand <u>sparse</u> row in mutual info matrix

$$I(Z_i, G_j)$$



Inf. theoretic based validation

Interventional robustness

PIDA $(L \mid g_I, g_J^\Delta) := d(\mathbb{E}[Z_L \mid do(G_I \leftarrow g_I)], \mathbb{E}[Z_L \mid do(G_I \leftarrow g_I, G_J \leftarrow g_J^\Delta)])$

$\hookrightarrow$ Postinterventional disagreement (in Z_L given to g_J^Δ given g_I)

EMPIDA $\rightarrow$ Expected Maximum PIDA

$$\text{EMPIDA}(L \mid I, J) := \mathbb{E}_{g_I}[\sup_{g_J^\Delta} \text{PIDA}(L \mid g_I, g_J^\Delta)]$$

IRS $\rightarrow$ Interventional Robustness Score

$$\text{IRS}(L \mid I, J) := 1 - \frac{\text{EMPIDA}(L \mid I, J)}{\text{EMPIDA}(L \mid \emptyset, \{1 \dots K\})} \in [0, 1]$$

$\hookrightarrow$ 1.0 for no harm done by G_J changes

$\hookrightarrow$ similar to R^2 for regression

L ... feature indices

G_I ... group of causal factors

g_I ... realization for a factor

Disentanglement score $\rightarrow$ Z_i robustly isolated from any causes

$$D_l := \max_{i \in \{1 \dots K\}} \text{IRS}(\{l\} \mid \{i\}, \{1 \dots K\} \setminus \{i\})$$

Independent Component Analysis (ICA)

↳ [nonlinear] latent var. model

Problem: How to balance disentanglement & reconstruction?

ICA

- Latent distrib ... Gaussian
- Observed vars related linearly to latent
- Independent → factorizes: $p(z) = \prod_{j=1}^{m} p(z_j)$

  ☺ To work, needs non-Gaussian latent,
  bc. Isotropic Gaussian (PCA) is rotation inv.
  bc. Σ is ~~given by~~ invariant.

  R orth $\Rightarrow X = XR \Rightarrow \Sigma = \frac{1}{n} XX^T = \frac{1}{n} X \underbrace{R^T R}_{=I} X^T$

$\Rightarrow$ Common choice $p(z_j) = \frac{1}{\pi \cosh(z_j)} = \frac{1}{\pi(e^{z_j} + e^{-z_j})}$

↳ heavy tails!

☺ Independent FA: all factors modeled by GMMs, Loglikelihood maximized by EM, reconstruction of latent vars using variational inference.

E.g. de-mix two people speaking (2 mics) → don't have original speech nor mixing weights
↳ constants, but unknown

$\Rightarrow$ ignore temporal, assume iid samples
2 lat vars (2 ppl), 2 observed vars (2 mics) | + SGD

→ Attempts a more "explicit" factor representation than current methods.

$\boxed{\text{WTF}}$ → Not sure how to compare this to current disentangled repr. learning. !