Static Facts:

$x,y\in\mathbb{R}^m$, $A\in\mathbb{R}^{m\times m}$

$\frac{dx^Ty}{dx}=y$ $\frac{dx^Tx}{dx}=2x$

$\frac{dx^TA}{dx}=A$ $\frac{dx^TAx}{dx}=(A+A^T)x$

Covariance:

$Cov(X,Y)=E[(X-E[X])(Y-E[Y])]$

For random vectors $X\in\mathbb{R}^m$, $Y\in\mathbb{R}^n$ then the m×n covariance matrix

$Cov(X,Y)=E[XY^T]-E[X]E[Y]^T$

Correlation:

$\rho_{X,Y}=\frac{cov(X,Y)}{\sigma_X\sigma_Y}=\frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X\sigma_Y}$

Variance $=E[X^2]-E[X]^2$

$E_{x\sim p(x)}[f(x)]=\int f(x)p(x)dx$

$E_{x\sim q(x|z)}[f(x)]=\int f(x)q(x|z)dx$

Exponential families of x given param η: $p(x|\eta)=h(x)g(\eta)\exp\{\eta^Tu(x)\}$

$u(x)$: natural statistic (sufficient)

$h(x)$: base measure (often constant)

$g(\eta)$: normalizer.

$g(\eta)\int h(x)\exp\{\eta^Tu(x)\}dx=1$.

E.g. Bernulli: $p(x|\mu)=\mu^x(1-\mu)^{1-x}$

$h(x)=1$, $u(x)=x$, $\eta=\ln\left(\frac{\mu}{1-\mu}\right)$.

$g(\eta)=\sigma(-\eta)=\frac{1}{1+e^{\eta}}$

Multinomial: $p(x_1,\dots,x_M|\mu)=\frac{N!}{\prod x_k!}\prod_{k=1}^{M}\mu_k^{x_k}$

$h(x)=\frac{N!}{\prod x_k!}$, $u(x)=\vec{x}$, $\eta=\begin{bmatrix}\ln\mu_1\\ \vdots\\ \ln\mu_K\end{bmatrix}$,

$g(\eta)=\begin{bmatrix}e^{\eta_1}\\ \vdots\\ e^{\eta_K}\end{bmatrix}$, where $\sum e^{\eta_i}=1$.

Gaussian: $p(x|\mu,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}\exp\{-\frac{1}{2\sigma^2}(x-\mu)^2\}$

$h(x)=(2\pi)^{-\frac{1}{2}}$, $g(\eta)=(-2\eta_2)^{\frac{1}{2}}\exp\left(\frac{\eta_1^2}{4\eta_2}\right)$

$\eta^T=\begin{bmatrix}\frac{\mu}{\sigma^2} & \frac{-1}{2\sigma^2}\end{bmatrix}$ $u(x)=\begin{bmatrix}x\\ x^2\end{bmatrix}$

Multivariate normal μ,Σ (cov)

$P(x_1,\dots,x_K|\mu,\Sigma)=\frac{1}{\sqrt{(2\pi)^k|\Sigma|}}\exp\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\}$

$h(x)=(2\pi)^{-\frac{k}{2}}$, $g(\eta)=\begin{bmatrix}-\frac{1}{2}\eta_2^{-1}\eta_1\\ -\frac{1}{2}\eta_2^{-1}\end{bmatrix}$

$\eta=\begin{bmatrix}\Sigma^{-1}\mu\\ -\frac{1}{2}\Sigma^{-1}\end{bmatrix}$ $u(x)=\begin{bmatrix}x\\ xx^T\end{bmatrix}$

Poission: $h(x)=\frac{1}{x!}$, $u(x)=x$, $\eta=\ln\lambda$

$g(\eta)=e^{\eta}$

Beta: $h(x)=\frac{1}{x(1-x)}$, $u(x)=\begin{bmatrix}\ln x\\ \ln(1-x)\end{bmatrix}$.

$\eta=\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$, $g(\eta)=\begin{bmatrix}\eta_1\\ \eta_2\end{bmatrix}$

MLE for Exponential family:

$-\nabla\ln g(\eta_{ML})=\frac{1}{N}\sum_{n=1}^{N}u(x_n)$

$P(\theta|d)=\frac{P(d|\theta)p(\theta)}{P(d)}$ $P(d|\theta)$ likelihood.

$P(\theta)$ prior. $P(\theta|d)$ posterior.

$\theta_{MLE}=\arg\max_\theta P(d|\theta)$

$=\arg\max_\theta\sum\log P(x_i|\theta)$

Global Markov property.

$s\perp t\mid V\setminus\{s,t\}$ pairwise Markov property

$\frac{\partial\sum\log P(x_i|\theta)}{\partial\theta}=0$, find $\hat{\theta}$.

(e.g. $\hat{\mu}$ and $\hat{\sigma}$ for normal).

MAP $\theta_{MAP}=\arg\max_\theta P(\theta|d)$

$=\arg\max_\theta\sum\log P(x_i|\theta)P(\theta)$

Conjugate prior: Conjugate priors induce a posterior of the same form: Gaussian for Gaussian, Gamma for poisson likelihood, Beta for Bernulli/binomial.

$\forall$ Exp family $\exists$ conjugate prior of $P(\eta|\chi,\nu)=f(\chi,\nu)g(\eta)^{\nu}\exp\{\nu\eta^T\chi\}$

multiply by likelihood, posterior

$P(\eta|D,\chi,\nu)\propto g(\eta)^{\nu+N}\exp\{\eta^T(\nu\chi+\sum_{n=1}^{N}u(x_n))\}$

$s\perp t\mid V\setminus\{s,t\}\Leftrightarrow G_{st}=0$

Discriminative:

$P(y|x)=P(k|x)=\frac{f(x,k,\theta)}{\sum_k f(x,k,\theta)}$

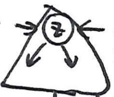
Generative:

$P(y|x)=\frac{P(x,y)}{P(x)}=\frac{P(x|k)P(k)}{P(x)}$

D-seperation: $X\to Z\to Y$ blocked

$X\leftarrow Z\to Y$

$X\to Z\leftarrow Y$



if z is not and no descent of z is observed, then $X\perp Y$.

Regression: $p(y|x)=\frac{p(y,x)}{p(x)}=\frac{p(y,x)}{\int p(y,x)dy}$

classification: $p(c|x)=\frac{p(c,x)}{p(x)}=\frac{p(c,x)}{\sum_c p(c,x)}$

clustering $p(c|x)=\frac{p(c,x)}{p(x)}$, c unknown.

Density estimation: $p(y|x)=\frac{p(y,x)}{p(x)}$

y unobserved.

If something is always unobserved, they are called hidden/latent.

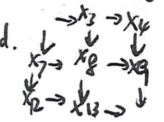
Complexity: DAG: m variables, each variable takes k different values.

Naive inference: Cost $O(k)$ operation to update each of $O(k^{m-1})$ table entries.

Markov blanket $\Leftrightarrow$ Independence

Markov blanket: parents, children, coparents.

$mb(8)=\{3,7\}\cup\{9,13\}\cup\{12,4\}$.



undirected graph also called markov random field (MRF)

In UGM: all neighbours.

$X_A\perp X_B|X_C$ iff C seperates A and B

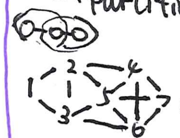
~~DAG~~ DGM→UGM: moralization: marry unmarried coparents together pairwisely.

UGM→DGM: Triangulation.

$\psi_c(y_c|\theta_c)$ is a non negative potential (energy) function or factor.

Ham: $P(y|\theta)=\frac{1}{Z(\theta)}\prod_{c\in C}\psi_c(y_c|\theta_c)$.

C is maximal cliques, y is all variables, $Z(\theta)=\sum_y\prod_{c\in C}\psi_c(y_c|\theta_c)$ called partition function.



$P(y)\propto\psi_{123}(y_1,y_2,y_3)\psi_{235}(y_2,y_3,y_5)\psi_{245}(y_2,y_4,y_5)\psi_{356}(y_3,y_5,y_6)\psi_{4567}(y_4,y_5,y_6,y_7)$.

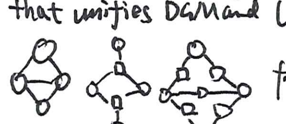
$P(A,B,C,D)=\frac{1}{Z}\psi_{a,b}(A,B)\psi_{b,c}(B,C)\psi_{c,d}(C,D)\psi_{a,d}(A,D)$.

if $\psi_{xyz}(x,y,z)=\psi_{x,y}(x,y)\psi_{y,z}(y,z)$,

then $\psi_{x,y,z}(x=1,y=1,z=1)=\psi_{x,y}(x=1,y=1)\cdot\psi_{y,z}(y=1,z=1)$

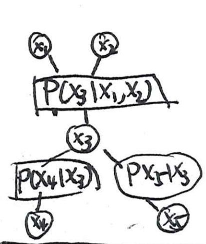
$\mathcal{H}$ markov net over X. U=u be the context. ~~Residue~~ reduced net $\mathcal{H}[u]$ is a markov net over nodes W=X−U where exists an edge between iff there is an edge between in $\mathcal{H}$.

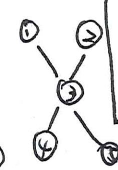
a factor graph is a graphical model that unifies DGM and UGM.



f

for directed model: one factor per (conditional distribution) CPD. And connect the factor to all the variables that uses the CPD.





Variable elimination: (Naive) $P(X_{i+1})=\sum_{X_i}P(X_{i+1}|X_i)P(X_i)$

$P(Y|E=e)=\frac{P(Y,e)}{P(e)}$

Complexity: m variables, each has k different values.

$P(X_i)$: k. CPD $P(X_{i+1}|X_i)$: k^2

k^2 multiplication and $k(k-1)$ add

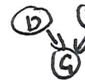
cost of total chain: $O(nk^2)$

full joint: $O(k^n)$.

Sum-product inference:

$\Gamma(Y)=\sum_Z\prod_{\phi\in\Phi}\phi(Z_{scope[\phi]\cap Z},Y_{scope[\phi]\cap Y})\ \forall Y$, where Φ is the set of potentials/factors.

For DGM, Φ is CPD. $N_i=3$ here.



$\phi_G(G,D,I)$. $Z=X-Y$

For UGM, Φ is the set of potentials. $Z=\sum_Y\Gamma(Y)$.

Variable Elim:

let N_i be the # of entries in the factor ϕ_i, let $N_{max}=\max_i N_i$

Total cost: $O(mk^{N_{max}})$.

(here n: # of variables, m # of init factors. N_{max} can be as large as n.).

In UGM, N_{max} = Size of largest clique.

p(x)	x,z~p(x,z)	x~p(x)	p(z)	p(x|z)	p(z…
DGM	✓	✓	✓	✓	✗
UGM	✗	✗	✗	✗	✓

$p(x)=\int p(x|z)dz$

$p(x)=\frac{1}{Z}p(x,z)$

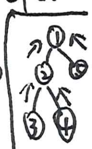
Message Believe Propagation:

Message from j to $i\in N(j)$ is:

$m_{j\to i}(x_i)=\sum_{x_j}\phi_j(x_j)\phi_{ij}(x_i,x_j)\cdot\prod_{k\in N_j/i}m_{k\to j}(x_j)$. Each message is a vector with one value for each state of x_i. In order to have $m_{j\to i}$, we must have $m_{k\to j}$ for $k\in N_j\setminus i$. Thus we need a specific ordering of messages.

We want $P(X_1)$. | T is the tree.

$P(x_1)\propto\sum_{x_2\dots x_n}\prod_i\phi_i(x_i)\prod_{(i,j)\in T}\phi_{i,j}(x_i,x_j)$



$m_{5\to1}(x_1)=\sum_{x_5}\phi_5(x_5)\phi_{15}(x_1,x_5)$

$m_{3\to2}(x_2)=\sum_{x_3}\phi_3(x_3)\phi_{13}(x_1,x_3)$

...

$m_{2\to1}(x_1)=\sum_{x_2}\phi_2(x_2)\phi_{12}(x_1,x_2)$

$m_{3\to2}(x_2)m_{4\to2}(x_2)$.

$P(x_1)\propto\phi_1(x_1)m_{2\to1}(x_1)m_{5\to1}(x_1)$. $Z=\sum_{x_1}$

~~P~~ choose root r. leaf to r, r to leaf.

Running time is 2 times the cost of VE $= O(nk^2)$, n = # of nodes, k = # states per node.

KL-Divergence:

$D(p\|q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$, measures the expected number of bits required to describe samples from $p(x)$ using a code based on q instead of p. $D(p\|q) \ge 0$ $\forall p, q$. $= 0$ iff $p = q$. Asymmetric.

punish when p high, q low. So, see the unnormalized dist'n.

$J(q) = \sum_x q(x) \log \frac{q(x)}{\tilde{p}(x)} = \dots \frac{q(x)}{Z\,p(x)}$

$= KL(q\|p) - \log Z$. Since Z is constant, we force q to become p by minimize $J(q)$.

$J(q) \ge -\log(Z) = -\log P(D)$.

negative log likelihood. Z

$J(q) = E_q[\log q(x)] + E_q[\log \tilde{p}(x)]$

$= -H(q) + E_q[\underset{\text{Energy}}{E(x)}]$

$= E_q[-\log P(D)] + KL(q\|p)$

Monte Carlo: Problem:

Sampleing from $p(x)$. (we can only compute $p(x)$ for a given x!)

calculate $\Phi = \underset{x \sim p(x)}{E}[\phi(x)] = \int \phi(x) p(x)\,dx$.

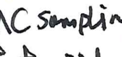
Simple monte Carlo:

if $x^{(r)} \sim P(x)$, then to find $E_p[\phi(x)]$.

$\int \phi(x) p(x)\,dx \sim \hat{\Phi} = \frac{1}{R} \sum_r \phi(x^{(r)})$

then $E[\hat{\Phi}] = E[\hat{\Phi}]$, $var[\hat{\Phi}] \propto \frac{\sigma^2}{R}$,

where $\sigma^2 = \int (\phi(x) - \bar{\phi})^2\,dx$. Note: accuracy of MC depend on var of $\phi(x)$, not dim of X. But, obtaining independent samples in high dim 在哪里都很稀薄!



Why is sampling from p hard?

assume $\tilde{P}(x)$ ez. $P(x) = \frac{\tilde{p}(x)}{Z}$, but $Z = \sum_i \tilde{P}(x_i)$. Even if know Z, correct samples from P will tend to come from places in x-space where $P(x)$ is big but how can we find those places without eval $P(x)$ everywhere? (In high dim?)

$Z = \int \tilde{p}(x)\,dx$. Compute Z cost (dimension of x).

~~Rejection Sampling~~: has a $Q(x)$. assume $\exists$ constant c, s.t. $c\tilde{Q}(x) > \tilde{P}(x)$ for all x. (Z_Q is the multiplicative factor, $Q(x)$ is the proposal density). Generate 2 random numbers.

1st: $x \sim Q$ $\Rightarrow$ eval $c\tilde{Q}(x)$

2nd: $u \sim [0, c\tilde{Q}(x)]$ uniform.

eval $\tilde{P}(x)$.

if $\tilde{P}(x) < u$, reject

if $\tilde{P}(x) > u$, accept, add x to our samples $\{x^{(r)}\}$, 和 $P(x)$

work well if Q is a good approx of P. Else, rejection rate will be large. In high dim, c will be forced too large. rejection rate so high, find c is hard, too.

- Importance sampling:

not soln to problem 1, it is soln to problem 2 ($\phi(x)$).

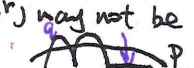
Still, $Q(x) = \frac{\tilde{Q}(x)}{Z_Q}$. Q is called sampler density.

$x \sim Q$, sample $\{x^{(r)}\}_{r=1}^R$.

$W_r = \frac{\tilde{P}(x^{(r)})}{\tilde{Q}(x^{(r)})}$ (weights).

$\hat{\Phi} = \frac{\sum_r W_r \phi(x^{(r)})}{\sum_r W_r}$.

problem: it is hard to estimate how reliable the estimator $\hat{\Phi}$ is (the variation of $\hat{\Phi}$ is hard to estimate) w_r and $w_r \phi(x^r)$ may not be a good guide).



↓: dramatically wrong! Also, in high dim, likely $\hat{\Phi}$ will be dominated by a few large W_r. The hyper-sphere case: all on surface

- Metropolis-Hasting: Q proposed depending on current state $x^{(t)}$

$Q(x'; x^{(t)})$ may be simple as gaussian centred on $x^{(t)}$.

Pending new state: $x' \sim Q(x'; x^{(t)})$

$a = \frac{\tilde{P}(x')}{\tilde{P}(x^{(t)})} \frac{Q(x^{(t)}; x')}{Q(x'; x^{(t)})}$. if $a \ge 1$, accept. Else, accept with probability a.

If accepted, set $x^{(t+1)} = x'$.

If rejected, set $x^{(t+1)} = x^{(t)}$

Add x' to $\{x^{(r)}\}$ nomatter what.

Samples are dependent. As an e.g. of MCMC, x^t depend on x^{t-1}. need long time to be "independent" difficult to tell whether converged. Given enough time, get all!

Use for high dim. (Where the scale ϵ is small, so that it won't jump to low-prob area ezly.) But long-time random walk.

HMM:

stationary: generative process doesn't ~~depend~~ evolve through time. Non-stationary: ... does ...

In general, Mth order: $P(x_t | x_{t-m}, \dots, x_{t-1})$

$P(x_1, \dots, x_T) = \prod_{t=1}^{T} P(x_t | x_{t-1}, \dots, x_1)$

FOMC: $P(x_t | x_{t-1})$. Each pair of output is a training case.

$P(x_t = B | x_{t-1} = A) = \#[t, \text{s.t. } x_t = B, x_{t-1} = A] / \#[x_{t-1} = A]$.

Higher order: May counts may be zero in training data set.

# of parameters:

1st order: $K(K-1)$.

Mth order: $K^m(K-1)$, $x_t \in \{1, \dots, K\}$.

initial state: $\pi_i = P(z_1 = i)$. (Hidden is 1st order).

Transition Prob: $T(i,j) = P(z_t = j | z_{t-1} = i)$

Emission Prob: $\varepsilon_i(x_t) = P(x_t | z_t = i)$

inital: 打电话前的天气. Transition: 雨→晴的概率. Emission: 雨天出去的概率. output is not any order.

Joint: $P(\vec{x}, \vec{z}) = P(z_1) P(x_1 | z_1) \prod_{t=2}^{T} P(z_t | z_{t-1}) P(x_t | z_t)$ → change.

SVI:

Inference: $P(\vec{z} | \vec{x}) = \frac{P(\vec{x}, \vec{z})}{\int P(\vec{x}, \vec{z})\,dz} = \tilde{P}(\vec{z})$ ez to eval density

↓ true posterior, ↓ hard.

so,

1. Introduce a variational dist'n: $q_\phi(\vec{z} | \vec{x}) = q(\vec{z} | \vec{x}; \phi)$.

(e.g. if variational Gaussian, ϕ would be mean and variance).

2. optimize ϕ to minimize distance between q and p.

$KL(q(y) \| p(y)) = \underset{y \sim q(y)}{E}\left[\log \frac{q(y)}{p(y)}\right]$

ELBO: ① $P(\vec{z} | \vec{x}) = \frac{P(\vec{x}, \vec{z})}{P(\vec{x})}$

$KL(q_\phi(\vec{z}|\vec{x}) \| P(\vec{z}|\vec{x})) = \underset{\vec{z} \sim q_\phi(\vec{z}|\vec{x})}{E}$ ①

$\left[\log q_\phi(\vec{z}|\vec{x}) \frac{1}{P(\vec{z}|\vec{x})}\right]$

$= \underset{q}{E}\left[\log\left(q_\phi(\vec{z}|\vec{x}) \frac{P(\vec{x})}{P(\vec{x}, \vec{z})}\right)\right]$ ①

$= E_q \log\left(\frac{q_\phi(\vec{z}|\vec{x})}{P(\vec{x}, \vec{z})}\right) + E_q[P(\vec{x})]$ ↳ not depend on q!

$= KL(q_\phi(\vec{z}|\vec{x}) \| P(\vec{x}, \vec{z})) + \log(P(x))$.

⇓ constant

Evidence Lower BOund.

ELBO $= -E_q[\log P(\vec{z}, \vec{x}) - \log q(\vec{z}|\vec{x})]$

so, maximizing ELBO is minimizing KL posterior ①. = minimizing distance.

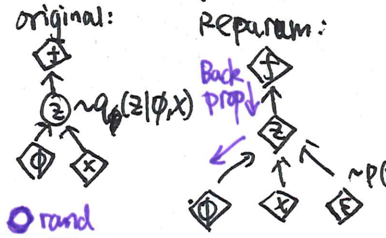
Also,

ELBO $= E_q[\log p(x|z)] - KL(q(z|x) \| p(z))$

Reparametrization trick:

In order to do backprop (BP cannot go through random node

original: Reparam:



○ rand

◇ determ.

- we can learn $q(z|x)$ in a very flexible way using NN to output param ϕ for q.
- NN takes x as input, output ϕ.
- combine deep learning and graphial models
- Remember we estimate ELBO by sampling from q.

ELBO: $\mathcal{L}(\phi) =$ (1) MC, (2) H, (3) KL.

$\underset{z \sim q}{E}[\log p(x|z) + \log p(z) - \log q_\phi(z|x)]$

$\underset{z \sim q}{E}[\quad] + H[q_\phi]$

$\underset{z \sim q}{E}[\quad] - KL(q_\phi(z|x) \| p(z))$

when $q_\phi(z) = p(z|x)$, MC has zero variance. When optimizing, $\nabla_\phi \mathcal{L}(\phi)$ use reparam trick (Back prop)

$\nabla_\phi \mathcal{L}_{MC}(\phi) =$ Reinforce with score func.

~~$\nabla_\phi E \log p(x, z)$~~

$\underset{z \sim q_\phi}{E}\Big[\underbrace{(\log p(x|z) - \log q_\phi(z|x))}_{\text{high variance}} \underbrace{\nabla_\phi \log q_\phi(z|x)}_{\text{score function}}\Big]$

if z is deteministic fn of ϕ and unbiased, push ∇_ϕ inside,

$\underset{\epsilon \sim p(\epsilon)}{E}[\nabla_\phi \log p(g(\epsilon, \phi), x) - \log p(g(\epsilon, \phi) | x)]$

lower variance ($g(\epsilon, \phi) = z$).

MC sampling unbiased estimator of $\mathcal{L}$. iid sample $z^{(r)} \sim q$, take $\mathbb{E}$.

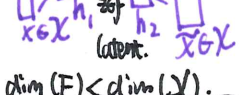
Alg: | sample $z^{(r)} \sim q_\phi(z|x)$
| compute $\log p(z^r, x) - \underbrace{\log q_\phi(z^r | x)}_{\text{score}}$

R times.

$\nabla_\phi \hat{\mathcal{L}} = \frac{1}{R} \sum$. ∇_ϕ by Auto diff.

Sample from MC ⇒ stochastic.

VAE Encoder decoder,



encoder: $g(x) \Rightarrow z$

decoder: $f(z) \sim \tilde{X}$.

$\dim(\mathcal{F}) < \dim(\mathcal{X})$.

MC is asymptotically exact.

SVI simple form underestimates posterior variance, but it can measure inference progress, use fancy tools (Adam).