Deep Bayes (HSE+)

conditional = $\frac{\text{joint}}{\text{marginal}} = \frac{P(x,y)}{P(y)}$

$P(x,y,z) = P(x|y,z)\,P(y|z)\,P(z)$

↳ one dim cond. distr

$P(y) = \int P(x,y)\,dx$

- estimate $P(y|x)$ from $P(x,y,z)$ w/o knowing z.

$$P(y|x) = \frac{P(y,x)}{P(x)} = \frac{\int P(x,y,z)\,dz}{\int P(x,y,z)\,dy\,dz}$$

(In theory)

Bayes theorem

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

Posterior = $\frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$.

Sathvik Udupa

Statistical inference

data $X = (x_1, \dots x_n) \Leftarrow P(x|\theta)$, θ ?

2 methods

Frequentist / Clasical :

MLE.

$$\theta_{ML} = \arg\max P(X|\theta) = \arg\max \prod_{1}^{n} P(x_i|\theta)$$

$$= \arg\max \sum_{i}^{n} \log P(x_i|\theta)$$

Bayesian :

encode uncertainity about θ in a prior $P(\theta)$

then

$$P(\theta|x) = \frac{\prod_{i=1}^{n} P(x_i|\theta) P(\theta)}{\int \prod_{i=1}^{n} P(x_i|\theta) P(\theta) d\theta}$$

Coin tossing ex :

2 tosses. Outcome H. H

estimate prob θ of landing H.

Frequentist : $\theta_{ML} = 1$ (Both are heads)

Bayesian :

take prior distribution (prob for coin tosses)

In Freq. random variable is not predictable. Simply collect statistics on data.

In Bayesian, random variable is outcome of a deterministic process, though some of the factors are known to observer.

Almost all data is random from Bayesian point of view.

Radiactive decay, quantum mechanics not bayesian

In MLE, no of data $\gg$ dim of θ.
In Bayesian, $\forall n$.

Freq framework is a limit case of Bayesian

$$\lim_{n/d \to \infty} p(\theta|x, \dots) = \delta(\theta - \theta_{ML})$$

↳ centered around θ_{ML}

There is no contradiction b/w classical & Bayesian.

Advantage of Bayesian

- prior info
- prior is a form of Regularisation
- point estimate of θ posterior contains info about uncertainity.

Probabilistic ML model

$x \rightarrow$ observed data (features)

$y \rightarrow$ hidden / latent

$\theta \rightarrow$ model parameters

Discriminative model

$$P(y, \theta \mid x)$$

$$= P(y \mid x, \theta) P(\theta)$$

usually prior distn θ is assumed indep. of x.

generative

$$P(x, y, \theta) = P(x, y \mid \theta) P(\theta)$$

can generate new $P(x, y)$

Training BY models

given (X_{tr}, Y_{tr}) & model $P(y, \theta \mid x)$

training: Bayesian inf over θ

$$P(\theta \mid X_{tr}, Y_{tr}) = \frac{P(Y_{tr} \mid X_{tr}, \theta) P(\theta)}{\int P(Y_{tr} \mid X_{tr}, \theta) P(\theta) d\theta}$$

for testing:

$$P(y \mid x, X_{tr}, Y_{tr}) = \int P(y \mid x, \theta) P(\theta \mid X_{tr}, Y_{tr}) d\theta$$

↓

weighting across the learned distribution

The lower integrals may be intractable

Conjugate distributions

$P(y)$ & $P(x|y)$ are conjugate iff $P(y|x)$ belongs to same parametric family as $P(y)$.
Then it can be integrated.

Eg: for coin toss

$$P(x, \theta) = P(x|\theta)P(\theta)$$

Bern $P(x|\theta) = x(1-\theta)^{1-x}$.

now $P(\theta)$ the prior should be blong to paramatic family of $P(x|\theta)$, which is bernoulli.

what distn is conjugate to bernoulli?

Beta distn matches requirements

$$\text{Beta}(\theta|a,b) = \frac{1}{B(a,b)} \theta^{a-1}(1-\theta)^{b-1}$$

and this can be tweaked to make it bell shaped to model coin toss.

$$P(\theta) = C\theta^{\alpha}(1-\theta)^{\beta}$$

$$P(\theta|x) = C'P(x|\theta)P(\theta) = C'\theta^{x}(1-\theta)^{1-x}\frac{\theta^{a-1}(1-\theta)^{b-1}}{B(a,b)}$$

$$= C''\theta^{a'}(1-\theta)^{\beta'}$$

$\therefore$ then two are conjugate.

$$P(\theta|x) = \frac{1}{z} P(x|\theta) P(\theta) = \frac{1}{z} \prod P(x_i|\theta) P(\theta)$$

$$= \frac{1}{z} \pi \theta^{x_i} (1-\theta)^{1-x_i} \frac{1}{B(a,b)} \theta^{a-1} (1-\theta)^{b-1}$$

$$= \frac{1}{z'} \theta^{a + \sum x_i - 1} (1-\theta)^{b+n-\sum x_i - 1}$$

$$= \text{Beta}(\theta|a', b')$$

new parameters

$$a' = a + \sum x_i \qquad b' = b + n - \sum x_i$$

Conjugate distributions

Likelihood $P(x\|y)$	y	Conjugate prior $P(y)$
gaussian	μ	gaussian
gaussian	σ^{-2}	gamma
gaussian	(μ, σ^{-2})	gaussian-gamma
Multivariate gaussian	Σ^{-1}	Wishart
Bernoulli	p	Beta
Multinoulli	$(p_1, \ldots, p_m)$	Dirichlet
Poisson	λ	Gamma
Uniform		

What to do if there is no conjugacy?

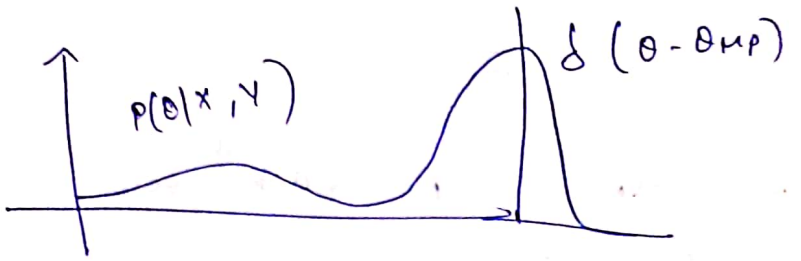
- Max posterior estimation

$\theta_{MP} = \arg\max P(\theta | x_{tr}, y_{tr}) = \arg\max P(y_{tr} | x_{tr}, \theta) P(\theta)$

~~this~~ ~~cor~~
L_2 Reg corresponds to this

In testing:

$P(y | x, x_{tr}, y_{tr}) = P(y | x, \theta_{MP})$



Bayesian Reasoning

1) dark mark

stays 20% if the dies.

stays 100% if he is alive

survival chance ~~0.01~~ 1%.

what is P of being alive if mark is present

$x \in (0,1)$ 1 if he is alive

$y \in (0,1)$ 1 if mark is visible

$P(y=1 | x=1) = 1$ $P(y=1 | x=0) = 0.2$ $P(x=1) = \frac{1}{10}$

$P(x=1 \mid y=1) = ?$

$$P(x=1 \mid y=1) = \frac{P(y=1 \mid x=1)P(x=1)}{\sum P(y=1 \mid x=j)P(x=j)}$$

$$= \frac{1 \times \frac{1}{101}}{1 \times \frac{1}{101} + \frac{1}{5} \times \frac{100}{101}}$$

$$= \frac{1}{21}$$

2) $X = \{x_1, x_2 \ldots x_N\}$ — N indep dice rolls

$N_k = \sum_{n=1}^{N} I(x_n = k)$ - counts

$P(x|\theta) = \prod_{k=1}^{K} \theta_k^{N_k}$ - multinomial likelihood

$\theta \in S_k$

MLE $\theta_{ML} = \arg\max \log P(x|\theta)$

Refer github. [Bookmark]

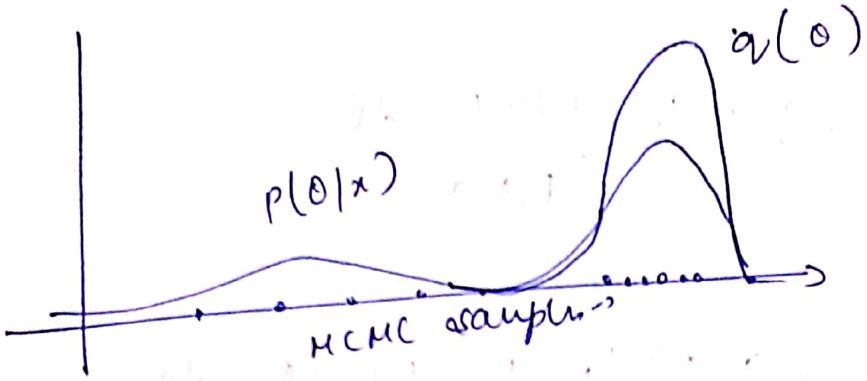
Approximate inference.

variational inference (VI)

- Approx $P(\theta|x) \approx q(\theta) \in \mathcal{Q}$
- Biased
- Faster & more scalable.

Monte Carlo Markov chain MCMC

- Samples from un normalized $p(\theta|x)$
- unbiased
- lot of samples needed



Dissimilarity measure for VI

- can be done using KL divergence.

$$\therefore F(q) := KL(q(\theta) || p(\theta|x)) \rightarrow \min_{q(\theta) \in \mathcal{Q}}$$

$$\therefore = \int q(\theta) \log \frac{q(\theta)}{p(\theta|x)} d\theta$$

Properties

~~to compute~~

$KL(q||p) \geq 0$

$KL(q||p) \geq 0 \Leftrightarrow q = p$

$KL(q||p) \neq KL(p||q)$

to compute KL div, $P(\theta|x)$ is needed. Though if it is obtained, then KL div is not needed.

Some maths

$$\log P(x) = \int q(\theta) \log (P(x))\, d\theta \qquad \text{(Identity)}$$

$$= \int q(\theta) \log \frac{P(x,\theta)}{P(\theta|x)}\, d\theta$$

$$= \int q(\theta) \log \frac{P(x,\theta) q(\theta)}{P(\theta|x) q(\theta)}\, d\theta$$

$$= \int q(\theta) \log \frac{P(x,\theta)}{P(\theta|x) q(\theta)}\, d\theta + \int q(\theta) \log \frac{q(\theta)}{P(\theta|x)}\, d\theta$$

$$\log P(x) = L(q(\theta)) + KL(q(\theta) \| P(\theta|x))$$

$\log P(x)$ ↓ doesn't depend on q

$L(q(\theta))$ ↓ Evidence lower bound (ELBO)

$KL(q(\theta) \| P(\theta|x))$ ↓ KL div depends on q

Instead of mini KL div, we can maximin ELBO in above eq

In ELBO, everything is computable. No true posterior estimate is needed

$$\mathcal{L}(q(\theta)) = \int q(\theta) \log p(x|\theta)\, d\theta + \int q(\theta) \log \frac{p(\theta)}{q(\theta)}\, d\theta$$

$$= \underbrace{E_{q(\theta)} \log p(x|\theta)}_{\text{encouraging MLE}} - \underbrace{KL(q(\theta) \| p(\theta))}_{\text{to maximise } \mathcal{L}, \text{ minimise this, So make it 0 (converge to prior)}}$$

If $\mathcal{L}$ is maximised over all distributions, & then we will arrive at true posterior distn $p(\theta|x)$. (not actually true since it is intractable)

how to perform optimisation wrt distn?

- Mean field approx MFA
- ② Parametric approx

① $q(\theta) = \prod q_j(\theta_j)$, $\theta = [\theta_1, \theta_2 \cdots, \theta_m]$

q is factorised as product of m multiplies

② $q(\theta) = q(\theta|\lambda)$

① MFA

all θ_i's are assumed ~~independent~~ independent of each other

(Simpler approx)

Using Block coordinate assent:

At each step, fix all factors $\{q_i(\theta_i)\}_{i \neq j}$ except one and optimize wrt to it

$$L(q(\theta)) \longrightarrow \max_{q_j(\theta_j)}$$

do it for all $j \in m$ till convergence

$$L(q(\theta)) = E \log p(x, \theta) - E \log q(\theta)$$

$$= \quad " \quad - \sum E_{q_k(\theta_k)} \log q_k \theta_k$$

$$= E_{q_j(\theta_j)} \left[E_{q_{i \neq j}} \log p(x, \theta) \right] - E_{q_j(\theta_j)} \log q_j(\theta_j) + C$$

Consider

$$\{r_j(\theta_j) = \frac{1}{Z_j} \exp\left(E_{q_{i \neq j}} \log(x, \theta) \right)\} \quad -①$$

$r_j \rightarrow$ pdf

$$= E_{q_j(\theta_j)} \log \frac{r_j(\theta_j)}{q_j(\theta_j)} + C = -KL(q_j(\theta_j) \| r_j(\theta_j)) + C$$

$\downarrow$

for -ve KL, optimal $q_j = r_j$

$$q_j(\theta) = r_j(\theta_j) = \frac{1}{Z_j} \exp\left(E_{q_{i \neq j}} \log p(x, \theta \right.$$

Update each q_j, repeat till convergence. This is guaranteed to converge.

Conditions to compute (1)

$p(x, \theta) = p(x|\theta)p(\theta)$ $\quad \theta = [\theta_1, \dots \theta_m]$

- Conditional conjugacy of likelihood and prior on each θ_j conditioned on all other $\{\theta_i\}_{i \neq j}$

$p(\theta_j | \theta_{i \neq j}) \in A(\alpha)$, $p(x|\theta_j, \theta_{i \neq j}) \in B(\theta_j) \rightarrow$

- The prior wrt each subgroup θ_j given all other $\theta_{i \neq j}$ fixed, $p(\theta_j|x, \theta_{i \neq j}) \in A(\alpha')$ and MLE fn as a fn of θ_j given all other $\theta_{i \neq j}$ fixed, form conjugate pair of distri.

Then posterior distri $i \neq j$ belongs to same parametric family as prior, this should hold true for each subgroup. ~~Then~~ if this holds true, then update equation (1) can be done in closed form

In practice,

$$\log q_j(\theta_j) = E_{q_{i \neq j}} \log p(x, \theta) + \text{const}$$

Inference methods. Summary

Full Bayesian inference : $p(\theta|x)$

MAP : $p(\theta|x) \approx \delta(\theta - \theta_{MAP})$

Mean field variational Inference : $p(\theta) \approx q(\theta) = \prod_i q_i(\theta_i)$

Parametric variational inference : $p(\theta|x) \approx q(\theta) = q(\theta_i)$

Latent Variable Models and EM

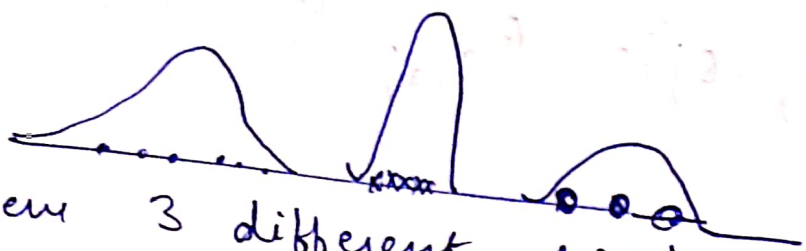
LVM eg:

i) $x_i \sim N(x_i|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{(x-\mu)^2}{-2\sigma^2}\right)$

estimate μ & σ

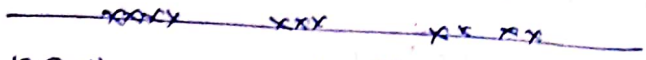
Soln: MLE eq for μ & σ.

ii)

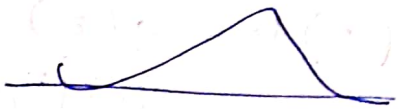


Here 3 different distri, but their samples can know which belongs to which. Same solution.

iii)



now we don't know which one was derived from which gaussian
- we can try to fit one gaussian.



but not the best choice because there is data missing b/w many points etc

- To solve this, establish a LVM.

for each x_i, establish an additional latent variable z_i which denotes index of gaussian from which x was derived.
ib data is assumed IID (independent identical dist)

$$P(X, Z|\theta) = \prod p(x_i, z_i|\theta)$$

$$= \prod p(x_i|z_i, \theta) p(z_i|\theta)$$

$$= \prod \pi_{z_i} N(x_i|\mu_{z_i}, \sigma^2_{z_i})$$

$\pi_j = P(z_j = j)$ an prior of j^{th} gaussian

If both X & Z are known, $\theta = \{\mu, \sigma, \pi\}$

$$\theta_{ML} = \arg\max_{\theta} P(X, Z|\theta).$$

$$= \arg\max_{\theta} \log P(X, Z|\theta)$$

But Z is not known.
Then we need to maxi wrt θ, the log of incomplete likelihood.

$$\log p(x|\theta) = \int q(z) \log \frac{p(x,z|\theta)}{q(z)} dz \rightarrow \text{ELBO}$$

$$+ \int q(z) \log \frac{q(z)}{q(z|x,\theta)} dz \rightarrow (2)$$

$\downarrow$ KL (non negative)

$$= L(q,\theta) + KL(q||p) \geq L(q,\theta)$$

Instead of optimising log of incomplete likelihood $\log p(x|\theta)$, we optimise ELBO wrt θ & $q(z)$.

The block-coordinate algorithm is known as EM algorithm

Variational lower bound definition

fn $g(\xi, x)$ a VLB of a fn $f(x)$ iff
- for all ξ & $\forall x$, it follows $f(x) \geq g(\xi, x)$
- for any x_0, there exists $\xi(x_0)$ such that $f(x_0) = g(\xi(x_0), x_0)$

If such a VLB exists, then instead of solving $f(x) \rightarrow \max_x$

we can perform block-coordinate update

EM algorithm

$$L(q,\theta) = \int q(z) \log \frac{p(x,z|\theta)}{q(z)} dz \to \max_{q,\theta}$$

Start with θ_0 & iterate

E-step:

$$q(z) = \arg\max_q L(q,\theta_0) = \arg\min_q KL(q||p)$$

$$= p(z|x,\theta_0)$$

due to decomposition, eq (4), max of 1st is min of 2nd.

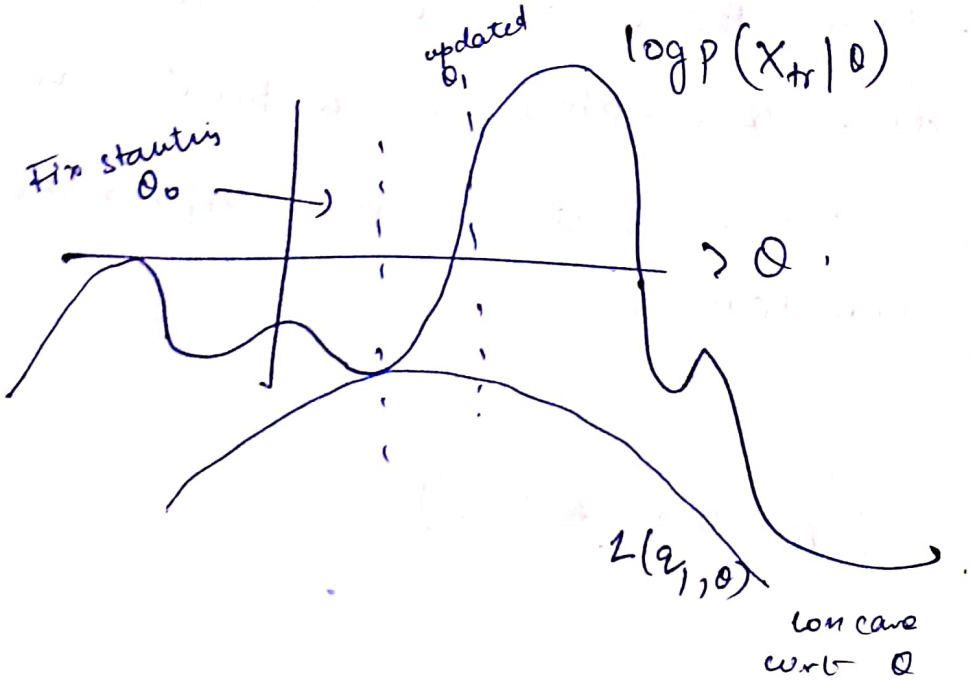
M-step

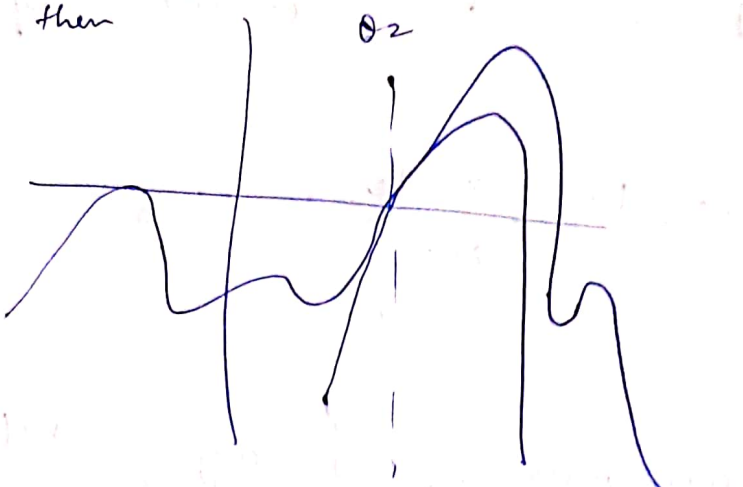
fix q.

$$\theta_* = \arg\max_\theta L(q,\theta) = \arg\max_\theta E_z \log p(x,z|\theta)$$

log of likelihood is usually concave fn.

maximisation of concave fn, is a convex optim problem





Converges to local maximum.

Benifits of EM

- E & M steps can be done in closed form (eg: mix of gaussians)
- allows to build complicated models using mix of simple distributions
- If true posterior is intractable, then closed q(z) can be solved by optimisation
- Allows to proces missing data by treating them as latent variables

discrete latent distn of variables is a finite mix of distn

Continuous latent variable is representation using

eg: PCA, ICA.

- [problem set 2]

Stochastic variational inference and VAE

Continuous latent variable

$$p(x_i|\theta) = \int p(x_i, z_i|\theta)\, dz_i = \int p(x_i|z_i, \theta) p(z_i|\theta) dz_i$$

$$q(z_i) = p(z_i|x_i, \theta) = \frac{p(x_i|z_i, \theta)\, p(z_i|\theta)}{\int p(x_i|z_i, \theta)\, p(z_i|\theta) dz_i}$$

Ex: PCA

$x \in R^D$, $z \in R^d$ $\quad D \gg d$

$$p(X, Z|\theta) = \prod p(x_i|z_i, \theta) p(z_i|\theta)$$

assume standard gaussian as latent

$$= \prod N(x_i|Wz_i + \mu, \sigma^2 I)\, N(z_i|0, I)$$

$\theta \to D \times d$ matrix W, D dim μ σ.

Can use EM to find $\arg\max_\theta p(X|\theta)$

Advantages of prob PCA:

- Closed form soln has $O(np^2)$, EM has $O(nDd)$
- Can process missing parts in x and present parts in z
- Can determine d if $p(\theta)$ is established
- Can be generalized to more general models such as mixture of PCA

Mixture of PCA

- Two types of latent ~~mode~~ variables:
discrete $t \in \{1, \dots, k\}$ & $z \in \mathbb{R}^d$

t is the index of corresponding subspace.

$$p(x, z, T|\theta) = \prod p(x_i|t_i, z_i, \theta) p(t_i|\theta)$$

$$= \prod \mathcal{N}(x_i|V_t z_i + \mu_{t_i}, \sigma_{t_i}^2 I) \mathcal{N}(z_i|0,1)^{\pi_{t_i}}$$

E

$$q(z, T) = \prod q(z_i, t_i) = \prod p(z_i, t_i|x_i, \theta)$$

$$= \prod \frac{\mathcal{N}(x_i|V_t z_i + \mu_t, \sigma_t^2 I) \mathcal{N}(z_i|0,1) \pi_{t_i}}{\sum_{k=1}^{K} \int x_i|_{k z_i} + \mu_k, \sigma_k^2 I) \mathcal{N}(z_i|0,1) \pi_k dz_i}$$

M

$$E_{z,T} \log p(X, Z, T|\theta)$$

$$= \sum_{i=1}^{n} E_{z_i, t_i} \left(\log p(x_i|t_i, z_i, \theta) + \log p(z_i|\theta) \right.$$

$$\left. + \log p(t_i|\theta) \right) \to \max_{\theta}$$

non-linear generalisation of PCA → VAE

$$p(x, z|\theta) = \prod_i p(x_i|z_i, \theta) p(z_i)$$

$$= \prod_{i=1}^{n} \left(\prod_{j=1}^{D} N(x_{ij} | \mu_j(z_i), \sigma_j^2(z_i)) \right) N(z_i|0)$$

generator

likelihood prior

here μ, σ is substituted by some non linear fn of z. μ_j & σ_j are o/p of a NN, parameterised by θ.

It takes z_i as i/p.

allows us to construct very non linear low-dim manifolds of latent rep of objects

no more conjugacy due to N.N

$$\int p(x, z|\theta)\, dz = \prod \int p(x_i|z_i;\theta)\, p(z_i)\, dz_i$$

↓

not optimisable

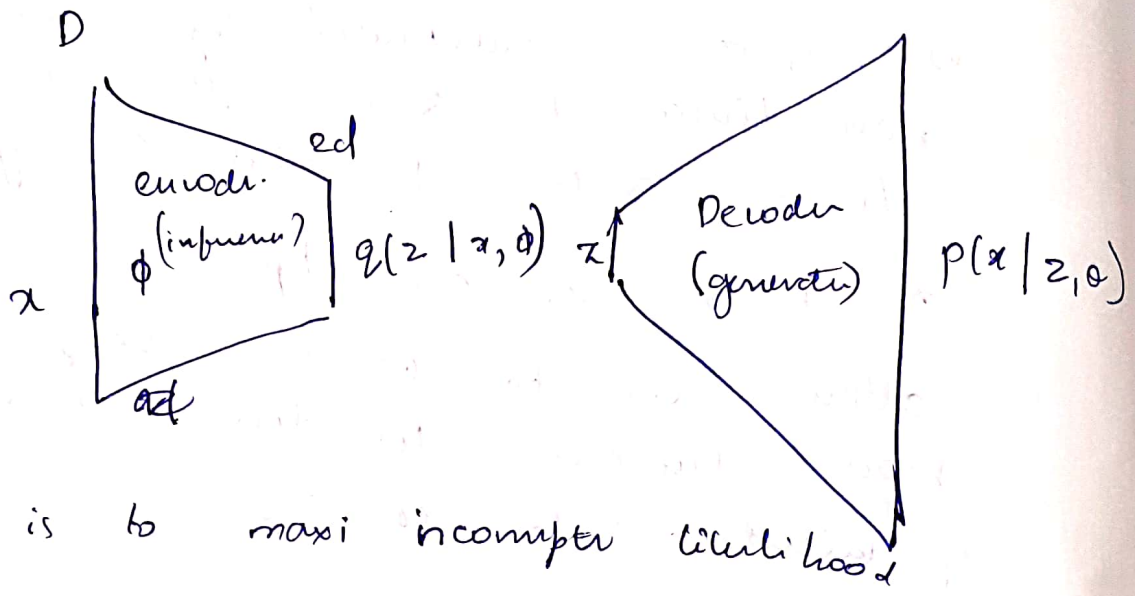
if we use EM, even it is intractable for true posterior distri

using variational bayes true posterior can be approximated.

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

$$= \prod \mathcal{N}(z_{ij}|\mu_j(x_i), \sigma_j^2(x_i))$$

this μ & σ is given by another neural n/w. with weight ϕ.



goal is to maxi incomplete likelihood.

$$q(z|x,\phi) = \arg\min_{\phi} KL(q(z|x,\phi) \| p(z|x,\theta))$$

This is equaliant to optimising ELBO,

$$L(\phi,\theta) = \int q(z|x,\phi) \log \frac{p(x|z,\theta)\, p(z)}{q(z|x,\phi)}\, dz$$

$\rightarrow \max_{\phi,\theta}$

Block cord- opti

wrt encodr $\phi \rightarrow \epsilon$

Block cord opti

wrt decodr $\theta \rightarrow M$

- Training data is assumed to be large
- integral in ELBO still intractable.

Soln:
Compute stochastic gradients using mini batch & monte carlo estimation.

Practical Session: VAE
(notebook tasks)
1. loss fn for VAEs in pytorch.
2. DRAW.

Discrete variable models

Why discrete?
- Easier to interpret
- Allow the model to make a discrete choice
  - hard attn
    binary masks where to look at
    attn classifies masked images
- To get certain properties for discrete preds
  - GANs for text

Relax objective over discrete random variables into an objective over continuous random samples & use reparametrisation trick.

Keep discrete in test case
for reparametrisation, $f(z)$ should be differentiable.

Gumbel Max trick

$z \sim \text{categorical}(\pi_1, \ldots, \pi_k)$

can be computed as

$$z \overset{d}{=} \arg\min_k \frac{\varepsilon_k}{\pi_k}, \quad \varepsilon_k \sim \text{Exp}(1)$$

↓

This will have categorical distribution

apply $-\log$ to make it computationally stable

$$z \overset{d}{=} \arg\max_k \underbrace{[\log \pi_k - \log \varepsilon_k]}_{\xi_k} \quad \xi \sim \text{Exp}(1)$$

argmax is not differentiable.

∴ approx argmax with softmax

Softmax with temp to relax it.

$$\text{Softmax}_\tau(x)_j := \frac{\exp(x_j/\tau)}{\sum \exp(x_k/\tau)}$$

temp allows to control sharpness of softmax

at $\tau \to 0$, argmax = softmax.

at $\tau = \infty$, softmax leads to uniform normal

assume discrete z is a one-hot vector & replace with a continuous relaxation $\tilde{z}$

$$\tilde{z}(\gamma, \pi) := \text{softmax}_\tau(\log \pi_1 + \gamma_1 + \ldots \log \pi_k + \gamma_k)$$

Then can reparameterized based gradient estimator.

Choosing temp

- bais / variance tradeoff
  small temp leads to high var, low bias
  large temp leads to low var, high bias
- grid search.

- Toy example -> implement.

going for cont distr for training & discrete distr for testing, then is the mismatch

Using Reinforce with reduce in Variance:

Control variates:

consider $b(z)$ with tractable expectation

$$\mu := E_{q(z)} b(z)$$

$$E_{q(z)} f(z) = E_{q(z)}[(f(z) - b(z)) + \mu]$$

↓

might be lower var estimator if $f(z)$ & $b(z)$ are positively correlated

- Unbiased
- lower bound on β
- convinient if $b(z)$ is zero-mean
- Can take several samples to reduce variance further.

b(z) is called 'baseline fn' in Reinforce

choosing baseline:

- constant

$b(z) = c$

$$(f(z) - c)\frac{\partial}{\partial \phi} \log q_\phi(z) + \underbrace{\frac{\partial}{\partial \phi_{q(z)}} E_{\epsilon} c}_{0}$$

formula for optimal constant such that var is minimal

$$c = \frac{\sum \text{cov}\left[f(z)\frac{\partial}{\partial \phi_d}\left[\log q_\phi(z)\frac{\partial}{\partial \phi_d}\log q_\phi(z)\right]\right]}{\sum \text{Var}\left[\frac{\partial}{\partial \phi_d}\log q_\phi(z)\right]}$$

- can be estimated by moving average, but variance might go higher.

- b(z) as a first-order taylor series expansion of f(z) at some point μ.

$$b(z) = f(\mu) + \frac{\partial f}{\partial z}(\mu)^T (z - \mu)$$

$\mu = z = \mu(\phi) = E_{q_\phi(z)}\, z$

- Backpropagate through mean, then fine tune inaccuracies with reinforce
- taylor series approx is expensive

- variance minimisation
  - minimising grad of var.
    $var[g(z,\phi)] = Eg(z,\phi)^2 - (Eg(z,\phi))^2$
  - In general minimising variance leads to increase in bias

- more complicated gradient estimator - REBAR
  - uses gumbel - relaxed as baseline
  - Practical

Fairness in ML

- Bias is everywhere
- uncontrolled bias will lead to unfairness

Source of unfairness

- Bias from data
- Bias from algorithm

Training data

- Sampling bias
  not representative of the overall target population
- Selective label
  only observe outcome of one side of decision

- Proxy labels
  eg: want to predict who will commit crime but have only data on who is arrested.

From algo

- when there is class imbalance (could be due to sampling error)
- feedback effect
  model at time t+1 uses data of time t

- In imSitu situation recognition dataset, 33% more women than men in data this amplifies to 68% by the algo.

- In Adult Income dataset (UCI), 30% of man income is high while 11% of female has high income. The skewness ratio is amplified from 3:1 to 5:1.

Enforcing fairness

Fairness difinition

- Discrimination in the law.
  - treating ppl from diff groups diffently (direct discri)
  - seemingly treat all groups equally but consequences are diffent (indirect)
- To remove direct, remove info on sensitive grouping.
- Indirect: enfore equality of outcomes.

enforcing equality on outcomes?
enfore
- same no of positive predictions (statistical demographic parity)
  - this produces lower accuracy. (Reason we are enforcing against biased data → forcing misclassification)
- True positive rate outcome should be same.
  enforce true positive rate same for all groups.
- Equalised odds
  True positive rate & false positive rate should be same for all classes.

Statistical fairness notions.

- Statistical parity $(\hat{Y} \perp S)$: $Pr(\hat{Y}=1|S=0) = Pr(\hat{Y}=1|S=1)$

↓ independent

prediction $\hat{Y}$

sensitive group S

$\hat{Y}$ independent of S

- Equalised odds $(\hat{Y} \perp S | Y)$

$Pr(\hat{Y}=y|S=0, Y=y) = Pr(\hat{Y}=y|S=1, Y=y)$

Conditional independence.

$\hat{Y}$ ⊥ S given true label info.

forcing TP & TN to be equal.

- Predictive parity:

$Pr(Y=y|S=0, \hat{Y}=y) = Pr(Y=y|S=1, \hat{Y}=y)$

Reverse the conditional independence.

Not possible to enforce all 3.

By Bayes Rule.

Equalised odds & predictive parity can be enforced at same time if priors are equal (Perfect dataset) or if TPR=1, FPR=0 (Perfect classifier)

equalised odds and statistical parity at once only in case of perfect classifier.

(8.1)

Problem Statement

total applicants 20k.

50% blue group

50% green group,

acceptance rate 50%.

entrance test:

80% who passed will graduate

10% who failed could graduate

60% of blue will pass the test

40% of green will pass the test

confusion tables:

Statistical parity

blue.

	Accepted	not
graduate	4000 (80%)	1200
doesn't	1000 (20%)	3800.

5000

green

	Accepted	not
graduate	3300	500 (10%)
doesn't	1700	4500 (90%)

5000

10% of qualified blue rejected while 10% of unqualified green accepted

equality of opportunity

blue

	Accepted	not
graduate	4440	760
doesn't	1110	3690

5510

green

	Accepted	not
graduate	3245	555
doesn't	1205	4995

55% of blue & 44.5% of green, TRL: 85.4%, for both

Predictive parity

60:40 accept based on test only.

	accepted	not
graduate	4800	400
doesn't	1200	3600

6000

	accepted	not
graduate	3200	600
doesn't	800	5400

4000

this can reinforce bias
(what if blue is well educated etc)
(if more represented people from green group are to be in future, other approaches)

- Only group effect is observed, not wrt each individual

- Can't use data for individual fairness, eg: other records like expert data from other sources to tackle this
- need to talk with domain expert to decide on which notion.

Algorithmic fairness methods

- pre processing
- post processing
- in processing (with learning)

Pre-processing

- Reweight training data (maintain class balance)

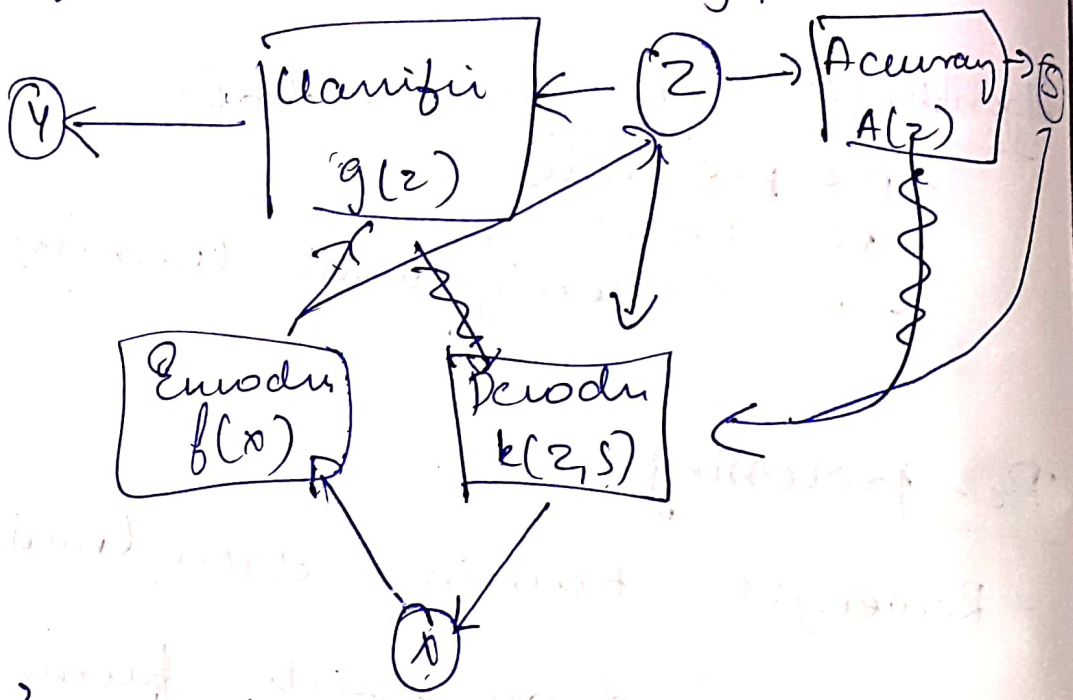
weights based on which fairness definition to be followed.

$$W(S=0, Y=1) = \frac{Pr(Y=1)\,Pr(S=0)}{Pr(Y=1, S=0)}$$

↓ for lesser data class, larger weight, so more times fed to model

- From reweighting to resampling.
  - Sampling data with replacement according to weights

- Learn a new, fair representation of data.
  - using adversarial learning.



Learn z, make z is informative wrt Y, but cannot predict S

~~of~~ Limitation

- latent embeddings still has some info on S if a diff classifier is used on z.

In processing

- specify fairness metric as constraint
- optimize for accuracy under the condition
- Then lower accuracy

Constraint:

loss for $L(\theta)$

normal: $\min_{\theta} L(\theta)$

Statistical parity:

subject to $P(\hat{y}=1|s=0) = P(\hat{y}=1|s=1)$.

- Not convex

- Need to find a better way.

- Return a set of models that trade-off b/w fairness & accuracy

- Likelihood with depending on s

$s: P(y=1|x,\theta,s)$.

Introduce latent target label $\tilde{y}$

$Pr[y=1|x,\theta,s] = \sum_{\tilde{y}} Pr(y=1|\tilde{y},s) Pr(\tilde{y}|x,\theta,s)$

Post-processing.

Using S at prediction time,
Use 2 diff. thresholds.

Interpretability in fairness

a fair rep. $T_w(x)$ has sensitive removed but is dependent on x

- Separate fair & unfair bits
- use ~~generative training~~ adversarial learning to generate data points with pre-specified sensitive attributes

GANs

Parameterisation

Samples from two distributions $p(x)$ & $q(x)$ taken.

parameterising model
- Parameterise density function

$$q_\theta(x) = N(x; \theta, I)$$

- Define distribution $q(z)$ implicitly

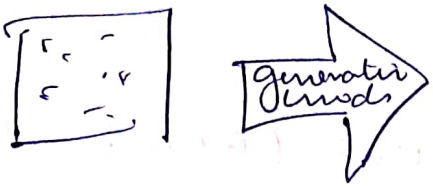
$$z \sim N(0, 1)$$

$$G_\theta(z) \sim q_\theta(x)$$

mapping z & x, simple sampling

Images

Noise $\sim N(0, I)$



Classifier

$$f_\phi(x) = p_\phi(y=1|x)$$

Loss :

BCE

$$L(\phi, \theta) = E_{p(x)}[-\log f(x)] + E_{p(z)}[-\log(1 - f_\phi(G_\theta(z)))]$$

update classifier

$$\phi^* = \arg\max_\phi L(\phi, \theta)$$

update generator

$$\theta^{new} = \theta^{old} + \frac{\partial L(\phi^*, \theta^{old})}{\partial \theta}$$

repeat.

learning scheme

1. update guide

- $\frac{p(x)}{p(x) + q(x)}$
- $\frac{p(x)}{q(x)}$
- $p(x) - q(x)$
- $D(p||q)$

2. update generator.

- Move $q(x)$ closer to $p(x)$

3. Repeat.

Prescribed (eg VAE)

- direct access to both prior, likelihood, conditional distributions

Implicit (GAN):

- evaluate & sample prior $p(z)$, posterior can only be sampled. Can approx $q(x)$, $q(z|x)$ from samples.

for classifier

Instead of $\min L(\phi, \theta)$

$$\max -L(\phi, \theta) = -\log 4 + 2D_{JS}(p||q_\theta$$

↓

Estimate distance $G(w)$ $P(x)$ and $q(x)$

update generator

- Same
- minimize distance.

This becomes a min max problem a/w classifier & generator

F- divergence & w- distance is taken.

F- divergence:

for $P(x)$ & $Q(x)$.

$$D_f(P||Q) = \int_x f\left(\frac{P(x)}{Q(x)}\right) Q(x)\, dx$$

different forms of f gives different divergences.

$f(t) = t\log(t)$ → KLD.

$$D_f(P||Q) = KL(P||Q)$$

$f(t) = -\log(t)$ → Reverse KLD

$$D_f(P||Q) = KL(Q||P)$$

$f(x) = \frac{1}{2}|t-1|$ → total variation

$$D_f(P||Q) = \frac{1}{2}\int_x |P(x) - Q(x)|\, dx$$

W- distance

cost of transporting x to y → $c(x, y)$

eg: $c(x, y) = \|x - y\|$

optimal transport:

$$T(P,Q) = \inf_{\gamma \in \Gamma(x\sim P, y\sim Q)} E_{(x,y)\sim\gamma}[c(x,y)]$$

- $\Gamma(x\sim P, y\sim Q)$ ↓ Set of all joint distns of (x,y) with P & Q.
- $E_{(x,y)\sim\gamma}[c(x,y)]$ ↓ expected cost

OT dual:

- Primal

$$T(P,Q) = \inf_{E\in(\ )} E_{x\sim P}[c(x,y)]$$

- Dual (W-1 metric)

$$T(P,Q) = W_1(P,Q) = \sup_{\|f\|_L \le 1} E_{x\sim P} f(x) - E_{x\sim Q} f(x)$$

↓ Lipschitz constant ≤ 1

↓ b/w continuity and smoothness.

↙ can be enforced by 2 methods

① clip weights of n/w

hope o/ps of n/w for similar points do not deviate much

② gradient penalty.

Evaluate expectation over set of $\tilde{x}$.

~~$\lambda E_{\tilde{x} \sim P(\tilde{x})}$~~

$$\lambda E_{\tilde{x} \sim P(\tilde{x})} (\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2$$

$\tilde{x}$ → convex combination b/w real & fake data points.

enforce gradient of the discriminator to be 1 almost everywhere.

Optimal Transport vs f-D

$$Z \sim U[0,1]$$

$$V = (0,2) \quad \delta \text{ fn at } 0$$

$$Q = (\theta, 2) \quad \delta \text{ fn at } \theta$$

$W_1(P,Q) = \theta$ → gives conti signal

$$JS(P\|Q) = \begin{cases} \log(2) & \theta \neq 0 \\ 0 & \theta = 0 \end{cases}$$ → grad = 0

$$KL(P\|Q) = \begin{cases} \infty & \theta \neq 0 \\ 0 & \theta = 0 \end{cases}$$ don't use due to ∞,

Practical tricks to make GAN training easier

(Always use these)

- Spectral Normalisation
  - grad penalty - is a strict constraint
    - introduces significant bias
  - easier way to introduce L-1 conti

- idea is n/w is composition of dom fns, if we can make each of this L-1 conti, n/w can be a L-1 conti fn. So linear fn can be made L-1 fn. It is good with some non-linearity.
- Can conv to made L-1 conti?

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix} \quad x$$

$$\begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} \quad w$$

$x \circ w_2$ $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ $\begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix}$ $x \circ w_2$

w

x

$x \circ w_2$ $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

↓

linear o/p.

$$x \circ w_2 \begin{bmatrix} x_1w_1 + x_2w_2 + x_4w_3 + x_5w_5 & x_2w_1 + \dots \\ x_4w_1 + \dots & x_5w_1 + \dots \end{bmatrix}$$

↓

linear o/p

reformat it into multi of some matrix & some vector

$$\begin{bmatrix} w_1 & w_2 & 0 & w_3 & w_4 & 0 & 0 & 0 & 0 \\ 0 & w_1 & w_2 & 0 & w_3 & w_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_1 & w_2 & 0 & w_3 & w_4 & 0 \\ 0 & 0 & 0 & 0 & w_1 & w_2 & 0 & w_3 & w_4 \end{bmatrix}$$

Aw

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix} \tilde{x}$$

∴ $\tilde{x}$ now $A\omega \circ \tilde{x}$ → reshape this from (4×1) to (2×2)

L-1 conti of a linear operator

$$\|f(x_1) - f(x_2)\|_2 \leq L \|x_1 - x_2\|_2$$

(can take any norm)

$$\|Ax_1 - Ax_2\|_2 \leq L \|x_1 - x_2\|_2$$

$$\frac{\|Ax\|_2}{\|x\|_2} \leq L$$

σ_{max} = $\sup_x \underbrace{\frac{\|Ax\|_2}{\|x\|_2}} \leq L$

Spectral norm of matrix equal to max singular value of matrix

$$1 = \sup_x \frac{\left\| \frac{A}{\sigma_{max}} x \right\|_2}{\|x\|_2} \leq L$$

→ this satisfy L-1 conti

Singular value:

using SVD.

$A = U \Sigma V^T$ → right singular

↙ left singular (orthogonal)

$\sigma = U^T A V$

U & V can be used to find σ_{max}.

to find σ_{max}:

power iteration

assume EVD for simplicity

$$A = Q \Lambda Q^{-1}$$

$A = Q \Lambda Q^{-1}$

let x_1 be init a randn with unit norm

Run the iterations.

1. $x_{i+1} = A x_i$

2. $x_{i+1} = \frac{x_{i+1}}{\|x_{i+1}\|_2}$

we converge to EV, q corresponding to λ_{max}

$\lambda_{max} = q^T A q$.

Similarly σ_{max} is computed.

All o/ps are diff wrt A, & W

This is done one step at each G.D step at gen / discri. Keep the prious q and use them to get better approx.

Theory Vs practice

Instead of A_W, a reshaped version W is used

$C_{out} H_{out} W_{out} \times C_{in} H_{in} W_{in}$ ↑ $C_{out} \times C_{in} K_h K_w$

- resulting σ_{max} is a lower bound to true σ_{max} using this power iteration method,
- σ_{max} is equal only for a linear layer.
- this outperms true spectral norm

game theory view on GANS

In theory.

ψ^* : $\arg\max_\psi L(\theta^{old}, \psi)$ → optimal discri

$\theta^{new} = \theta^{old} - \alpha \nabla_\theta L(\theta^{old}, \psi^*)$ → gen single step

$\theta^{old} = \theta^{new}$

$D(P||\theta)$

- For most Div, optimal value will be constant. (zero grad)
- In practice, we don't get exact Div, we estimate it which has non zero grad

In practice:

$\psi_{new} = \psi^{old} + \alpha \nabla_\psi L(\theta^{old}, \psi^{old})$

$\theta^{new} = \theta^{old} - \alpha \nabla_\theta L(\theta^{old}, \psi^{new})$ → alternating G-D

$\theta^{old} = \theta^{new}$, $\psi^{old} = \psi^{new}$

L → minimise lower bound on div

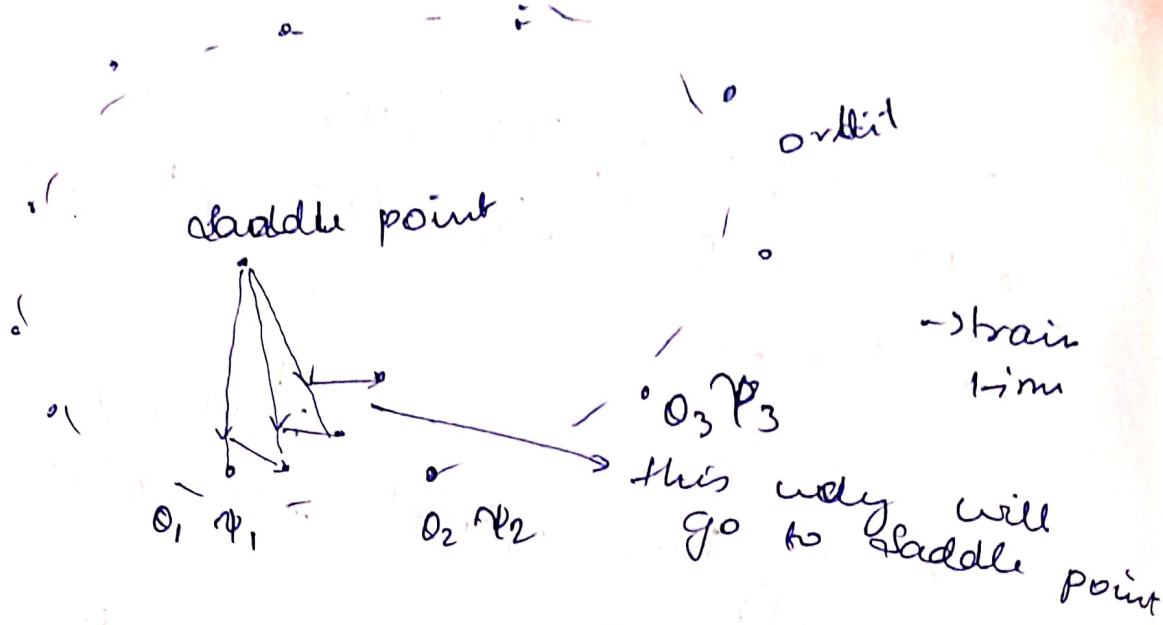
Simplifying Loss

$$\min_\theta \max_\psi E_{x \sim \delta_\theta} f(\psi, x) + E_{z \sim \delta_\theta} f(\psi, z)$$

$$\min_\theta \max_\psi f(\theta) + f(\psi\theta)$$

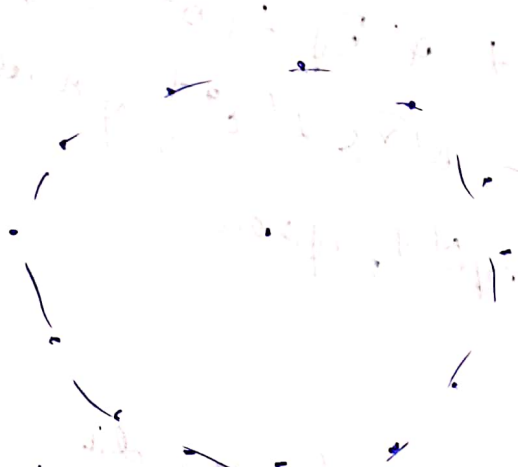
$$= \min_\theta \max_\psi f(\psi\theta)$$

Exponential moving average.



$\hat{\theta}_{i+1} = (1-\alpha)\hat{\theta}_i + \alpha\theta_{i+1}$ → EMA of weights

weight averaging.



average all points

$\theta_2 \frac{1}{N-n} \sum_{n}^{N} \theta_i$

When to start averaging?

- have to compute the orbit
- and keep going cyclic
- but can't track this (tricky)

Soln

Stop at some epoch, collect weights & take avg.

Quality Measurements

- Best metrics are hand crafted & problem specific.
  (Human evaluation for eg:)
- Metrics for Images:

Inception Score (IS)

- Evaluates objectiveness & class distri
- Assumption - Single object in image
- take pretrained n/w, gen samples, run n/w on samples to get prob on each class on pretrained data
- Each prediction should have low entropy (high prob for single class)
- For collection of images, marginal distri should be high.

$$\log IS = E_x KL(p(y|x) || p(y))$$

$$= \sum_x p(x) \sum_y p(y|x) \log \frac{p(y)}{p(y|x)}$$

$$= \underbrace{\sum_y \log p(y) \sum_x p(y|x) p(x)}$$

$$= \text{entropy} \quad \underbrace{-\sum_x \sum_y p(y|x) p(x) \log p(y|x)}_{H(y|x)}$$

$$= H(y) - H(y|x)$$

Frechet Inception Distance (FID)

- Measures quality of image.
- Compare activations of pre-trained convnet b/w real & generated.

$$FID = \|\mu_r - \mu_g\|^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

Compare only μ mean & pairwise correlations

Applications

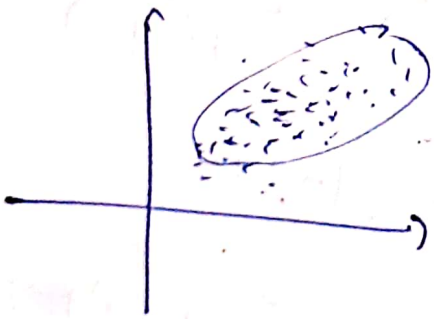
- 2k inception quality image generations (StyleGAN)
- extend datasets
- super resolution, text2speech etc.

Normalising flows.

Probabilistic view on generative modeling.

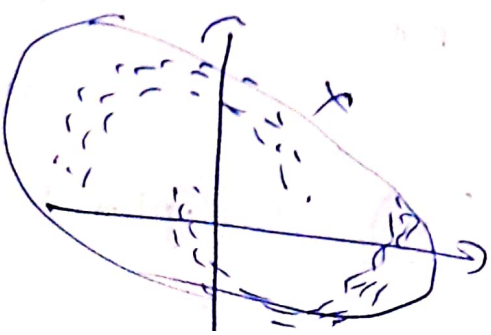
- learning distr on data

$p_{model}(x) \approx p_{data}(x)$



→ can use gaussian distr

$N(x | \mu, \sigma^2)$



→ can't use gaussian

more applications

- Probability estimate (out of distr estim distr)
- Representation learning (semi- supervised)

grad based training of probabilistic GMs

samples from distr $X = (x_1, x_2, \dots, x_N)$ $x_i \sim P_{data}(x)$

parametric model $P_\theta(x)$

- assume model is suitable for grad opti by θ
- prob constraints (valid pdf output)
- desirable to have a tractable sampling procedure.

Train using MLE

$$\log P_\theta(X) = \sum \log P_\theta(x_i) \to \max_\theta$$

solved using S.G. ascent, Adam etc,

Eg:

$$P_\theta(x) = \mathcal{N}(x | \mu, \Sigma) \qquad \theta = (\mu, \Sigma)$$

MLE: $\sum_i \log \mathcal{N}(x_i | \mu, \Sigma) \to \max_{\mu, \Sigma}$

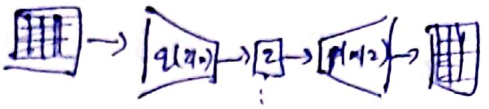
update μ, Σ

$\mu = [2 \;\; 2]$

$\Sigma = [[0.4 \;\; 0.2]$

Choosing model for complex data.

Variational AEs (semi-implicit model)



$p(z)$

$\log p_\theta(x) = \log E_{p(z)} p_\theta(x|z)$

↓

unbiased estimate

Variational inference and

$\log p_\theta(x) \geq L(x, \theta) \to \max_\theta$

→ an unbiased estimate, apply non linear fn ($\log$) then it won't remain unbiased.

expressive and tractable pdf

$p_\theta(x)$ = Normalising flow

- expressive parametric model
- Tractable & exact $p_\theta(x)$
- fast & tractable sampling

Normalising flows NF

Change of variable formula

1d - case

$z \sim p(z)$

Given $p(z)$

$z = f(x)$

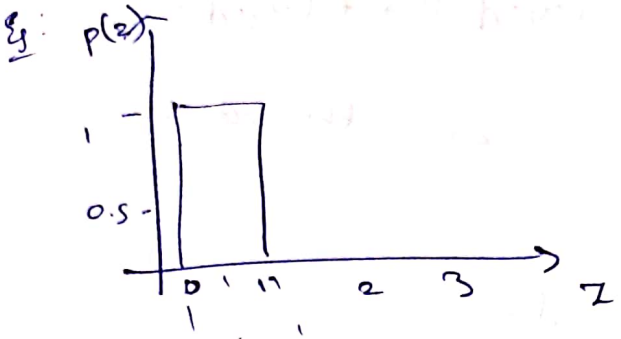
$p(x) = ?$

$$p(x)|dx| = p(z)|dz|$$

$$p(x) = p(z)\left|\frac{dz}{dx}\right|$$

$$= p(f(x))\left|\frac{df(x)}{dx}\right|$$

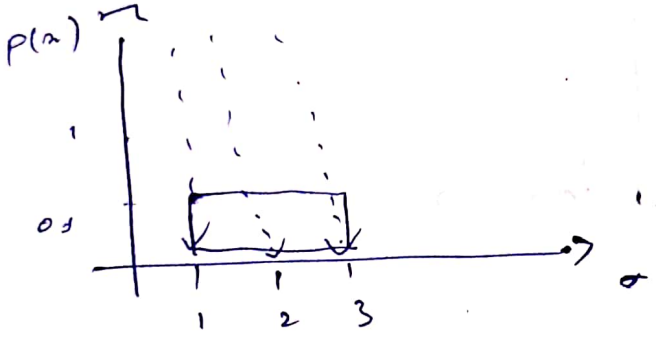
Eg:



$x = 2z + 1$

$z = \left(\frac{x-1}{2}\right)$

$\frac{df(x)}{dx} = 0.5$



prob goes down

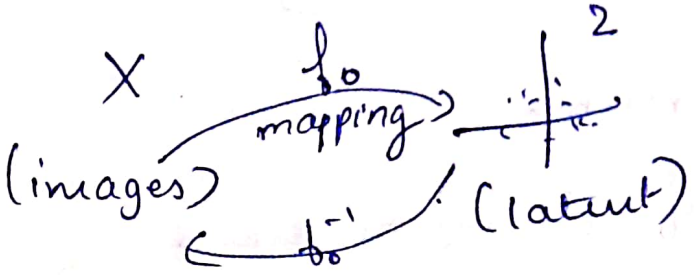
N-d case

$$p(x) = p(f(x))\left|\det \frac{\partial f(x)}{\partial x^T}\right|$$

↓

Jacobian.

NF concepts



f_θ Should be reversible

$$p(x) = p(f(x))\left|\det \frac{\partial}{\partial}\right.$$

Coupling layer.

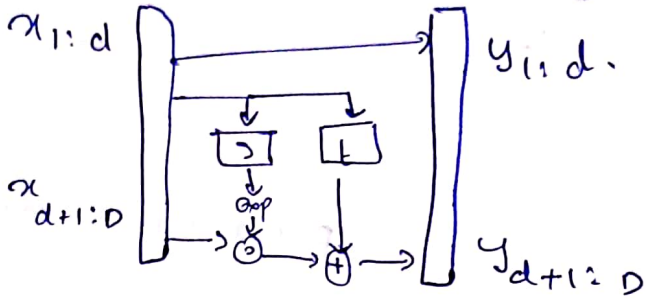
i/p x , o/p y.

Forward:

$$y_{1:d} = x_{1:d}$$

$$y_{d+1:D} = x_{d+1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d})$$

s and t are conditioned on the d components



Inverse:

$$x_{1:d} = y_{1:d}$$

$$x_{d+1:D} = (y_{d+1:D} - t(y_{1:d})) \odot \exp(-s(y_{1:d}))$$

∴ It is reversible

Jacobian:

$$\frac{\partial y}{\partial x^T} = \begin{bmatrix} I_d & 0 \\ \frac{\partial y_{d+1:D}}{\partial x^T_{1:d}} & \text{diag}(\exp[s(x_{1:d})]) \end{bmatrix}$$

This is a lower Δ matrix.

∴ det = product of diag elements

$$\det \left| \frac{\partial y}{\partial x^T} \right| = \exp \sum_j s(x_{1:d})_j$$

- can't stack coupling layers directly.
- so add permutation in between coupling layers

model.

$f_\theta(x): R^D \to R^D$ $\quad f_\theta(x) = f^N \circ \dots \circ f^1(x)$

(coupling layers)

$$\log p_\theta(x) = \log p(f_\theta(x)) + \log \left| \det \frac{\partial f_\theta(x)}{\partial x^T} \right|$$

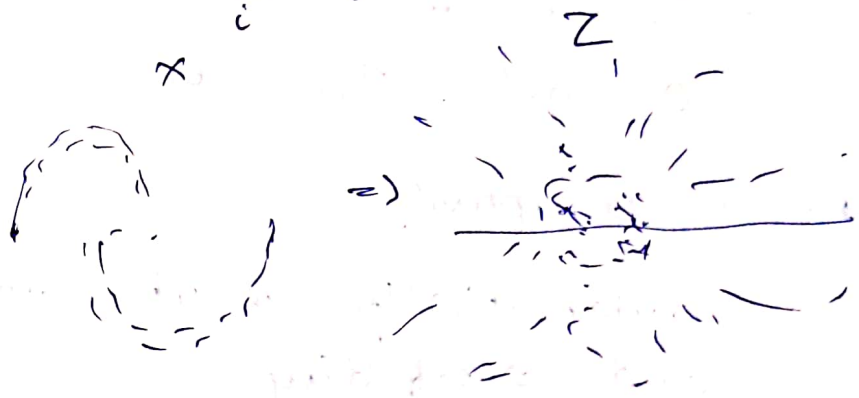
$$= \log p(f(x)) + \sum_1^N \log \left| \det \frac{\partial f^i}{f^{i-1}} \right|$$

$$\log p_\theta(x) = \sum_i \log p_\theta(x_i) \to \max_\theta$$

Inference

$x \sim \tilde{p}_x$

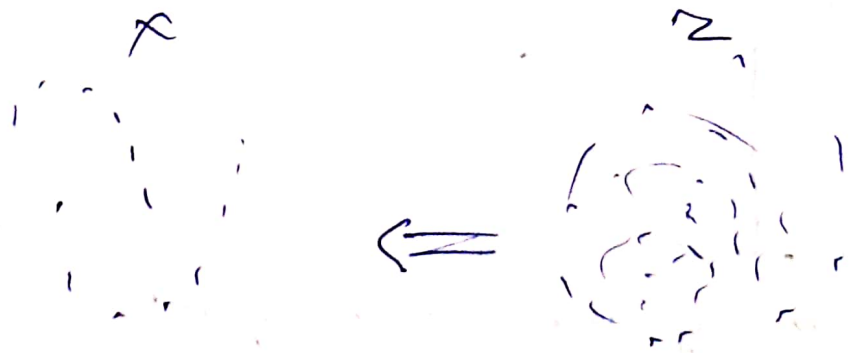
$z = f(x)$



generation

$z \sim p_z$

$x = f^{-1}(z)$



each coupling layer perform zoom scaling and shifting.

Measuring quality

Continuous data → avg test log likelihood

- Normalising flow can be applied to variational inference (eg in VAE after encoder)

Gaussian Process and Bayesian optimization

Applications

- Real valued regression
- Classification
- Ordinal regression

Model complexity:

- automatically select model complexity and overfitting.

gaussian distribution

divide gaussian vector f into 2 sub vectors

$p(f_1, f_2) \sim \mathcal{N}(f_1, f_2 | \mu, \Sigma)$

Joint

also divide μ & Σ into μ_1, μ_2 & $\Sigma_{11}, \Sigma_{12}, \Sigma_{21}, \Sigma_{22}$

$$\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$$

$$p(f_1) = \int p(f_1, f_2) df_2 = N(f_1 | \mu_1, \Sigma_{11})$$

gaussian process model

- assume data is generated by a conti. stochastic process.
- assume covariance matrix is from a kernel function

$$K_2 \{K_{ij}\} = \{K\{x_i, x_j\}\}$$

- squared exponential kernel

$$K(x, x') = \sigma_f^2 \exp\left\{ - \frac{\sum_1^d (x_i - x_j')^2}{2r^2} \right\}$$

a GP need not use the gaussian kernel.

- more general ex:

$$K(x, x') = \sigma_f^2 \exp\left\{ - \frac{|x - x'|}{r} \right\} + \sigma_1^2 + \sigma_2^2 \delta(x, x')$$

σ_f^2 - marginal func variance
σ_1^2 - variance of bias
σ_2^2 - noise variance
r - lengthscale
$\nu \geq 1$ - roughness

- Probabilistic prior assumptions in form of covariance

- new covariance can be constructed from old

GP regression

data $S_m = \{X, y\} = \{(x_i, y_i)\}_{i=1}^{m}$

Model

$$y_i = f(x_i) + \varepsilon_i$$

↳ gaussian white noise $\mu = 0, \sigma_n^2$

$$f \sim GP(\cdot|0, K)$$

- $f(x_i)$s are generated & its noisy version is observed.

prior $P(f) = N(f|0, K)$

model noise & likelihood.

$$P(y|f) = N(y|f, \sigma^2 I_m)$$

marginal likelihood

$$P(y) = \int P(y|f)\, P(f)\, df = N(0, K + \sigma^2 I_m)$$

Prediction

$$y_* = f_* + \varepsilon_* \qquad f_* = f(x_*)$$

$$P(y, f_*) = N\left(0, \begin{bmatrix} K + \sigma^2 I_m & k_* \\ k_*^T & k_{**} \end{bmatrix}\right)$$

$k_* = \{K(x_*, x_i)\}_1^m$ $\quad K_{**} = K(x_*, x_*)$.

$$p(f_*|y) = N(f_*|\mu_*, \sigma_*^2)$$

where $\mu_* = k_*^T [K + \sigma^2 I_m]^{-1} y$, $\sigma_*^2 = K_{**} - k_*^T [K + \sigma^2 I_m]^{-1} k_*$

Even in a GP with no parameter, there can be hyperparameter σ^2: noise variance & params of cov fn.

Learning GP model

- ability to choose hyper param and cov directly from data
- minimise negative log marginal likelihood $L(\theta)$ wrt cov fn params & noise θ.

$\therefore p(y) = N(0, K + \sigma^2 I_m)$

then

$$L = -\log p(y|\theta)$$

$$= \underbrace{+\frac{1}{2} \log \det C(\theta)}_{\text{Reg}} + \underbrace{\frac{1}{2} y^T C^{-1}(\theta) y}_{\text{data fit}} + \frac{m}{2} \log(2\pi)$$

$C = K + \sigma^2 I_m$

- minimisation of $L(\theta)$ is a non convex optim problem
- gradients

$$\frac{\partial L}{\partial \theta} = \frac{1}{2} tr\left(C^{-1}\frac{\partial C}{\partial \theta_i}\right) - \frac{1}{2} y^T C^{-1} \frac{\partial C}{\partial \theta_i} C^{-1} y$$

$\downarrow$

$O(n^3)$ $\quad C \to n \times n$

Cost can be approximated using random fourier features, special quadrature based formulas. So overall complexity reduces to linear (almost?)

Bayesian optim

Global optim:

Consider $f: X \to \mathbb{R} \quad X \subseteq \mathbb{R}^d$

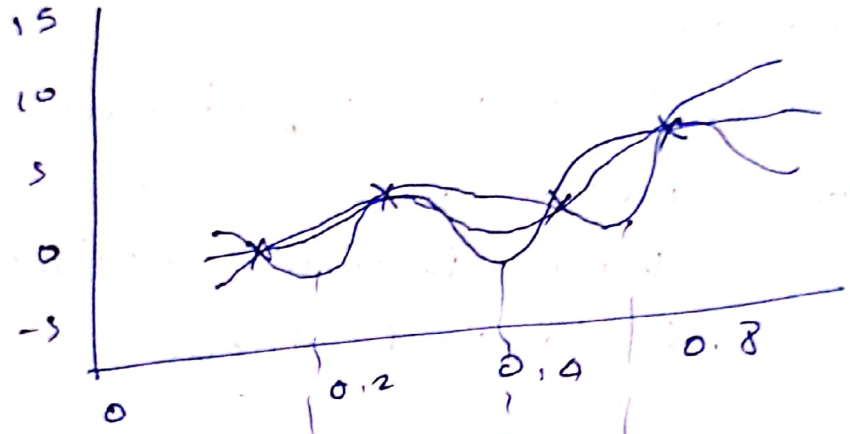
$$x_{min} = \arg\min_{x \in X} f(x)$$

- f is multimodal, expensive to evaluate
- So random search can't be done
- grads can't be computed numerically.
- Real world problems often deal with this, Eg: optim of struct of aircraft, lot of params, single sim might take days.
- Eg: optim geometry of a struct to minimize the mass, and strength is big, very time consuming, physical sims needed on each optim.

Eg: Modern DNNs.

- params:
  - No of layers
  - Units per layers
  - Reg. Coeffs
  - Lr etc.
- might take a lot of time
- how to get best struct?

Eg: design of grafts to used in heart surgery

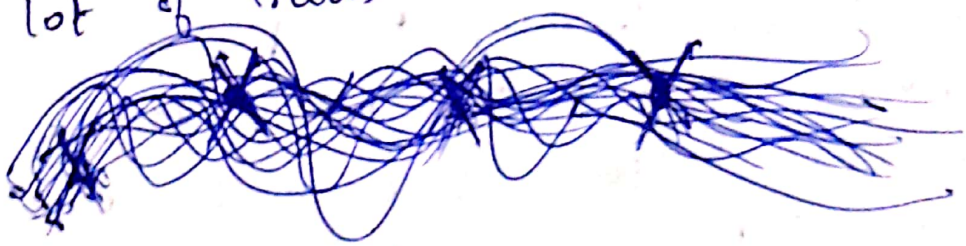


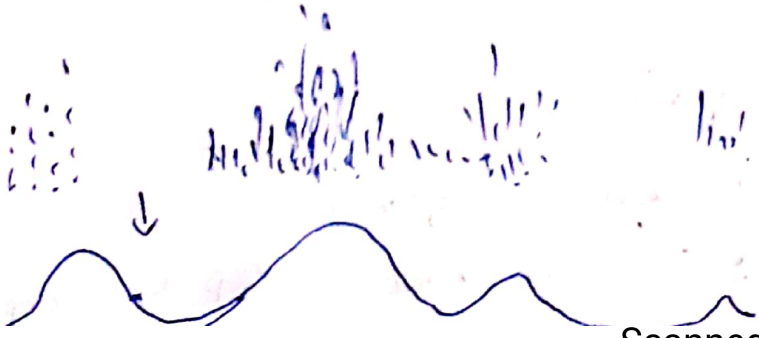
4 points

-> 3 diff minimas
- for 3 f(x)

the more f(x) we optimise, the more locations of minima found.

after lot of runs





-> minima concentrates

-> gaussian prom

① is the posterior given the minima concentrated
(-but we don't consider global minima?)
- prior assumption on f.
- samples values of f.
- Calculate posterior distri & position of minima
- sample around the locations of minima
- add them new observations & iterate
- On this, we need to support exploration and exploitation
  - exploration -> var is large (big uncertainty)
  - exploitation -> where mean is low.
    (should indicate optimum is located there)

- Consider some heuristic criterion to sample points
- on gaussian assumes, if the f(x) is generated by some gaussian process then if it follows the ~~process~~ criterion mentioned below, the proc will converge ~~to~~ in finite no of steps to global minima. Speed of conv is log, lower than G.D.

- take f(x)
- sample from f(x)
- Using this, construct GP to approx f(x)

- Using one of the criterion, select the next point which will evaluate the $f(x)$. Repeat.

Criterion to select candidate:
(acquisition fns)

← $\alpha_{LCB}(x) = -\mu_*(x) + \xi \cdot \sigma_*(x)$ ← user defined param for fn

↓

-ve sign because maximin μ

- have to find real minima

- $\sigma(x)$ is the uncertainty estimate to add the error in prediction.
- can be considered as a confidence interval for predictor positions of min value.

[ERP upper (lower) Confidence Band]

- Expected Improvement
  - prob criterion
  - estimates avg value of improv over existing obs.

$\alpha(x) = y_{best} - \mu_*(x)$, $y_{best} = \min_{i=1,\dots,m} y_i$,

↓

have to minimise this

$\alpha_{EI}(x) = \int \max(0, y_{best} - y_*) p(y_*|x)\, dy_*$?

↓

we don't care about -ve

- maximum prob of improvement

$$\gamma(x) = \frac{\mu(x) - y_{best}}{\sigma(x)} \rightarrow \text{normalised va}$$

$$\alpha_{MPI}(x) = P(f(x) < y_{best}) = \phi(\gamma(x))$$

Bayesian

Instead of optim initial fn, construct aquaition fn

$$x_{t+1} = \underset{x \in X}{\arg\max}\ \alpha(x | S_t)$$

$\alpha(x)$ → not so expensive

grads are typically available

- no need to find global optim for many cases, just need better than dig to getting to a local is fine

Tuning hyperp of SVM

$n_{samples} = 2500$

$n_{feature} = 41$

$n_{informative} = 15$

$n_{redundant} = 5$

- optimise wrt $x = (C, \gamma)$

C - penalisation

γ - kernal width

- target fn $L(x)$ is AUC, 3 fold CV

- put random values of C, V, then estimate AUC.
- Construct gaussian process using minima obtained
- optimize on a criterion (Expected Improv) & obtain point for next step. Repeat

About GP

- Simple to use
- not many params
- can estimate uncertainty

Deepbayes (continued)

Deep Gaussian Processes

linear models

- linear combination of basis fns

$$f(x) = w^T \psi(x)$$

- $\psi(x) = (\psi_1(x) \dots \psi(D))^T$.

- In regression, obs are modelled as

$$p(y \mid w, X, \lambda) = \mathcal{N}(\phi w, \lambda I)$$

$$\phi = \phi(X) = \begin{bmatrix} \psi_1(x_1) & \dots & \psi_D(x_1) \\ \vdots & & \\ \psi_1(x_N) & \dots & \psi_D(x_N) \end{bmatrix}$$

ϕw -> mean of latent fn.

I/ps $X = (x_1, \dots x_N)^T$.

$y = (y_1, \dots, y_N)^T$.

$w = (w_1 \dots w_D)^T$

quadratic loss => maximising likelihood

given a $p(w)$ prior, we learn $p(w \mid y, X)$

$$p(w \mid y, X) = \frac{p(y \mid X, w) p(w)}{\int p(y \mid X, w) p(w) dw}$$

let $p(w) \sim \mathcal{N}(0, S)$

- prior of f is again gaussian

$$p(f) = N(0, K)$$

$$K = \text{Cov}(f) = E[\phi w w^T \phi^T] = \phi S \phi^T$$

we can solve either by

$$p(y|w, x, \lambda) = N(\phi w, \lambda I) \quad p(w) = N(0, S)$$

or

$$p(y|w, x, \lambda) = N(f, \lambda I) \quad p(f) = N(0, K)$$

(prior in terms of f)

- Bayesian linear regression is a solved problem we can analytically compute posterior etc.
- But, we need to specify basis functions
- Now we can work with functions such as cov K, instead of parameters. Using this, we can choose K from inputs and evaluate ϕ instead of other way around. K can be defined such that ϕ is infinite dim

Eg: $$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2}\right)$$

↓

exp doesn't allow disentanglement of relation b/w x_i & x_j.

Taylor expansion is needed. This turns out to be scalar product of things that go to infinite dim.

- There are other kernels satisfying this property.

- In Bayesian LR.

prior is G, likelihood is G so.

posterior must be G

overw: - posterior $p(w|x,y,\lambda) = N(\mu_w, \Sigma_w)$

Cov $$\Sigma_w = \left(\frac{1}{\lambda}\phi^T\phi + S^{-1}\right)^{-1}$$

mean $$\mu_w = \frac{1}{\lambda}\Sigma_w\phi^T y$$

Predictions

$$p(y_*|x,y,x_*,\lambda) = N\left(\psi(x_*)^T\mu_w, \lambda + \psi(x_*)^T\Sigma_w\psi(x_*)\right)$$

can be rewritten using K.

over f.

$$p(f|x,y,\lambda) = N(\mu_f, \Sigma_f)$$

$$\Sigma_f = K - K(K+I)^{-1}K$$

$$\mu_f = K(K+I)^{-1}y.$$

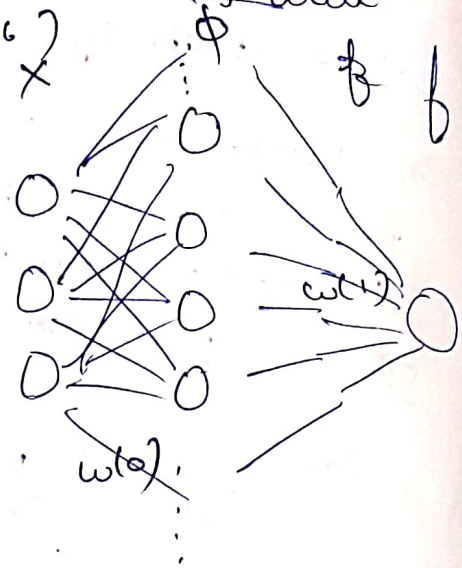
$$p(y_*|x,y,x_*,\lambda) = N\left(k_*^T(K+I)^{-1}y, \lambda + k_*^T(K+I)^{-1}k_*\right)$$

Both are equivalent.

$\psi(\cdot)$ costs $O(D^2)$ mem, $O(D^3)$ time

$k(\cdot,\cdot)$ costs $O(N^2)$ mem, $O(N^3)$ time

gaussian processes as infinitely wide shallow nn (1996)



- Central limit theorem implies that f is gaussian
- f has zero mean
- $\text{cov}(f) = \mathbb{E}_{p(w^{(0)}, w^{(1)})}[\phi(x w^{(0)}) w^{(1)} w^{(1)T} \phi(x w^{(0)})^T]$

$$= \alpha_1 \mathbb{E}_{p(w^{(0)})}[\phi(x w^{(0)}) \phi(x w^{(0)})^T]$$

- Some choices of ϕ leads to analytic expression of known kernels.

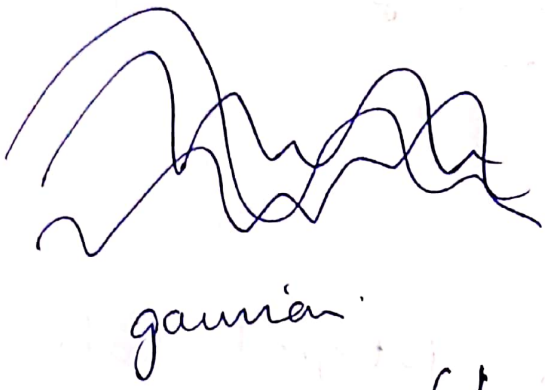
Challenges & limitations on GP

- Kernel design $k(\cdot,\cdot|\theta)$ (before it was basis fn) (can ; have parameters for more flexibility)
- GPs can be too expensive
- GPs might not be tractable (says for classification).
  ↓
  need approx.

Eg:

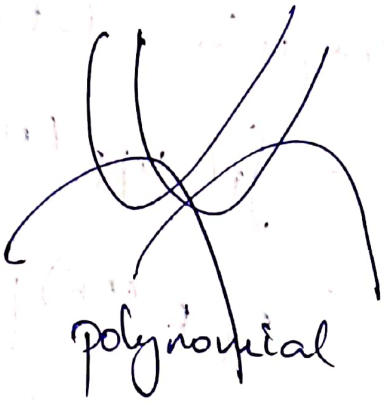
$$p(y|x,\theta) = \int p(y|f)\, p(f|x,\theta)\, df$$

- can't compute (no conjugacy)
- Bernoulli other distri
- gaussian



gaussian

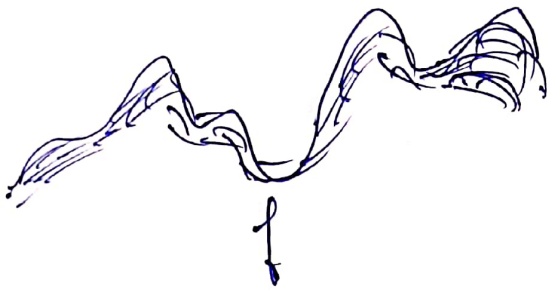
Matern



polynomial

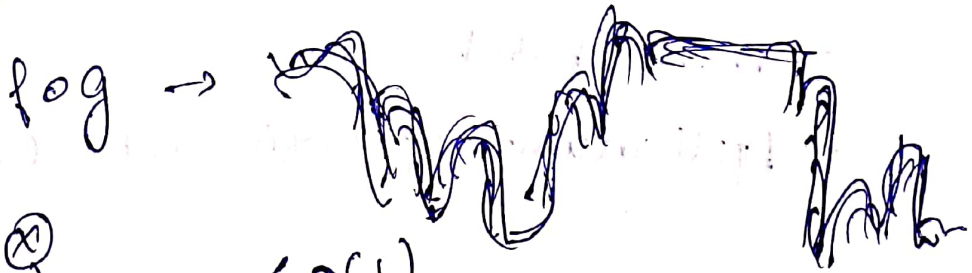
Deep GP for large Rep Power

- composition of processes

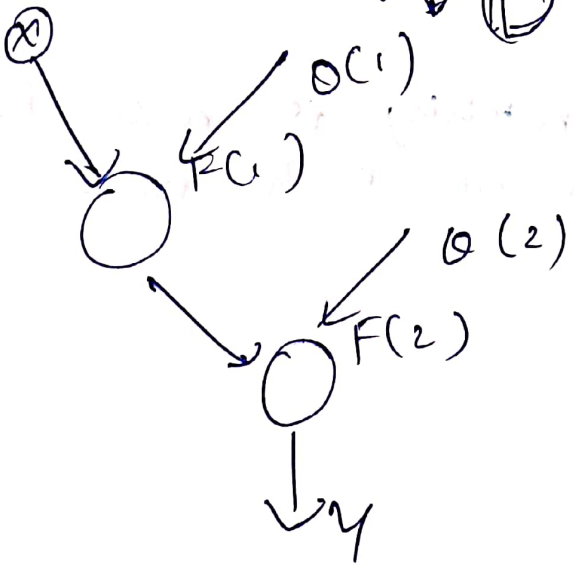


f

g



$f \circ g \rightarrow$



- increase the rep power of GP
- also complicate the model

Pathologies of Deep gaussian process

- give input to later layer to help better representation.

learning deep gaussian process

- Inference requir. integral Calc.

$$p(Y|X,\theta) = \int p(Y|F^{(N_h)},\theta^{(N_h)}) \times p(F^{(N_h)}|F^{(N_h-1)},\theta^{(N_h-1)}) \times \dots \times p(F^{(1)}|X,\theta^{(0)}) dF^{(N_h)} \dots dF$$

---

why ppl do DL & not GP?

- automatic diff.
- Reg
- GPUs, TPUs
- mini batch
- Applications specific reps (CNNs)

- In GPs, mini batching is really hard to do, need to invert K to mini batch.

Main ingredients for Inference for DGPs

- posterior $p(f|X, y, \theta) \approx q(f)$

(can be non gaussian as likelihood might not be gaussian)

(so approx it with a gaussian)

~~other approxs which deal~~

approx to gaussian:

- Laplace approx
- Expectation propagation
- Variational Bayes
- MCMC

- Now, scale model to lot of datapoints

Sparse GP ⇒ Nystrom approx

introduce $M < N$ new latent variables u at locations Z.

$$K \approx K_{xz} K_{zz}^{-1} K_{zz}$$

- Fully Independent Training Conditionals.

diagonal of K is not correct, so

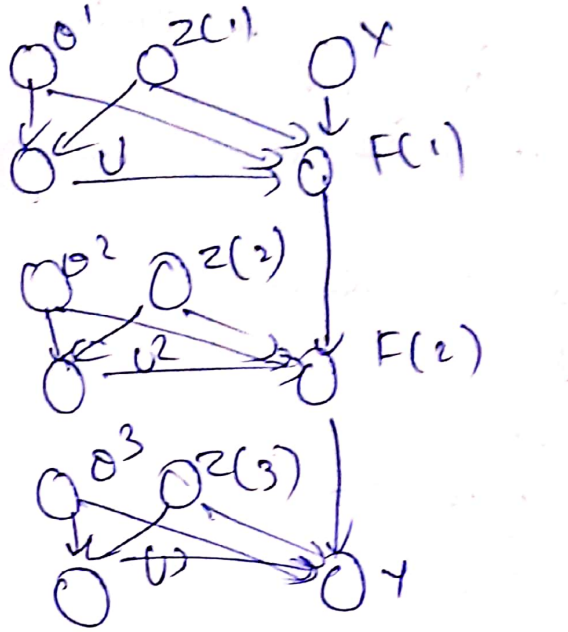
- partially independent training Conditionals (PITC is done)

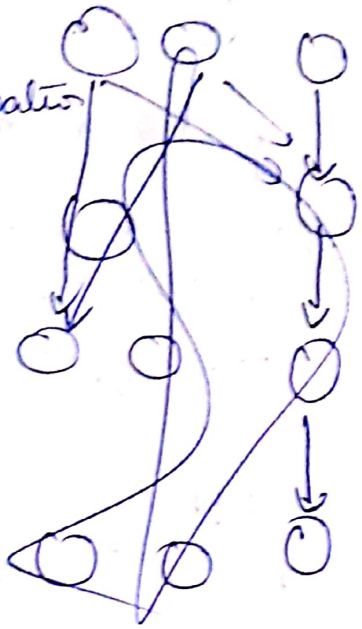
- Variationally sparse GP
  - introduces distribution and $q(f, u) \approx p(f|u)\, q(u)$
- Random feature expansion
- Kronecker (Toeplitz) Tensor Structured GPs

Inducing

- Inference for DGPs

Scalable expectation propagation



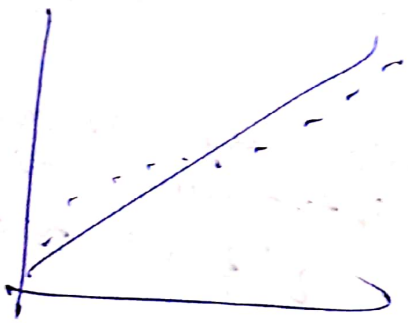


- new rand variabls z at every law.
- hyperparams for kernels & latent vars

- DGPs with rand features beyond CNNs
- Can be learnt using Stochastic Variational

(Bound on fourier transform due to shift invariant)

Calibration as a measure of Quantification of Uncertainty



→ for deep CNNs.

Post Calibration fixes it.

what is the principled way of developing classifiers, which by construction calibrated

- gaussian proces + CNN.

CNN layers → DGP → O/P

- Bayesian CNNs are calibrated
  - Need to be bayesian wrt filters
  - have to regularise filters.
  - Eg: Using Monte Carlo Dropout

another method
- Replace linear with Random Feature Expansions

- some new trends mentioned.

Adaptive skip grams

Representation learning
- dependent on feature
- K-means might give similar colour data not similar semantic data.

Rep L for words

- goal is to learn close embeddings for semantically similar words.

- Skipgram.
  - learn i/p embed & o/p embed
  - take current word x and task context words y (words surrounding x)

- $p(y|x) = \prod_j p(y_j|x)$
- assume they are independent
- based on distributional hypothesis: Similar words appear on similar context.

$$\theta \leftarrow \theta + \eta \nabla \log p(y_i|x_i)$$

then pick next x & repeat till convergence

- learns rich embeddings
- sparse gradients
- Very efficient parallel training.

Is it good enough?

- for some words, only one meaning was learnt.
- for some words, diff meanings might get mixed up in one vector.

Why is it happing?

- word ambiguity

Soln:

- latent variable model

training using EM

observed $D = \{(x_i, y_i)\}_1^N$

latent $Z = \{z_i\}_1^N$ each meaning in local context, local

$\Pi = \{\pi_w\}_1^V$ meaning probability, global

param $\theta = \{in_{w,k}\}_{w=1\,k=1}^{V,K} \cup \{out_w\}_1^V$ (embeddings)

toget $\log p(D|\theta)$

$$= \log \int p(\pi|\alpha) \int p(z|\pi) \prod_i^N p(y_i|x_i, z_i, \theta) \, dz \, d\pi$$

$$\rightarrow \max_\theta$$

↓

intractable.

So consider a fully factored posterior approx

$$q(z, \pi) = \prod q(\pi_w) \prod q(z_i) \approx p(z, \pi | D, \theta)$$

Then introduce variational bound.

Then,

E-step

- assume current approx of $q(\pi)$
- solve $q_*(z) = \prod q^*(z_i) = \arg\max_{q(z)} L(q(\pi), q(z))$
- Perform word sense disambiguation

M-step

- stochastic update of global params
  - Due to conjugacy $q(\pi_w) = \text{Dirichlet}(\gamma_{w1}, \ldots \gamma_{wk})$
- VLB has a finite-dim numeric parameterization.

$$L^*(\gamma, \theta) = E_q [\underbrace{\log p(\pi|k) - \log q(\pi|\gamma)}_{-KL \text{ b/w Dirichlet}} + \underbrace{\sum\sum q^*(z_i=k) \log p(y_i | x_i, k, \theta)}_{\text{weighted skip gram pred}}] + \underbrace{H(q^*(z))}_{\text{constant}}$$

SVI for Multi-meaning Skipgram model

(Stochastic variational inference)

- Sample a context $(x, y) \sim D$
- Dis ambiguate word meaning

$$q(z) \propto \exp\left(E_{q(\pi_x)} \log(z|\pi) + \log p(y|x, z, \theta)\right)$$

- update word embeddings

$$\theta \leftarrow \theta + \eta \nabla \sum_{1}^{K} q(z=k) \log p(y|x, z=k, \theta)$$

- update params of $q(\pi_x | \gamma_x)$

$$\gamma_{x,k} \leftarrow + \eta \underbrace{(q(z=k) n_x - \gamma_{x,k})}_{\text{natural grad.}}$$

Choosing number of meanings

- Fix k for each word (not very good)
- Use a heuristic, based on word freq (bit non trivial)
- external knowledge, speech tagging.

Given a distn of points, how many clusters?

traditional model selection

- define a finite mixture model

$\pi | \alpha \sim \text{Dir}(\alpha_1, \ldots \alpha_K)$

$\phi_k \sim H, k = 1, \ldots, K$ (cluster params)

$H \rightarrow$ prior distn over cluster params

$\mu_k, \Sigma_k = P_H(\mu, \Sigma)$

$z_i | \pi \sim \text{Categorical}(\pi)$ (for each data point, choose a cluster from categorial distn)

$x_i | z_i, \phi \sim p(x | \phi_i)$

- For each K measure $\log p(x_{val} \mid x_{train})$
- This is computationally inefficient.
- Not very elegant (have to make sure val data comes from same distn)

Model selection with Dirichlet process

- Define a countably infinite mixture model

$$G = \sum_{k=1}^{\infty} \pi_k \delta(\phi - \phi_k)$$

infinite number of cluster params, each with its weight.

- $p(x_i \mid G) = \sum_{k=1}^{\infty} \pi_k p(x_i \mid \phi_k)$

- choose a nonparametric prior.

$$G \mid H, \alpha \sim DP(H, \alpha)$$

(prior over cluster param; scalar value; ↳ Dirichlet process)

- select the model one-shot

$$p(x \mid X) = \int \underbrace{p(G \mid x)}_{\text{model}} p(x \mid G)\, dG.$$

$$p(G \mid x) \propto \underbrace{p(G)\, p(X \mid G)}_{\text{model selection}}$$

What is a good non parametric prior?
- should allow potential infinite number of clusters.
- for a finite dataset, then no of clusters $K \ll n$.
- Model complexity to grow with more data

Two representations of Dir Proc.
- Chinese Restaurant process
- Stick breaking process.

Chinese Restaurant Process

- imagine you are first customer & restaurant is empty. Take a table.
- next customer has a choice, to sit with you on same table or pick a new table.
- So at every time step new customer comes and has a choice

$$p(z_i = k \mid z_{<i}) \propto \begin{cases} n_k, & k \leq K \\ \alpha, & k = K+1 \end{cases}$$

$$p(z) = \prod_{i}^{N} p(z_i \mid z_{<i})$$

- treat then z as cluster assignments.
- each table is a cluster
- for each non empty table, sample cluster params from prior H.
- Then for generating, sample from corresponding mixture ~~param~~ component

- So hmm, we need to replace old prior over word over topic prob with the CRP prior.
- But data is not anymore IID. Because z_is are all conditioned on each other.
- So not convenient for SVI to train.

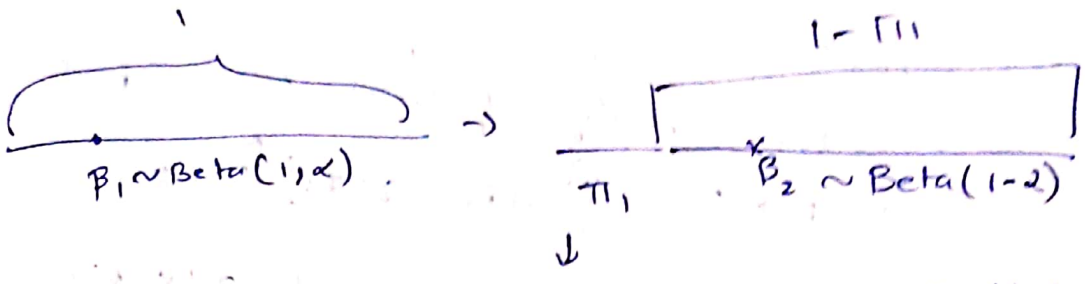
Stick Breaking Process.

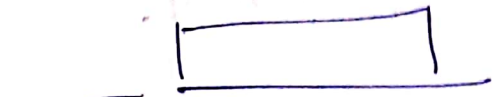
Length 1

On beginning, there is whole stick, then decide where to break it by

breaking proportion from Beta distri. $\beta(1,\alpha)$.

$\beta \in (0,1)$





π_1 π_2 → prior of mean 2.

Repeat this infinitely.

$\beta_k \sim Beta(1,\alpha)$ $k \to \infty$.

$\phi_k \sim H$, $k = 1, 2, \ldots \infty$

π_k prior prob of cluster.

$$\pi_k = \beta_k \prod_{l<k} (1-\beta_l), = \beta_k \left(1 - \sum_{l<k} \pi_l\right)$$

$z_i | \pi \sim$ categorical (π)

$x_i | z_i \sim p(x_i | \phi_k)$

$$G = \sum_{1}^{\infty} \pi_k \delta(\phi - \phi_k) \sim DP(H, \alpha) \quad z \sim CRP(\alpha)$$

In CRP, cluster assignment was sequential.

In SBP, it is done independently.

Adaptive - skipgram

- based on infinite mixture.
- prior over infinite many meanings
- Automatic model selection.
- Control of meaning granularity.

to compute it practically:

Variational inference for SBP

- finite dim var approx
- assum fully factored var approx

$$q(\beta, \phi, z) = \prod_{1}^{\infty} q(\beta_k) q(\phi_k) \prod_{1}^{N} q(z_i)$$

$$\approx p(\beta, \phi, z | x, \nu)$$

- still infinitely many β, ϕ.

$$q(\beta_T) = \delta(\beta_T = 1)$$

$$q(\beta_k) = \underbrace{\text{Beta}(1, \alpha)}_{\text{prior}}, \quad q(\phi_k) = p_H(\phi_k), k > T$$

We don't care after T

- for all $k > T$, ~~prior~~ posterior over β_k is the same as prior and for $k > T$, posterior over ϕ is same as prior

- Due to prior conjugacy:

$$q(\beta_k) = \text{Beta}(\beta_k | a_k, b_k)$$

lower bound decomposes

$$L(q) = E_q[\log p(x | \beta, \pi, z) - KL(q(\beta, \pi, z) || p(\beta, \pi, z))]$$

↓ likelihood ↓ KL(posterior || prior)

$$E_q\Big[- \sum_{1}^{T} KL(q(\beta_k) q(\phi_k) || p(\beta_k) p(\phi_k)) - \sum_{T+1}^{\infty} KL(q(\beta_k) q(\phi_k) || p(\beta_k) p(\phi_k)) \Big] - KL(q(z) || p(z | \beta))$$

$\to 0$

Markov Chain Monte Carlo

- problem with ELBO maximisation, $q(\theta|\lambda)$ could be a poor approx for the true posterior.

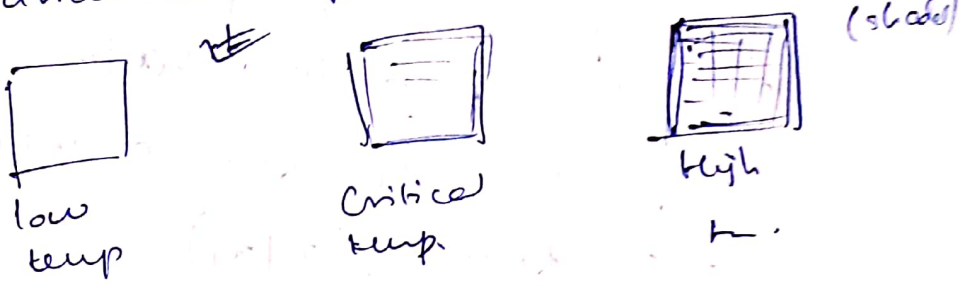
- There are cases where good approx are necessary.

Ising model

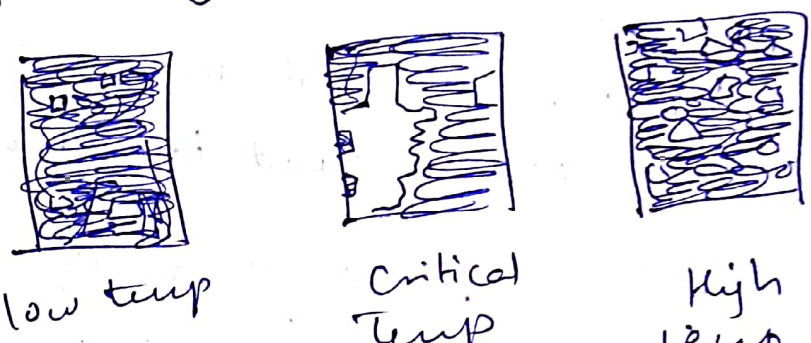
- model from statistical physics that try to describe magnetic property of solids depending on temp. In a short interval of temperature, magnet loses its properties (phase transition effect)

- Variational inference for Ising model

variational approx



from true posterior



Variational proxy is too poor for this model

- If variational proxy is more complex, it not efficient to estimate

MCMC

- able to model easily and to estimate needed statistics using this
- going to construct a samples from the distribution.

given $p(x) = \frac{1}{z}\tilde{p}(x)$, $z = \int \tilde{p}(x)dx$

MCMC → $(x_1, x_2, \dots x_m) \sim p(x)$ using only $\tilde{p}(x)$

given these samples

$$E_{p(x)} f(x) = \int f(x)p(x)dx \simeq \frac{1}{m}\sum_{i}^{m} f(x)$$

↓ Statistics

- to drawing samples using some specially construct markov chains.
  Transition probability $q(x|y)$
  - generate x_1 from a initial distri $p_0(x)$
  - x_2 from $q(x|x_1)$
  - x_3 from $q(x|x_2)$
  - ⋮

$x_1, x_2 \dots$ are not independent, but can still be used to estimate $E_{p(x)}f(x)$

- first samples can be really far from true
- In practice, first set of samples are dropped.

How to converge?

Markov chain properties

- invariant distribution

  Given a sample from true distri., transitions using a transition prob, it should give sample from our distribution.
  In practice, this is ensured by sampler.

- ergodic

  initial $P_0(x)$.

  $P_i(x)$, i^{th} step of a MC.

  Then

  $$P_i(x) \to \Pi(x) \quad , \quad i \to \infty \quad \forall P_0(x)$$

gibbs sampling

generate $p(x) = \frac{1}{2}\tilde{p}(x)$ $\quad x = (x_1, \ldots x_k)$

In this, consider 1D conditionals

$x_1^{new} \sim p(x_1 | x_2 \ldots x_k)$ (first coord of new point)

$x_2^{new} \sim p(x_2 | x_1^{new}, x_3, \ldots x_k)$ (second coord of new point)

$\vdots$

$x_k^{new} \sim p(x_k | x_1^{new}, x_3^{new} \ldots x_{k-1}^{new})$

$x^{new} = [x_1^{new}, x_2^{new} \ldots x_k^{new}]$

- can be proved about its invariance.

- if all $p(x_{ik} | x_{\setminus i}) > 0$, then ergodic

gibbs sampling : conclusions

- for discrete, conti, very general
- no params for tuning
- inefficient for high dims.

Metropolis - Hastings Sampling

goal is to generate $p(x) = \frac{1}{z}\tilde{p}(x)$

- introduce proposal distri $r(x|y)$. (this can be anything)
- to make it invariant, add a rejection step.
  - sample $x_{trial} \sim r(x|x_{old})$
  - accept or reject.

$$A = \min\left(1, \frac{\tilde{p}(x_{trial})\, r(x_{old}|x_{trial})}{\tilde{p}(x_{old})\, r(x_{trial}|x_{old})}\right)$$

  - $x_{new} = x_{trial}$ with prob A

    $x_{new} = x_{old}$ with prob $1-A$

Metropolis sampling

proposal is symmetric

$r(x|y) = r(y|x)$

$$A = \min\left(1, \frac{\tilde{p}(x_{trial})}{\tilde{p}(x_{old})}\right)$$

- generate $x_{trial} \sim r(x|x_{old})$
- if $\tilde{p}(x_{trial}) \geq \tilde{p}(x_{old})$ $x_{new} = x_{trial}$

  else $x_{new} = x_{trial}$ with prob $\frac{\tilde{p}(x_{trial})}{\tilde{p}(x_{old})}$ else $x_{new} = x_{old}$

Hamiltonian Mcmc

- particular case of Metropolis-Hastings but with good proposal distri.
- In MH, r doesn't use anything from desired distri

Hamiltonian equation

Consider simple physical setting,

$x \rightarrow$ particle

$p \rightarrow$ momentum

$p = m \frac{dx}{dt}$

$U(x) \rightarrow$ potential E

$K(p) = \frac{p^T p}{m}$ - Kinetic E

$H(x, p) = U(x) + U(p)$

↓

Hamiltonian

Hamiltonian eqs:

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial p_i} \qquad \frac{dp_i}{dt} = \frac{\partial H}{\partial x_i}$$