Week 1 19.08.2018

# Introduction to Bayesian methods and conjugate priors

## ① Think Bayesian

Principle 1: Use prior knowledge

Principle 2: Chose answer that explains observations the most.

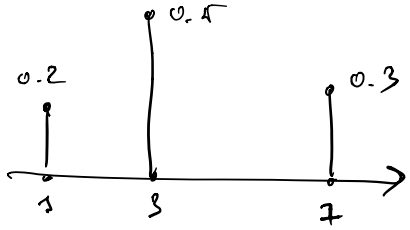
Principle 3: Avoid making extra assumptions

## ② Review of probability

random variables → discrete
→ continuous

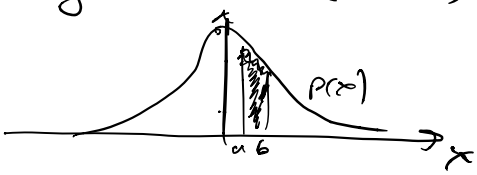
### Discrete: Probability mass function (PMF)

$$P(x) = \begin{cases} 0.2 & x=1 \\ 0.5 & x=3 \\ 0.3 & x=7 \\ 0 & \text{otherwise} \end{cases}$$




### Continuous: Probability Density function (PDF)

$$P(x \in [a,b]) = \int_a^b p(x)\,dx$$



## Independence

$X \perp Y \iff P(X, Y) = P(X)P(Y)$

joint; marginals

## Conditional probability

$$\mathbb{P}(X \mid Y) = \frac{\mathbb{P}(X, Y)}{\mathbb{P}(Y)}$$

conditional; joint; marginal

## Chain rule

$$\mathbb{P}(X, Y) = \mathbb{P}(X \mid Y)\mathbb{P}(Y)$$

$$\mathbb{P}(X, Y, Z) = \mathbb{P}(X \mid Y, Z) \cdot \mathbb{P}(Y \mid Z) \cdot \mathbb{P}(Z)$$

$$\mathbb{P}(X_1, \dots X_n) = \prod_{i=1}^{n} \mathbb{P}(X_i \mid X_1, \dots X_{i-1})$$

## Sum rule

$$p(X) = \int_{\mathbb{R}} p(X, Y)\, dy$$

## Bayes theorem

$\Theta$ - parameters  X - observations

$$\mathbb{P}(\Theta \mid X) = \frac{\mathbb{P}(X, \Theta)}{\mathbb{P}(X)} = \frac{\mathbb{P}(X \mid \Theta)\mathbb{P}(\Theta)}{\mathbb{P}(X)}$$

$\mathbb{P}(\Theta \mid X)$; $\mathbb{P}(X)$; $\mathbb{P}(X)$

posterior  evidence  likelihood  prior

## ③ Bayesian approach to statistics

| Frequentist | Bayesian |
|---|---|
| objective | subjective |
| $\theta$ is fixed | $\theta$ is random |
| $X$ is random | $X$ is fixed |
| $\lvert X\rvert \gg \lvert\Theta\rvert$ | for any $\lvert X\rvert$ |
| max likelihood | bayes theorem: |
| $\hat{\theta} = \arg\max_\theta \mathbb{P}(X\mid\theta)$ | $\mathbb{P}(\theta\mid X) = \frac{\mathbb{P}(X\mid\theta)\,\mathbb{P}(\theta)}{\mathbb{P}(X)}$ |

Classification

$$\mathbb{P}(\theta \mid X_{tr}, y_{tr}) = \frac{\mathbb{P}(y_{tr}\mid X_{tr},\theta)\;\mathbb{P}(\theta)}{\mathbb{P}(y_{tr}\mid X_{tr})}$$

training ↗

$$\mathbb{P}(y_{ts}\mid X_{ts}, X_{tr}, y_{tr}) = \int \mathbb{P}(y_{ts}\mid X_{ts},\theta)\,\mathbb{P}(\theta\mid X_{tr}, y_{tr})\,d\theta$$

prediction ↗

Regularization

$$P(\theta|X) = \frac{P(X|\theta)\,P(\theta)}{P(X)} \leftarrow \text{regularizer}$$

## On-line learning

$$P_k(\theta) = P(\theta|x_k) = \frac{P(x_k|\theta)\,P_{k-1}(\theta)}{P(x_k)}$$

$P_k(\theta)$: new prior; $P(\theta|x_k)$: posterior; $P(x_k|\theta)$: likelihood; $P_{k-1}(\theta)$: prior

④ How to define a model

## Bayesian network

Nodes: random variables
Edges: direct impact

(I)

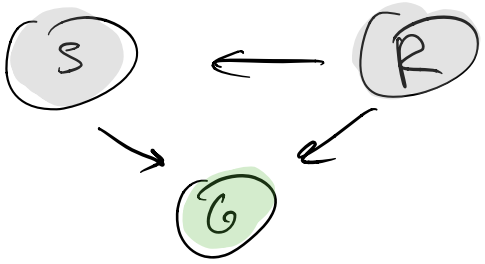

(II)

## Probabilistic model from BN

model: joint probability over all variables

$$\mathbb{P}(X_1, \dots X_n) = \prod_{k=1}^{n} \mathbb{P}(X_k \mid Pa(X_k))$$

parent

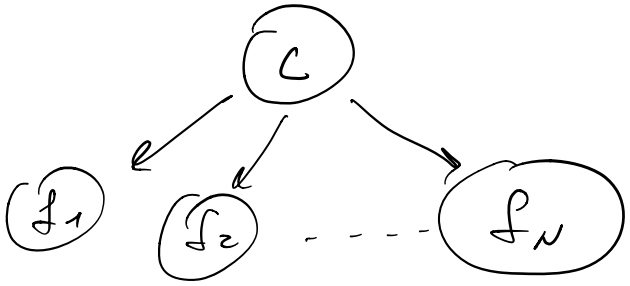



$$Pa(G) = \{R, S\}$$

$$\mathbb{P}(S, R, G) = \mathbb{P}(G \mid S, R) \cdot \mathbb{P}(S \mid R) \cdot \mathbb{P}(R)$$

probability model.

## Naive Bayes classifier




[features]

$$\mathbb{P}(C, f_1, \dots f_N) = \mathbb{P}(C) \prod_{i=1}^{N} \mathbb{P}(f_i \mid C)$$

naïve

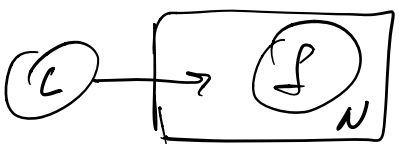
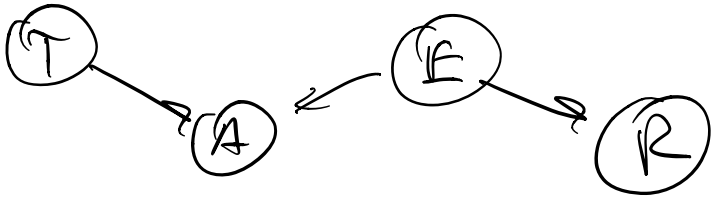



plate notation

## ⑤ Example: thief & alarm

Model:




$$P(t, a, e, r) = P(t) \cdot P(e) \cdot P(a \mid t, e) P(r \mid e)$$

Distributions

Priors

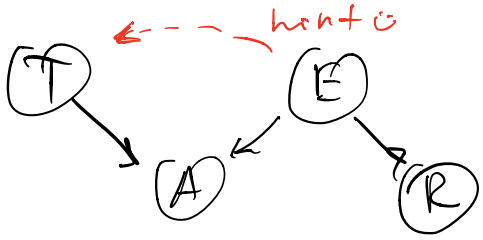
| | |
|---|---|
| $P(t=1)$ | $10^{-3}$ |
| $P(e=1)$ | $10^{-2}$ |

| $P(a=1 \mid T, e)$ | $e=0$ | $e=1$ |
|---|---|---|
| $t=0$ | 0 | 1/10 |
| $t=1$ | 1 | 1 |

| $P(R \mid E)$ | |
|---|---|
| $E=0$ | 0 |
| $E=1$ | 1/2 |

$T=1$ T

$T=0$ $\bar{T}$




| Priors | |
|---|---|
| $P(T)$ | $10^{-3}$ |
| $P(E)$ | $10^{-2}$ |

| $P(A\mid T,E)$ | $\bar{E}$ | $E$ |
|---|---|---|
| $\bar{T}$ | 0 | 1/10 |
| $T$ | 1 | 1 |

| $P(R\mid E)$ | |
|---|---|
| $E$ | 1/2 |
| $\bar{E}$ | 0 |

$$P(T\mid A)=\frac{P(T,A)}{P(A)}=\frac{P(T,A,E)+P(T,A,\bar{E})}{P(A,E)+P(A,\bar{E})}=$$

$$=\frac{P(T,A,E)+P(T,A,\bar{E})}{P(T,A,E)+P(T,A,\bar{E})+P(\bar{T},A,E)+P(\bar{T},A,\bar{E})}=$$

($P(T,A,E)\approx 2\cdot 10^{-5}$; $P(\bar{T},A,\bar{E})=0$)

$$P(T,A,E)=P(A\mid T,E)\cdot P(T,E)=$$

$$=P(A\mid T,E)\cdot P(T\mid E)\cdot P(E)=$$

$$=P(A\mid \underset{1}{T},E)\cdot \underset{10^{-3}}{P(T)}\cdot \underset{10^{-2}}{P(E)}=10^{-5}$$

$$P(\bar{T},A,\bar{E})=P(A\mid \bar{T},\bar{E})\cdot P(\bar{T},\bar{E})=$$

$$=P(A\mid \bar{T},\bar{E})\cdot P(\bar{T}\mid \bar{E})\,P(\bar{E})=P(A\mid \bar{T},\bar{E})\cdot P(\bar{T})\,P(\bar{E})=0$$

($P(\bar{T})=1-10^{-3}$, $P(\bar{E})=1-10^{-2}$)

$$P(T,A,\bar{E})=P(A\mid T,\bar{E})\cdot P(T,\bar{E})=$$

$$=P(A\mid T,\bar{E})\cdot P(T\mid \bar{E})\,P(\bar{E})=$$

$= P(A|T,\bar{E}) \cdot P(T) \cdot P(\bar{E}) = 1 \cdot 10^{-3} (1-10^{-2}) =$

$= 1 \cdot 10^{-3} - 10^{-5} = 0.001 - 0.00001 = 99 \cdot 10^{-4}$

(≈) 50%.

$$P(T|A,R) = \frac{P(T,A,R)}{P(A,R)} =$$

$$= \frac{P(T,A,R,\bar{E}) + P(T,A,R,E)}{P(A,R,\bar{T}) + P(A,R,T)} =$$

$$= \frac{P(T,A,R,\bar{E}) + P(T,A,R,E)}{P(A,R,\bar{T},E) + P(A,R,\bar{T},\bar{E}) + P(A,R,T,\bar{E}) + P(A,R,T,E)} \quad (2)$$

$P(T,A,R,\bar{E}) = P(A|R,T,\bar{E}) \cdot P(R,T,\bar{E}) =$

$= P(A|R,T,\bar{E}) \cdot P(R|T,\bar{E}) \cdot P(T,\bar{E}) =$

$= P(A|R,T,\bar{E}) \cdot P(R|\bar{T},\bar{E}) P(T|\bar{E}) P(\bar{E})$

$\qquad\qquad\qquad\qquad\qquad\qquad\quad 0 \qquad 10^{-3} \qquad (1-10^{-2})$

(≠) 1%

## ⑥ Example: linear regression

### Univariate normal

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\mathbb{E}\, x = \mu$$

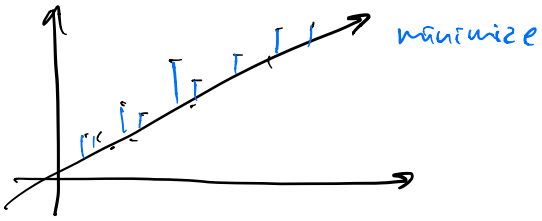
$$\mathrm{Var}[x] = \sigma^2$$

### Multivariate normal

$$\mathcal{N}(x \mid \mu, \Sigma) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right]$$

$$\mathbb{E}\, x = \mu \qquad \mathrm{Cov}[x] = \Sigma$$

### Linear regression

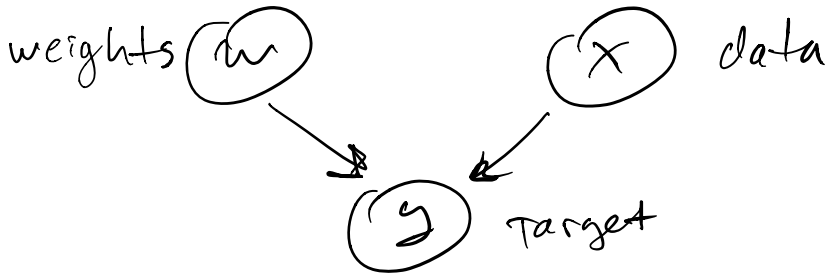



least squares problem

$$L(w) = \sum_{i=1}^{N} (w^T x_i - y_i)^2 = \|w^T X - y\|^2 \to \min_w$$

$$\hat{w} = \arg\min_w L(w)$$

### Model




$$P(w, y \mid X) = P(y \mid X, w)\, P(w)$$

$$P(y \mid w, X) = \mathcal{N}(y \mid w^T X, \sigma^2 I)$$

$$P(w) = \mathcal{N}(w \mid 0, \gamma^2 I)$$

$$P(w \mid y, X) = \frac{P(y, w \mid X)}{P(y \mid X)} \to \max_w$$

$$P(y, w \mid X) = p(y \mid X, w) \cdot p(w) \to \max_w$$

$$\log P(y, w \mid X) = \log \left[ P(y \mid X, w) \cdot p(w) \right]$$

$$\log \left[ P(y \mid X, w) \cdot p(w) \right] =$$

$$= \log P(y \mid X, w) + \log P(w) =$$

$$= \log C_1 \cdot \exp\left( -\frac{1}{2} (y - w^T X)^T \left[ \sigma^2 I \right]^{-1} (y - w^T X) \right) +$$

$$+ \log C_2 \exp\left( -\frac{1}{2} w^T \left[ \gamma^2 I \right]^{-1} w^T \right) =$$

$$= -\frac{C_1}{2C^2}(y - w^T x)^T (y - w^T x) - \frac{C_2}{2\sigma^2} w^T w =$$

$$= -\frac{1}{2C^2}\|y - w^T x\|^2 - \frac{1}{2\sigma^2}\|w\|^2 \quad \nearrow \text{max} \quad \Big| \cdot -2C^2$$

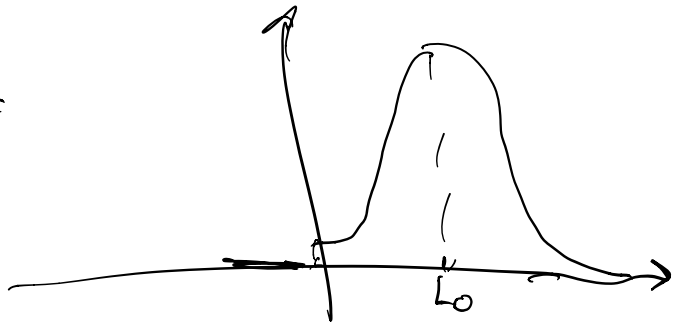
$$\min_w \quad \|y - w^T x\|^2 + \frac{C^2}{\sigma^2}\|w\|^2$$

$$\frac{C^2}{\sigma^2} = \lambda$$

$L_2$

$p(O, L, L_0, T, t) =$

$= p(O \mid L, L_0, T, t)$




$p(O|L)\ p(L|L_0, temp) \cdot p(temp|L_0, T) p(L_0) p(T)$

$p(L, L_0, Tem)$

(2) chose correct statements

$p(a|b) = \int p(a|b,c)\, dc = \ (-)$

$= \int \frac{p(a,b,c)}{p(b,c)}\, dc = \frac{p(a|b,c)\, p(b|a)\, p(c)}{}$

$p(a|b) = \frac{p(a,b)}{p(b)}$

$\int \frac{p(a,b,c)}{p(b)}\, dc$

$p(a|b) = \int p(a,c|b)\, dc$

$\frac{p(a,b)}{p(b)}$ $\int \frac{p(a,c,b)}{p(b)}\, dc$ (v)

$p(a|b)$ $\int p(a|b,c)\,p(c)\,dc$

$\frac{p(a,b)}{p(b)}$ $\int \frac{p(a,b,c)}{p(b,c)} \cdot p(c)\,dc$

$\int \frac{p(a,b,c)}{p(b)} = \int \frac{p(a,b,c)}{p(b)p(c)} p(c)\,dc$

$p(a,b|c)$ $p(a|b,c)\,p(b|c)$

$\frac{p(a,b,c)}{p(c)}$ $\frac{p(a,b,c)}{p(b,c)} \cdot \frac{p(b,c)}{p(c)}$

(3)

$p(a|b,c)$ $p(a|b)\,p(a|c)$

$\frac{p(a,b,c)}{p(b,c)\,p(a|b,c)}$ $\frac{p(a,b)}{p(b)} \frac{p(a,c)}{p(c)}$ (=)

$\int p(a,b,c)\,dc$ $\int p(a,b)\,p(a,c)\,dc$

$p(a|b) = p(a,b)\,p(a)$

1

$p(a|b,c)$ $\quad \dfrac{p(b|a,c)\cdot p(a|c)}{\int p(b|a',c)\,p(a'|c)\,da'}$

$$p(a|b,c) = \frac{p(a,b,c)}{p(b,c)} = p(b|a,c$$

$$p(a|b,c) = \frac{p(a,b,c)}{p(b,c)} = \frac{p(b|a,c)\cdot p(a|c)p(c)}{p(b,c)} =$$

$$= \frac{p(b|a,c)\,p(a|c)\,p(c)}{\int p(a',b,c)\,da'} =$$

$$= \frac{p(b|a,c)\,p(a|c)\,\cancel{p(c)}}{\int p(b|a',c)\,p(a'|c)\,\cancel{p(c)}\,da'}$$

$p(a|b)$ $\quad \dfrac{p(a,c|b)}{p(c|a,b)}$

$\dfrac{p(a,b)}{p(b)}$ $\quad \dfrac{\cancel{p(a,c,b)}}{p(b)} - \dfrac{p(a,b)}{\cancel{p(c,a,b)}}$

$p(a|b)p(b) + p(a|\bar{b})p(\bar{b})$ $\quad p(a)$

$\sum_{\forall b'} p(a|b')p(b') \; \not\# =$

$= \sum_{\forall b'} p(a,b) = p(a)$

$p(a,b) \; \epsilon p(a|b)$

$\sum_{\forall b'} p(a|b') = \sum_{\forall b'} \frac{p(a,b')}{p(b')}$

# Conjugate priors

## 1. Analytical inference

Posterior distribution

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

($P(\theta)$: prior; $P(X|\theta)$: likelihood; $P(X)$: evidence)

what is $P(X)$?

### Maximum a posteriori

$$\theta_{MP} = \arg\max_{\theta} P(\theta|X)$$

$$\theta_{MP} = \arg\max_{\theta} \frac{P(X|\theta)P(\theta)}{P(X)}$$

$$\theta_{MP} = \arg\max_{\theta} P(X|\theta)P(\theta)$$

optimization problem

### MAP: problems

Not invariant to reparametrization


$p_x \mapsto p_{f(x)}$

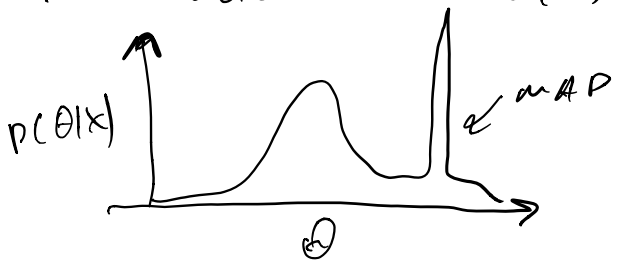
maximum has changed

Can't use as prior

$$p_k(\theta) = \frac{p(x_k|\theta)\, p_{k-1}(\theta)}{p(x_k)}$$

$$p_k(\theta) = \frac{p(x_k|\theta)\, \delta(\theta - \theta_{MP})}{p(x_k)} = \delta(\theta - \theta_{MP})$$

MAP is a solution to $L(\theta) = \mathbb{I}[\theta \neq \theta^*]$ min




Summary:

Pros

- Easy to compute

Cons:

- Not invariant to reparametrization
- Can't use as a prior
- Finds untypical point
- Can't compute credible intervals

# Conjugate distributions

$$P(\theta|X) = \frac{P(X|\theta)\ P(\theta)}{P(X)}$$

$P(\theta)$ ← our own choise

$P(X|\theta)$ — fixed by model

$P(X)$ — fixed by data

$P(\theta)$ is conjugate to $P(X|\theta)$:

$$P(\theta|X) = \frac{P(X|\theta)\ P(\theta)}{P(X)} \leftarrow A(\nu)$$

$P(\theta|X)$ ↑ $A(\nu')$

$A$ - same family of distributions

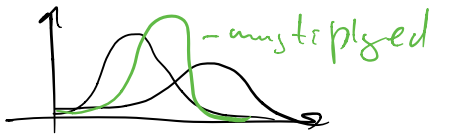
## Example

$P(X|\theta) = \mathcal{N}(X|\theta, \sigma^2)$ $\quad A(\nu)$ - ?

$$P(\theta|X) = \frac{P(X|\theta)\ P(\theta)}{P(X)} \leftarrow \mathcal{N}(X|\theta, \sigma^2)$$

Two gaussians

$P(X_1) \sim \mathcal{N}(\mu_1, \sigma_1^2)$ $\quad P(X_2) \sim \mathcal{N}(\mu_2, \sigma_2^2)$




$\mathcal{N}(X|\theta, \sigma^2)$ ↑ $\quad \leftarrow \mathcal{N}(\theta|w, s^2)$

solution: $P(\theta|X) = \frac{P(X|\theta)\ P(\theta)}{P(X)}$

$P(\theta)$ ↑ $\mathcal{N}(\theta|a, b^2)$

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} = \frac{\mathcal{N}(x|\theta,1)\,\mathcal{N}(\theta|0,1)}{p(x)}$$

$$p(\theta|x) \sim e^{-\frac{1}{2}(x-\theta)^2} \cdot e^{-\frac{1}{2}\theta^2} \sim e^{-(\theta-\frac{x}{2})^2}$$

$$p(\theta|x) = \mathcal{N}\left(\theta \,|\, \frac{x}{2}, \frac{1}{2}\right)$$

## Examples: Normal, precision

### 3. Distributions: Gamma

$$\Gamma(\gamma|a,b) = \frac{b^a}{\Gamma(a)}\gamma^{a-1} \cdot e^{-b\gamma} \qquad \gamma, a, b > 0$$

$$\mathbb{E}[\gamma] = a/b \qquad \text{Mode}[\gamma] = \frac{a-1}{b}$$

$$\text{Var}[\gamma] = a/b^2$$

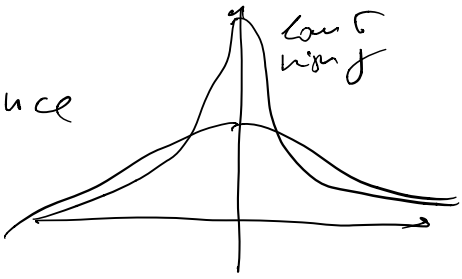
example: you run from $\pm$ 100m a day
(expectation, std)

$\frac{a}{b} = 5000 \qquad \frac{a}{b^2} = 100$

### 4. Example: Normal, precision

Precision $\gamma = \frac{1}{\sigma^2}$ — variance

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



$$\mathcal{N}(x|\mu,\lambda^{-1}) = \frac{\sqrt{\lambda}}{\sqrt{2\pi}} e^{-\lambda \frac{(x-\mu)^2}{2}}$$

$$\mathcal{N}(x|\mu,\lambda^{-1}) \sim \lambda^{\frac{1}{2}} e^{-b\lambda}$$

wrong

$$p(\lambda) \sim \lambda^{\frac{1}{2}} e^{-b\lambda}$$

$$p(\lambda|x) = \frac{p(x|\lambda)\,p(\lambda)}{p(x)} \sim \lambda\, e^{-\lambda\left(b + \frac{(x-\mu)^2}{2}\right)}$$

$$p(\lambda) \sim \lambda^{a-1} e^{-b\lambda}$$

$$\underline{p(\lambda) = \Gamma(\lambda|a,b) \sim \lambda^{a-1} b^{-b\lambda}}$$

$$p(\lambda|x) \sim p(x|\lambda)\,p(\lambda) \sim \left(\lambda^{\frac{1}{2}} e^{-\lambda \frac{(x-\mu)^2}{2}}\right) \cdot \left(\lambda^{a-1} e^{-b\lambda}\right)$$

$$p(\lambda|x) \sim \lambda^{\frac{1}{2}+a-1} e^{-\lambda\left(b + \frac{(x-\mu)^2}{2}\right)}$$

$$\underline{p(\lambda|x) = \Gamma\left(a + \tfrac{1}{2},\ b + \frac{(x-\mu)^2}{2}\right)}$$

## Example: Bernoulli

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

$x \in [0,1] \qquad a,b > 0$

$$B(a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}$$

$$\mathbb{E}[x] = \frac{a}{a+b} \qquad \text{Mode}[x] = \frac{a-1}{a+b-2}$$

$$\text{Var}[x] = \frac{ab}{(a+b)^2(a+b-1)}$$

## Beta prior

$$p(X|\theta) = \theta^{N_1}(1-\theta)^{N_0}$$

$$p(\theta) = B(\theta|a,b) \sim \theta^{a-1}(1-\theta)^{b-1}$$

$$p(\theta|X) \sim p(X|\theta)\, p(\theta)$$

$$p(\theta|X) \sim \theta^{N_1}(1-\theta)^{N_0} \cdot \theta^{a-1}(1-\theta)^{b-1}$$

$$p(\theta|X) \sim \theta^{N_1+a-1}(1-\theta)^{N_0+b-1}$$

$$p(\theta|X) = B(N_1+a, N_0+b)$$

## 7. Summary

$$\underbrace{p(\theta|X)}_{\text{(posterior)}} = \frac{p(X|\theta)\,p(\theta)}{\underbrace{p(X)}_{\text{evidence}}}$$

choose prior

pros:
- Exact posterior
- Easy for on-line learning

e.g. $p(\theta|X) = B(N_1+a, N_0+b)$

cons:
- conjugate prior may be inadequate.

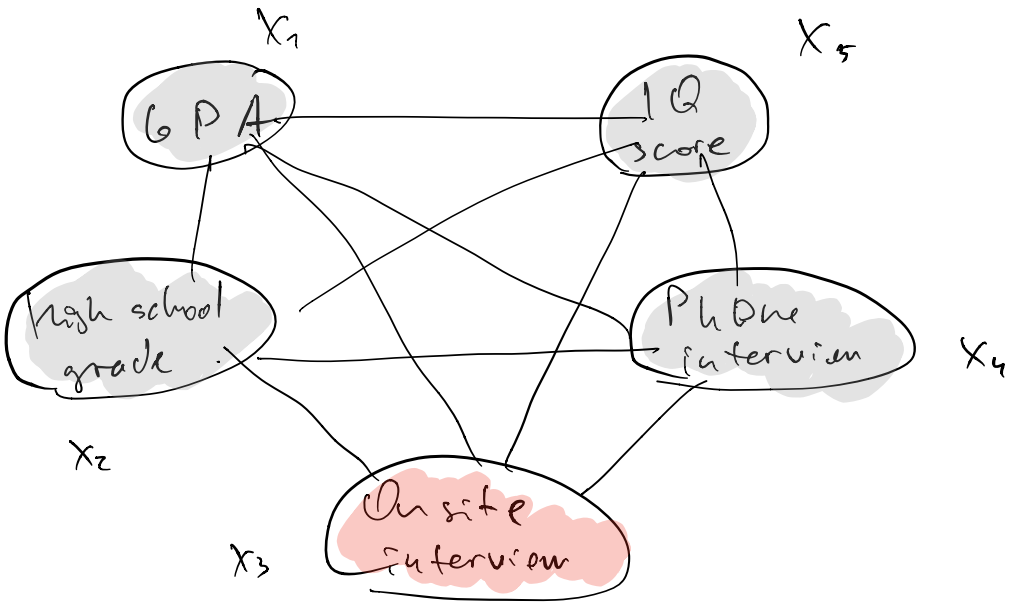
Week 2 23.08.2018

# Expectation-Maximization algorithm

## Latent Variable models

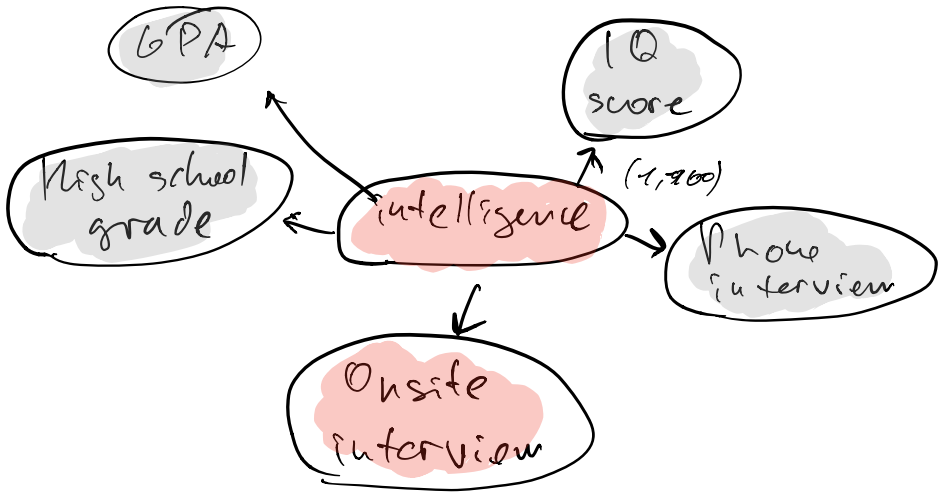
Latent (hidden) variable is a variable that you never observe

probabilistic model for hiring




$$p(x_1, x_2, x_3, x_4, x_5) = \frac{\exp(-w^T x)}{Z}$$ - impractical

let's introduce new variable: intelligence




$$p(x_1, x_2, x_3, x_4, x_5) = \sum_{I=1}^{100} p(x_1, x_2, x_3, x_4, x_5 | I)\, p(I) =$$

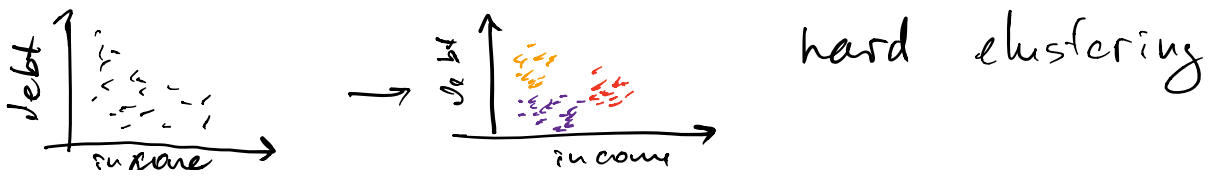
$$= \sum_{I=1}^{100} p(x_1|I) \dots p(x_5|I)\, p(I)$$

## Latent variable

pros:
- Simpler models (less edges)
- Fewer parameters
- Are sometimes meaningful

cons:
- harder to work with

## Probabilistic clustering




hard clustering

soft clustering

$p(\text{cluster idx} \mid x)$

hard clustering

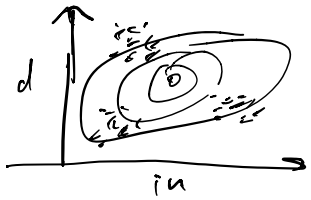
$\text{cluster idx} = f(x)$

## Gaussian Mixture Model

probabilistic model of data

$p(x \mid \theta) = \mathcal{N}(x \mid \mu, \Sigma)$

$\theta = \{\mu, \Sigma\}$




$p(x \mid \theta) = \pi_1 \cdot \mathcal{N}(x \mid \mu_1, \Sigma_1) + \pi_2 \cdot \mathcal{N}(x \mid \mu_2, \Sigma_2) +$

$+ \pi_3 \cdot \mathcal{N}(x \mid \mu_3, \Sigma_3) \quad , \sum \pi_i = 1$

$\theta = \{\pi_1, \pi_2, \pi_3, \mu_1, \mu_2, \mu_3, \Sigma_1, \Sigma_2, \Sigma_3\}$

## Training GMM

$\max_{\theta} p(X \mid \theta) = \prod_{i=1}^{N} p(x_i \mid \theta) =$ ( dataset has N points )

$= \prod_{i=1}^{N} (\pi_1 \cdot \mathcal{N}(x_i \mid \mu_1, \Sigma_1) + \dots)$

subject to $\sum \pi_i = 1 \; , \forall k \; \pi_k \geq 0$

$\Sigma_k \succ 0$; ( SGD can't solve problem with this constrain)

## Summary

- GMM is a flexible probability distribution
- It is hard to fit (train) with SGD

## Introducing latent variable

$$p(x|\theta) = \pi_1 \cdot \mathcal{N}(x|\mu_1, \Sigma_1) + \pi_2 \cdot \mathcal{N}(x|\mu_2, \Sigma_2) + \\ + \pi_3 \cdot \mathcal{N}(x|\mu_3, \Sigma_3)$$


$$p(t=c|\theta) = \pi_c$$

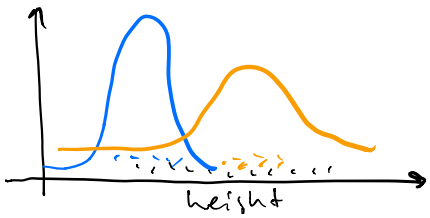
$$p(x|t=c,\theta) = \mathcal{N}(x|\mu_c, \Sigma_c)$$

$$p(x|\theta) = \sum_{c=1}^{3} p(x|t=c,\theta) \cdot p(t=c|\theta)$$

$$p(x|\theta) = \int p(x|t,\theta)\, p(t|\theta)\, dt$$

## Expectation Maximization

Dataset: $\{x_1, \dots x_N\}$



How to estimate parameter $\theta$?

If sources $t$ are known, easy:

$$p(x \mid t=1, \theta) = \mathcal{N}(x \mid \mu_1, \sigma_1^2)$$

$$\mu_1 = \frac{\sum_{\text{blue } i} x_i}{\#\text{ of blue points}} \qquad \sigma_1^2 = \frac{\sum_{\text{blue } i} (x_i - \mu_1)^2}{\#\text{ of blue points}}$$

if we know only: $t_i : p(t_i \mid x_i, \theta)$

$$\mu_1 = \frac{\sum_i p(t_i=1 \mid x_i, \theta)\, x_i}{\sum_i p(t_i=1 \mid x_i, \theta)}$$

$$\sigma_1^2 = \frac{\sum_i p(t_i=1 \mid x_i, \theta)\, (x_i - \mu_1)^2}{\sum_i p(t_i=1 \mid x_i, \theta)}$$

But, we don't know the sources

Given: $p(x \mid t=1, \theta) = \mathcal{N}(-2, 1)$

Find: $p(t=1 \mid x, \theta)$

$$p(t=1 \mid x, \theta) = \frac{p(x \mid t=1, \theta)\, p(t=1 \mid \theta)}{Z}$$

## Chicken and egg problem

- Need Gaussian parameters to estimate sources
- Need sources to estimate Gaussian parameters

## EM algorithm

1. Start with 2 randomly placed Gaussians parameters $\theta$
2. Until convergence repeat:
   a) For each point compute $p(t=c \mid x_i, \theta)$: does $x_i$ look like it came from cluster $c$?
   b) Update Gaussian parameters $\theta$ to fit points assigned to them.

## Example of GMM training

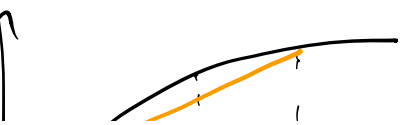
....

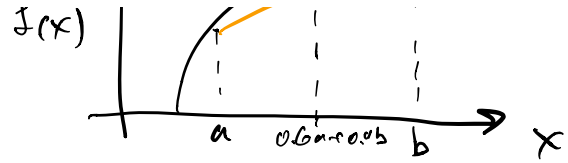
## Expectation Maximization algorithm

## Jensen's inequality & Kullback-Leibler divergence

## General form of EM

## concave functions






def: $f(x)$ is concave if

$\forall\ a, b, \alpha: \quad f(\alpha a + (1-\alpha) b) \geq \alpha f(a) + (1-\alpha) f(b)$

$0 \leq \alpha \leq 1$

## Jensen's inequality

$$f(\alpha a + (1-\alpha) b) \geq \alpha f(a) + (1-\alpha) f(b)$$

$0 \leq \alpha \leq 1$

$$f(\alpha_1 \cdot a_1 + \alpha_2 \cdot a_2 + \alpha_3 \cdot a_3) \geq \alpha_1 \cdot f(a_1) + \alpha_2 f(a_2) + \alpha_3 f(a_3)$$

$\alpha_1 + \alpha_2 + \alpha_3 = 1 \quad \alpha_k \geq 0$

Let's call $\alpha_k$ - probability

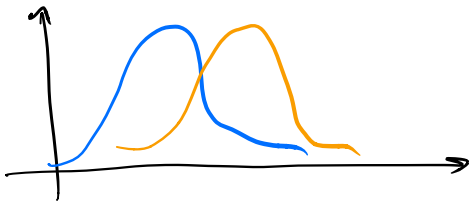
$p(t = a_1) = \alpha_1$

$p(t = a_2) = \alpha_2$

$p(t = a_3) = \alpha_3$

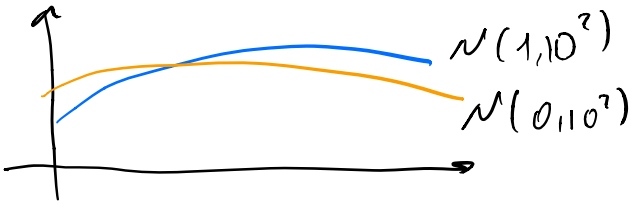
$$f(\mathbb{E}_{p(t)}\, t) \geq \mathbb{E}_{p(t)}\, f(t) \quad : f \text{ - concave}$$

## Kull back - Leibler divergence



$\mathcal{N}(0,1)$

$\mathcal{N}(1,1) \quad KL(q_1 \| p_1) = 0.5$




$KL(q_2 \| p_2) = 0.005$

$$KL(q_\theta \| p) = \int q_\theta(x) \log \frac{q_\theta(x)}{p(x)} dx$$

$$KL(q_\theta \| p) = \mathbb{E}_{q_\theta} \log \frac{q_\theta(x)}{p(x)}$$

1. $KL(q_\theta \| p) \neq KL(p \| q_\theta)$
2. $KL(q_\theta \| q_\theta) = 0$
3. $KL(q_\theta \| p) \geq 0$

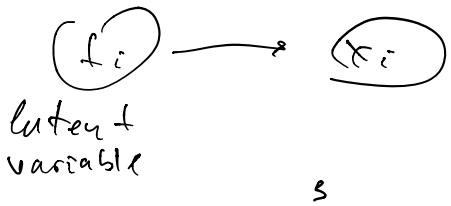
proof: $-KL(q_\theta \| p) = -\mathbb{E}_{q_\theta} \left[ \log \frac{q_\theta}{p} \right] =$

$$= \mathbb{E}_{q_\theta} \left( -\log \frac{q_\theta}{p} \right) = \mathbb{E}_{q_\theta} \left( \log \frac{p}{q_\theta} \right) \leq$$

$$\leq \log \mathbb{E}_{q_\theta} \left( \frac{p}{q_\theta} \right) = \log \int q_\theta(x) \frac{p(x)}{q_\theta(x)} dx = \log 1 = 0$$

## Expectation-Maximization algorithm

### General form EM




$$p(x_i \mid \theta) = \sum_{c=1}^{S} p(x_i \mid t_i = c, \theta)\, p(t_i = c \mid \theta)$$

$$\max_\theta p(X|\theta)$$

$$\max_\theta \log p(X|\theta) = \log \prod_{i=1}^{N} p(x_i|\theta) =$$

(assume independence)

$$= \sum_{i=1}^{N} \log p(x_i|\theta)$$

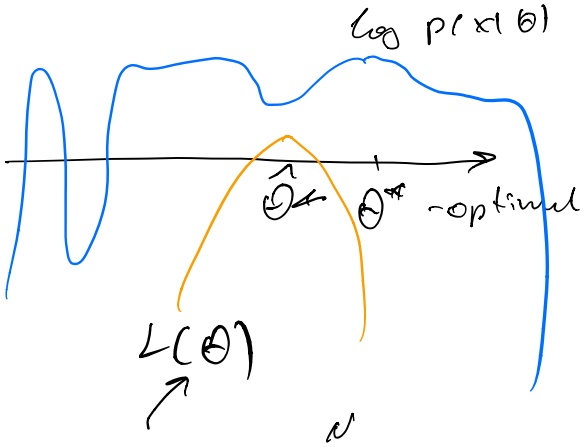
$$\log p(X|\theta) = \sum_{i=1}^{N} \log p(x_i|\theta) =$$

$$= \sum_{i=1}^{N} \log \left( \sum_{c=1}^{3} p(x_i, t_i = c|\theta) \right)$$

$\geq$ Jensen's inequality

lower bound $\mathcal{L}(\theta)$

example:



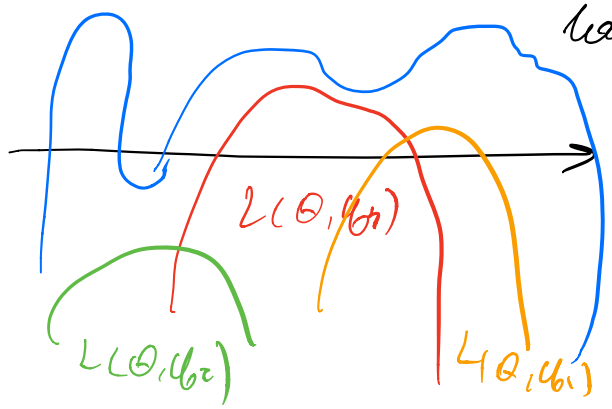

$$\log p(x|\theta) = \sum_{i=1}^{N} \log p(x_i|\theta) = \sum_{i=1}^{N} \log \sum_{t_i=1}^{3} p(x_i, t_i = c|\theta) \frac{q(t_i=c)}{q(t_i=c)}$$

$$= \sum_{i=1}^{N} \log \sum_{c=1}^{3} q(t_i=c) \frac{p(x_i, t_i=c|\theta)}{q(t_i=c)} \geq$$

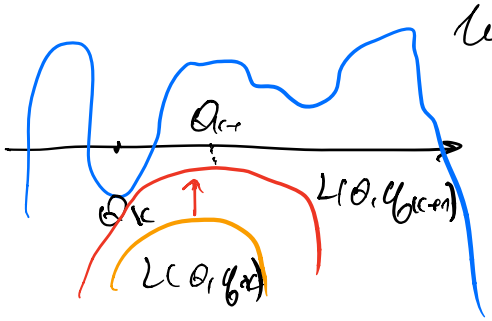
$$\log\left(\sum_c a_c v_c\right) \geq \sum_c a_c \log(v_c)$$

$$\geq \sum_{i=1}^{N} \sum_{c=1}^{3} q(t_i = c) \log \frac{p(x_i, t_i = c \mid \theta)}{q(t_i = c)}$$

$$\log p(x \mid \theta) \geq L(\theta, q) \text{ for any } q$$




family of lower bounds




$$q^{(t+1)} = \arg\max_q L(\theta^{(t)}, q)$$

## Summary of Expectation Maximization

$$\log P(x \mid \theta) \geq L(\theta, q) \text{ for any } q$$

└ variational lower bound

## E-step

$$q^{(t+1)} = \arg\max_{q} L(\theta^{(t)}, q)$$

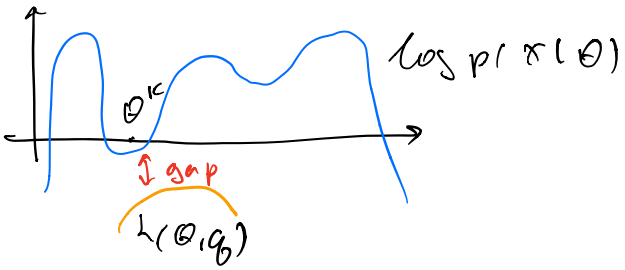
## M-step

$$\theta^{(t+1)} = \arg\max_{\theta} L(\theta, q^{(t+1)})$$

## E-step details

$$\log P(X|\theta) \geq L(\theta, q)$$

E-step: $\max_{q} L(\theta^{(t)}, q)$




$$GAP = \log P(X|\theta) - L(\theta, q) =$$

$$= \sum_{i=1}^{N} \log p(x_i|\theta) - \sum_{i=1}^{N} \sum_{c=1}^{3} q(t_i=c) \log \frac{p(x_i, t_i=c|\theta)}{q(t_i=c)} =$$

$$= \sum_{i=1}^{N} \left( \log p(x|\theta) \cdot \sum_{c=1}^{3} q(t_i=c) - \sum_{c=1}^{3} q(t_i=c) \log \frac{p(x_i, t_i=c|\theta)}{q(t_i=c)} \right) =$$

$$= \sum_{i=1}^{N} \sum_{c=1}^{3} q(t_i=c) \left( \log p(x_i|\theta) - \log \frac{p(x_i, t_i=c|\theta)}{q(t_i=c)} \right) =$$

$$= \sum_{i=1}^{N} \sum_{c=1}^{3} q(t_i=c) \left[ \log \frac{p(x_i|\theta)\, q(t_i=c)}{p(x_i, t_i=c|\theta)} \right] =$$

$$= \sum_{i=1}^{N} \sum_{c=1}^{3} q(t_i=c) \left[ \log \frac{q(t_i=c)}{p(t_i=c|x_i,\theta)} \right]$$

$$KL(q(t_i) \,||\, p(t_i=c|x_i,\theta))$$

$$GAP = \sum_{i=1}^{N} KL(q(t_i) \,||\, p(t_i=c|x_i,\theta)) \geq 0$$

↘ min $q$

## M-step details

$$L(\theta, q) = \sum_i \sum_c q(t_i=c) \log \frac{p(x_i, t_i=c|\theta)}{q(t_i=c)} =$$

$$= \sum_i \sum_c q(t_i=c) \log p(x_i, t_i=c|\theta) -$$

$$- \sum_i \sum_c q(t_i=c) \log q(t_i=c) \Leftarrow \text{const w.r.t } \theta$$

$$\Leftarrow \mathbb{E}_q \log p(X, T|\theta) + \text{const}$$

(Usually) concave function w.r.t $\Theta$, easy to optimize

## Expectation - Maximization algorithm

for $k = 1, \dots$

E. step

$$q^{k+1} = \arg\min KL[q(T) \| p(T | X, \Theta^k)]$$

$$\Leftrightarrow$$

$$q^{k+1}(t_i) = p(t_i | x_i, \Theta^k)$$

M - step

$$\Theta^{k+1} = \arg\max_{\Theta} \mathbb{E}_{q^{k+1}} \log p(X, T | \Theta)$$

## Convergency guaranties

$$\log p(X | \Theta^{k+1}) \geq L(\Theta^{k+1}, q^{k+1}) \geq$$

$$\geq L(\Theta^k, q^{k+1}) = \log p(X | \Theta^k)$$

$$\Downarrow$$

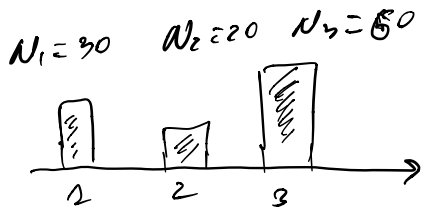
$$\log p(X | \Theta^{k+1}) \geq \log p(X | \Theta^k)$$

- One each iteration EM doesn't decrease the objective
- Guaranteed to converge to a local

maximum (or saddle point)

# Example: EM for discrete mixture.

## E-Step




EM example

you want to fit distribution to the data

$$p(x_i) = \lambda \cdot p_1(x_i) + (1-\lambda)\, p_2(x_i)$$

$p(x_i|\theta)$ everyone means

| | 1 | 2 | 3 |
|---|---|---|---|
| | $p_1$ | | |
| $t_i=1$ | $\alpha$ | $1-\alpha$ | 0 |
| $t_i=2$ | $p_2$ 0 | $1-\beta$ | $\beta$ |

find $\lambda, \alpha, \beta$ with EM

$\alpha_0 = \beta_0 = \lambda_0 = 0.5$


$p(t_i=1) = \lambda$

$p(t_i=2) = 1-\lambda$

$p(x_i | t_i=2) = p_2(x_i)$

## E step

$$\gamma(t_i=c) = p(t_i=c|x_i)$$

$$p(t_i=1 \mid x_i=1) = \frac{p(x_i=1 \mid t_i=1)\, p(t_i=1)}{p(x_i=1 \mid t_i=1)\, p(t_i=1) + p(x_i=1 \mid t_i=2)\, p(t_i=2)} =$$

$$= \frac{\alpha \cdot \beta}{\alpha \cdot \beta + \delta \cdot (1-\beta)} = 1$$

$$p(t_i=1 \mid x_i=3) = \frac{p(x_i=3 \mid t_i=1)\, p(t_i=1)}{p(x_i=3 \mid t_i=1)\, p(t_i=1) + p(x_i=3 \mid t_i=2)\, p(t_i=2)} =$$

$$= \frac{0}{\dots} = 0$$

$$p(t_i=1 \mid x_i=2) = \frac{(1-\beta)\beta}{(1-\alpha)\beta + (1-\beta)(1-\alpha)} = 0.5$$

## M-step

E-step result

$$q_\theta(t_i=1) = p(t_i=1 \mid x_i) = \begin{cases} 1, & x_i=1 \\ 0.5, & x_i=2 \\ 0, & x_i=3 \end{cases}$$

$$q_\theta(t_i=2) = 1 - q_\theta(t_i=1)$$

$$\max_{\alpha,\beta,\gamma} \sum_{i=1}^{N} \mathbb{E}_{q(t_i)} \log p(x_i \mid t_i)\, p(t_i) =$$

$$= \sum_{i=1}^{N} \mathbb{1}(t_i=1) \cdot \log p(x_i|t_i=1)\, p(t_i=1)^{\gamma} +$$

$$+ \sum_{i=1}^{N} \mathbb{1}(t_i=2) \cdot \log p(x_i|t_i=2)\, p(t_i=2)^{(1-\gamma)} =$$

$$= 30 \cdot 1 \cdot \underbrace{p(t_i=1|x_i=1)}_{1} \log \alpha \cdot \gamma +$$

$$+ 20 \cdot 0.5 \cdot \log(1-\alpha) \cdot \gamma +$$

$$+ 60 \cdot 0 \cdot \log(0) \cdot \gamma +$$

} first component

$$+ 30 \cdot p(t_i=2|x_i=1) \cdot \ldots \quad (=0) +$$

$$+ 20 \cdot 0.5 \cdot \log(1-\beta) \cdot (1-\gamma) +$$

$$+ 60 \cdot 1 \cdot \log(\beta) \cdot (1-\gamma)$$

} second component

$$=$$

$$= 30 \cdot \log \alpha \cdot \gamma + 10 \cdot \log(1-\alpha) \cdot \gamma + \text{const}(\alpha)$$

$$\nabla = 30 \frac{1}{\alpha} \gamma - 10 \frac{1}{1-\alpha} \gamma = 0$$

$$\frac{30}{\alpha} = \frac{10}{1-\alpha} \Rightarrow 30 - 30\alpha = 10\alpha \Rightarrow \alpha = \frac{30}{40}$$

$$\beta = \frac{6}{7} \quad \gamma = \frac{4}{11}$$

## Summary of Expectation Maximization

- Method for training Latent Variable Models


- Handles missing data
- Sequence of simple task instead of one hard
- Guarantees to converge
- Helps with complicated parameter constrains

$$\Sigma_c > 0$$

- Numerous extensions

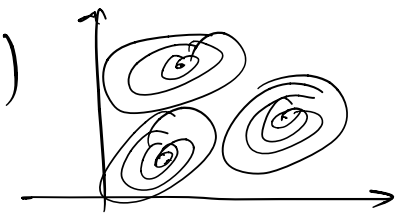
## Applications and examples

## General EM for GMM

GMM.

$$p(x|\theta) = \sum_{i=1}^{3} \pi_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

$$\theta = \{\pi_i, \mu_i, \Sigma_i\}$$



E-Step

EM: for each point compute
$q(t_i) = p(t_i | x_i, \theta)$

GMM: for each point compute
$p(t_o | x_i, \theta)$

M-Step

EM: Update parameters to maximize
$\max_\theta \mathbb{E}_q \log p(X, T | \theta)$

GMM: Update Gaussian parameters to fit points assigned to them

$$\mu_1 = \frac{\sum_i p(t_i = 1 | x_i, \theta)\, x_i}{\sum_i p(t_i = 1 | x_i, \theta)}$$

proof:

M-step for GMM

$$\max_\theta \mathbb{E}_q \log p(X, T | \theta) =$$

$$= \max_\theta \sum_{i=1}^{N} \mathbb{E}_{q(t_i)} \log p(x_i, t_i | \theta)$$

$$\sum_{i=1}^{N} \sum_{c=1}^{3} q(t_i = c) \log \left[ p(x_i | t_i, \theta)\, p(t_i, \theta) \right] =$$

$$= \sum_{i=1}^{N} \sum_{c=1}^{3} q(t_i=c) \log \left[ \frac{1}{z} \exp\left(-\frac{(x_i-\mu_c)^2}{2\sigma_c^2}\right) \cdot \pi_c \right] =$$

$$= \sum_{i=1}^{N} \sum_{c=1}^{3} q(t_i=c) \left[ \log \frac{\pi_c}{z} - \frac{(x_i-\mu_c)^2}{2\sigma_c^2} \right]$$

$$\frac{\partial}{\partial \mu_2} = \sum_{i=1}^{N} q(t_i=1) \cdot \frac{(x_i-\mu_1)}{\sigma_1^2} = 0 \quad \Big| \sigma_1^2$$

$$\sum_{i=1}^{N} q(t_i=1)(x_i-\mu_1) = 0$$

$$\sum q(t_i=1)\, x_i - \sum_{i=1}^{N} q(t_i=1)\, \mu_1 = 0$$

$$\mu_1 = \frac{\sum q(t_i=1)\, x_i}{\sum q(t_i=1)}$$

same for $\sigma_c^2$

$$\sigma_c^2 = \frac{\sum_i (x_i-\mu_c)^2 \cdot q(t_i=c)}{\sum_i q(t_i=c)}$$

$\pi_c \geq 0$
$\sum \pi_c = 1$ same with constrain

$$\pi_c = \frac{\sum_{i=1}^{N} \mathbb{1}(t_i = c)}{N}$$

## K-means from probabilistic perspective

1. Randomly initialize parameters $\theta = \{\mu_1, \ldots \mu_K\}$
2. Until convergance repeat:
   a) For each point compute closest centroid
   $$c_i = \arg\min_c \|x_i - \mu_c\|^2$$
   b) Update centroids
   $$\mu_c = \frac{\sum_{i: c_i = c} x_i}{\#\{i: c_i = c\}}$$

From GMM to K-means

- Fix covariances to be identical $\Sigma_c = I$
- Fix weight to be uniform
$$\pi_c = \frac{1}{\#\text{ of Gaussian}}$$

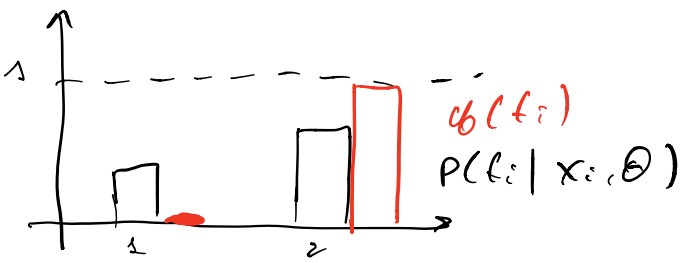
$$p(x_i \mid t_i = c, \theta) = \frac{1}{Z} \exp\left(-0.5 \|x_i - \mu_c\|^2\right)$$

E-Step:

$$q^{t+1} = \arg\min_{q \in Q} KL\left[q(T) \| p(T \mid X, \theta^t)\right]$$

where $Q$ is the set of delta-functions

example:




$$KL(q \| p) = \mathbb{E}_q \log \frac{q}{p} =$$

$$= 0 \cdots + 1 \cdot \log \frac{1}{0.6} = 0.52$$

E-step:

$$q^{(t+1)}(t_i) = \begin{cases} 1, & \text{if } t_i = c_i \\ 0, & \text{otherwise} \end{cases}$$

$$c_i = \arg\max_c p(t_i = c \mid x_i, \theta)$$

$$p(t_i \mid x_i, \theta) = \frac{1}{Z} p(x_i \mid t_i, \theta)\, p(t_i \mid \theta) =$$

$$= \frac{1}{Z} \exp(-0.5 \|x_i - \mu_c\|^2)\, \pi_c$$

$$c_i = \arg\max_c p(t_i = c \mid x_i, \theta) = \arg\min_c \|x_i - \mu_c\|^2$$

K-means M-Step

$$\max_\theta \mathbb{E}_q \log(X, T | \theta) =$$

$$= \max_\theta \underbrace{\sum_{i=1}^N \mathbb{E}_{q(t_i)} \log p(x_i, t_i | \theta)}$$

$$= \sum_{i=1}^N \sum_{c=1}^3 q(t_i = c) \log p(x_i | t_i, \theta)\ p(t_i | \theta) =$$

$$\mu_c = \frac{\sum_{i=1}^N q(t_i = c) \cdot x_i}{\sum_{i=1}^N q(t_i = c} = \frac{\sum_{i: c_i = c} x_i}{\sum_{i: c_i = c} 1}$$

$$q(t_i) = \begin{cases} 1, t_i = c_i^* \\ 0, t_i \neq c_i^* \end{cases}$$

$$\theta^{k+1} = \arg\max_\theta \mathbb{E}_{q^{t+2}} \log p(X, T | \theta)$$

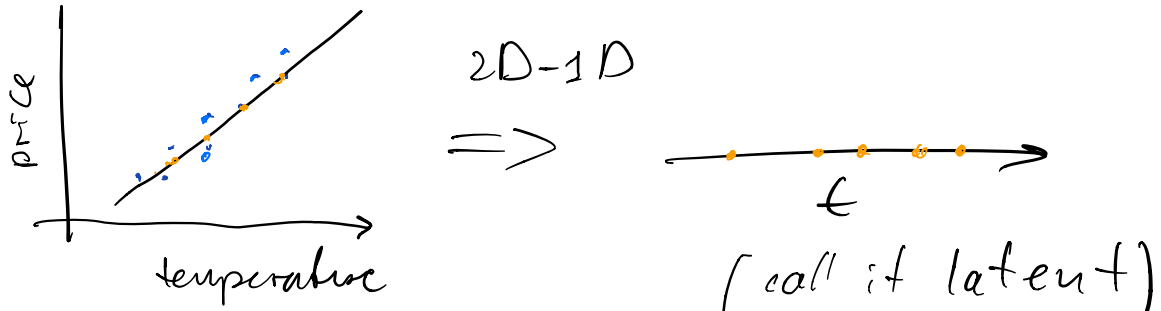
$$\mu_c^{k+1} = \frac{\sum_{i: c_i = c} x_i}{\sum_{i: c_i = c} 1}$$

Summary:

K-means is actualy EM for GMM but

- with fixed covariance matrices: $\Sigma_c = I$
- Simplified E-step (approximate $p(t_i | x_i, \theta)$ with delta function)

## Probabilistic PCA




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

prior

$x_i = (1,1) \cdot t_i + (2\sigma, \mu)$

$x_i = W t_i + b$

$x_i = W t_i + b + \varepsilon_i \quad , \varepsilon \sim \mathcal{N}(0, \Sigma)$

$t_i \longrightarrow x_i$

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

$p(x_i | t_i, \theta) = \mathcal{N}(W t_i + b, \Sigma)$

$$\max_{\theta} p(X|\theta) = \prod_{i=1}^{N} p(x_i|\theta) =$$

$$= \prod_i \underbrace{\int p(x_i|t_i,\theta)\, p(t_i)\, dt_i}_{\text{conjugacy} \quad \mathcal{N}(\mu_i, \Sigma_i)}$$

## EM for Probabilistic PCA

E-step

$$q_\theta(t_i) = p(t_i|x_i,\theta) = \frac{p(x_i|t_i,\theta)\, p(t_i)}{Z} =$$

$$= \mathcal{N}(\hat{\mu}_i, \hat{\Sigma}_i)$$

M-step

$$\max_{\theta} \mathbb{E}_{q(T)} \sum_i \log p(x_i|t_i,\theta)\; p(t_i) =$$

$$= \sum_i \mathbb{E}_{q(t_i)} \log\left(\frac{1}{Z}\exp(\cdot)\exp(\cdot)\right) =$$

$$= \sum_i \log\frac{1}{Z} + \sum_i \mathbb{E}_{q(t_i)} \log(\exp(\cdot)\exp(\cdot))$$

//

$$\sum_i \mathbb{E}_{q(f_i)} \underbrace{\log\left[\exp\left(-\frac{(x-w_i-b)^2}{2C^2}\right)\exp\left(-\frac{f_i^2}{2}\right)\right]}_{af_i^2+cf_i+d}$$

## Probabilistic formulation of PCA

- Allows for missing data
- Straightforward iterative scheme for large dimensionalities
- Can do mixture of PPCA
- Hyperparameter tuning (number of components or choose between diagonal and full covariance)

# Week 3

## Variational Inference & Latent Dirichlet Allocation

### Variational inference

### Why approximate inference?

$$p^*(z) = p(z|X) = \frac{p(X|z)\,p(z)}{p(X)}$$

- Easy for conjugate priors
- Hard otherwise

Example: $p(x|z) = \mathcal{N}(x \mid \mu(z), \sigma^2(z))$ [VAE]

$\mu(z)$, $\sigma^2(z)$: neural networks

VI

1. Select a family of distributions $Q$

   example: $\mathcal{N}(\mu, \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_n^2 \end{pmatrix})$

2. Find best approximation $q(z)$ of $p^*(z)$

$$KL[q(z) \,||\, p^*(z)] \to \min_{q \in Q}$$

#### choice of variational family

## Unnormalized distribution

$$p^*(z) = p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{\hat{p}(z)}{Z}$$

optimization

$$KL\left[q_\theta(z) \,\|\, \frac{\hat{p}(z)}{Z}\right] = \int q_\theta(z) \log \frac{q_\theta(z)}{\hat{p}(z)/Z} dz =$$

$$= \int q_\theta(z) \log \frac{q_\theta(z)}{\hat{p}(z)} dz + \int q_\theta(z) \log Z dz =$$

$$= KL\{q_\theta(z) \,\|\, \hat{p}(z)\} + \log Z$$

$$KL\{q_\theta(z) \,\|\, \hat{p}(z)\} \to \max_{q \in Q}$$

## Mean field approximation

1. Select a family of distributions $Q$

$$Q = \{q \mid q(z) = \prod_{i=1}^{d} q_i(z_i)\}$$ [distributions can be factorized]

2. find best approximation $q(z)$ of $p^*(z)$

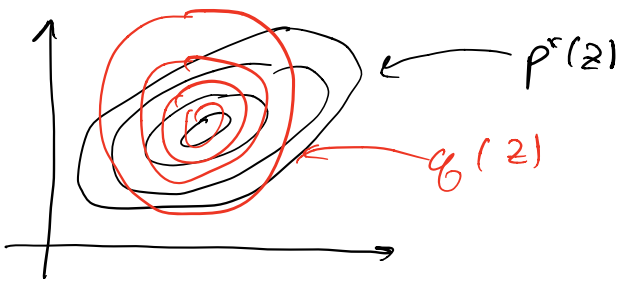
$$KL\{q_\theta(z) \,\|\, p^*(z)\} \to \min_{q \in Q}$$

Example:

$$p^*(z_1, z_2) \approx q_{\theta 1}(z_1)\, q_{\theta 2}(z_2)$$

$$p^*(z_1, z_2) = \mathcal{N}(0, \Sigma)$$

$$q_{\theta 1}(z_1)\, q_{\theta 2}(z_2) = \mathcal{N}\left(0, \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}\right)$$




Optimization

$$KL(q_\theta \| p^*) = KL\left(\prod_{i=1}^{d} q_i \| p^*\right) \to \min_{q_1, q_2 .. q_d}$$

coordinate descend:

1. $KL(q_\theta \| p^*) \to \min_{q_1}$
2. $KL(q_\theta \| p^*) \to \min_{q_2}$
3. $\vdots$

one step:

$$KL(q_\theta \| p^*) \to \min_{q_{1c}}$$

$$KL(q_\theta \| p^*) = KL\left[\prod_{i=1}^{d} q_i \| p^*\right] =$$

$$= \int \prod_{i=1}^{d} q_i(z_i) \log \frac{\prod_{i=1}^{d} q_i(z_i)}{p^*(z)} dz =$$

$$= \sum_{i=1}^{d} \int \prod_{j=1}^{d} q_j \cdot \log q_i \, dz - \int \prod_{j=1}^{d} q_j \cdot \log p^* \, dz =$$

$$= \int \prod_{j=1}^{d} q_j \log q_k \, dz + \sum_{i \neq k} \int \prod_{j=1}^{d} q_j \log q_i \, dz -$$

$$- \int \prod_{j=1}^{d} q_j \log p^* \, dz \quad (=)$$

const w.r.t. $k$

$$\int q_k \log q_k \underbrace{\left[ \int \prod_{j \neq k} q_j \, dz_{\neq k} \right]}_{1} dz_k =$$

$$= \int q_k \log q_k \, dz_k$$

$$\sum_{i \neq k} \int q_i \log q_i \, dz_i$$

$$(=) \int q_k \log q_k \, dz_k - \int q_k \left[ \int \prod_{j \neq k} q_j \log p^* \, dz_{\neq k} \right] dz_k = \quad + \text{const}$$

$$= \int q_k \left[ \log q_k - \underbrace{\int \prod_{j \neq k} q_j \log p^* \, dz_{\neq k}}_{h(z_k) = \mathbb{E}_{q_{\neq k}} \log p^*} \right] dz_k + \text{const} =$$

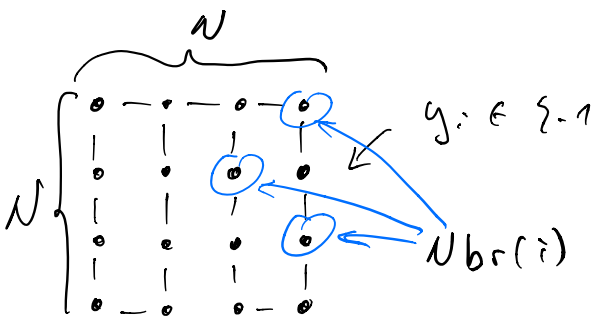
$$t(z_k) \quad \frac{e^{h(z_k)}}{\int e^{h(z_k)} \, dz_k} \quad \text{- our new distributions}$$

$$= \int q_k \log \frac{q_k}{t} \, dz_k + \text{const} \to \min$$

$$KL\left[q_k \| t\right] \to \min$$

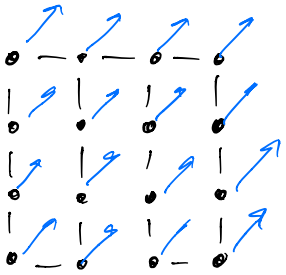
$$q_k = t \Rightarrow \boxed{\log q_k = \mathbb{E}_{q_{-k}} \log p^* + \text{const}}$$

## Example: Ising model



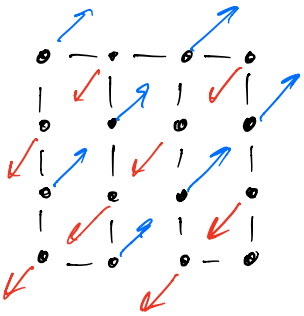

$$p(y) \propto \exp\left(\frac{1}{2} J \cdot \sum_i \sum_{j \in Nbr(i)} y_i y_j + \sum_i b_i y_i\right)$$

$$\phi(y)$$



Ferromagnetic

$J > 0$



Antiferromagnetic

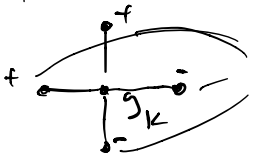
$J > 0$

normalization constant

$$p(y) = \frac{1}{Z}\phi(y) \qquad Z = \sum_y \phi(y) \qquad 2^{n^2} \text{ terms}$$

$$p(y) \approx q(y) = \prod_i q_i(y_i)$$

$$p(y) \propto \exp\left(\frac{1}{2} \cdot J \cdot \sum_i \sum_{j \in Nbr(y_i)} y_i y_j + \sum_i b_i y_i\right)$$




$Nbr(y_k)$

$q_k(y_k) - ?$

$$\log q_k = \mathbb{E}_{q_{-k}} \log p + const =$$

$$= \mathbb{E}_{q_{-k}} \left[\frac{1}{2} \cdot J \cdot \sum_i \sum_{j \in Nbr(y_i)} y_i y_j + \sum_i b_i y_i\right] + const =$$

$$= \mathbb{E}_{q_{-k}} \left[J \cdot \sum_{j \in Nbr(y_k)} y_k y_j + b_k y_k\right] + const =$$

$$= J \cdot \sum_{j \in Nbr(y_k)} y_k \underbrace{\mathbb{E} y_j}_{\mu_j} + b_k y_k + const =$$

$$= y_k \cdot \underbrace{\left(J \cdot \sum_{j \in Nbr(y_k)} \mu_j + b_k\right)}_{M} + const$$

$$q_k = C \cdot \exp(y_k \cdot M)$$

$$q_k(1) + q_k(-1) = 1$$

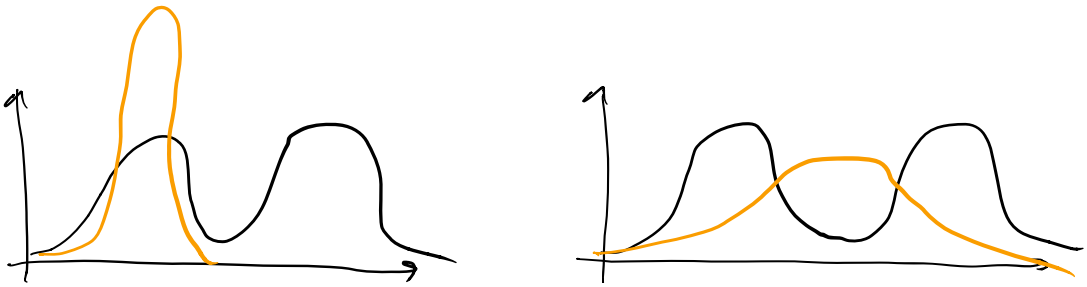
$$C \cdot e^M + C e^{-M} = 1 \qquad C = \frac{1}{e^M + e^{-M}}$$

$$q_{\mathbb{R}}(+1) = \frac{e^{m}}{e^{m}+e^{-m}} = \frac{1}{1+e^{-2m}} = \sigma(2m)$$

$$\mu_{\mathbb{R}} = 1 \cdot q(+1) + (-1)\, q(-1) = q(+1) - q(-1) =$$

$$= \frac{e^{m} - e^{-m}}{e^{m}+e^{-m}} = \tanh(m)$$

## Optimization solutions



| | | |
|---|---|---|
| x | capture statistics | ✓ |
| ✓ | mode has high probability | x |

$$KL[q \| p^{*}] = \int q(z) \log \frac{q(z)}{p^{*}(z)} dz = +\infty$$

($p^{*}(z) = 0$, $q(z) \neq 0$)

## Variational EM & Review

$$\log p(x|\theta) \geq L(\theta, q) = \mathbb{E}_{q(T)} \log \frac{p(x,T|\theta)}{q(T)} dT$$

- $\log p(x|\theta)$: marginal likelihood
- $L(\theta, q)$: variational lower bound
- $\max$

E-step

$$L(\theta, q) \to \max_{q} \iff KL[q_\theta(T) \,||\, p(T|X,\theta)] \to \min_{q}$$

M-step

$$L(\theta, q) \to \max_{\theta} \iff \mathbb{E}_{q(T)} \log p(X, T|\theta) \to \max_{\theta}$$

E-step

$$KL\{q_\theta(T) \,||\, p(T|X,\theta)\} \to \min_{q}$$

full posterior

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

variational inference

$$KL\{q_\theta(T) \,||\, p(T|X,\theta)\} \to \min_{q \in Q}$$

Model

Known: $X$ data

Unknown: $\theta$ parameters

Unknown: $T$ latent variables

**Methods**

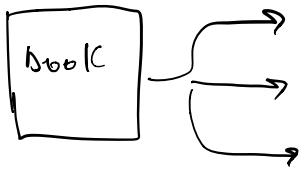
accurate ↑ inaccurate

slow ↓ fast

- Full inference $p(T, \Theta | X)$
- Mean field $q_\phi(T) q_\phi(\Theta) \approx p(T, \Theta | X)$
- EM algorithm $q_\phi(T), \Theta = \Theta_{MAP}$
- Variational EM $q_{\phi_1}(T_1) \dots q_{\phi_d}(T_d), \Theta = \Theta_{MAP}$
- Crisp EM $T = T_{MAP}, \Theta = \Theta_{MAP}$ [clusters]

# Latent Dirichlet Allocation

## Topic Modeling




60% detective
30% adventure
10% horror

Document is a distribution over topics

sport
20% football
10% hockey

economy
2% money
3% dollar

politics
10% president
4% USA

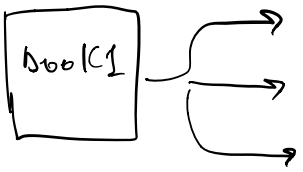
(topics)

Football player from USA has

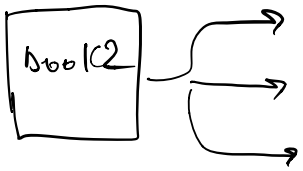
salary in dollars

**Topic is a distribution over words**

## Similarity




60% detective
30% adventure $= \begin{pmatrix} 0.6 \\ 0.3 \\ 0.1 \end{pmatrix} = a$
10% horror




62% detective
33% adventure $= \begin{pmatrix} 0.62 \\ 0.33 \\ 0.05 \end{pmatrix} = b$
5% horror

## Goals

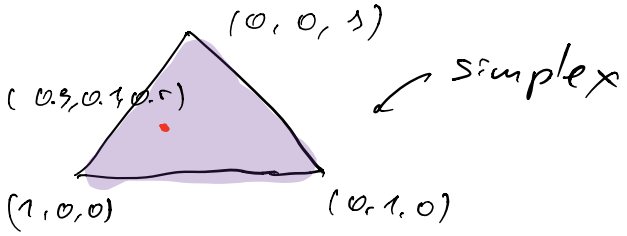
1. Constract topics
2. Assign topics to texts

## Dirichlet distribution

$$Dir(\theta|\alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

$\sum \theta_k = 1$

$\theta_k \geq 0$



## Statistics

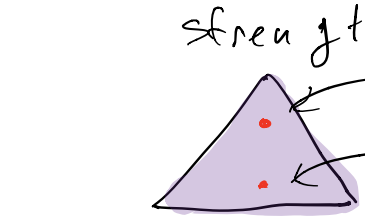
$$\mathbb{E}\,\theta_i = \frac{\alpha_i}{\alpha_0}$$

$$\mathrm{cov}(\theta_i, \theta_j) = \frac{\alpha_i \alpha_0 [i=j] - \alpha_i \alpha_j}{\alpha_0^2 (\alpha_0 + 1)}$$

$$\alpha_0 = \sum_{k=1}^{K} \alpha_k$$

## Example

massively multiplayer online role-playing game (MMORPG)




all players $\Rightarrow$ $\alpha = (3, 1, 1)$

get it from data

## Conjugate prior

$P(\theta)$ is conjugate to $P(X|\theta)$

$$P(\theta|X) = \frac{P(X|\theta)\,P(\theta)}{P(\theta)}$$

$P(\theta|X)$: $A(\nu)$

$P(X|\theta)\,P(\theta)$: $A(\nu)$

## Multinomial likelihood

$$P(X|\theta) = \frac{n!}{x_1! \dots x_K!} \theta_1^{x_1} \dots \theta_K^{\theta x_K}$$

$$p(\theta) = Dir(\theta|\alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

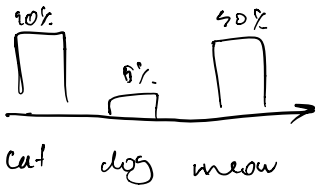
$$p(\theta|X) \propto \prod_{k=1}^{K} \theta_k^{\alpha_k + x_k - 1}$$

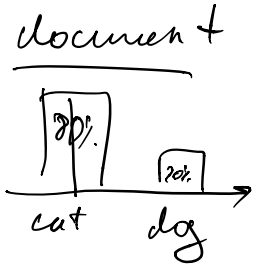
$$p(\theta|X) = Dir(\theta | (\alpha_k + x_k))$$

## Latent Dirichlet Allocation

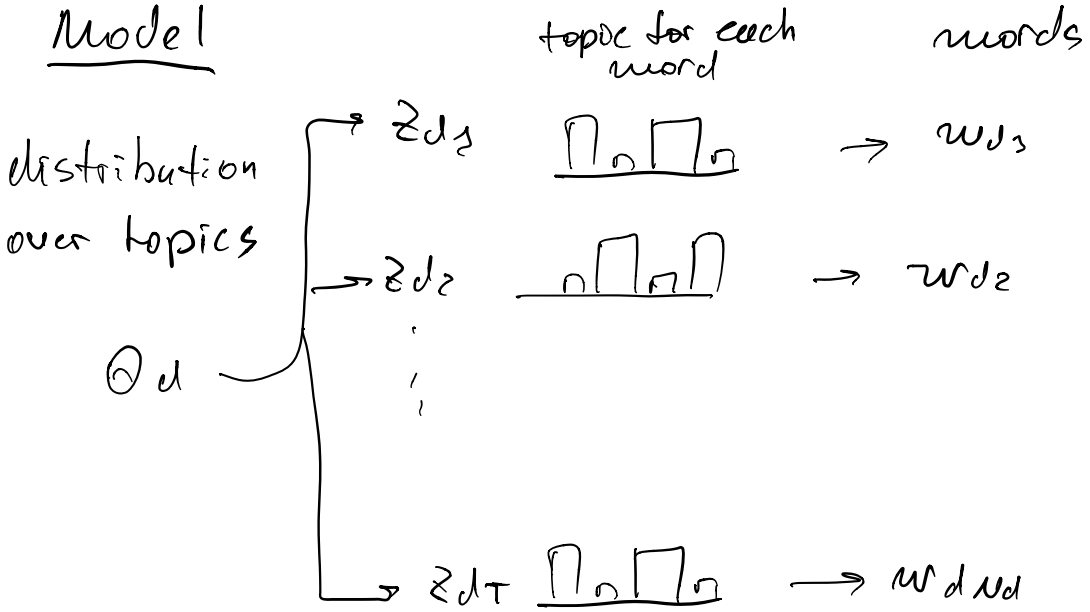
Document is a distribution over topics

Topic is a distribution over words



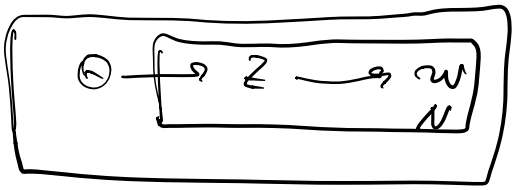



## Model




$z_{dn} \in \{1, \ldots T\}$ $\quad w_{dn} \in \{1 \ldots V\}$

## LDA Model




$$p(w, z, \theta) = \prod_{d=1}^{D} p(\theta_d) \prod_{n=1}^{N_d} p(z_{dn} | \theta_d)\, p(w_{dn} | z_{dn})$$

- $D$: for each document
- $N_d$: for each word
- $p(\theta_d)$: generate topic probabilities
- $p(z_{dn} | \theta_d)$: select topic
- $p(w_{dn} | z_{dn})$: select word from topic

$$p(W, Z, \Theta) = \prod_{d=1}^{D} p(\Theta_d) \prod_{n=1}^{N_d} p(z_{dn} \mid \Theta_d)\, p(w_{dn} \mid z_{dn})$$

$$p(\Theta_d) \sim \mathrm{Dir}(\alpha)$$

$$p(z_{dn} \mid \Theta_d) = \Theta_{d z_{dn}}$$

$$p(w_{dn} \mid z_{dn}) = \phi_{z_{dn} w_{dn}}$$

constraints:

$$\phi_{tw} \geq 0$$

$$\sum_{w} \phi_{tw} = 1$$

Known: $W$ data

Unknown: $\phi$ parameters, distribution over words for each topic

Unknown: $Z$ latent variables, topic of each word

Unknown: $\Theta$ latent variables, distribution over topics for each document

## LDA: E. step, theta

$$p(\Theta, z, w) = \prod_{d=1}^{D} p(\Theta_d) \prod_{n=1}^{N_d} p(z_{dn} \mid \Theta_d)\, p(w_{dn} \mid z_{dn}) =$$

$$= \prod_{d=1}^{D} p(\Theta_d) \prod_{n=1}^{N_d} \Theta_{d z_{dn}} \cdot \phi_{z_{dn} w_{dn}}$$

$p(W \mid \Phi) \to \max_{\Phi}$

E step: $KL(q(\Theta)q(Z) \| p(\Theta, Z \mid W)) \to \min$

$q(\Theta), q(Z)$

M-step

$$\mathbb{E}_{q(\Theta)q(T)} \log p(\Theta, Z, W) \to \max_{\Phi}$$

$$\log q(\Theta) = \mathbb{E}_{q(Z)} \log p(\Theta, Z \mid W) + \text{const} \quad (=)$$

$$p(\Theta, Z \mid W) = \frac{p(\Theta, Z, W)}{p(W)}$$ does not depend on $q(Z)$

$$(=) \quad \mathbb{E}_{q(Z)} \log p(\Theta, Z, W) + \text{const} \quad (=)$$

$$\log_D p(\Theta, Z, W) =$$

$$= \sum_{d=1}^{D} \left[ \sum_{t=1}^{T} (n_{dt} - 1) \log \theta_{dt} + \tau \right.$$ (constant w.r.t. $\Theta$)

$$\left. + \sum_{n=1}^{N_d} \sum_{t=1}^{T} [z_{dn} = t] (\log \theta_{dt} + \log \phi_{t w_{dn}}) \right] + \text{const}$$

$$= \mathbb{E}_{q(z)} \left( \sum_{d=1}^{D} \left[ \sum_{t=1}^{T} (\alpha_t - 1) \log \theta_{dt} + \sum_{n=1}^{N_d} \sum_{t=1}^{T} [z_{dn} = t] \log \theta_{dt} \right] \right) + \text{const} =$$

$$= \sum_{d=1}^{D} \left[ \sum_{t=1}^{T} (\alpha_t - 1) \log \theta_{dt} + \sum_{n=1}^{N_d} \sum_{t=1}^{T} \underbrace{\mathbb{E}_{q(z_{dn})} [z_{dn} = t]}_{\gamma_{dn}} \cdot \log \theta_{dt} \right] + \text{const} =$$

$$= \sum_{d=1}^{D} \sum_{t=1}^{T} \left[ (\alpha_t - 1) + \sum_{n=1}^{N_d} \gamma_{dn} \right] \cdot \log \theta_{dt} + \text{const}$$

$$q(\theta) = \prod_{d=1}^{D} \prod_{t=1}^{T} \theta_{dt}^{\left[\alpha_t + \sum_n \gamma_{dn} - 1\right]} \cdot \text{const}$$

$$\boxed{q(\theta) = \prod_{d=1}^{D} q(\theta_d), \quad q(\theta_d) = \text{Dir}(\theta_d \mid \alpha + \textstyle\sum \gamma_{dn})}$$

## LDA : E-step, z

$$\log q(z) = \mathbb{E}_{q(\theta)} \log p(\theta, z, w) + \text{const} =$$

$$p(\theta, z \mid w) = \frac{p(\theta, z, w)}{p(w)}$$ ← does not depend on $q(z)$

$$= \mathbb{E}_{q(\theta)} \left[ \sum_d \left[ \sum_t (\alpha_t - 1) \log \theta_{dt} + \sum_n \sum_t [z_{dn} = t] \left( \log \theta_{dt} + \log \varphi_{t w_{dn}} \right) \right] \right]$$

← does not depend on $z$

$$+\text{const} =$$

$$= \mathbb{E}_{q(\Theta)} \sum_d \sum_n \sum_t [z_{dn}=t] \left(\log \theta_{dt} + \log \varphi_{tw_{dn}}\right) + \text{const} =$$

$$= \sum_d \sum_n \sum_t [z_{dn}=t] \left[ \mathbb{E}_{q(\theta)} \log \theta_{dt} + \log \varphi_{tw_{dn}} \right] + \text{const}$$

$$q(z) = \prod_d^D \prod_n^{N_d} q(z_{dn}) + \text{const}$$

$$q(z_{dn}=t) = \frac{\varphi_{tw_{dn}} \cdot \exp\left(\mathbb{E}_{q(\theta_{dt})} \log \theta_{dt}\right)}{\sum_{t=1}^{T} \varphi_{tw_{dn}} \cdot \exp\left(\mathbb{E}_{q(\theta_{dt})} \log \theta_{dt}\right)} = \gamma_{dn}(t)$$

$$\gamma_{dn} = \mathbb{E}_{q(z_{dn})} [z_{dn}=t]$$

## LDA: M-step & Prediction

$$\mathbb{E}_{q(\Theta) q(T)} \log P(\Theta, z, w) \to \max_{\Phi}$$

$$\mathbb{E}_{q(\Theta) q(T)} \log P(\Theta, z, w) =$$

$$= \mathbb{E}_{q(\Theta) q(z)} \left[ \sum_d \left[ \sum_t (\alpha_t - 1) \log \theta_{dt} \right] + \sum_n \sum_t [z_{dn}=t] \left(\log \theta_{dt} + \log \varphi_{tw_{dn}}\right) \right] =$$

const $\Phi$ const $\Phi$

$$= \mathbb{E}_{q(\theta)q(z)} \left[ \sum_d \sum_n \sum_t [z_{dn}=t] \cdot \log \varphi_{tw_{dn}} \right] + \text{const} =$$

$$\max_{\varphi}$$

✓ $\varphi_{tw} \geq 0 \quad \forall t, w$

$$\sum_{w=1}^{V} \varphi_{tw} = 1 \quad \forall t$$

(using Lagrangian)

$$L = \mathbb{E}_{q(\theta)q(z)} \sum_d \sum_n \sum_t [z_{dn}=t] \log \varphi_{tw_{dn}} +$$

$$+ \sum_{t=1}^{T} \lambda_t \left( \sum_w^V \varphi_{tw} - 1 \right) =$$

$$= \sum_d \sum_n \sum_t \gamma_{dn}^t \cdot \log \varphi_{tw_{dn}} + \sum_t \lambda_t \left[ \sum_w^V \varphi_{tw} - 1 \right)$$

$$\frac{\partial L}{\partial \varphi_{tw}} = \sum_d \sum_n \varphi_{dn}^t \frac{1}{\varphi_{tw_{dn}}} \cdot [w_{dn} = w] +$$

$\varphi_{tw_{dn}} = \varphi_{tw}$

$$+ \lambda_t = 0$$

$$\Rightarrow \quad \varphi_{tw} = \frac{\sum_d \sum_n \gamma_{dn}^t \cdot [w_{dn} = w]}{-\lambda_t}$$

$$\sum_w \varphi_{tw} = \sum_w \frac{\sum_d \sum_n \gamma_{dtn} [w_{dn} = w]}{\lambda_t} = 1$$

$$\lambda_t = \sum_w \sum_d \sum_n \gamma_{dtn} [w_{dn} = w]$$

$$\varphi_{tw} = \frac{\sum_d \sum_n \gamma_{dtn} [w_{dn} = w]}{\sum_{w'} \sum_d \sum_n \gamma_{dtn} [w_{dn} = w']}$$

predictions

$p(\theta_{d^*}, z_{d^*} \mid W, \Phi)$ - approximate

$d^*$ - new document
$W$ - train data

$$KL\left(q(\theta_{d^*}) q(z_{d^*}) \,\|\, p(\theta_{d^*}, z_{d^*} \mid W, \Phi)\right) \to \min$$

$q(\theta_{d^*})$
$q(z_{d^*})$

## Extensions of LDA

$$p(W, Z, \Theta) = \prod_{d=1}^{D} p(\Theta_d) \prod_{n=1}^{W_d} p(z_{dn} \mid \Theta_d)\, p(w_{dn} \mid z_{dn})$$

$p(\Theta_d) \sim Dir(\alpha)$

$\alpha \uparrow$ => More topics for each document

$\alpha \downarrow$ => Less topics for each document

$\alpha$ can be selected as $p(W \mid \alpha) \to \max_{\alpha}$

## Sparsity of topics

Sparse prior on $\Phi$

$$p(W, Z, \Theta, \Phi) = \prod_{t=1}^{T} p(\Phi_t)\, p(W, Z, \Theta \mid \Phi)$$

$p(\Phi_t) \sim Dir(\beta)$

## Topics correlation

logistic normal distribution

$p(\Theta_d) \sim \mathcal{P}(\mathcal{N}(\mu, \Sigma))$

## Dynamic Topic Model

$p(B_{t\cdot}^{\uparrow t+1} \mid B_{t\cdot}^{\uparrow t}) \sim \mathcal{N}(B_{t\cdot}^{\uparrow t}, \sigma^2 I)$

$\Phi_{t\cdot}^{\uparrow t+1} = Soft\,max[B_{t\cdot}^{\uparrow t}]$

## Summary

- Many topics are interpretable
- Works well with rare words
- Fast even for huge text collections
- Multicore & distributed implementations
- Many features can be added with extensions

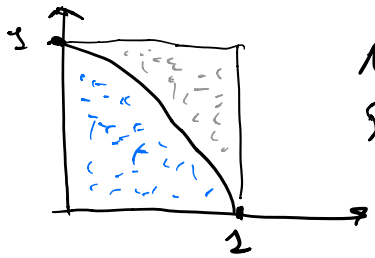
Week 4

# Markov chain Monte Carlo

## Monte Carlo estimation

- MCMC - silver bullet of probabilistic modeling

Estimate expected value by sampling




$M = 3000$

$\hat{\pi} \approx 3.128$

$$\frac{\pi}{4} = \mathbb{E}\left[x^2 + y^2 \le 1\right] \approx \frac{1}{M} \sum_{s=1}^{M} \left[x_s^2 + y_s^2 \le 1\right]$$

$x_s, y_s \sim U(0, 1)$

$\mathbb{E}_{p(x)} f(x)$ - general form

$$\mathbb{E}_{p(x)} f(x) \approx \frac{1}{M} \sum_{s=1}^{M} f(x_s), \quad x_s \sim p(x)$$

Why do we need to estimate expected values?

- Full Bayesian inference (week 1)

$p(y \mid x, Y_{train}, X_{train}) =$

$$= \int p(y \mid x, w)\, p(w \mid X_{train}, Y_{train})\, dw \quad (\approx)$$

$p(y \mid x, w)$ - neural network, $w$ - weight of nn

$$(\approx) \quad \mathbb{E}_{p(w \mid X_{train}, Y_{train})}\, p(y \mid x, w)$$

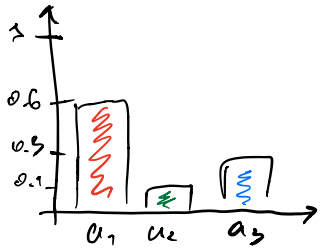
$$p(w \mid Y_{train}, X_{train}) = \frac{p(Y_{train}, X_{train}, w)\, p(w)}{Z}$$

- **M-step of EM-algorithm** (week 2)

$$\max_{\theta} \mathbb{E}_{\theta} \log p(X, T \mid \theta)$$

## Sampling from 1-d distributions

### 1-d sampling (discrete)




we have $U(0, 1)$


1-d discrete distributions with finite number of values are easy

At least then number of values is $< 10\,000$

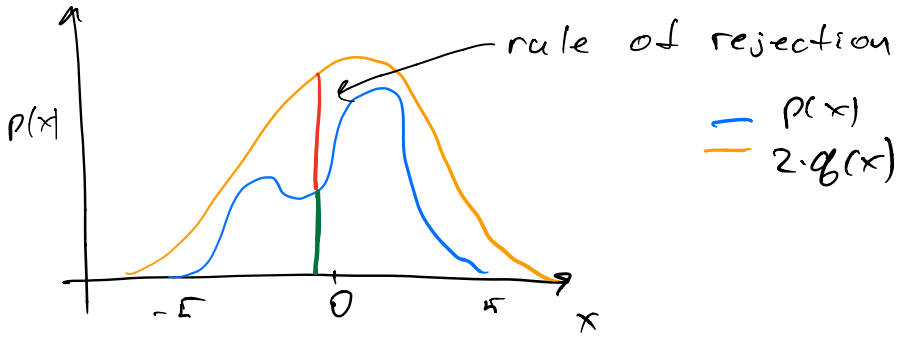
# continuous sampling

Sampling from Gaussian distribution

idea: central limit theorem

$$z = \sum_{i=1}^{12} x_i - 6 \quad , x_i \sim U(0,1)$$

$$p(z) \approx \mathcal{N}(0,1)$$




$\tilde{x} \sim q(x)$ $\quad y \sim U[0, 2q(\tilde{x})]$

Accept $\tilde{x}$ with probability $\frac{p(x)}{2q(x)}$ : if $y \leq p(x)$

How is it effective?

$$p(x) \leq M q(x)$$

Accept $\frac{1}{M}$ points on average

$$\frac{\hat{p}(x)}{z} \leq M q(x) \Rightarrow \hat{p}(x) \leq \underbrace{zM}_{\text{new } M} q(x)$$

works if you can upper bound your unnormalized distribution

## Summary of rejection sampling

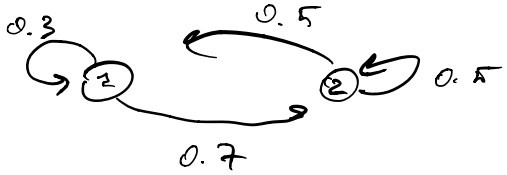
pros:

- Works for most distributions (even unnormalized)

cons:

- if $q$ and $p$ are too different ($M$ is large), rejects most of the points
- $M$ is large for $d$-dimensional distributions

## Markov Chains




$T(L \to L) = 0.3$

$T(L \to R) = 0.7$

$T(R \to L) = 0.5$

$T(R \to R) = 0.5$

T-transition probabilities

| | 1 | 2 |
|---|---|---|
| $x^1$ | 1 | 0 |
| $x^2$ | 0.3 | 0.7 |
| $x^3$ | $0.3^2 + 0.7 \cdot 0.5$ <br> 0.44 | $0.3 \cdot 0.7 + 0.7 \cdot 0.5$ <br> 0.56 |
| ⋮ | | |
| $x^n$ | | |

$p(x^3) = p(x^3 \mid x^2 = L)\, p(x^2 = L) +$

$+ p(x^3 \mid x^2 = R)\, p(x^2 = R)$

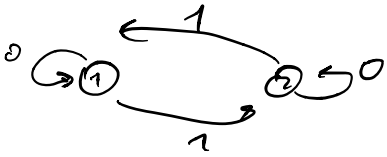
rule of sum prob

$p(L) \approx 0.42 \quad p(R) \approx 0.58$

- We want to sample from $p(x)$
- Build a Markov chain that converge to $p(x)$
- Start from any $x^0$
- For $k = 0, 1, \ldots$

$$x^{k+1} \sim T(x^k \to x^{k+1})$$

- Eventually $x^k$ will look like sample from $p(x)$




- markov chain is not converge

## Definition

A distribution $\pi$ is called stationary if

$$\pi(x') = \sum_x T(x \to x') \pi(x)$$

## Theorem:

if $T(x \to x') > 0$ for all $x, x'$ then exist unique $\pi$:

$$\pi(x') = \sum_x T(x \to x') \pi(x)$$

and Markov chain converges to $\pi$ from any starting point

# Gibbs sampling

$$p(x_1, x_2, x_3) = \frac{\hat{p}(x_1, x_2, x_3)}{Z}$$

start with $(x_1^0, x_2^0, x_3^0)$ e.g. $(0,0,0)$

$$x_1^1 \sim p(x_1 \mid x_2 = x_2^0, x_3 = x_3^0) = \frac{\hat{p}(x_1, x_2^0, x_3^0)}{Z_1}$$

$$x_2^1 \sim p(x_2 \mid x_1 = x_1^1, x_3 = x_3^0)$$

$$x_3^1 \sim p(x_3 \mid x_1 = x_1^1, x_2 = x_2^1)$$

For $k = 0, 1, 2 \ldots$

$$x_1^{k+1} \sim p(x_1 \mid x_2 = x_2^k, x_3 = x_3^k)$$

$$x_2^{k+1} \sim p(x_2 \mid x_1 = x_1^{k+1}, x_3 = x_3^k)$$

$$x_3^{k+1} \sim p(x_3 \mid x_1 = x_1^{k+1}, x_2 = x_2^{k+1})$$

## Proof of Gibs Sampling

$$p(x', y', z') = \sum_{x,y,z} q(x, y, z \to x', y', z')\, p(x, y, z)$$

(want to prove)

$$\sum_{x,y,z} p(x' \mid y = y, z = z)\, p(y' \mid x = x', z = z) \cdot p(z' \mid x = x', y = y') \cdot p(x, y, z) =$$

$$= p(z' \mid x=x', y=y') \sum_{x,y,z} p(x' \mid y=y, z=z)\, p(y' \mid x=x', z=z)\, p(x, y, z) =$$

$$= p(z' \mid x=x', y=y') \sum_{y,z} p(x' \mid y=y, z=z)\, p(y' \mid x=x', z=z) \underbrace{\sum_{x} p(x, y, z)}_{p(y,z)} =$$

$$= p(z' \mid x=x', y=y') \sum_{y,z} p(x' \mid y=y, z=z)\, p(y' \mid x=x', z=z)\, p(y, z) =$$

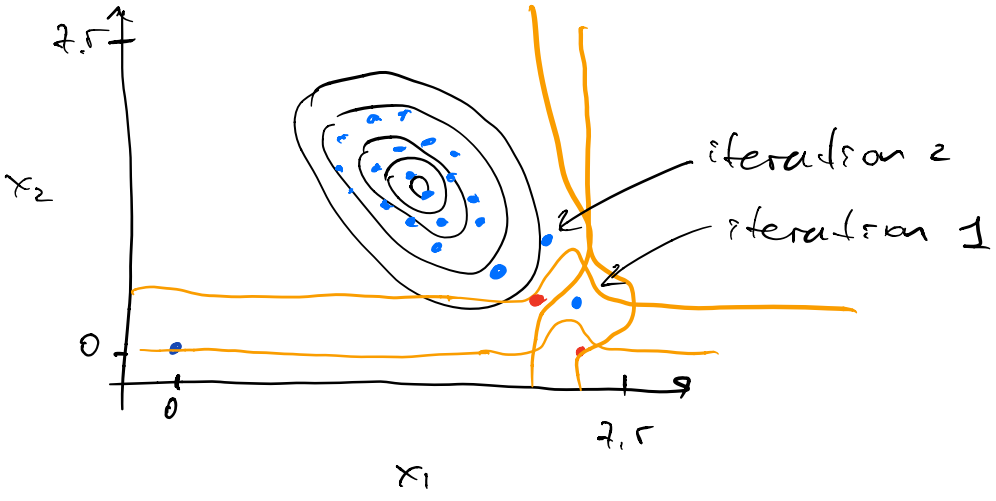
$$= p(z' \mid x=x', y=y') \sum_{z} p(y' \mid x=x', z=z) \cdot \sum_{y} p(x' \mid y=y, z=z)\, p(y, z) =$$

$$= p(z' \mid x=x', y=y') \sum_{z} p(y' \mid x=x', z=z)\, p(x', z) =$$

$$= p(z' \mid x=x', y=y') \sum_{z} p(y', x', z) =$$

$$= p(z' \mid x=y', y=y')\, p(y', x') = p(z', y', x')$$

## Example of Gibbs sampling



## Summary

pros:
- reduce multidimensional sampling to sequance of 1d sampling
- a few lines of code

cons:
- highly correlated samples
- slow convergence (mixing)
- not parallel

## Metropolis-Hastings

Apply rejection sampling to Markov Chains

For $k = 1, 2, \ldots$
- sample $x'$ from a wrong $Q(x^{(k)} \to x')$
- Accept proposal $x'$ with probability $A(x^{(k)} \to x')$
- Otherwise stay at $x^{(k)}$: $x^{(k+1)} = x^{(k)}$

$$T(x \to x') = Q(x \to x') A(x \to x') \text{ for all } x \neq x'$$

$$T(x \to x') = Q(x \to x) A(x \to x) + \sum_{x' \neq x} Q(x \to x') (1 - A(x \to x'))$$

How to choose $A$:

$$\pi(x') = \sum_x \pi(x) T(x \to x')$$

## Detailed Balance

$\pi(x') = \sum_x \pi(x) T(x \to x')$ - stationary distribution

If: ↙ detailed balance question

$$\pi(x) T(x \to x') = \pi(x') T(x' \to x)$$

Then:

$$\pi(x') = \sum_x \pi(x) T(x \to x')$$

Proof:

$$\sum_x \pi(x) T(x \to x') = \sum_x \pi(x') T(x' \to x)$$

$$\pi(x') \underbrace{\sum_x T(x' \to x)}_{1} = \pi(x')$$

How to choose $A$:

$$\pi(x) T(x \to x') = \pi(x') T(x' \to x)$$

# Metropolis-Hastings: choosing the critic

choosing a critic:

$$\pi(x)\underbrace{Q(x\mapsto x')}_{T(x\mapsto x')}A(x\mapsto x') = \pi(x')\underbrace{Q(x'\mapsto x)}_{T(x'\mapsto x)}A(x'\mapsto x)$$

$$\frac{A(x\to x')}{A(x'\mapsto x)} = \frac{\pi(x')\,Q(x'\to x)}{\pi(x)\,Q(x\to x')} = \rho < 1$$

assumption

$$A(x\to x') = \rho$$

$$A(x'\to x) = 1$$

$$\frac{A(x\to x')}{A(x'\mapsto x)} = \frac{\pi(x')\,Q(x'\to x)}{\pi(x)\,Q(x\to x')} = \rho > 1$$

$$A(x\mapsto x') = 1$$

$$A(x'\mapsto x) = 1/\rho$$

$$\Downarrow$$

$$A(x\to x') = \min\left\{1, \frac{\widehat{\pi}(x')\,Q(x'\to x)}{\widehat{\pi}(x)\,Q(x\mapsto x')}\right\}$$

we don't care about normalization const

## Choice of Q

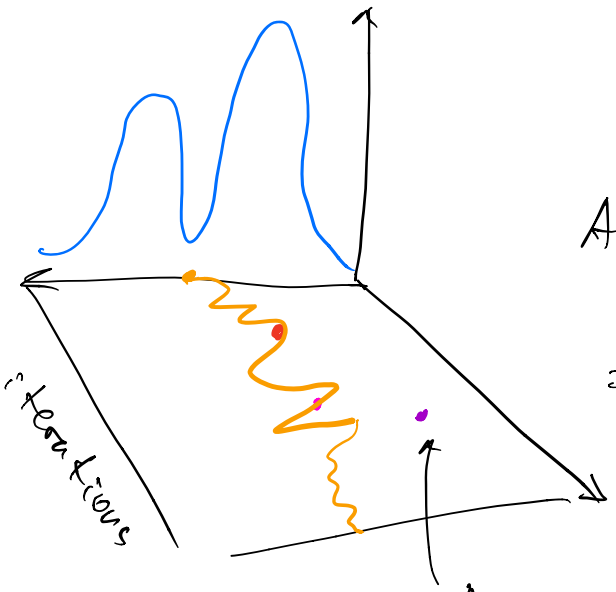
$Q(x \to x') > 0$

opposing forces:

- $Q$ should spread out, to improve mixing and reduce correlation
- But then acceptance probability is often low

## Example of Metropolis-Hastings

$$Q(x \to x') = \mathcal{N}(x, 1)$$




$$A(x \to x') = \min\left\{1, \frac{\pi(x')\,Q(x' \to x)}{\pi(x)\,Q(x \to x')}\right\} =$$

$$= \min\left\{1, \frac{\pi(x')}{\pi(x)}\right\} =$$

$$= \min\left\{1, \frac{0.27}{0.07}\right\} = 1$$

$$A(x \to x') = \min\left\{1, \frac{0.04}{0.28}\right\} = 0.13$$

$Q = \mathcal{N}(x, 0.02)$ – very low convergence
stay at the same region

$Q = \mathcal{N}(x, 100)$ – very low convergence
stay at same place

## Metropolis Hastings as correction scheme

Recall Gibbs sampling
Lets make it parallel

$$x_1^{k+1} \sim p(x_1 \mid x_2 = x_2^k, x_3 = x_3^k)$$

$$x_2^{k+1} \sim p(x_2 \mid x_1 = x_1^k, x_3 = x_3^k)$$

$$x_3^{k+1} \sim p(x_3 \mid x_1 = x_1^k, x_2 = x_2^k)$$

It's wrong now, but can correct with Metropolis Hastings!

Summary

Rejection sampling applied to Markov Chains

Pros:
- You can choose among family of Markov Chains
- Works for unnormalized densities
- Easy to implement

Cons:
- Samples are still correlated
- Have to choose among family of Markov Chains

Markov Chain Monte Carlo summary

Two MCMC approaches:

- Gibbs sampling - reducing multidimensional sampling to a sequence of 1d
- Metropolis Hastings - rejected sampling for Markov Chains

MC (monte carlo) → approximate via sampling

sampling via MC (markov chain)

## Monte Carlo vs Variational Inference

### Monte Carlo

$$\mathbb{E}_{p(x)} f(x) \approx \frac{1}{M} \sum_{s=1}^{M} f(x_s), \quad x_s \sim p(x)$$

unbiased estimation (larger $M \Rightarrow$ better accuracy)

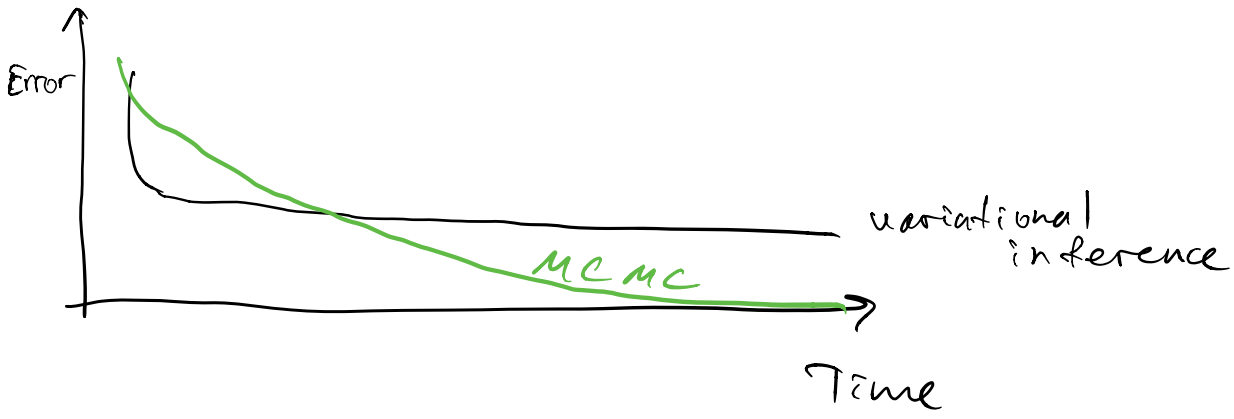
$$\mathbb{E}_{p(x)} \frac{1}{M} \sum_{s=1}^{M} f(x_s) = \mathbb{E}_{p(x)} f(x)$$

### Variational Inference

$$p(x) \approx q(x)$$

$$\mathbb{E}_{p(x)} f(x) \approx \mathbb{E}_{q(x)} f(x)$$

Schematic illustration




## Methods

best

- Full inference $p(T, \Theta | X)$
- Mean field $q_\phi(T) q_\phi(\Theta) \approx p(T, \Theta | X)$
- **MCMC** $T_s, \Theta_s \sim p(T, \Theta | X)$
- EM algorithm $q_\phi(T)$, $\Theta = \Theta_{MP}$
- Variational EM $q_{\phi_1}(T_1) \dots q_{\phi_d}(T_d)$, $\Theta = \Theta_{MP}$
- **MCMC EM** $T_s \sim p(T | \Theta, X)$, $\Theta = \Theta_{MP}$

worst

## Summary MCMC

Pros:

- easy to implement
- easy to parallelize
- unbiased estimates (wait more $\Rightarrow$ more accuracy)

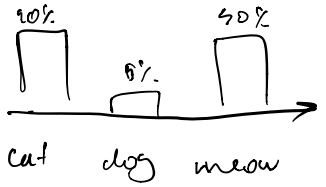
Cons:

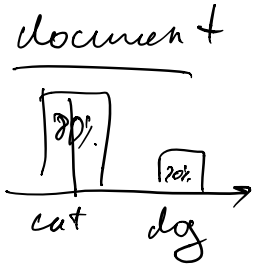
- usually slower than alternatives

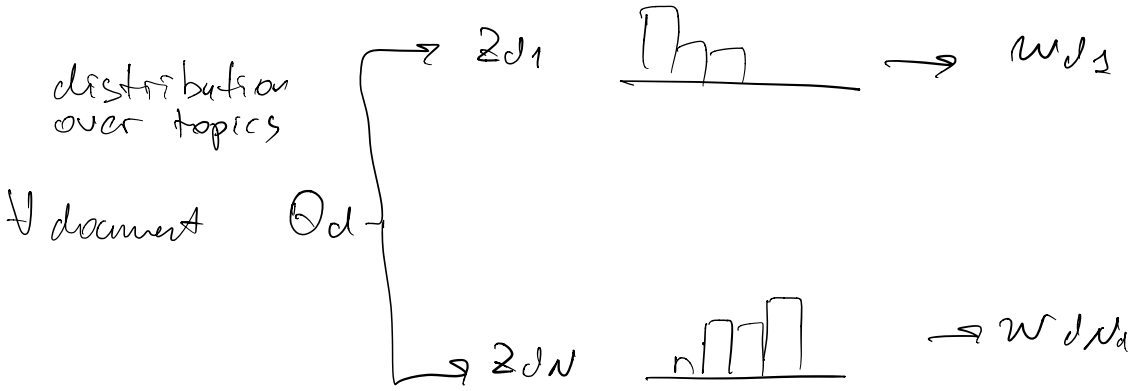
# MCMC to LDA

Document is a distribution over topics

Topic is a distribution over words




$z_{dn} \in \{1, \dots, T\}$ $w_{dn} \in \{1, \dots V\}$

## Mean field for LDA

$$\max_{\Phi} p(W \mid \Phi)$$

E-step $p(z, \Theta \mid W, \Phi) \approx q(z, \Theta)$

M-step $\max_{\Phi} \mathbb{E}_q \log p(z, \Theta, W \mid \Phi)$

# MCMC for LDA

Known: $W$ data

Unknown: $\Phi$ parameters, distribution over words for each topic

Unknown: $Z$ latent variables, topic of each word

Unknown: $\Theta$ latent variables, distribution over topics for each document

⇓ lets go full Bayesian

Known: $W$ data

Unknown: $\Phi$ latent variables, distribution over words for each topic

Unknown: $Z$ latent variables, topic of each word

Unknown: $\Theta$ latent variables, distribution over topics for each

$p(\Phi, \Theta, Z \mid W) \sim$ {bibbs Sampling}

Init: $\Phi^0, \Theta^0, Z^0$

$\Phi_1^1 \sim p(\Phi_1 \mid \Phi_2^0, \Phi_3^0, \ldots, \Theta^0, Z^0, W)$

$$\phi_2^1 \sim p(\phi_2 \mid \phi_1^1, \phi_3^0, \ldots, \Theta^0, z^0, w)$$

$$\vdots$$

$$\phi_i^1 \sim p(\phi_i \mid \phi_1^1, \ldots, \phi_{i-1}^1, \phi_{i+1}^0, \ldots \Theta^0, z^0, w)$$

$$\theta_i^1 \sim p(\theta_i \mid \phi^1, \theta_1^1, \ldots \theta_{i-1}^1, \theta_{i+1}^0, \ldots, z^0, w)$$

$$z_i^1 \sim p(z_i \mid \phi^1, \theta^1, z_1^1, \ldots z_{i-1}^1, z_{i+1}^0, \ldots, w)$$

for $k = 1, 2, \ldots$

## Collapsed Gibbs for LDA

Model

$p(\theta_d) = Dir(\beta)$ ← $p(\phi_k) = Dir(\alpha)$ ←

$p(z_d \mid \theta_d) = \theta_{d z_{dn}}$ ← $p(w_{dn} \mid z_{dn}, \phi) = \phi_{z_{dn} w_{dn}}$ ←

conjugate

can compute analytically

$p(\theta \mid z)$

$$p(z) = \int p(z \mid \theta) p(\theta) d\theta =$$

$$= \frac{p(z \mid \theta) p(\theta)}{p(\theta \mid z)}$$

conjugate

$$p(\phi \mid z, w)$$

$$p(w \mid z) = \frac{p(w \mid z, \phi)\, p(\phi)}{p(\phi \mid z, w)}$$

$$p(z \mid w) = \frac{p(w \mid z)\, p(z)}{c}$$

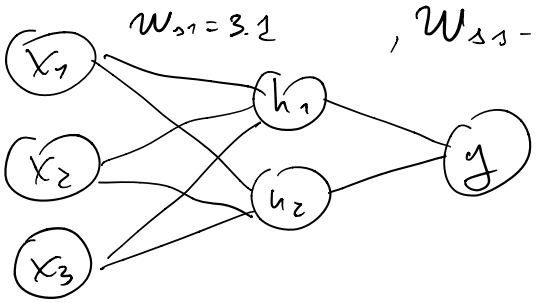
$p(z \mid w)$ ~ {Gibbs Sampling}

$$p(\phi \mid w) = \int p(\phi \mid w, z)\, p(z \mid w)\, dz =$$

$$= \mathbb{E}_{p(z \mid w)}\, p(\phi \mid w, z) \approx$$

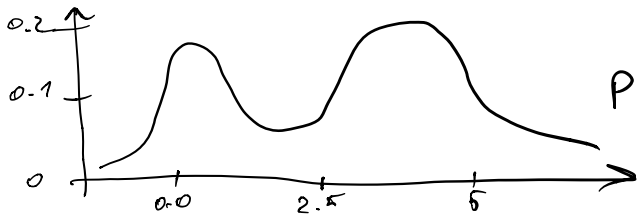
$$\approx p(\phi \mid w, \bar{z})$$

## Bayesian Neural Networks




, $w_{ij}$ - distribution in BNN

BNN uses distributions instead numbers




$$p(w_{31} \mid \Sigma_{train}, X_{train})$$

$$p(y \mid x, Y_{train}, X_{train}) =$$

$$= \int \underbrace{p(y \mid x, w)}_{NN\ output}\, p(w \mid Y_{train}, X_{train})\, dw =$$

$$= \mathbb{E}_{p(w \mid Y_{train}, X_{train})}\, p(y \mid x, w)$$

$$p(w \mid Y_{train}, X_{train}) \sim \{\text{Gibbs}\}$$

$$p(w \mid Y_{train}, X_{train}) = \frac{p(Y_{train} \mid X_{train}, w)\, p(w)}{Z}$$

problem: depends on the whole dataset!

## Langevin Monte Carlo

Gibbs and Metropolis Hastings can't do mini-batches

Say we want to sample from $p(w \mid D)$

Start from $w^0$

For $k = 1, \dots$

$$w^{k+1} = \underbrace{w^k + \varepsilon \nabla \log p(w^k \mid D)}_{\text{gradient ascent}} + \zeta^k =$$

$$\zeta^k \sim \mathcal{N}(0, 2\varepsilon I)$$

$$= w^{(k)} + \varepsilon \nabla \left( \log p(w^{(k)}) + \sum_{i=1}^{N} \log p(y_i | x_i, w^{(k)}) \right) + \zeta^{(k)}$$

weight decay $-c \|w^{(k)}\|^2$

usual cross entropy

scheme:

- initialize weight $w^0$
- Do say 100 iterations with usual SGD, but add Gaussian noise $\zeta^{(k)} \sim N(0, 2\varepsilon I)$ to each update
- After 100 epochs decide that Markov Chain converged and start collecting weights values
- For a new object predict compute average prediction of CNN with weight $w^{100}, w^{101}, \ldots w^{200}$ (↑ better)
- Train another CNN to mimic the ensemble

Week 5 20.10.2018

# Variational Autoencoder

## Scaling Variational Inference & Unbiased estimates

### Scale Variational Inference

People used to thing that Bayes is for small datasets

- Too slow for big data
- Not very beneficial anyway

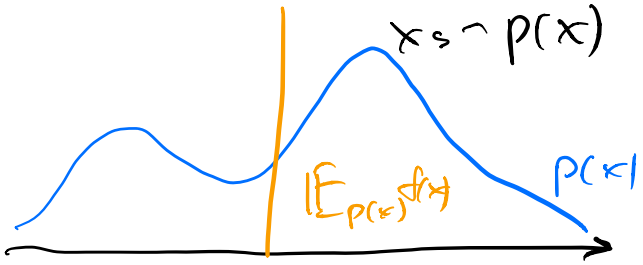
Things changed when Bayes met Deeplearning

- Understand why and how combine Deep Learning and Bayesian methods
- Learn how to synthesize images with VAE
- Learn state-of-the-art Bayesian neural networks and their applications

### Unbiased estimates

$$\mathbb{E}_{p(x)} f(x) \approx \frac{1}{M} \sum_{s=1}^{M} f(x_s) = R$$

R: random variable




$$\mathbb{E}_{p(x)} R = \mathbb{E}_{p(x)} f(x)$$

not unbiased

$$\log\left(\mathbb{E}_{p(x)} f(x)\right) \overset{?}{\approx} \log\left(\frac{1}{M} \sum_{s=1}^{M} f(x_s)\right) = G$$

$x_s \sim p(x)$

$$\mathbb{E}_{p(x)} G \neq \log\left(\mathbb{E}_{p(x)} f(x)\right)$$

- Estimator called unbiased if its expected value equals to thing it estimates.
- This is unbiased estimator:

$$\mathbb{E}_{p(x)} f(x) \approx \frac{1}{M} \sum_{s=1}^{M} f(x_s) = R$$

## Modeling a distribution of images

Let's start with fitting $p(x)$ into a dataset.

But why do we need it?

- Generate new data.
- Detect anomalies and outliers.
- Work with missing data.
- Represent your data in a nice way (e.g. model p(molecule) to search for drugs)

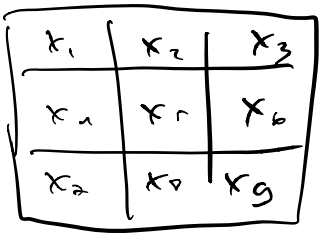
## How to model $p(x)$

- $\log \hat{p}(x) = CNN(x)$

$$p(x) = \frac{\exp(CNN(x))}{Z}$$

$Z$ - infeasible → billions of images.

- Use the chain rule




image

$$p(x_1, \ldots x_d) = p(x_1)\, p(x_2|x_1) \ldots \ldots p(x_d|x_1, \ldots x_{d-1})$$

pixel recurrent neural networks

$p(x_k | x_1, \dots x_{k-1}) = RNN(x_1, \dots, x_{k-1})$

cool, but slow to generate

- $p(x_1, \dots x_d) = p(x_1) \cdots p(x_d)$

too restrictive

- Mixture of several Gaussians (GMM)

still too restrictive

- Mixture of infinitely many Gaussians

$$p(x) = \int p(x|t)\, p(t)\, dt$$

Using CNNs with a mixture of Gaussians

$$p(x) = \int p(x|t)\, p(t)\, dt$$

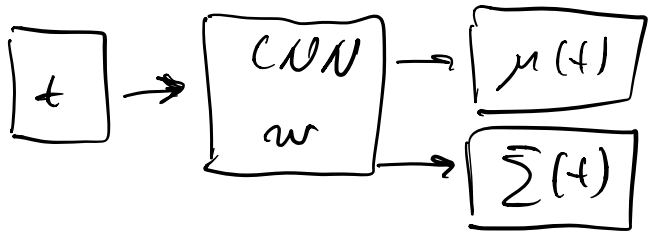
prior: $p(t) = \mathcal{N}(0, I)$

likelihood: $p(x|t) = \mathcal{N}(\mu(t), \Sigma(t))$

if: $\mu(t) = Wt + b$, $\Sigma(t) = \Sigma_0$ get PPCA (metric 2)

But if $x$ is image, why not

$\mu(t) = CNN_1(t)$ $\quad \Sigma(t) = CNN_2(t)$




$$p(x|w) = \int p(x|t,w)\,p(t)\,dt$$

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

$$p(x|t,w) = \mathcal{N}\left(\mu(t,w), \Sigma(t,w)\right)$$

$\Sigma(t,w)$: $\mathrm{diag}(\sigma^2(t,w))$

## Scaling up Expectation Maximization

$$\max_w p(x|w) = \int p(x|T,w)\,p(T)\,dt$$

Latent Variable model - use EM!

$\log p(x|w) \geq L(w,q)$ - E step

maximize $L(w,q)$ - M step
$w,q$

But E-step is intractable: Need to compute $p(T|x,w)$

MCMC - ?

$$\mathbb{E}_q \log p(X, T \mid w) \approx \frac{1}{M} \sum_{s=1}^{M} \log p(X, T_s \mid w)$$

$$T_s \sim q(T)$$

Variational EM!

$$\log p(X \mid w) \geq L(w, q)$$

$$\underset{w, q}{\text{maximize}} \quad L(w, q)$$

subject to: $q_i(f_i) = \hat{q}(t_{i1}) \dots \hat{q}(t_{im})$

but again intractable.

Scaling up Variational EM

$$\underset{w, q}{\text{maximize}} \quad L(w, q)$$

subject to: $q_i(f_i) = \hat{q}_i(t_{i1}) \dots \hat{q}_i(t_{im})$

$$\Downarrow$$

$$\underset{w, q_1, \dots q_N}{\text{maximize}} \quad L(w, q_1, \dots q_N)$$

subject to: $q_i(f_i) = \hat{q}_i(t_{i1}) \dots \hat{q}_i(t_{im})$

⇓ another aproximation

maximize $L(w, q_1, \dots q_N)$

$w, m_1, \dots m_N, s_1, \dots s_N$

subject to: $q_i(t_i) = \mathcal{N}(m_i, \text{diag}(s_i^2))$

But this may ~100 parameters for each training object

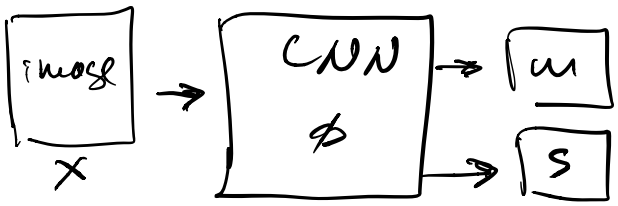
And is not clear what is $m, s$ for test objects

⇓

maximize $L(w, q_1, \dots q_N)$

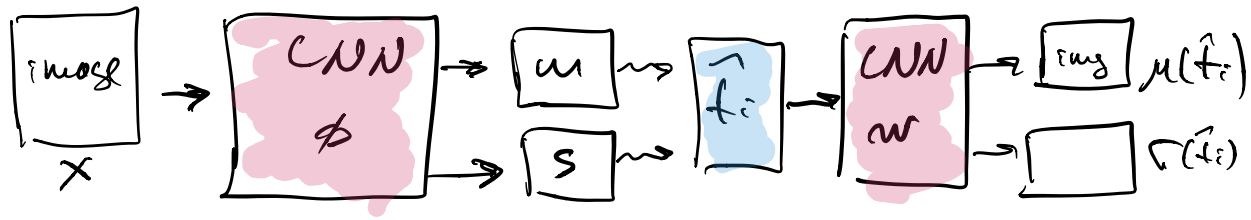
$w, \phi$

subject to $q_i(t_i) = \mathcal{N}(m(x_i, \phi), \text{diag}(s^2(x_i, \phi)))$

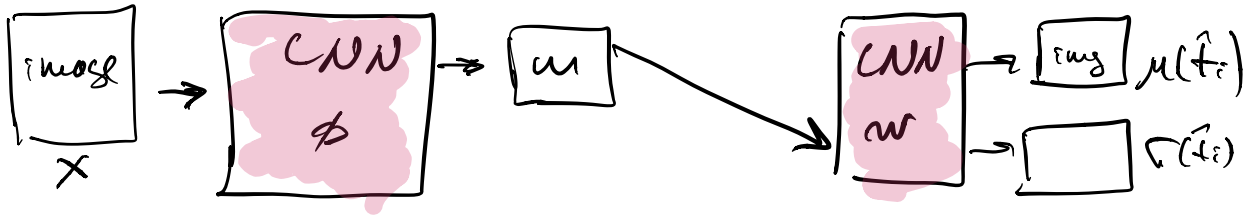



$$L(w, q_1, \dots q_N) = \sum_i \mathbb{E}_{q_i} \log \frac{p(x_i|t_i, w)\, p(t_i)}{q_i(t_i)}$$

$$\hat{t}_i \sim \mathcal{N}(m(x_i, \phi), \text{diag}(s^2(x_i, \phi)))$$




if $s(x)=0$ then $\hat{f}_i = m(x_i, \phi)$ : usual autoencoder




$$\max_{w,\phi} \sum_i \mathbb{E}_{q_i} \log \frac{p(x_i|f_i,w)\,p(f_i)}{q_i(f_i)} =$$

$$= \sum_i \mathbb{E}_{q_i} \log p(x_i|f_i,w) + \mathbb{E}_{q_i} \log \frac{p(f_i)}{q_i(f_i)}$$

$$-KL(q_i(f_i) \,\|\, p(f_i))$$

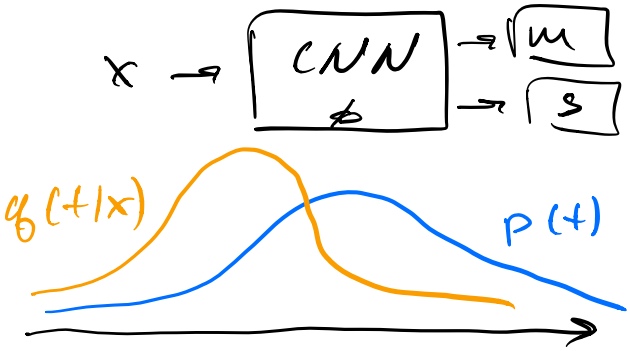
$-\|x_i - \mu(f_i)\|^2 + \text{const}$ if $\sigma(x_i)=1$ for simplicity

approximate posterior $q(f_i) \approx p(f_i|x_i,w)$

reconstruction loss

regularization

## Detecting outliers



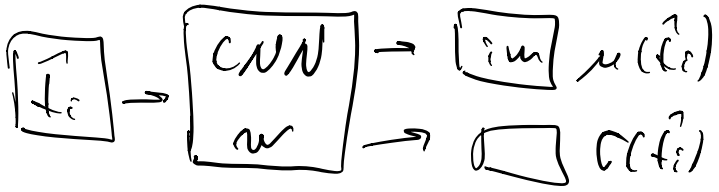

$$KL(q_\phi(t|x) \| p(t)) = 0.54$$

but for fraud $KL(q_\phi(t|x) \| p(t)) = 6.25$ for example

## Generating new samples

$$p(x|w) = \int p(x|t,w)p(t)dt$$

$\hat{t}_i \sim \mathcal{N}(0, I)$




## Gradient of decoder

$$\max_{w,\phi} \sum_i \mathbb{E}_{q_i} \log p(x_i|t_i,w) - KL(q_i(t_i) \| p(t_i))$$

easy and analytical

$$KL(q_i(t_i) \| p(t_i)) =$$

$$= \sum_j \left( -\log \sigma_j(t_i) + \frac{\sigma_j^2(t_i) + \mu_j^2(t_i) - 1}{2} \right)$$

$$f(w, \phi) = \sum_i \mathbb{E}_{q_i} \log p(x_i | t_i, w)$$

$$q_i(t_i) = q_\phi(t_i | x_i, \phi) = \mathcal{N}(m_i, \text{diag}(s_i^2))$$

$$f(w, \phi) = \sum_i \mathbb{E}_{q_\phi(t_i|x_i,\phi)} \log p(x_i | t_i, w)$$

$$\nabla_w f(w, \phi) = \nabla_w \sum_i \mathbb{E}_{q_\phi(t_i|x_i,\phi)} \log p(x_i|t_i, w) =$$

$$= \nabla_w \sum_i \int q_\phi(t_i|x_i, \phi) \log p(x_i|t_i, w) \, dt_i =$$

$$= \sum_i \int \nabla_w [q_\phi(t_i|x_i, \phi) \cdot \log p(x_i|t_i, w)] \, dt_i =$$

(if this is smoothe fuction)

$$= \sum_i \int q_\phi(t_i|x_i, \phi) \nabla_w \log p(x_i|t_i, w) \, dt_i =$$

$$= \sum_i \mathbb{E}_{q_\phi(t_i|x_i,\phi)} \nabla_w \log p(x_i|t_i, w) \approx$$

$$\approx \sum_i \nabla_w \log p(x_i | \hat{t}_i, w), \quad \hat{t}_i \sim q_\phi(t_i|x_i, \phi)$$

[approximate by sampling]

$$\nabla_w f(w,\phi) \approx \sum_{i=1}^{N} \nabla_w \log p(x_i \mid \hat{t}_i, w) \approx$$

$\hat{t}_i \sim q(t_i \mid x_i, \phi)$ — gradient of standart NN

$\hat{t}_i$ - sample

$$\approx \frac{N}{n} \sum_{s=1}^{n} \nabla_w \log p(x_{i_s} \mid \hat{t}_{i_s}, w)$$

Stochastic gradient of standart NN

$i_s \sim U\{1, \dots, N\}$

## Log derivative trick

$$\nabla_\phi f(w,\phi) = \nabla_\phi \sum_i \mathbb{E}_{q(t_i \mid x_i, \phi)} \log p(x_i \mid t_i, w) =$$

$$= \sum_i \int \nabla_\phi q(t_i \mid x_i, \phi) \log p(x_i \mid t_i, w)\, dt_i =$$

(only for a smoothe functions)

$$= \sum_i \int \frac{q(t_i \mid x_i, \phi)}{q(t_i \mid x_i, \phi)} \nabla_\phi q(t_i \mid x_i, \phi) \log p(x_i \mid t_i, w)\, dt_i =$$

$$\nabla \log g(\phi) = \frac{\nabla g(\phi)}{g(\phi)}$$

$$= \sum_i \int q(t_i \mid x_i, \phi) \nabla \log q(t_i \mid x_i, \phi) \log p(x_i \mid t_i, w)\, dt_i =$$

$$= \sum_i \mathbb{E}_{q(t_i \mid x_i, \phi)} \nabla \log q(t_i \mid x_i, \phi) \log p(x_i \mid t_i, w)\, dt_i$$

(log-derivative trick) → approximate with samples

## Reparametrization trick

$$\nabla_\phi \mathcal{L}(w, \phi) = \sum_i \nabla_\phi \mathbb{E}_{q(t_i|x_i,\phi)} \log p(x_i|t_i, w) =$$

$$= \sum_i \mathbb{E}_{q(t_i|x_i,\phi)} \nabla \log q(t_i|x_i,\phi) \log p(x_i|t_i,w) dt_i \quad (1)$$

$\log p(x_i|t_i,w)$ - very big negative number, because of net doesn't know new images, so variance of estimation $\nabla_\phi \mathcal{L}(w,\phi)$ will be huge.

$$t_i \sim q(t_i|x_i,\phi) = \mathcal{N}(m_i, \text{diag}(s_i^2))$$

$$t_i = \varepsilon_i \odot s_i + m_i = g(\varepsilon_i, x_i, \phi)$$

$$\varepsilon_i \sim p(\varepsilon_i) = \mathcal{N}(0, I)$$ ← deterministic function

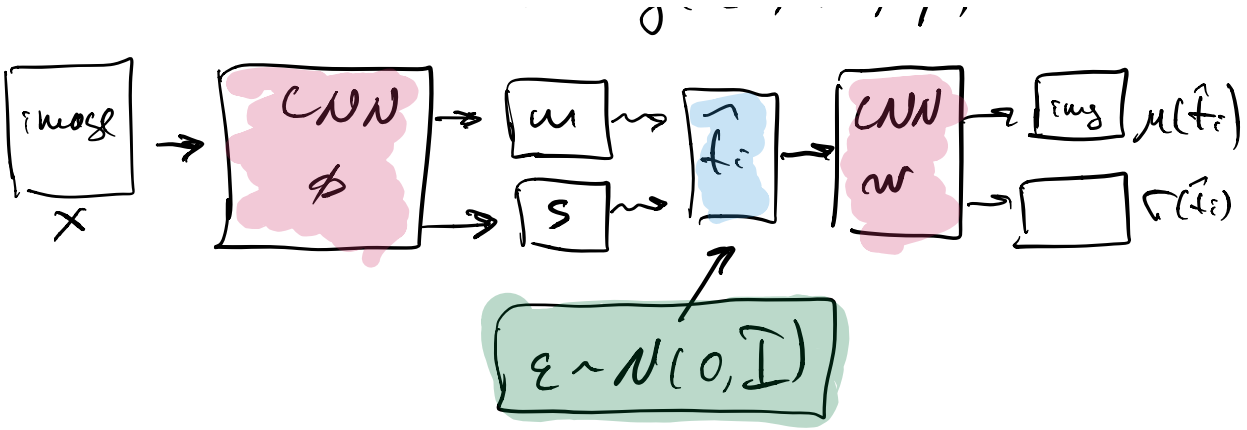
$$(1) = \sum_i \nabla_\phi \mathbb{E}_{p(\varepsilon_i)} \log p(x_i | g(\varepsilon_i, x_i, \phi), w) =$$

$$= \sum_i \int p(\varepsilon_i) \nabla_\phi \log p(x_i | g(\varepsilon_i, x_i, \phi), w) d\varepsilon_i =$$

$$= \sum_i \mathbb{E}_{p(\varepsilon_i)} \nabla_\phi \log p(x_i | g(\varepsilon_i, x_i, \phi), w)$$

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

$t_i = \varepsilon_i \odot s_i + m_i = g(\varepsilon_i, x_i, \phi)$




## Variational Autoencoder summary

- Infinite mixture of Gaussians
- To learn: EM + approximate q with Gaussians + stochastic variational inference
- Like plain autoencoder, but with noise and KL regularization
- Generates nice images

# Variational Dropout

## Learning with priors

### Ideal Bayesian learning

- $(X, Y)$ and probabilistic classifier $p(y|x,w)$
- define a reasonable prior $p(w)$
- Training stage

$$p(w|X,Y) = \frac{p(Y|X,w)\,p(w)}{\int p(Y|X,w)\,p(w)\,dw}$$

posterior

usually intractable

- Test stage

$$p(y^*|x^*, X, Y) = \int p(y^*|x^*,w)\,p(w|X,Y)\,dw$$

### Variational Inference

Inference Optimization

$p(w|X,Y)$ - original problem

$$q(w|\phi) = \arg\min_{\phi} D(q(w|\phi), p(w|X,Y))$$

variational distribution family

distance measure between 2 distributions

If $D(q(w|\phi), p(w|X, Y)) =$

$= KL[q(w|\phi) \| p(w|X, Y)]$

Then optimization problem is equivalent to maximizing ELBO (evidence lower bound)

$$L(\phi) = \int q(w|\phi) \log \frac{P(Y|X,w)p(w)}{q(w|\phi)} dw$$

$\downarrow$

$\max$

$\neq$

steel intractable but

Nice properties of ELBO

$$L(\phi) = \int q(w|\phi) \log \frac{P(Y|X,w)p(w)}{q(w|\phi)} dw$$

- We can use mini-batching for optimization since

$$\log P(Y|X,w) = \sum_{i=1}^{n} \log p(y_i|x_i, w)$$

- We can use MC estimates for computing stochastic gradient - especially effective when reparameterization trick is applicable

$$L(\phi) = \int r(\varepsilon) \log \frac{p(\Sigma \mid X, W(\varepsilon,\phi))\, p(W(\varepsilon,\phi))}{q(W(\varepsilon,\phi) \mid \phi)}\, d\varepsilon$$

- The richer is variational family $\{q(W|\phi)\ \phi \in \Phi\}$ the better we approximate true posterior
- Can split ELBO into data term and KL-term

$$L(\phi) = \underbrace{\int q(W|\phi) \log p(\Sigma|X,W)\, dW}_{\text{data term}} - \underbrace{KL(q(W|\phi) \,||\, p(W))}_{\text{regularizer}}$$

## Dropout as Bayesian procedure

- Popular regularization technique
- Injects noise to the weights or activations at each iteration during training
- The magnitude of noise is defined by dropout rate

## Gaussian drop out

We inject multiplicative Gaussian noise with mean 1 and variance $\alpha$

$$w_{ij} = \theta_{ij}\,\varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(\varepsilon_{ij} \mid 1, \alpha)$$

Then we compute stochastic gradient of log-likelihood given a noisified weight $W$

$$\text{stoch grad}_\theta \log p(Y \mid X, W) =$$

$$= \text{stoch grad}_\theta \log p(Y \mid X, \theta\hat{\varepsilon})$$

$$\hat{\varepsilon} \sim \mathcal{N}(\varepsilon \mid 1, \alpha I)$$

But we obtain exactly the same stochastic gradient if we consider

$$\text{stoch grad}_\theta \int \mathcal{N}(W \mid \theta, \alpha\theta^2) \log p(Y \mid X, W) dW$$

where

$$\mathcal{N}(W \mid \theta, \alpha\theta^2) = \prod_{ij} \mathcal{N}(w_{ij} \mid \theta_{ij}, \alpha\theta_{ij}^2)$$

$$\text{stoch grad}_\theta \int \mathcal{N}(\varepsilon \mid 1, \alpha) \log p(Y \mid X, W) dW =$$

= stoch grad$_\theta$ $\log p(Y|X,W)$
where $\hat{\varepsilon} \sim \mathcal{N}(\varepsilon|1, 2I)$

## Gaussian dropout optimizes

$$\int q(W|\theta,\alpha) \log p(Y|X,W)\,dW \to \max_\theta$$

Looks similar to the data term in ELBO with variational distribution

$$q(W|\theta,\alpha) = \mathcal{N}(W|\theta, \alpha\theta^2) = \prod_{ij} \mathcal{N}(w_{ij}|\theta_{ij}, \alpha\theta_{ij}^2)$$

But where is KL-term?

What if KL-term depends only on $\alpha$?

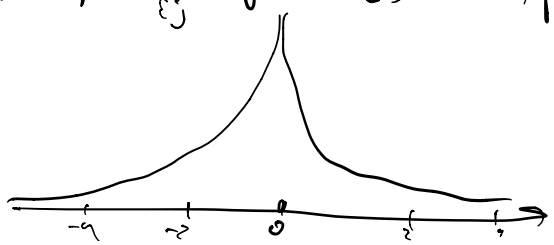
$$KL[q(W|\theta,\alpha)\|p(W)] = R(\alpha)$$

Then Gaussian dropout is exactly variational Bayesian inference!

## Log-uniform prior

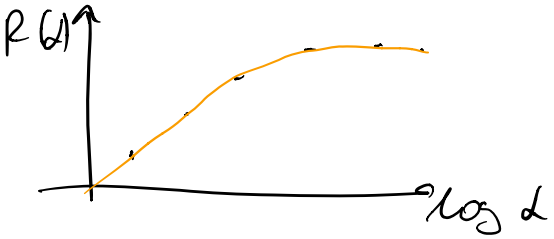
Such prior $p(w)$ exists and is very interpretable

$$p(W) = \prod_{ij} p(w_{ij}) \quad , p(w_{ij}) \propto \frac{1}{|w_{ij}|}$$



## KL-term

- It can be shown that with such choice of prior KL-term depends only on $\alpha$
- Although KL is intractable it can be approximated well a differentiable function




## Conclusion

- Dropout prevents overfitting
- It injects noise during training
- Gaussian dropout corresponds to variational Bayesian inference with log-uniform prior
- Now we may construct various generalizations of dropout

# Sparse variational dropout

## Variational dropout

- Gaussian dropout with fixed noise variance is equivalent to optimizing ELBO

$$\mathcal{L}(\theta) = \int q(w|\theta,\alpha) \log p(Y|X,w)\, dw - R(\alpha)$$

$$\max_{\theta}$$

where $R(\alpha)$ stands for KL-term

- Why not to optimize it w.r.t. noise variance $\alpha$ as well?

- This will only improve our variational approximation

$$\Downarrow$$

$$\mathcal{L}(\theta) = \int q(w|\theta,\alpha) \log p(Y|X,w)\, dw - R(\alpha)$$

$$\max \theta, \alpha$$

- Note it was not possible until we came with Bayesian interpretation of Gaussian dropout

Our variational approximation for the true posterior

$$q(w|\theta,\alpha) = \prod_{ij} \mathcal{N}(w_{ij}|\theta_{ij}, \alpha\theta_{ij}^2)$$

⇓ individual dropout rates

$$q(w|\theta,\alpha) = \prod_{ij} \mathcal{N}(w_{ij}|\theta_{ij}, \alpha_{ij}\theta_{ij}^2)$$

## Sparsification property

Remember that maximum of $R(\alpha)$ is achieved at infinity

We also show that

$$\alpha_{ij} \to +\infty \Rightarrow \theta_{ij} \to 0 \text{ and } \alpha_{ij}\theta_{ij}^2 \to 0$$

This means that

$$\lim_{\alpha_{ij} \to +\infty} q(w_{ij}|\theta_{ij}, \alpha_{ij}) = \delta(0)$$

we may remove the corresponding weight!

## Sparse variational dropout

- Assign log-uniform prior over the weight $p(w)$
- Fix variational family of distributions

$$q(w|\theta, A) = \prod_{ij} \mathcal{N}(w_{ij}|\theta_{ij}, \alpha_{ij}\theta_{ij}^2)$$

- Perform variational inference

$$L(\theta, A) = \int q(W|\theta, A) \log P(Y|X, W) dW - R(A)$$

$$\max_{\theta, A}$$

- Remove all weights whose $\alpha_{ij} >> 1$

Up to 33.3% of the weights become irrelevant without accuracy drop!

## Conclusions

- Bayesian dropout allows to remove high redundancy of modern DNN
- Variational Bayesian inference is highly scalable procedure that allows to optimize millions of variational parameters
- One of many examples of successful combination of Bayesian methods and deep learning

Quize: Categorical Reparametrization

with Gumbel - Softmax

3, 4, 8

read the paper

Week 6 5.11.2018

# Gaussian processes & Bayesian optimization

## Non parametric methods

### Parametric methods

① Define parametric model
$p(y \mid X, \theta)$

② Find best parameters using MAP estimation:

$$p(\theta \mid y, X) \rightarrow \max_{\theta}$$

Parametric methods: fixed number of parameters

Non-parametric: Number of parameters depends on dataset size

### K-nearest neighbors

$$y = \frac{1}{k} \sum_{i=1}^{k} y_{(i)}$$

### Nadaraya-Watson

$$y(x) = \sum_{i=1}^{N} w_i(x) y_i \quad ; \quad w_i(x) = \frac{K(x, x_i)}{\sum_{i=1}^{N} K(x, x_i)}$$

## Some kernels

$$k(x_1, x_2) = e^{-\frac{1}{2c^2}\|x_1 - x_2\|^2}$$

$$k(x_1, x_2) = [\![\, \|x_1 - x_2\| < h \,]\!]$$

P:

- limited complexity
- fast inference
- slow learning

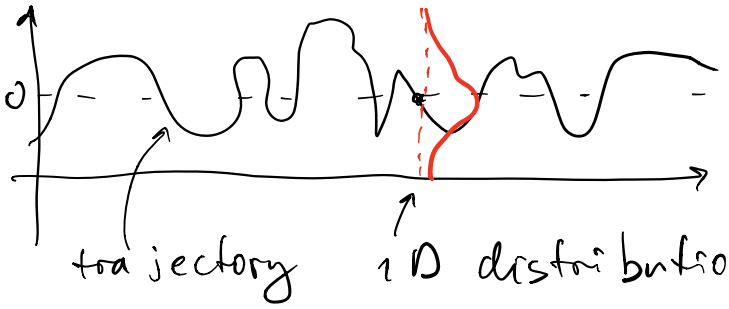
non P:

- arbitrary complex
- Need to process all data for prediction
- Learning: remember all data

## Gaussian processes

### Random process

For any $x \in \mathbb{R}^d$ assign random variable $f(x)$




### Gaussian process

Random process $f$ is Gaussian, if for any finite number points, their joint distribution is normal

$\forall n \in \mathbb{N}$, $\forall x_1, x_2, \dots x_n \in \mathbb{R}^d$

$$\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \sim \mathcal{N}$$

Parameters:

$\mathbb{E} f(x) = m(x)$

$Cov[f(x_1), f(x_2)] = K(x_1, x_2)$

Finally:

$$\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \sim \mathcal{N}\left( \begin{pmatrix} m(x_1) \\ \vdots \\ m(x_n) \end{pmatrix}, \begin{pmatrix} K(x_1, x_1) & \dots & K(x_1, x_n) \\ \vdots & & \vdots \\ K(x_n, x_1) & & K(x_n, x_n) \end{pmatrix} \right)$$

## Stationary process

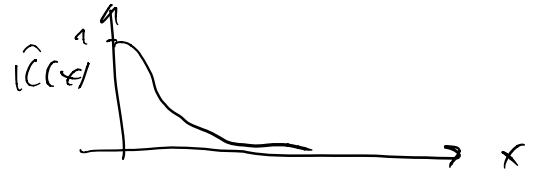
Random process $f$ is stationary, if its finite-dimentional distributions depend only on relative position of the points

Gaussian process is stationary, if

$m(x) = const$

$K(x_1, x_2) = \hat{K}(x_1 - x_2)$

Variance: $Var[f(x)] = \hat{K}(0)$



## kernel

RBF : $\hat{K}(x_1 - x_2) = \sigma^2 \exp\left(-\frac{(x_1 - x_2)^2}{2\ell^2}\right)$

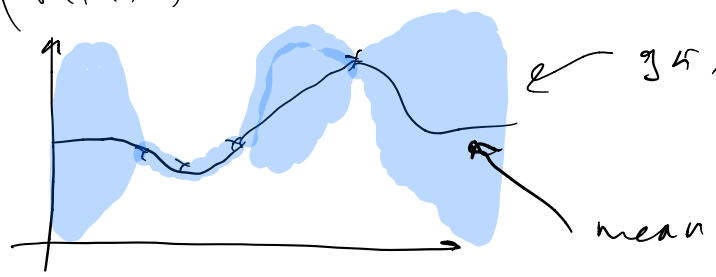
Rational Quadratic: $\hat{K}(x_1 - x_2) = \sigma^2 \left(1 + \frac{(x_1 - x_2)^2}{2\alpha\ell^2}\right)^{-\alpha}$

White noise : $\hat{K}(x_1 - x_2) = \sigma^2 \delta(x_1 - x_2)$

## GP for machine learning

Task

$\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \Rightarrow$ ~~$f(x) = ?$~~ $\Rightarrow p(f(x) \mid f(x_1) .. f(x_n)) = ?$




## Prediction

$f(x)$ is a Gaussian Process with stationary prior, $m(x) = 0$

$p(f(x) \mid f(x_1), \ldots f(x_n)) = \frac{p(f(x), f(x_1), .. f(x_n))}{p(f(x_1), .. f(x_n))} =$

$$= \frac{\mathcal{N}(f(x), f(x_1), \dots f(x_n) \mid 0, \hat{C})}{\mathcal{N}(f(x_1), \dots f(x_n) \mid 0, C)} \quad (=)$$

$$C = \begin{pmatrix} k(0) & \cdots & k(x_1 - x_n) \\ \vdots & & \\ k(x_n - x_1) & \cdots & k(0) \end{pmatrix}$$

$$\hat{C} = \begin{pmatrix} k(0) & k^T \\ k & C \end{pmatrix} \qquad k = \begin{pmatrix} k(x - x_1) \\ \vdots \\ k(x - x_n) \end{pmatrix}$$

$$(=) \; \mathcal{N}(f(x) \mid \mu, \sigma^2)$$

$$\mu = k^T C^{-1} f \qquad f = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$$

$$\sigma^2 = k(0) - k^T C^{-1} k$$

## Preprocessing

- For from data: $\mathbb{E}\, f(x) = 0$, $\mathrm{Var}\{f(x)\} = k(0)$
- Remove trend, seasonality
- Substract mean and normalize

## Derivation of main formula

$$p(f(x) \mid f(x_1), \dots f(x_n)) = \frac{p(f(x), f(x_1), \dots f(x_n))}{p(f(x_1), \dots f(x_n))} =$$

$$= \frac{\mathcal{N}\left(\begin{bmatrix} f^* \\ f \end{bmatrix} \mid 0, \begin{bmatrix} k(0) & k^T \\ k & C \end{bmatrix}\right)}{\mathcal{N}(f \mid 0, C)} \quad \propto$$

$$\propto \exp\left(-\frac{1}{2}\begin{bmatrix} f^* \\ f \end{bmatrix}^T \begin{bmatrix} k(0) & k^T \\ k & C \end{bmatrix}^{-1} \begin{bmatrix} f^* \\ f \end{bmatrix} + f^T C^{-1} f\right) =$$

$$= \exp\left(-\frac{1}{2}\begin{bmatrix} f^* & f^T \end{bmatrix} \begin{bmatrix} k(0) & k^T \\ k & C \end{bmatrix}^{-1} \begin{bmatrix} f^* \\ f \end{bmatrix} + f^T C^{-1} f\right) =$$

$$\begin{bmatrix} d & b^T \\ b & A \end{bmatrix}$$

$$= \exp\left(-\frac{1}{2}\left[\, d\,(f^*)^2 + 2 b^T f \cdot f^* + \text{const}\,\right]\right) \quad \propto$$

$$\propto \exp\left(-\frac{1}{2d^{-2}}\left[\, f^* + \frac{b^T f}{d}\,\right]^2\right) =$$

$$\mu = -\frac{b^T f}{d} \qquad \sigma^2 = d^{-1}$$

$$= \mathcal{N}(f^* \mid \mu, \sigma^2)$$

Let's find $b, d$

$$\begin{bmatrix} k(0) & k^T \\ k & C \end{bmatrix} \begin{bmatrix} d & b^T \\ b & A \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & I \end{bmatrix}$$

$$\begin{cases} k(0)\, d + k^T b = 1 \\ \end{cases} \Rightarrow k(0)\, d - k^T C^{-1} k d = 1$$

$$\begin{cases} k(0)\, b^T + K^T A = 0 \\ \underline{d\,K + C\,b = 0} \\ K\, b^T + C A = I \end{cases} \Rightarrow \quad b = -C^{-1} K d$$

$$d = \frac{1}{k(0) - K^T C^{-1} K}$$

$$\Rightarrow \quad b = -\frac{C^{-1} K}{k(0) - K^T C^{-1} K}$$

$$\Rightarrow \quad \sigma^2 = k(0) - K^T C^{-1} K$$

$$\mu = -\frac{b^T f}{d} =$$

## Nuance of GP

### Noisy observations

$\hat{f}(x) = f(x) + \varepsilon$ - independent Gaussian noise

$\varepsilon \sim \mathcal{N}(0, s^2)$

$m(x) = 0$

$\hat{k}(x_i - x_j) = k(x_i - x_j) + s^2 \mathbb{I}(x_i = x_j)$

## Kernel parameters

$$k(x_1, x_2) = \sigma^2 \exp\left(-\frac{(x_1 - x_2)^2}{2\ell^2}\right) + s^2 \mathbb{1}\{x_i = x_j\}$$

$$p(f(x_1), f(x_2), \dots f(x_n) \mid \sigma^2, \ell, s^2) \to \max_{\sigma^2, \ell, s^2}$$

$$\mathcal{N}(f(x_1), f(x_2), \dots f(x_n) \mid \varphi, C) \to \max_{\sigma^2, \ell, s^2}$$

optimize with gradient ascent

## Classification

$y \in \{-1, 1\}$

Latent process: $f(x)$

Class probabilities: $p(y \mid f) = \frac{1}{1 + \exp(-yf)}$

Training:

- Approximate latent process from data

$$p^*(f(x)) = p(f(x) \mid y_1(x_1), \dots y_n(x_n))$$

- Compute predictions

$$\pi(x) = \int p(y(x) \mid f(x)) \cdot p^*(f(x))\, df(x)$$

## Inducing inputs

$$\mu = k^T C^{-1} f \qquad \leftarrow O(n^3)$$

$$\sigma^2 = k(0) - k^T C^{-1} k$$

Idea:
- Replace dataset with small number of points (like SVM)
- Predict using those points

Speed:
- Precomputing $O(m^2 n)$
- Mean $O(m)$
- Variance $O(m^2)$

## Bayesian optimization

## Black-box optimization

$$f(x) \to \max_x$$

Gradient is known:
- Gradient descent with restarts

Gradient is unknown:
- Numerically estimate gradient
- Grid search / Random search

Cases:
- x geographic coordinates, f(x) - amount of oil
  1 sample = $10^6$ $
- x hyperparameters of NN, f(x) - objective
  1 sample = 10 hours

- x drug, f(x) - effectiveness against disease
  1 sample = 2 month, $10k, life of a rat

Goal: Optimize with minimum number of trials

Surrogate model $\hat{f} \approx f$
- Approximates true function
- Cheap to evaluate

Acquisition function: $\mu(x)$
- Estimate profit for optimization
- Uses surrogate model

## Surrogate model

$$\hat{f} \approx f$$

Should model arbitrary complex functions
=> Nonparametric method

Profitable to estimate uncertainty
=> Gaussian process

## Acquisition function

Exploration:
Search in regions with high uncertainty

Exploitation:

Search in regions with high estimated value

## Maximum probability of improvement (MPI)

Current best value: $f^*$

$$\mu(x) = \mathbb{P}(\hat{f}(x) \geq f^* + \varepsilon) = \Phi\left(\frac{\mathbb{E}\hat{f}(x) - f^* - \varepsilon}{\mathrm{Var}[\hat{f}(x)]}\right)$$

Works well if value of maximum is known

## Upper confidence bound (UCB)

$$\mu(x) = \mathbb{E}\hat{f}(x) + \gamma \mathrm{Var}[\hat{f}(x)]$$

## Expected improvement (EI)

$$\mu(x) = \mathbb{E}\max(f(x) - f^*, 0) =$$

$$= \mathrm{Var}[\hat{f}(x)] \cdot [z \cdot \Phi(z) + \phi(z)]$$

$$z = \frac{\mathbb{E}\hat{f}(x) - \mu(x)}{\mathrm{Var}[\hat{f}(x)]}$$

most widely used

Example:

Start with few points

while not converged:

1. Train GP
2. Find maximum of $\mu(x)$ using e.g. gradient ascent
3. Evaluate function at maximum of $\mu(x)$

## Random search vs Gaussian processes

| RS | GP |
|---|---|
| + Parallelizable | - Hard to parallelize experiments |
| - Needs many more points for high dimensions | + Require less points on average |
| Any function | Each evaluation is expensive |

**Train BM << Train NN**

## Applications of Bayesian optimization

### Hyperparameter tuning

Network parameters
- Number of layers
- Layer sizes
- Dropout on/off

- Batch normalization on/off

Training parameters
- Learning rate
- Momentum

Usually finds better optima than when tuned by hand

Honest comparison with other methods in research

## Discrete and continuous variables

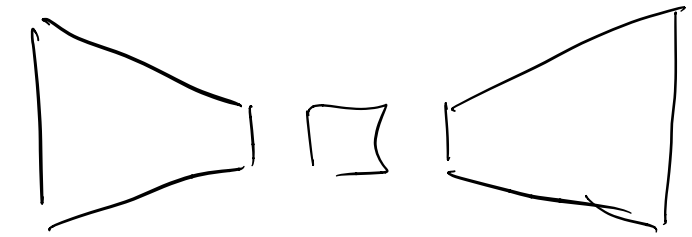
- Treat discrete variables as continuous when fitting process
- Maximize $\mu(x)$ for each possible value of discrete variables
- Multi-armed bandit: all variables are discrete

Final Task 28.12.2018
6.01.2019

(1) 25 samples from VAE

3 points

vae




(2) Search procedure.