Bayesian Framework

Jin Young Choi

Seoul National University

Artificial Intelligence

인공지능의 핵심 요소

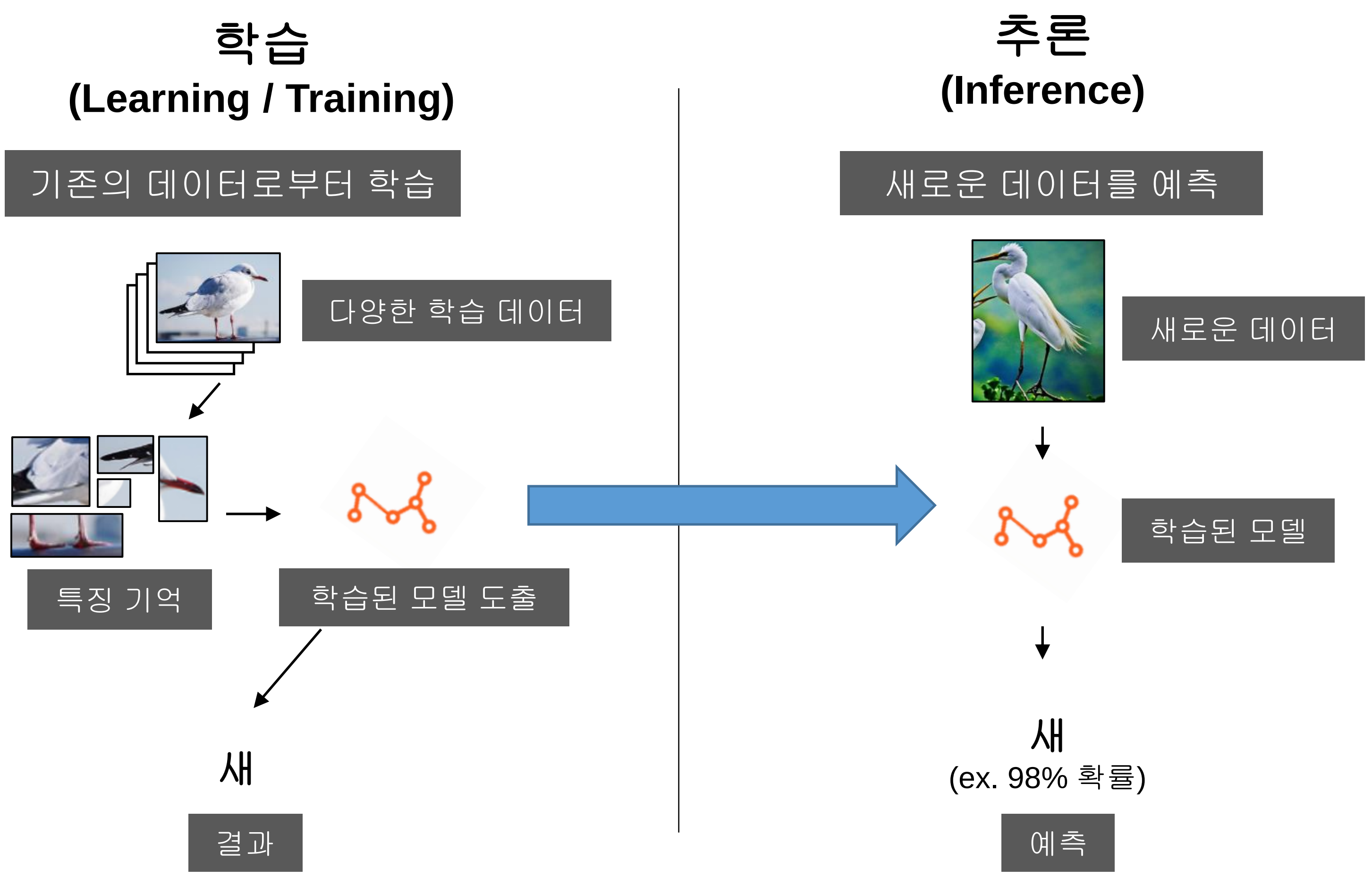
학습
(Learning / Training)

경험, 관측, 예시로부터
지식을 기억하는 과정

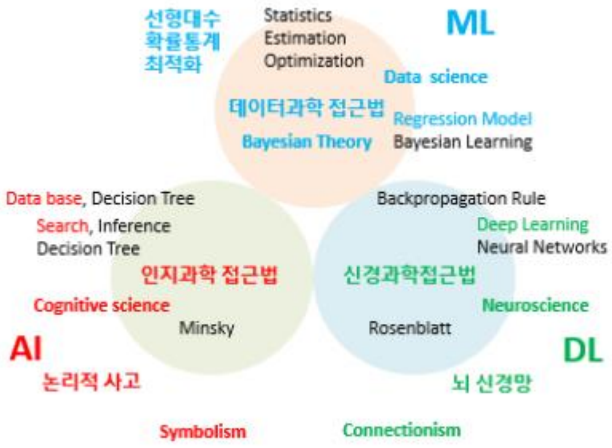
추론
(Inference)

기억된 지식에 기반하여
질문 또는 의문에 대한 답을 발견

Artificial Intelligence



Artificial Intelligence



인지과학 접근법: 기호주의(Symbolism)

학습

대상 객체의 명칭과 특징들을 데이터베이스(DB)에 저장
의사결정나무(Decision Tree) 구조로도 저장할 수 있음

특징

객체	날개	비행	지느러미	몸매	…
매	큼	가능	없음	날렵함	…
돌고래	작음	못함	존재	뚱뚱함	…

명칭

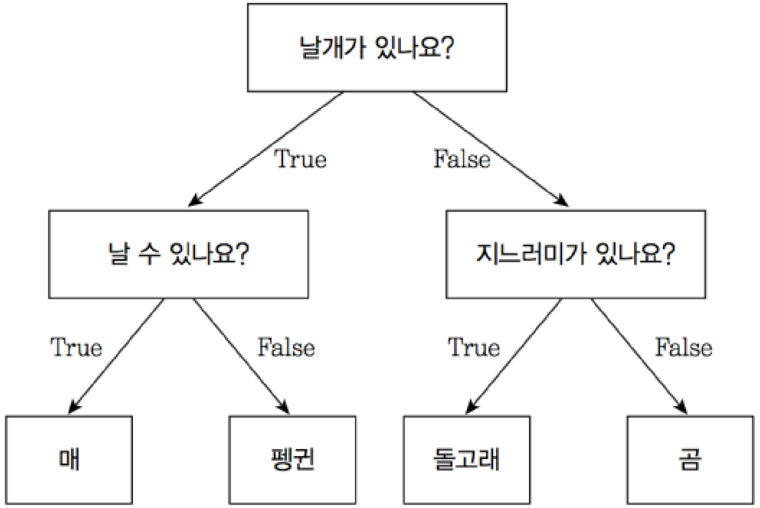


표 형태의 데이터베이스 (DB) 예시

의사결정나무(Decision Tree)의 예시

Artificial Intelligence

인지과학 접근법: 기호주의(Symbolism)

추론

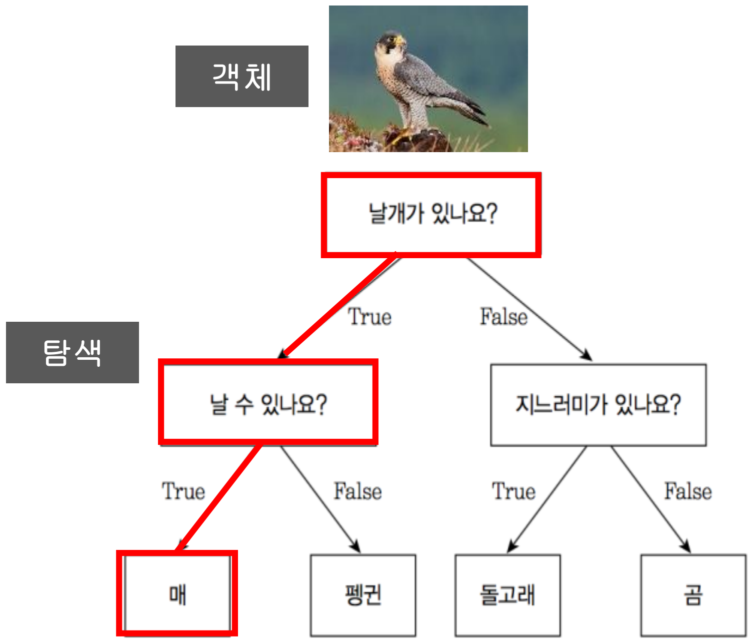
주어진 객체로부터 저장된 지식을 탐색하여
가장 특징이 비슷한 객체의 명칭을 찾아내는 것

특징

명칭

객체	날개	비행	지느러미	몸매	…
매	큼	가능	없음	날렵함	…
돌고래	작음	못함	존재	뚱뚱함	…

특징 매칭 탐색
Binary Search
Relational Data Base



스무고개식 탐색
Breath First Search
Depth First Search

Artificial Intelligence

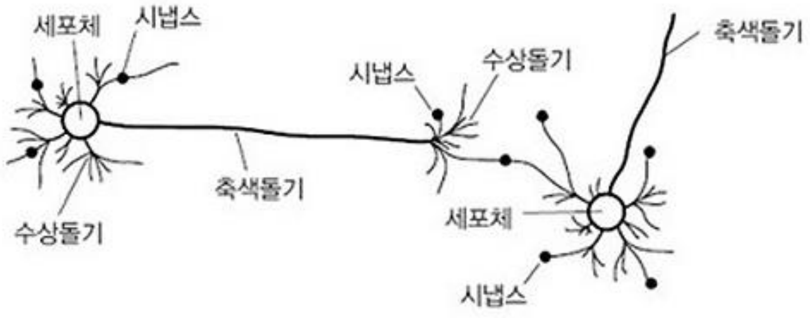
신경과학 접근법: 연결주의(Connectionism)

학습 & 추론

신경세포(뉴런)의 연결 모델을 모방한 인공신경망 모델
뉴런과 뉴런 사이의 시냅스의 연결 가중치 값 (W)이 변동되면서 학습

$$o = f(W, x) = P_W(A|B)$$

학습: 사후 확률이 출력되도록 학습 후,
추론: 새로운 특징이 입력되면, 어떤 객체인지 판단

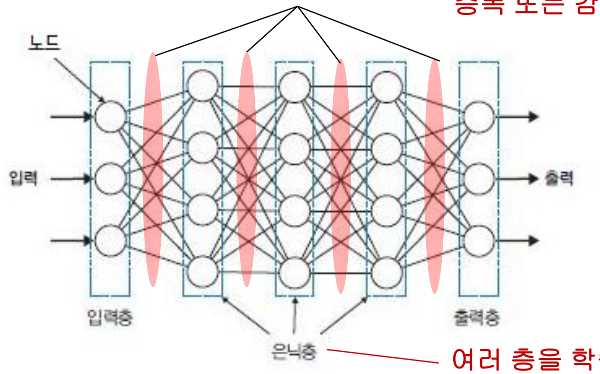


생물학적 신경망

생물학적 신경망	인공신경망
세포체(뉴런)	노드
수상돌기	입력
시냅스	가중치
축색돌기	출력

가중치 (W)

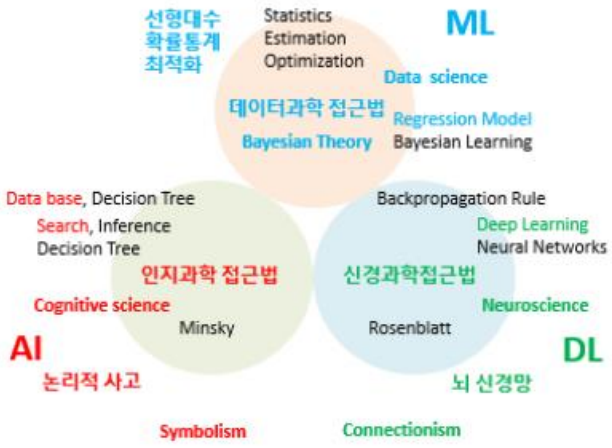
가중치에 의해 신호가
증폭 또는 감소



여러 층을 학습하는 경우를
딥러닝이라고 함

인공신경망

Artificial Intelligence



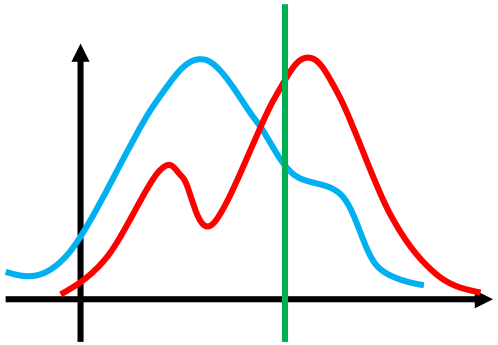
데이터과학 접근법: 확률통계법

학습

베이지안 이론(**Bayesian Theory**): 머신러닝의 대표적 이론
조건부 확률의 확률 분포를 추정하여 기억

> 다음의 조건부 확률이 주어졌을 때,
> 암환자 여부에 따른 각각의 검사결과 분포를 학습

- 암환자(***A***)가 ***B***라는 특징을 가질 확률: $P(B|A)$
  (=암환자 (A)에서 ***B***라는 특징이 발견될 확률)
- 정상인(***A***ᶜ)이 ***B***라는 특징을 가질 확률: $P(B|AC)$
  (= 정상인(***A***ᶜ) 에서 ***B***라는 특징이 발견될 확률)



$\boldsymbol{B = b}$

- 빨간색: 암환자에서 발견되는 ***B*** 특징의 분포 : $P(B|A)$
- 파란색: 정상인에서 발견되는 ***B*** 특징의 분포 : $P(B|Ac)$

- 암에 걸릴 확률: $P(A)$
- 암에 안걸릴 확률: $P(A^C)$=1−$P(A)$

데이터과학 접근법: 확률통계법

추론

주어진 조건부 확률의 분포로부터
베이스 룰에 의해 사후 확률을 구할 수 있음

암환자가 **B**라는 특징을 가질 확률

암에 걸릴 확률

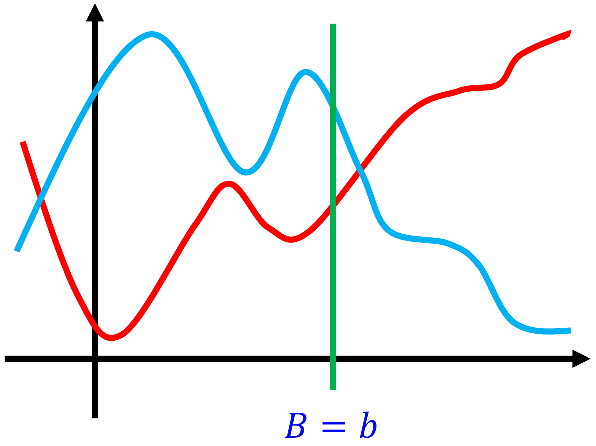
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

앞의 학습과정에서 기억된 확률 분포를 사용

암 검사에서 **B**라는 특징이 발견되었을 때, 실제로 그 사람이 암환자일 확률

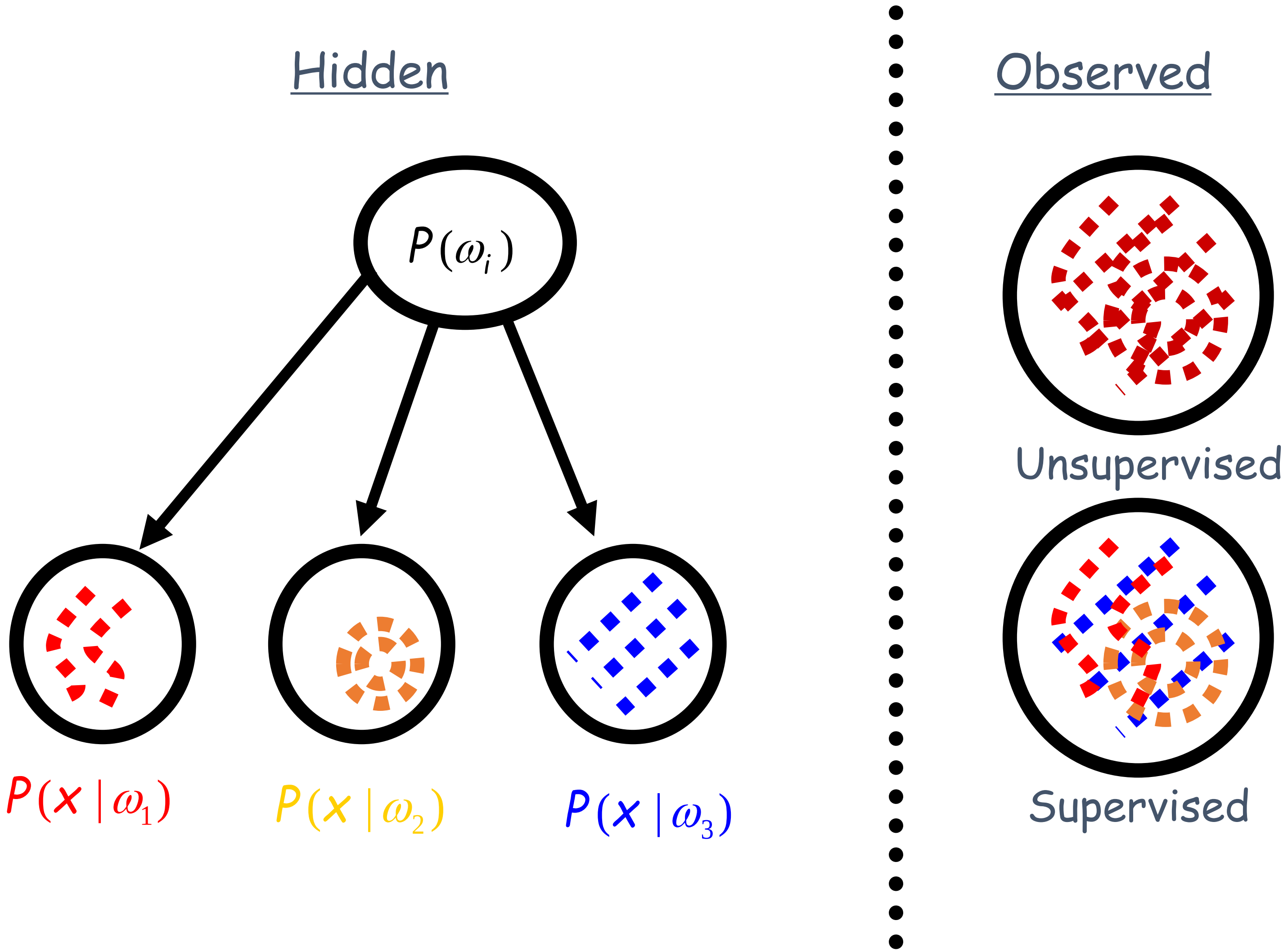
암 검사에서 **B**라는 특징이 발견될 확률

베이스 법칙
(Bayes Rule)



- 빨간색: 특징값(B)에 따른 암환자일 확률분포
- 파란색: 특징값(B)에 따른 정상인일 확률분포

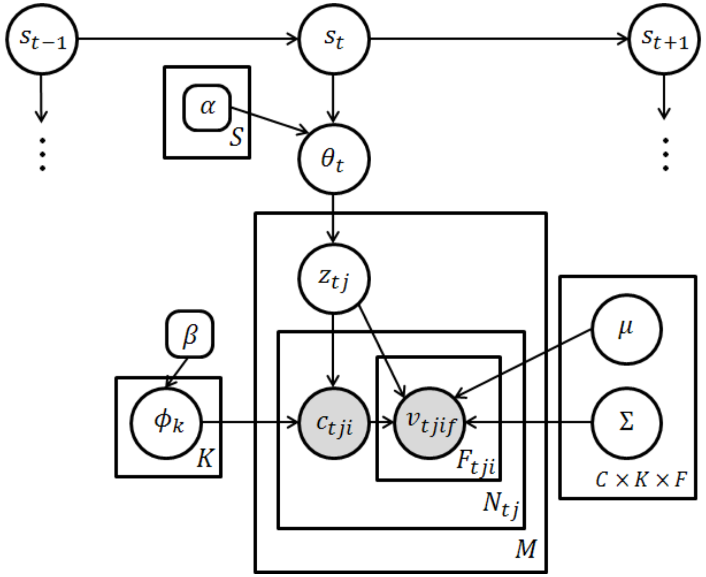
Supervised/Unsupervised Learning



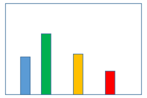
Bayesian networks: Traffic Pattern Analysis

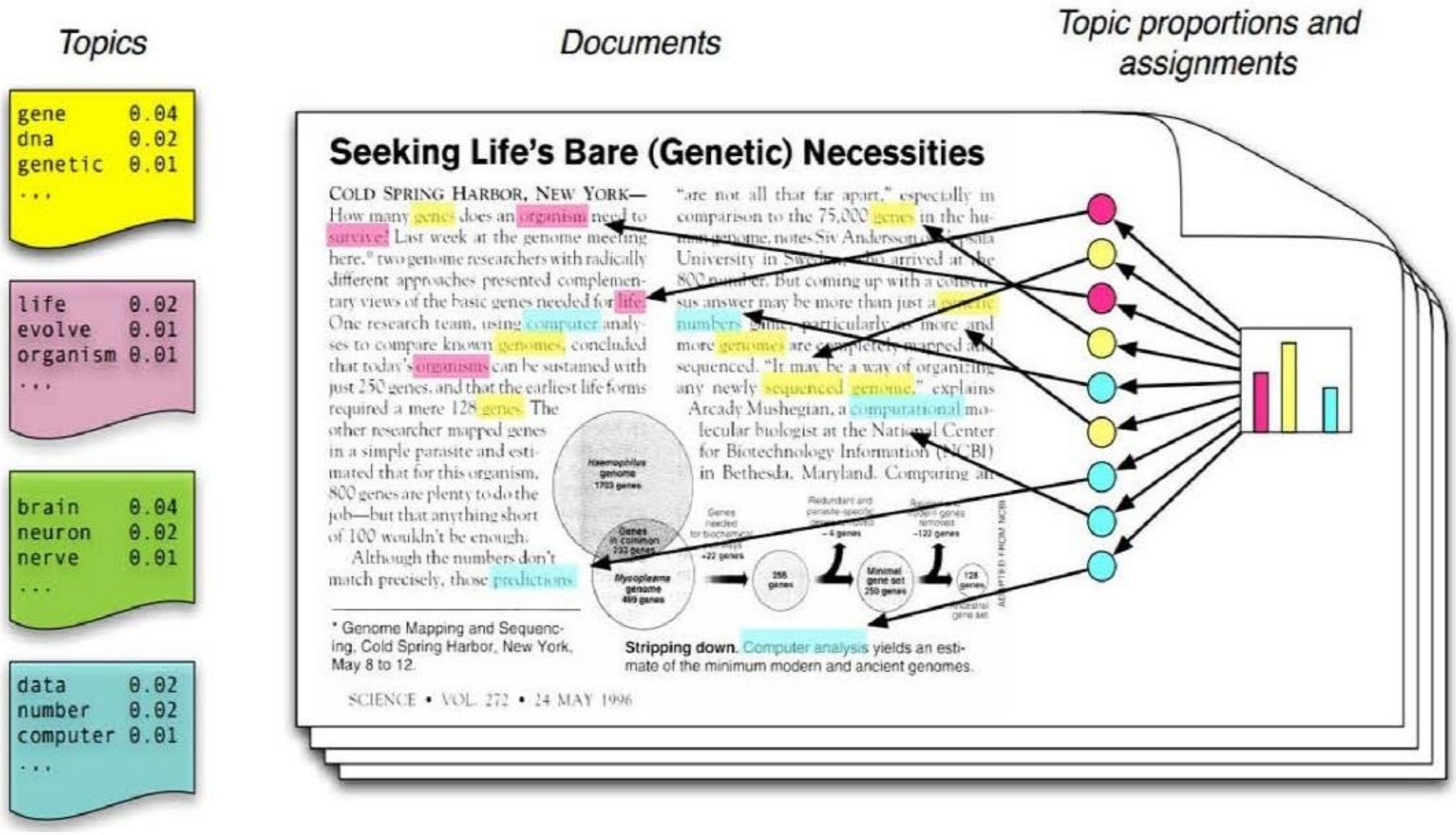
- Surveillance in crowded scenes

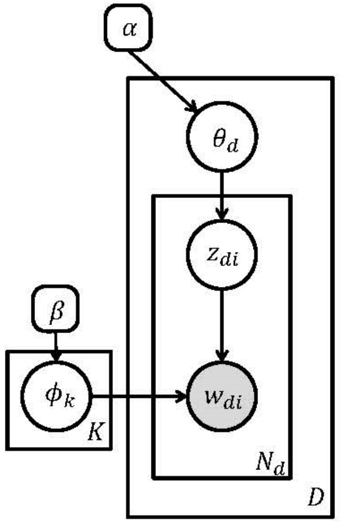
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$



Bayesian networks (Topic Modelling)



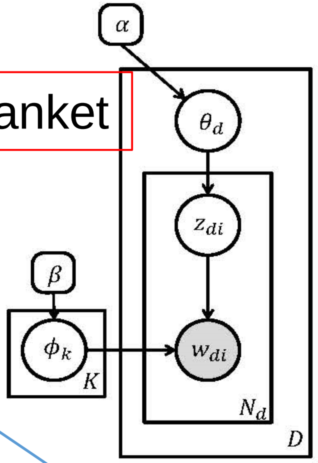




Bayesian Learning for Unsupervised Learning

- Markov Chain Monte Carlo (MCMC) framework

Markov Blanket



$p(z|w, \phi_k, \theta_k)$

Posteriors

$$p(\theta_d|z,\alpha) = \frac{\overbrace{p(z|\theta_d)}^{Multinomial}\ \overbrace{p(\theta_d|\alpha)}^{Dirichlet}}{p(z|\alpha)}$$

$$= Dir(\theta_d|\boxed{h_\theta(d,\cdot)} + \alpha),$$

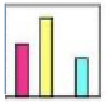
$$p(\phi_k|z,w,\beta) = Dir(\phi_k|\boxed{h_\phi(k,\cdot)} + \beta).$$

$$h_\theta(d,k) = \sum_{i=1}^{N_d} \delta\left[z_{di} - k\right]$$

$$h_\phi(k,v) = \sum_{d=1}^{D}\sum_{i=1}^{N_d} \delta\left[w_{di} - v\right]\delta\left[z_{di} - k\right];$$

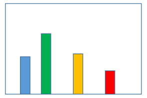
w: 1, 3, 2, 3, 3, 5, 4, 1, 6
z : 1, 2, $\boxed{2}$, 2, 1, 1, 2, 1, 2
h_θ(d,2): 5

w: 1, 3, 2, 3, 3, 5, 4, 1, 6
z : 1, 2, $\boxed{2}$, 2, 1, 1, 2, 1, 2
h_ϕ(1,3): 1
h_ϕ(2,3): 2

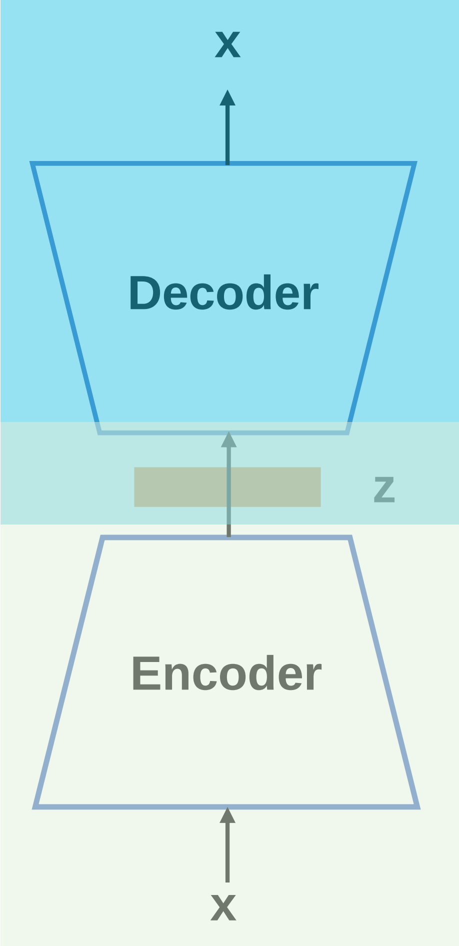


$$\hat{\theta}_d(k) = E[\theta_d(k)|h_\theta(d,\cdot)+\alpha] = \frac{h_\theta(d,k)+\alpha(k)}{\sum_{k=1}^{K}\left[h_\theta(d,k)+\alpha(k)\right]},$$

$$\hat{\phi}_k(v) = E[\phi_k(v)|h_\phi(k,\cdot)+\beta] = \frac{h_\phi(k,v)+\beta(v)}{\sum_{v=1}^{V}\left[h_\phi(k,v)+\beta(v)\right]}.$$



Variational Auto-encoder (VAE)



Reconstruction Loss

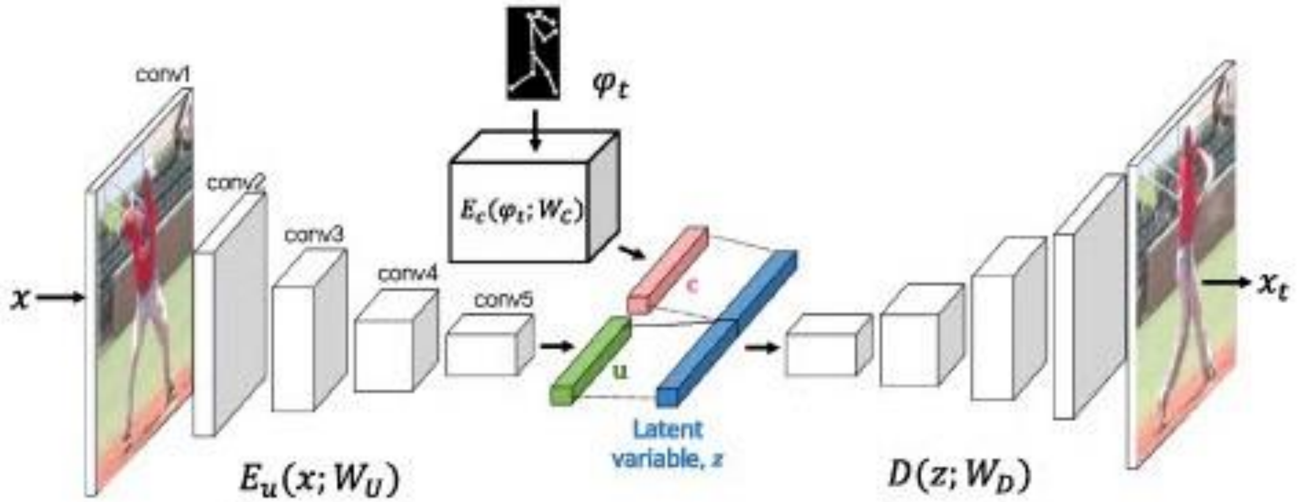
$$Loss = -E_{q_\varphi(z|x)} log P_\theta(x|z) + D_{KL}(q_\varphi(z|x)||P_\theta(z))$$

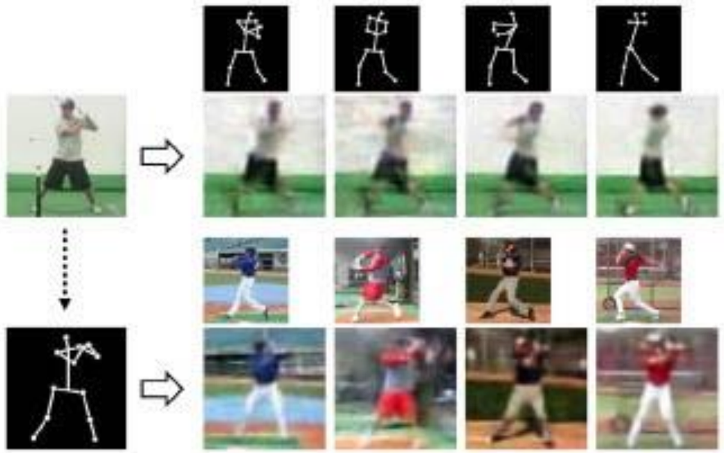
Variational Inference

$p_\theta(x|z)$: a multivariate Gaussian (real-valued data)

a Bernoulli (binary-valued data)

Pose Transformer





$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{ref} + \mathcal{L}_{pose} + \mathcal{L}_{id}.$$

$$\mathcal{L}_{ref} = -\mathbb{E}_{q_\phi(z|x_a^k, \varphi_a^k)}[\log p_\theta(x_a^k|z)] + D_{KL}\left(q_\phi(z|x_a^k, \varphi_a^k) \parallel p_\theta(z)\right).$$

$$\mathcal{L}_{pose} = -\mathbb{E}_{q_\phi(z|x_a^k, \varphi_a^l)}[\log p_\theta(x_a^l|z)] + D_{KL}\left(q_\phi(z|x_a^k, \varphi_a^l) \parallel p_\theta(z)\right) + \lambda_u \cdot D_{KL}\left(q_\phi(u|x_a^l) \parallel q_\phi(u|x_a^k)\right).$$

$$\mathcal{L}_{id} = -\mathbb{E}_{q_\phi(z|x_b^{k'}, \varphi_a^k)}[\log p_\theta(x_b^{k'}|z)] + D_{KL}\left(q_\phi(z|x_b^{k'}, \varphi_a^k) \parallel p_\theta(z)\right) + \lambda_c \cdot D_{KL}\left(q_\phi(c|\varphi_b^{k'}) \parallel q_\phi(c|\varphi_a^k)\right)$$

Supervised/Unsupervised Learning

- Supervised Learning
  - Labeled Deep learning
  - Labeled Density Estimation (Parametric)
- Un/Semi/Self-supervised Learning
  - Clustering
  - Bayesian Network Learning
  - Variational Auto-Encoder (VAE)
  - Active Learning (Uncertainty)
- Background Techniques
  - Entropy (Uncertainty)
  - Cross-Entropy, K-L Divergence
  - Bayesian Decision, Bayes Rule
  - Parametric Density Estimation (MLE, Bayesian Learning)
  - Non-parametric Density Estimation (EM, MCMC)

Information

- Discrete random variable X is defined in the sample set Ψ

$$\Psi = \{x_k | k = 0, \pm 1, \ldots, \pm K\}$$

- Event $X = x_k$ occurs with probability $p_k = P(X = x_k)$

- **Information** ≡ surprise ≡ uncertainty
  The amount of information of the event is related to the *inverse* of the probability of occurrence. That is, the lower the probability p_k is, the more "surprise" there is, and the more "information".

$$I(x_k) = \log(\frac{1}{p_k}) = -\log p_k$$

내일도 지구가 회전한다 $p_k = 1$: 정보(×), surprise(×)
내일 미국이 북한을 공격한다 $p_k \ll 1$: 정보(O), surprise(O)

Information

- base=2 ⇒ 정보단위 bits
- base=e ⇒ 정보단위 nats
- 32 bit : 한 code의 정보는 $I(x_k) = -\log(\frac{1}{2^{32}}) = 32$

① $I(x_k) = 0$ for $p_k = 1$
② $I(x_k) \geq 0$ for $0 \leq p_k \leq 1$
③ $I(x_k) \geq I(x_i)$ for $p_k \leq p_i$

- Entropy : a measure of the *average amount of information conveyed per message*, i.e., expectation of Information

$$H(X) = E[I(X)] = \sum_{k=-K}^{K} p_k I(x_k) = -\sum_{k=-K}^{K} p_k \log p_k$$

Information

- Maximum entropy : when p_k is equiprobable.

$$0 \leq H(X) \leq -\sum_{k=-K}^{K} \frac{1}{2K+1} log \frac{1}{2K+1} = \log(2K+1)$$

- $H(X) = 0$ for an event that $p_k = 1$ o/w $p_k = 0$

- Theorem (Gray 1990): Relative entropy (or Kullback – Leibler divergence)

Discrete: $D_{p\|q} = \sum_k p_k \log(\frac{p_k}{q_k}) \geq 0$

where p_k is probability mass ftn. (pmf), q_k is reference pmf

Continuous: $D_{p\|q} = \int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$

where $p(x)$ is probability density ftn. (pdf), $q(x)$ is reference pdf.

Information

- Relative entropy (or Kullback – Leibler divergence) for neural networks

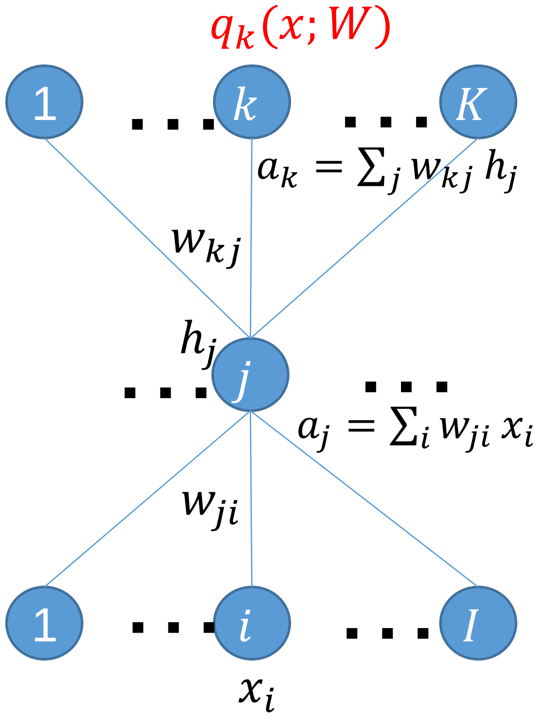
$$D_{p\|q}(\mathrm{X}; W) = \sum_k \sum_{x\in\mathrm{X}} p_k(x) \log\left(\frac{p_k(x)}{q_k(x;W)}\right) = \sum_k [\sum_{x\in\mathrm{X}} p_k(x) \log p_k(x) - \sum_{x\in\mathrm{X}} p_k(x) \log q_k(x;W)]$$

- Cross entropy for one-hot classification by deep learning (softmax activation)

$$C_{p\|q}(\mathrm{X}; W) = \sum_k [-\sum_{x\in\mathrm{X}} p_k(x) \log q_k(x;W)]$$
$$= \sum_k [-E_{p_k(x)} \log q_k(x;W)]$$

- Cross entropy for multi-label classification by deep learning(sigmoid activation)

$$C_{p\|q}(\mathrm{X}; W) = \sum_k [-\sum_{x\in\mathrm{X}} [p_k(x) \log q_k(x;W) + (1-p_k(x)) \log(1-q_k(x;W))]]$$
$$= \sum_k [-E_{p_k(x)} \log q_k(x;W) - E_{1-p_k(x)} \log(1-q_k(x;W))]$$



Backpropagation Learning Rule

- For multi-label classification (ex, output: 0110100), sigmoid activation function is used and the loss is defined by the cross entropy loss function: $E(w) = -\sum_k^K [t_k \log o_k(x,w) + (1-t_k)\log(1-o_k(x,w))]$, where

  • $o_k = \sigma(a_k) = \frac{1}{1+e^{-a_k}}$. Then find $\frac{\partial E}{\partial a_k}$.

Sol.)

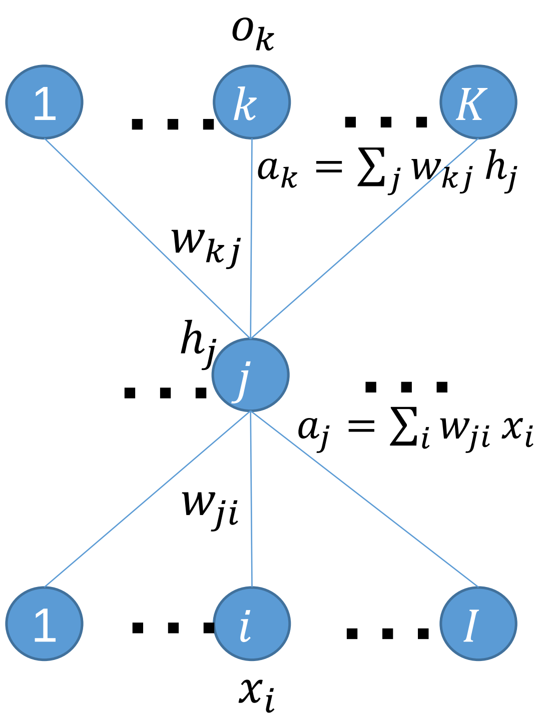
$$\frac{\partial E}{\partial a_k} = \frac{\partial E}{\partial o_k}\frac{\partial o_k}{\partial a_k}.$$

$$\frac{\partial o_k}{\partial a_k} = \sigma(a_k)(1-\sigma(a_k)) = o_k(1-o_k).$$

$$\begin{aligned}\frac{\partial E}{\partial a_k} &= -t_k\frac{1}{o_k}\frac{\partial o_k}{\partial a_k} - (1-t_k)\frac{-1}{1-o_k}\frac{\partial o_k}{\partial a_k}\\ &= -t_k\frac{1}{o_k}o_k(1-o_k) - (1-t_k)\frac{-1}{1-o_k}o_k(1-o_k)\\ &= -t_k(1-o_k) + (1-t_k)o_k = o_k - t_k = -(t_k - o_k) = -\delta_k\end{aligned}$$

$$\frac{\partial E_d}{\partial w_{kj}} = \frac{\partial E_d}{\partial a_k}\frac{\partial a_k}{\partial w_{kj}} = \frac{\partial E_d}{\partial a_k} h_j$$

$$\Delta w_{kj} = -\eta\frac{\partial E_d}{\partial w_{kj}} = \eta\delta_k h_j$$



J. Y. Choi. SNU

Backpropagation Learning Rule

- For multi-class classification (ex, [0 0 0 1 0 0]), the softmax activation function is used and the loss is defined by the cross entropy loss function: $E(w) = -\sum_i^K t_i \log(o_i(x,w))$, where $o_k(x,w) = \frac{e^{a_k}}{\sum_j e^{a_j}}$. The target value $t_k \in \{0,1\}$ is labelled by 1 hot vector. Then find $\frac{\partial \mathrm{E}}{\partial a_k}$.

  Sol.)

$$\frac{\partial E_n}{\partial a_k} = \frac{\partial}{\partial a_k}\left(-\sum_i^K t_i \log\left(\frac{e^{a_i}}{\sum_j e^{a_j}}\right)\right)$$

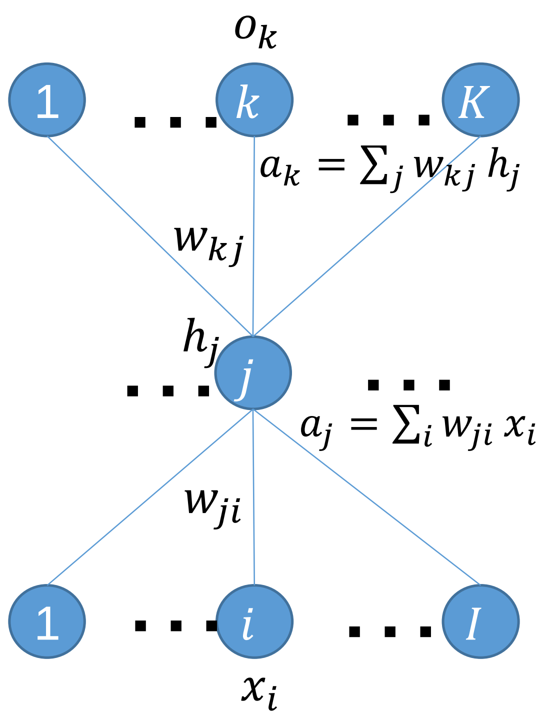
$$= \frac{\partial}{\partial a_k}\left(-\sum_i^K [t_i \log(e^{a_i}) - t_i \log(\textstyle\sum_j e^{a_j})]\right)$$

$$= \frac{\partial}{\partial a_k}\left(-\sum_i^K [t_i a_i - t_i \log(\textstyle\sum_j e^{a_j})]\right) = -t_k + \sum_i t_i \frac{e^{a_k}}{\sum_j e^{a_j}}$$

$$= -t_k + \frac{e^{a_k}}{\sum_j e^{a_j}} \sum_i t_i = o_k - t_k = -(t_k - o_k) = -\delta_k$$

$$\frac{\partial E_d}{\partial w_{kj}} = \frac{\partial E_d}{\partial a_k}\frac{\partial a_k}{\partial w_{kj}} = \frac{\partial E_d}{\partial a_k} h_j$$

$$\Delta w_{kj} = -\eta \frac{\partial E_d}{\partial w_{kj}} = \eta \delta_k h_j$$



J. Y. Choi. SNU

Mutual Information

- Conditional Entropy (조건부 불확실성의 량)

  Y 가 관측되고 난 후의 X 의 정보기대치 (Entropy)

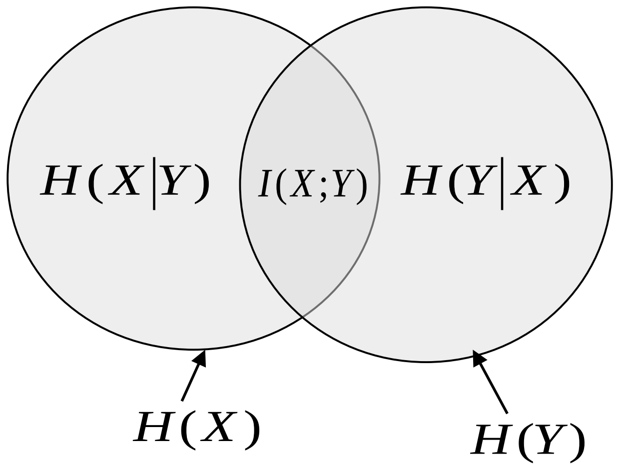
  Y 와 연관이 있는 X 의 정보는 제외

- Theorem (Gray 1990)

  $H(X|Y) = H(X,Y) - H(Y)$

  $0 \leq H(X|Y) \leq H(X)$

$\leftarrow p(x|y) = \dfrac{p(x,y)}{p(y)}$



- Joint Entropy

  $H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log \underbrace{p(x,y)}$

  → Joint probability mass(or density) function

Mutual Information

- Mutual Information: Output Y 의 관측에 의해 알 수 있는 X 의 uncertainty (정보)

$$I(X;Y) = H(X) - H(X|Y)$$

$$= H(X) + H(Y) - H(X,Y)$$

$$= -\sum_{x \in X} p(x) \log(p(x)) - \sum_{y \in Y} p(y) \log(p(y))$$

$$+ \sum_{x \in X} \sum_{y \in Y} p(x,y) \log(p(x,y))$$

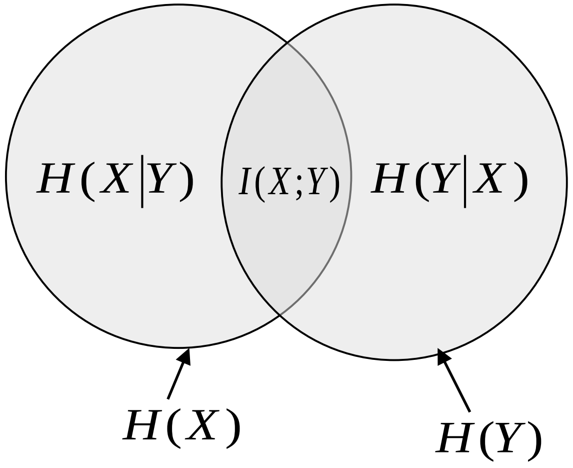
$$= \sum_{x \in X} \sum_{y \in Y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

$$p(x) = \sum_{y \in Y} p(x,y)$$

$$p(y) = \sum_{x \in X} p(x,y)$$

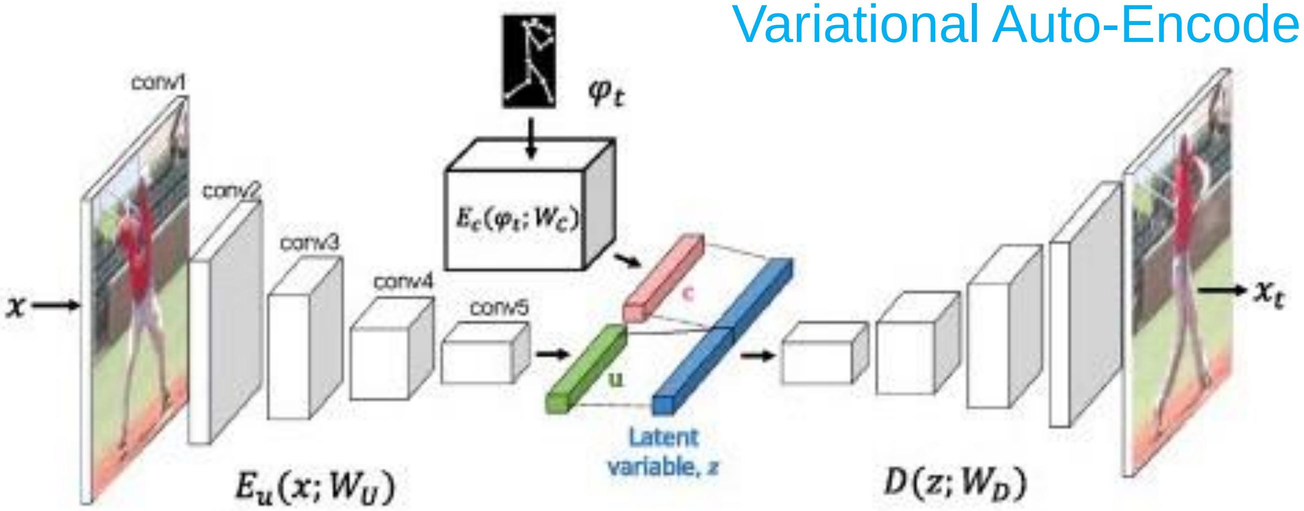
- KL-divergence & Independence ?

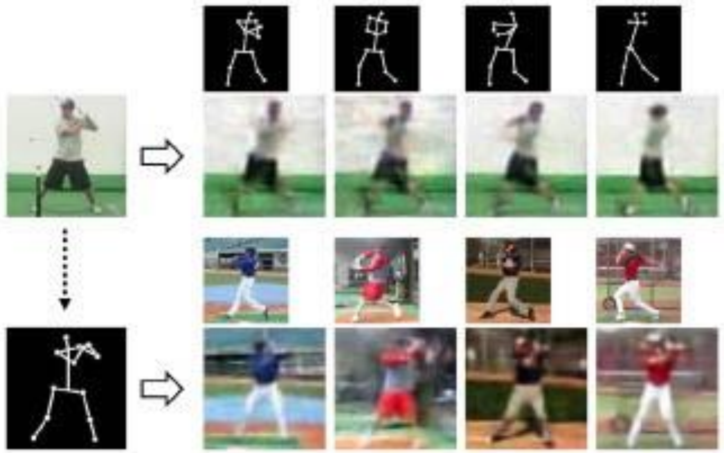
$$H(X) = I(X,X)$$



Pose Transformer

Variational Auto-Encoder (VAE)





$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{ref} + \mathcal{L}_{pose} + \mathcal{L}_{id}.$$

$$\begin{aligned}\mathcal{L}_{ref} = & -\mathbb{E}_{q_\phi(z|x_a^k, \varphi_a^k)}[\log p_\theta(x_a^k|z)] \\ & + D_{KL}\left(q_\phi(z|x_a^k, \varphi_a^k) \parallel p_\theta(z)\right).\end{aligned}$$

$$\begin{aligned}\mathcal{L}_{pose} = & -\mathbb{E}_{q_\phi(z|x_a^k, \varphi_a^l)}[\log p_\theta(x_a^l|z)] \\ & + D_{KL}\left(q_\phi(z|x_a^k, \varphi_a^l) \parallel p_\theta(z)\right) \\ & + \lambda_u \cdot D_{KL}\left(q_\phi(u|x_a^l) \parallel q_\phi(u|x_a^k)\right).\end{aligned}$$

$$\begin{aligned}\mathcal{L}_{id} = & -\mathbb{E}_{q_\phi(z|x_b^{k'}, \varphi_a^k)}[\log p_\theta(x_b^{k'}|z)] \\ & + D_{KL}\left(q_\phi(z|x_b^{k'}, \varphi_a^k) \parallel p_\theta(z)\right) \\ & + \lambda_c \cdot D_{KL}\left(q_\phi(c|\varphi_b^{k'}) \parallel q_\phi(c|\varphi_a^k)\right)\end{aligned}$$

Exercise

- 프리미어리그에서 아스널과 토트넘이 경기를 하고 있다. TV를 보며 마음 껏 떠들 수 있도록 자리가 마련된 치킨집의 식객 30명과 바로 옆 삽겹살 집 식객 60명이 응원전을 펼치고 있다. 치킨집 사람들에게 어느 팀을 응원하는지 물었을 때 토트넘 10명, 아스널을 20명이 응원한다고 답했다.삼겹살 집에서는 각 팀을 몇 명이 응원하고 있는지 확인하지 못했다.
- 치킨 집에서 ‘토트넘’을 응원한다는 답변에 담긴 정보량(Information Gain)은?

Exercise

- 프리미어리그에서 아스널과 토트넘이 경기를 하고 있다. TV를 보며 마음 껏 떠들 수 있도록 자리가 마련된 치킨집의 식객 30명과 바로 옆 삼겹살 집 식객 60명이 응원전을 펼치고 있다. 치킨집 사람들에게 어느 팀을 응원하는지 물었을 때 토트넘 10명, 아스널을 20명이 응원한다고 답했다.삼겹살 집에서는 각 팀을 몇 명이 응원하고 있는지 확인하지 못했다.
- 치킨 집에서 ‘토트넘’을 응원한다는 답변에 담긴 정보량(Information Gain)은?

- 일목요연하게 내용 정리.

Exercise

- 프리미어리그에서 아스널과 토트넘이 경기를 하고 있다. TV를 보며 마음 껏 떠들 수 있도록 자리가 마련된 치킨집의 식객 30명과 바로 옆 삼겹살 집 식객 60명이 응원전을 펼치고 있다. 치킨집 사람들에게 어느 팀을 응원하는지 물었을 때 토트넘 10명, 아스널을 20명이 응원한다고 답했다.삼겹살 집에서는 각 팀을 몇 명이 응원하고 있는지 확인하지 못했다.
- 치킨 집에서 '토트넘'을 응원한다는 답변에 담긴 정보량(Information Gain)은?

- 일목요연하게 내용 정리.
- $Information = -\log p(X = x)$

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60 - n)$명

Exercise

- 프리미어리그에서 아스널과 토트넘이 경기를 하고 있다. TV를 보며 마음 껏 떠들 수 있도록 자리가 마련된 치킨집의 식객 30명과 바로 옆 삼겹살 집 식객 60명이 응원전을 펼치고 있다. 치킨집 사람들에게 어느 팀을 응원하는지 물었을 때 토트넘 10명, 아스널을 20명이 응원한다고 답했다.삼겹살 집에서는 각 팀을 몇 명이 응원하고 있는지 확인하지 못했다.
- 치킨 집에서 '토트넘'을 응원한다는 답변에 담긴 정보량(Information Gain)은?

- 일목요연하게 내용 정리.
- $Information = -\log P(x)$

$P(X = \text{토트넘} | Y = \text{치킨집}) = 1/3$

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

Exercise

- 프리미어리그에서 아스널과 토트넘이 경기를 하고 있다. TV를 보며 마음 껏 떠들 수 있도록 자리가 마련된 치킨집의 식객 30명과 바로 옆 삼겹살 집 식객 60명이 응원전을 펼치고 있다. 치킨집 사람들에게 어느 팀을 응원하는지 물었을 때 토트넘 10명, 아스널을 20명이 응원한다고 답했다.삼겹살 집에서는 각 팀을 몇 명이 응원하고 있는지 확인하지 못했다.
- 치킨 집에서 '토트넘'을 응원한다는 답변에 담긴 정보량(Information Gain)은?

- 일목요연하게 내용 정리.

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

- $Information: I(x) = -\log P(x)$

$P(X = \text{토트넘} | Y = \text{치킨집}) = 1/3$

$I(X = \text{토트넘} | Y = \text{치킨집}) = \log 3$

Exercise

- 치킨집에서 토트넘 응원하는 경우를 $X = 0$, 아스널 응원하는 경우를 $X = 1$이라 할 때 우측 표가 지닌 X의 엔트로피는?

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60 - n)$명

Exercise

- 치킨집에서 토트넘 응원하는 경우를 $X = 0$, 아스널 응원하는 경우를 $X = 1$이라 할 때 우측 표가 지닌 X 의 엔트로피는?

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60 - n)$명

- $Entropy$: $H(X) = -\sum_{x \in X} p(x) \log p(x)$

Exercise

- 치킨집$(Y = 0)$에서 토트넘 응원하는 경우를 $X = 0,$ 아스널 응원하는 경우를 $X = 1$이라 할 때 우측 표가 지닌 X 의 엔트로피는?

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60 - n)$명

- $Entropy$: $H(X) = -\sum_x p(x) \log p(x)$

- $H(x|Y = 0) = -\sum_x p(x|Y = 0) \log p(x|Y = 0)$

- $H(x|Y = 0) = -\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3}$

Exercise

- KL-Divergence의 의미를 생각할 때 각 음식점 에서 두팀을 응원할 확률분포간의 KL-divergence, 즉 $D_{P(X|Y=0)||P(X|Y=1)}$을 최소로 하는 n 값을 구하시오.

$D_{P(Y=0)||P(Y=1)}$을 최소로한다는 것은 각 음식점에서 두팀을 응원할 확률 분포가 같게 된다는 의미이다.

즉, $P(X|Y=0) = P(X|Y=1)$

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

Exercise

- KL-Divergence의 의미를 생각할 때 각 음식점 에서 두 팀을 응원할 확률분포 간의 KL-divergence, 즉 $D_{P(X|Y=0)||P(X|Y=1)}$을 최소로 하는 n 값을 구하시오.

$D_{P(Y=0)||P(Y=1)}$을 최소로한다는 것은 각 음식점에서 두팀을 응원할 확률 분포가 같게 된다는 의미이다.

즉, $P(X|Y=0) = P(X|Y=1)$

$$\frac{1}{3} = \frac{n}{60}, \qquad \frac{2}{3} = \frac{60-n}{60} \rightarrow n = 20$$

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

Exercise

- $D_{P(X|Y=0)||P(X|Y=1)}$을 최소로 하는 n 값을 최적화 방법으로 구하시오.

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

$$n^* = \underset{n}{\operatorname{argmin}}\, D_{P(X|Y=0)||P(X|Y=1)}$$

Exercise

- $D_{P(X|Y=0)||P(X|Y=1)}$을 최소로 하는 n 값을 최적화 방법으로 구하시오.

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

$$n^* = \underset{n}{\operatorname{argmin}}\, D_{P(X|Y=0)||P(X|Y=1)}$$

$$D_{P(X|Y=0)||P(X|Y=1)} = \sum_x P(X=x|Y=0)\log\frac{P(X=x|Y=0)}{P(X=x|Y=1)}$$

Exercise

- $D_{P(X|Y=0)||P(X|Y=1)}$을 최소로 하는 n 값을 최적화 방법으로 구하시오.

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

$$n^* = \underset{n}{\operatorname{argmin}}\, D_{P(X|Y=0)||P(X|Y=1)}$$

$$D_{P(X|Y=0)||P(X|Y=1)} = \sum_x P(X=x|Y=0)\log\frac{P(X=x|Y=0)}{P(X=x|Y=1)}$$
$$= 1/3\log\frac{\frac{1}{3}}{\frac{n}{60}} + 2/3\log\frac{\frac{2}{3}}{\frac{60-n}{60}}$$

Exercise

- $D_{P(X|Y=0)||P(X|Y=1)}$을 최소로 하는 n 값을 최적화 방법으로 구하시오.

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

$$n^* = \underset{n}{\operatorname{argmin}}\, D_{P(X|Y=0)||P(X|Y=1)}$$

$$D_{P(X|Y=0)||P(X|Y=1)} = \sum_x P(X=x|Y=0)\log\frac{P(X=x|Y=0)}{P(X=x|Y=1)}$$

$$= 1/3\log\frac{\frac{1}{3}}{\frac{n}{60}} + 2/3\log\frac{\frac{2}{3}}{\frac{60-n}{60}}$$

$$\frac{d}{dn}D_{P(X|Y=0)||P(X|Y=1)} = \frac{n}{60}\left(-\frac{20}{n^2}\right) + \frac{60-n}{60}\left(\frac{40}{(60-n)^2}\right) = -\frac{1}{3n} + \frac{2}{3(60-n)} = \frac{-60+3n}{3n(60-n)} = 0 \rightarrow n = 20$$

Exercise

- $D_{P(X|Y=0)||P(X|Y=1)}$을 이용하여 구한 n이 참값이라고 할 때, 위 표가 지닌 응원팀(X)과 음식점(Y)에 관한 Mutual Information I(X, Y)을 수식을 사용하지 않고 개념적으로 구하시오. 그리고 수식을 사용하여 구하여 개념적으로 구한 경우와 비교하시오.

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

Exercise

- $D_{P(X|Y=0)||P(X|Y=1)}$을 이용하여 구한 n이 참값이라고 할 때, 위 표가 지닌 응원팀(X)과 음식점(Y)에 관한 Mutual Information I(X, Y)을 수식을 사용하지 않고 개념적으로 구하시오. 그리고 수식을 사용하여 구해보고 개념적으로 구한 경우와 비교하시오.

응원팀과 음식점은 서로 독립이다. 그 이유는 음식점에 따라 두 팀을 응원하는 확률 분포가 달라지지 않기 때문이다. 따라서 Mutual Information은 0 이다.

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

Exercise

- $D_{P(X|Y=0)||P(X|Y=1)}$을 이용하여 구한 n이 참값이라고 할 때, 위 표가 지닌 응원팀(X)과 음식점(Y)에 관한 Mutual Information $\mathrm{I}(X,Y)$을 수식을 사용하지 않고 개념적으로 구하시오. 그리고 수식을 사용하여 구해보고 개념적으로 구한 경우와 비교하시오.

응원팀과 음식점은 서로 독립이다. 그 이유는 음식점에 따라 두 팀을 응원하는 확률 분포가 달라지지 않기 때문이다. 따라서 Mutual Information은 0 이다.

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

$$\mathrm{I}(X,Y) = \sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} = \sum_{x,y} p(x|y)p(y)\log\frac{p(x|y)p(y)}{p(x)p(y)}$$

$$= \frac{1}{3}\frac{1}{3}\log\frac{\frac{11}{33}}{\frac{11}{33}} + \frac{2}{3}\frac{1}{3}\log\frac{\frac{21}{33}}{\frac{21}{33}} + \frac{1}{3}\frac{2}{3}\log\frac{\frac{12}{33}}{\frac{12}{33}} + \frac{2}{3}\frac{2}{3}\log\frac{\frac{22}{33}}{\frac{22}{33}} = 0\,.$$

Exercise

- Mutual Information과 Conditional Entropy의 관계에 의하여 $H(X|Y)$을 구하시오.

	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

Exercise

- Mutual Information과 Conditional Entropy의 관계에 의하여 $H(X|Y)$을 구하시오.

$$\mathrm{I}(X,Y) = \mathrm{H}(X) - \mathrm{H}(X|Y) = 0$$

X \ Y	치킨집	삼겹살집
토트넘 응원자	10	n명
아스널 응원자	20	$(60-n)$명

$$\mathrm{H}(\mathrm{X}|Y) = \mathrm{H}(X) = -\sum_x p(x)\log p(x) = -\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3}$$

Bayesian Decision

- Question:
  - There live two kinds of fishes in a lake: tuna or salmon.
  - If you catch a fish by fishing, is the fish likely to be tuna or salmon?

Bayesian Decision

- We have experienced that salmon has been caught in 70% and tuna in 30%.
- What is the next fish likely to be?

Bayesian Decision

- If other types of fish are irrelevant:

$$p(\omega = \omega_1) + p(\omega = \omega_2) = 1,$$

ω is random variable, ω_1 and ω_2 denote salmon and tuna.

- Probabilities reflect our prior knowledge obtained from past experience.
- Simple Decision Rule:
  - Make a decision without seeing the fish.
  - Decide ω_1 if $p(\omega = \omega_1) > p(\omega = \omega_2)$

    ω_2 otherwise.

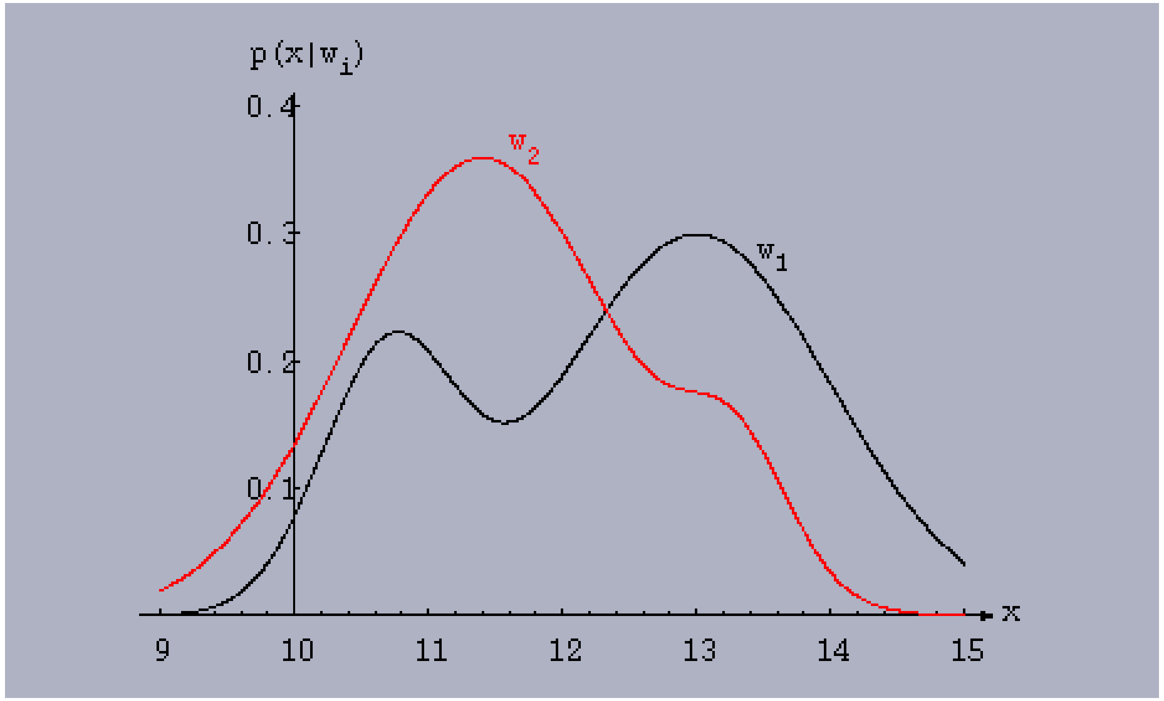
Bayesian Decision

- In general, we will have some features and more information.
- Feature: lightness measurement = x
  - Different fish yields different lightness readings (x is a random variable)

Bayesian Decision

- Define
  - $p(x|\omega_i)$= Class Conditional Probability Density
  - The difference between $p(x|\omega_1)$ and $p(x|\omega_2)$ describes the difference in lightness between tuna and salmon.

Bayesian Decision



- Hypothetical class-conditional probability
- Density functions are normalized (area under each curve is 1.0)

Bayesian Decision

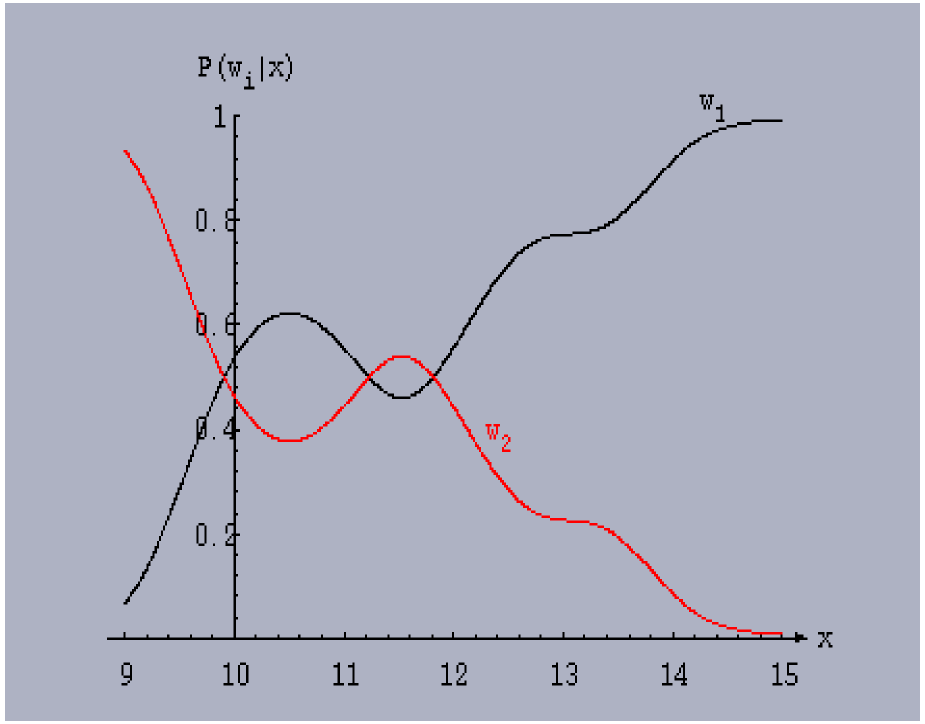
- Suppose that we know
  - The prior probabilities $p(\omega_1)$ and $p(\omega_2)$
  - The conditional densities $p(x|\omega_1)$ and $p(x|\omega_2)$
  - Measure lightness of a fish $= x$
- What is the category of the fish with lightness of x ?
- The probability that the fish has category of ω_i is $p(\omega_i|x)$.

Bayes Rule

- $p(\omega_i|x) = \frac{p(x|\omega_i)p(\omega_i)}{p(x)}$,

  where $p(x) = \sum_j p(x, \omega_j) = \sum_j p\left(x\middle|\omega_j\right)p(\omega_j)$.
- $Posterior = \frac{Likelihood * Prior}{Evidence}$
- $p(x|\omega_i)$ is called the ***likelihood*** of ω_i with respect to x.
  - The ω_i category for which $p(x|\omega_i)$ is large is more "likely" to be the true category
- $p(x)$ is the ***evidence***
  - How frequently is a pattern with feature value x observed.
  - Scale factor that the posterior probabilities sum to 1.

Bayes Rule



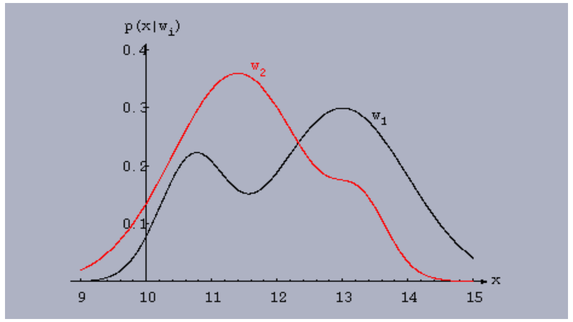
- Posterior probabilities for the particular priors $p(\omega_1) = 2/3$ and $p(\omega_2) = 1/3$. At every x the posteriors sum to 1.

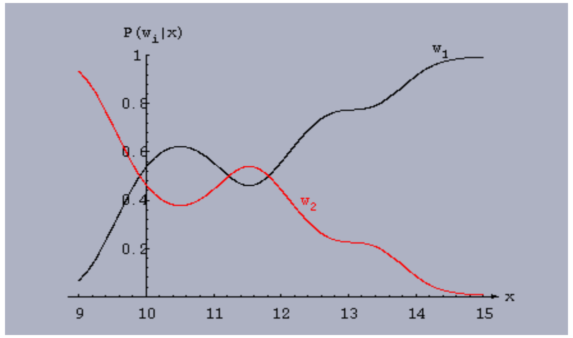
Bayes Decision (Minimal probability error)

- Likelihood Decision:
  - ω_1 : $if\ p(x|\omega_1) > p(x|\omega_2)$
  - ω_2 : *otherwise*
- Posteriori Decision:
  - ω_1 : $if\ p(x|\omega_1)p(\omega_1) > p(x|\omega_2)p(\omega_2)$
  - ω_2 : *otherwise*
- Decision Error Probability
  - $p(error|x) = \min(p(\omega_1|x), p(\omega_2|x))$

where the decision error is given by

$$p(error|x) = \begin{cases} p(\omega_2|x) & if\ we\ decide\ \omega_1\ for\ \ \omega_2 \\ p(\omega_1|x) & if\ we\ decide\ \omega_2\ for\ \ \omega_1 \end{cases}$$





Exercise

- 지금까지 샌디에고 만에서 잡힌 연어의 20%가 40cm 이하였고, 잡힌 농어의 30%가 40cm 이하였다. 또한 연어와 숭어의 잡힌 비율은 7:3이었다. 잡힌 물고기의 크기가 40cm 이하인데, 연어와 숭어 둘 중 하나로 보인다. 연어인지, 농어인지 판단해 보시오.

Exercise

- 지금까지 샌디에고 만에서 잡힌 연어의 20%가 40cm 이하였고, 잡힌 농어의 30%가 40cm 이하였다. 또한 연어와 숭어의 잡힌 비율은 7:3이었다. 잡힌 물고기의 크기가 40cm 이하인데, 연어와 숭어 둘 중 하나로 보인다. 연어인지, 농어인지 판단해 보시오.

- Sol.
  - ✓ (힌트) 문장의 수치에 해당하는 내용과 질문을 수식으로 표현해 보세요.
  - ✓ 연어: $X = 0$, 농어: $X = 1$, 크기: Y.
  - ✓ $P(Y \leq 40cm|\, X = 0) = 0.2$, $P(Y \leq 40cm|\, X = 1) = 0.3$, $P(X = 0) = 0.7$, $P(X = 1) = 0.3$

Exercise

- 지금까지 샌디에고 만에서 잡힌 연어의 20%가 40cm 이하였고, 잡힌 농어의 30%가 40cm 이하였다. 또한 연어와 숭어의 잡힌 비율은 7:3이었다. 잡힌 물고기의 크기가 40cm 이하인데, 연어와 숭어 둘 중 하나로 보인다. 연어인지, 농어인지 판단해 보시오.

- Sol.
  - ✓ (힌트) 문장의 수치에 해당하는 내용과 질문을 수식으로 표현해 보세요.
  - ✓ 연어: $X = 0$, 농어: $X = 1$, 크기: Y.
  - ✓ $P(Y \leq 40cm| \, X = 0) = 0.2$, $P(Y \leq 40cm| \, X = 1) = 0.3$, $P(X = 0) = 0.7$, $P(X = 1) = 0.3$
  - ✓ 질문: posteriori: $P(X = 0 \,|\, Y \leq 40cm) =?$, $P(X = 1 \,|\, Y \leq 40cm) =?$

$$p(\omega_i|x) = \frac{p(x|\omega_i)p(\omega_i)}{p(x)}$$

Exercise

- 지금까지 샌디에고 만에서 잡힌 연어의 20%가 40cm 이하였고, 잡힌 농어의 30%가 40cm 이하였다. 또한 연어와 숭어의 잡힌 비율은 7:3이었다. 잡힌 물고기의 크기가 40cm 이하인데, 연어와 숭어 둘 중 하나로 보인다. 연어인지, 농어인지 판단해 보시오.

- Sol.
  - ✓ (힌트) 문장의 수치에 해당하는 내용과 질문을 수식으로 표현해 보세요.
  - ✓ 연어: $X = 0$, 농어: $X = 1$, 크기: Y.
  - ✓ $P(Y \leq 40cm | X = 0) = 0.2$, $P(Y \leq 40cm | X = 1) = 0.3$, $P(X = 0) = 0.7$, $P(X = 1) = 0.3$
  - ✓ 질문: posteriori: $P(X = 0 \,|\, Y \leq 40cm) =?$, $P(X = 1 \,|\, Y \leq 40cm) =?$
  - ✓ $P(X = 0 \,|\, Y \leq 40cm) = \frac{P(Y \leq 40cm | X=0) P(X=0)}{P(Y \leq 40cm)} = \frac{0.2 \times 0.7}{0.2 \times 0.7 + 0.3 \times 0.4} = 0.54$

Exercise

- 지금까지 샌디에고 만에서 잡힌 연어의 20%가 40cm 이하였고, 잡힌 농어의 30%가 40cm 이하였다. 또한 연어와 숭어의 잡힌 비율은 7:3이었다. 잡힌 물고기의 크기가 40cm 이하인데, 연어와 숭어 둘 중 하나로 보인다. 연어인지, 농어인지 판단해 보시오.

- Sol.
  - ✓ (힌트) 문장의 수치에 해당하는 내용과 질문을 수식으로 표현해 보세요.
  - ✓ 연어: $X = 0$, 농어: $X = 1$, 크기: Y.
  - ✓ $P(Y \le 40cm| \, X = 0) = 0.2$, $P(Y \le 40cm| \, X = 1) = 0.3$, $P(X = 0) = 0.7$, $P(X = 1) = 0.3$
  - ✓ 질문: posteriori: $P(X = 0 \,|\, Y \le 40cm) =?$, $P(X = 1 \,|\, Y \le 40cm) =?$
  - ✓ $P(X = 0 \,|\, Y \le 40cm) = \frac{P(Y \le 40cm|X=0)P(X=0)}{P(Y \le 40cm)} = \frac{0.2 \times 0.7}{0.2 \times 0.7 + 0.3 \times 0.4} = 0.54$
  - ✓ $P(X = 1 \,|\, Y \le 40cm) = \frac{P(Y \le 40cm|X=1)P(X=1)}{P(Y \le 40cm)} = \frac{0.3 \times 0.3}{0.2 \times 0.7 + 0.3 \times 0.4} = 0.46$
  - ✓ Bayes decision 에 의해 연어라고 판단한다.

General Formulation

- Let $\{\omega_1, \ldots, \omega_c\}$ be the finite set of c categories.
- Let $\{\alpha_1, \ldots, \alpha_a\}$ be the finite set of a possible actions.

  Ex. Action α_i $=$ deciding that the true state is ω_i or others.
- The risk function $\lambda(\alpha_i|\omega_j)$ = risk incurred for taking action when the state of nature is ω_j.
- $x = d$ –dimensional feature vector (random variable)
- $p(x|\omega_i) =$ likelihood probability density function for x for given ω_i
- $p(\omega_i)$ $=$ prior probability that nature is in state ω_i.

Conditional Risk

- After the observation, the expected risk (conditional risk) is given by

$$R(\alpha_i|x) = \sum_{j=1}^{c} \lambda(\alpha_i|\omega_j)p(\omega_j|x)$$

- The decision action $\alpha(x)$ for given x is given

$$\alpha(x) = arg \min_{\alpha_i} R(\alpha_i|x) = arg\min_{\alpha_i} \sum_{j=1}^{c} \lambda(\alpha_i|\omega_j)p(\omega_j|x)$$

Two-Category Classification

- Action α_1 = deciding that the true state is ω_1
- Action α_2 = deciding that the true state is ω_2
- Let $\lambda_{ij} = \lambda(\alpha_i|\omega_j)$ be the risk incurred for deciding ω_i when true state is ω_j.
- *The conditional risks:*

$$R(\alpha_1|x) = \lambda_{11}p(\omega_1|x) + \lambda_{12}p(\omega_2|x)$$
$$R(\alpha_2|x) = \lambda_{21}p(\omega_1|x) + \lambda_{22}p(\omega_2|x)$$

- Decide ω_1 if $R(\alpha_1|x) < R(\alpha_2|x)$

  or if $(\lambda_{21} - \lambda_{11})p(\omega_1|x) > (\lambda_{12} - \lambda_{22})p(\omega_2|x)$

  or if $(\lambda_{21} - \lambda_{11})p(x|\omega_1)p(\omega_1) > (\lambda_{12} - \lambda_{22})p(x|\omega_2)p(\omega_2)$

  and ω_2 , *otherwise*

Two-Category Likelihood Ratio Test

- Under reasonable assumption that $\lambda_{12} > \lambda_{22}$ and $\lambda_{21} > \lambda_{11}$, (why?)

  decide ω_1 if $\frac{p(x|\omega_1)}{p(x|\omega_2)} > \frac{(\lambda_{12}-\lambda_{22})p(\omega_2)}{(\lambda_{21}-\lambda_{11})p(\omega_1)} = T$

  and ω_2 , *otherwise.*

- The ratio $\frac{p(x|\omega_1)}{p(x|\omega_2)}$ is called the *likelihood ratio*.

- We can decide ω_1 if the likelihood ratio exceeds a threshold T value that is independent of the observation x.

Minimum-Error-Rate Classification

- To give an equal cost to all errors, we define zero-one risk function as

$$\lambda(\alpha_i|\omega_j) = \begin{cases} 0, & i = j \\ 1, & i \neq j \end{cases}, \quad \text{for } i, j = 1, \dots, C$$

- The conditional risk representing error rate is

$$R(\alpha_i|x) = \sum_{j=1}^{c} \lambda(\alpha_i|\omega_j) p(\omega_j|x)$$
$$= \sum_{j\neq i}^{c} p(\omega_j|x) = \sum_{j}^{c} p(\omega_j|x) - p(\omega_i|x) = 1 - p(\omega_i|x)$$

- To **minimize** $R(\alpha_i|x)$, we **maximizes** $p(\omega_i|x)$

  Decide ω_i if $p(\omega_i|x) > p(\omega_j|x)$, for all $j \neq i$

  (same as Bayes' decision rule)

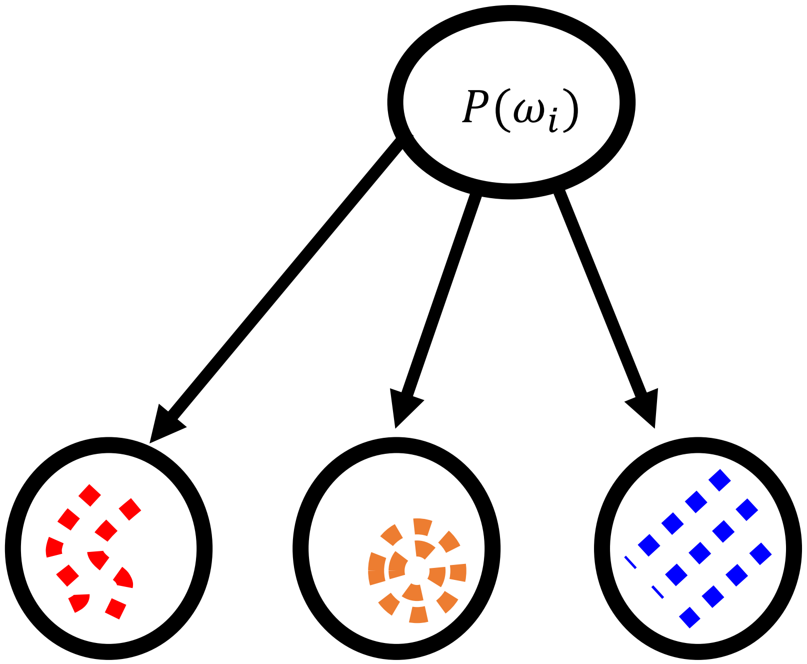
Density Estimation for Supervised Learning

Jin Young Choi

Seoul National University

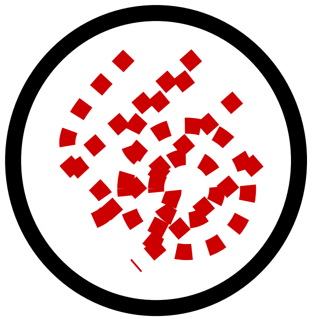
Learning From Observed Data

Hidden



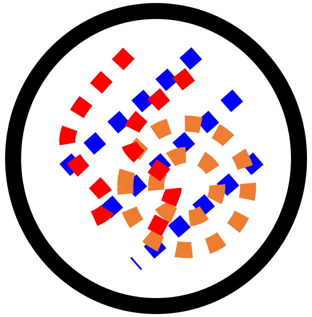
$P(x|\omega_1)$ $P(x|\omega_2)$ $P(x|\omega_3)$

Observed



Unsupervised

Clustering
Auto-encoding



Supervised

Likelihood Estimation
Error backpropagation

Parametric Learning for supervised learning

- Assume specific parametric distributions with parameters:

  $$p(x|\omega_i) \approx p(x|\theta_i), \theta_i \in \Theta \subset R^p$$

- Estimate parameters $\hat{\theta}(D)$ from training data D

- Replace true value of class-conditional density with approximation and apply the Bayesian framework for decision making.

Parametric Learning

- Suppose we can assume that the relevant (class-conditional) densities are of some parametric form. That is,
- $$p(x|\omega) \approx p(x|\theta), where\ \theta \in \Theta \in R^p$$
- Examples of parameterized densities:
  - Binomial: x has m 1's and $(n-m)$ 0's
  - $$p(x|\theta) = \binom{n}{m}\theta^m(1-\theta)^{n-m}, \Theta = [0,1]$$
  - Exponential: Each data point x is distributed according to
  - $$p(x|\theta) = \theta e^{-\theta x}, \Theta = (0,\infty)$$
  - Normal:
- $$p(x|\theta) \sim N(\mu, \sigma^2)$$

▪Multinomial pmf : $\Omega = \{k_1, \cdots, k_m |\ k_i = 0,\ 1,\ 2, \cdots, n\}$

$$p(k_1, \cdots, k_m) = \begin{cases} \frac{n!}{k_1! \cdots k_m!} p_1^{k_1} \cdots p_m^{k_m}, & k_i = 0,1, \cdots, n \\ 0, & \text{else} \end{cases}$$

Parametric Learning

- Bayesian Decision

$$arg \max_i p(\omega_i|x) \propto p(x|\omega_i)p(\omega_i)$$

- Maximum Likelihood Estimation

$$arg \max_{\theta_i} p(D|\theta_i) \leftarrow p(x|\omega_i) \approx p(x|\theta_i), \theta_i \in \Theta \subset R^p$$

- Maximum A-Posteriori Estimation

$$arg \max_{\theta_i} p(\theta_i|D), \theta_i = constant$$

- Bayesian Learning (not Estimation)

Find $p(\theta_i|D), \theta_i = random\ variable$

Maximum Likelihood Estimation

- The samples are i.i.d.

$$j^{th}\ class\ set\ D_j = \left\{x_l \middle| (x_l, \overline{\omega}_l) \in S_j\right\},\ S_j \in S = \{(x_l, \overline{\omega}_l) | l = 1, \dots, N\}$$

- The i.i.d. assumption implies that

$$p\left(D_j \middle| \theta_j\right) = \prod_{x \in D_j} p(x|\theta_j)$$

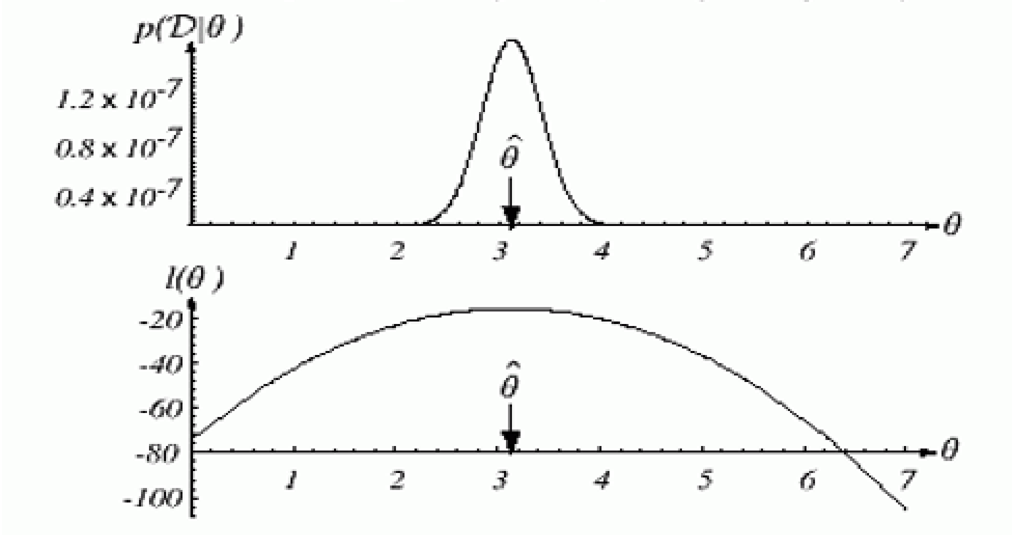
- Let D be a generic sample set of size $n = |D|$
- Log-likelihood function:

$$l(\theta; D) \equiv \ln p(D|\theta) = \sum_{k=1}^{n} \ln p(x_k|\theta)$$

- The log-likelihood function is identical to the logarithm of the probability density (or mass) function, but is interpreted as a function of parameter θ

Log-Likelihood Illustration

- $p(D|\theta)$ is not convex, but $\ln p(D|\theta)$ is convex (quadratic) for normal dist.



$$p(D|\theta) = \prod_{x \in D} p(x|\theta)$$

$$l(\theta; D) \equiv \ln p(D|\theta) = \sum_{k=1}^{n} \ln p(x_k|\theta)$$

Maximum Likelihood Estimation (MLE)

- The "most likely value" for MLE is given by

$$\hat{\theta}(D) = arg\max_{\theta \in \Theta} l(\theta; D)$$

- Gradient function

$$\nabla_\theta l(\theta; D) = \left[\frac{\partial l(\theta; D)}{\partial \theta_1}, \ldots, \frac{\partial l(\theta; D)}{\partial \theta_p}\right]$$

- Necessary condition for MLE (if not on border of domain Θ)

$$\nabla_\theta l(\theta; D) = 0$$

Example of Maximum Likelihood

The Gaussian Case:

$$p(x) = \frac{1}{\sqrt{(2\pi)^d}\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

- Log-likelihood is

$$l(\mu, \Sigma; D) = \ln p(D|\mu, \Sigma) = \ln \prod_{k=1}^{n} p(x_k|\mu) = \sum_{k=1}^{n} \ln p(x_k|\mu)$$

where

$$\ln p(x_k|\mu, \Sigma) = -\frac{1}{2}(x_k - \mu)^T\Sigma^{-1}(x_k - \mu) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma|$$

- For a sample point x_k, we have

$$\nabla_\mu \ln p(x_k|\mu, \Sigma) = \Sigma^{-1}(x_k - \mu)$$

- The maximum likelihood estimate for μ must satisfy

$$\sum_{k=1}^{n} \Sigma^{-1}(x_k - \mu) = 0 \rightarrow \hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k \quad \text{(sample mean)}$$

Example of Maximum Likelihood

The Gaussian Case:

- Log-likelihood is

$$l(\mu, \Sigma; D) = \ln p(D|\mu, \Sigma) = \ln \prod_{k=1}^{n} p(x_k|\mu) = \sum_{k=1}^{n} \ln p(x_k|\mu)$$

where $\ln p(x_k|\mu, \Sigma) = -\frac{1}{2}(x_k - \mu)^t \Sigma^{-1}(x_k - \mu) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma|$

$$\boxed{\nabla_\Sigma \ln|\Sigma| = \frac{1}{|\Sigma|} adj(\Sigma) = \Sigma^{-1}}$$

- For a sample point x_k, we have

$$\nabla_\Sigma \ln p(x_k|\mu, \Sigma) = \frac{1}{2}\Sigma^{-2}(x_k - \mu)(x_k - \mu)^t - \frac{1}{2}\Sigma^{-1} = 0$$

- The maximum likelihood estimate for Σ must satisfy

$\hat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n}(x_k - \hat{\mu})(x_k - \hat{\mu})^t$ (sample covariance)

Example of Maximum Likelihood

The Gaussian case

- For the multivariate case, it is easy to show that the MLE estimates are given by

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k\,, \qquad \hat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n}(x_k - \hat{\mu})(x_k - \hat{\mu})^t$$

- The MLE for Σ is biased

$$E\left[\frac{1}{n}\sum_{k=1}^{n}(x_k - \hat{\mu})(x_k - \hat{\mu})^t\right] = \frac{n-1}{n}\Sigma \neq \Sigma$$

- Unbiased estimate

$$\hat{\Sigma} = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \hat{\mu})(x_k - \hat{\mu})^t$$

$$E(\boldsymbol{e}^T\boldsymbol{e}) = Tr(\mathbf{I} - \mathbb{H})E(\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}) = (n-p-1)n\sigma^2$$
$$\rightarrow E(\boldsymbol{e}^T\boldsymbol{e}/(n-p-1)) = n\sigma^2$$
$$E(\boldsymbol{e}\boldsymbol{e}^T) = Tr(\mathbf{I} - \mathbb{H})E(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) = (n-p-1)\sigma^2\mathbf{I}$$
$$\rightarrow E(\boldsymbol{e}\boldsymbol{e}^T/(n-p-1)) = \sigma^2\mathbf{I}$$

Exercise

- 서울대학교병원에 암진단을 받으러 온 사람은 10만명이다. 그중 1000명이 암환자로 판명이 난다. 암진단은 피검사를 하여 암표지자 값을 보고 판단을 한다. 이 암표지자를 확률 변수 x 로 했을 때, 정상인과 암환자 모두 다음의 매개변수 θ 로 표현되는 확률 분포는 $p(x|\theta) = \theta x e^{-\theta x}, \quad for\ x > 0\ and\ \theta > 0$의 형태를 가지고 있다. 샘플은 i.i.d. 특성을 만족한다. 정상인의 암표지자 값을 평균하면 0.01이 되고 암환자의 암표지자를 평균하면 0.1이 된다. 정상인과 암환자의 분포를 가장 잘 나타내는 $\hat{\theta}_1, \hat{\theta}_2$ 를 MLE 방법으로 추정하시오.

Exercise

- 서울대학교병원에 암진단을 받으러 온 사람은 10만명이다. 그중 1000명이 암환자로 판명이 난다. 암진단은 피검사를 하여 암표지자 값을 보고 판단을 한다. 이 암표지자를 확률 변수 X 로 했을 때, 정상인과 암환자 모두 다음의 매개변수 θ 로 표현되는 확률 분포는 $p(X = x|\theta) = \theta x e^{-\theta x}, \quad for\ x > 0\ and\ \theta > 0$의 형태를 가지고 각 샘플은 i.i.d. 라고 가정한다. 정상인의 암표지자 값을 평균하면 0.01이 되고 암환자의 암표지자를 평균하면 0.1이 된다. 정상인과 암환자의 분포를 가장 잘 나타내는 $\hat{\theta}_1, \hat{\theta}_2$ 를 MLE 방법으로 추정하시오.

- Sol. (힌트) 문장의 수치에 해당하는 내용과 질문을 수식으로 표현해 보세요.

Exercise

- 서울대학교병원에 암진단을 받으러 온 사람은 10만명이다. 그중 1000명이 암환자로 판명이 난다. 암진단은 피검사를 하여 암표지자 값을 보고 판단을 한다. 이 암표지자를 확률 변수 x 로 했을 때, 정상인과 암환자 모두 다음의 매개변수 θ 로 표현되는 확률 분포는 $p(x|\theta) = \theta x e^{-\theta x}, \quad for\ x > 0\ and\ \theta > 0$의 형태를 가지고 각 샘플은 i.i.d. 라고 가정한다. 정상인의 암표지자 값을 평균하면 0.01이 되고 암환자의 암표지자를 평균하면 0.1이 된다. 정상인과 암환자의 분포를 가장 잘 나타내는 $\hat{\theta}_1, \hat{\theta}_2$ 를 MLE 방법으로 추정하시오.

- Sol. (힌트) 문장의 수치에 해당하는 내용과 질문을 수식으로 표현해 보세요.
  - ✓ 정상인의 샘플 $n = 99000$ 개, 즉 D_i =$\{x_i|i = 1, \dots, n\}$ 를 가지고 MLE를 수행하자. Likelihood function 은 아래와 같이 정의한다.

$$p(D_i|\theta) = \prod_{i=1}^{n} \theta x_i e^{-\theta x_i}\, \delta,$$ δ is constant and can be omitted without loss of generality

Exercise

- 서울대학교병원에 암진단을 받으러 온 사람은 10만명이다. 그중 1000명이 암환자로 판명이 난다. 암진단은 피검사를 하여 암표지자 값을 보고 판단을 한다. 이 암표지자를 확률 변수 x 로 했을 때, 정상인과 암환자 모두 다음의 매개변수 θ 로 표현되는 확률 분포는 $p(x|\theta) = \theta x e^{-\theta x}, \quad for\ x > 0\ and\ \theta > 0$의 형태를 가지고 각 샘플은 i.i.d. 라고 가정한다. 정상인의 암표지자 값을 평균하면 0.01이 되고 암환자의 암표지자를 평균하면 0.1이 된다. 정상인과 암환자의 분포를 가장 잘 나타내는 $\hat{\theta}_1, \hat{\theta}_2$ 를 MLE 방법으로 추정하시오.

- Sol. (힌트) 문장의 수치에 해당하는 내용과 질문을 수식으로 표현해 보세요.
  - ✓ 정상인의 샘플 $n = 99000$ 개, 즉 D_i ={$x_i | i = 1, \dots, n$} 를 가지고 MLE를 수행하자. Likelihood function 은 아래와 같이 정의한다.

$$p(D_i|\theta) = \prod_{i=1}^{n} \theta x_i e^{-\theta x_i}$$

  - ✓ 양변에 log를 위하여 log-likelihood 를 구하면 다음과 같다.

$$l(\theta) = n \log\theta + \sum_{i=1}^{n} \log x_i - \theta \sum_{i=1}^{n} x_i$$

Exercise

- 서울대학교병원에 암진단을 받으러 온 사람은 10만명이다. 그중 1000명이 암환자로 판명이 난다. 암진단은 피검사를 하여 암표지자 값을 보고 판단을 한다. 이 암표지자를 확률 변수 x 로 했을 때, 정상인과 암환자 모두 다음의 매개변수 θ 로 표현되는 확률 분포는 $p(x|\theta) = \theta x e^{-\theta x}, \quad for\ x > 0\ and\ \theta > 0$의 형태를 가지고 각 샘플은 i.i.d. 라고 가정한다. 정상인의 암표지자 값을 평균하면 0.01이 되고 암환자의 암표지자를 평균하면 0.1이 된다. 정상인과 암환자의 분포를 가장 잘 나타내는 $\hat{\theta}_1, \hat{\theta}_2$ 를 MLE 방법으로 추정하시오.

- Sol. (힌트) 문장의 수치에 해당하는 내용과 질문을 수식으로 표현해 보세요.
  - ✓ 정상인의 샘플 $n = 99000$ 개, 즉 D_i ={$x_i | i = 1, \ldots, n$} 를 가지고 MLE를 수행하자. Likelihood function 은 아래와 같이 정의한다.

$$p(D_i|\theta) = \prod_{i=1}^{n} \theta x_i e^{-\theta x_i}$$

  - ✓ 양변에 log를 위하여 log-likelihood 를 구하면 다음과 같다.

$$l(\theta) = (n \log \theta + -\theta \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} \log x_i$$

  - ✓ 양변에 θ 에 대해 미분하여 그 값이 0이 되도록 θ 를 구하면

$$\frac{d}{d\theta} l(\theta) = \frac{n}{\theta} - \sum_{i=1}^{n} x_i = 0, \quad \rightarrow \quad \hat{\theta}_{MLE} = \frac{n}{\sum_{i=1}^{n} x_i}$$

Exercise

- 서울대학교병원에 암진단을 받으러 온 사람은 10만명이다. 그중 1000명이 암환자로 판명이 난다. 암진단은 피검사를 하여 암표지자 값을 보고 판단을 한다. 이 암표지자를 확률 변수 x 로 했을 때, 정상인과 암환자 모두 다음의 매개변수 θ 로 표현되는 확률 분포는 $p(x|\theta) = \theta x e^{-\theta x}, \quad for\ x > 0\ and\ \theta > 0$의 형태를 가지고 각 샘플은 i.i.d. 라고 가정한다. 정상인의 암표지자 값을 평균하면 0.01이 되고 암환자의 암표지자를 평균하면 0.1이 된다. 정상인과 암환자의 분포를 가장 잘 나타내는 $\hat{\theta}_1, \hat{\theta}_2$ 를 MLE 방법으로 추정하시오.

- Sol. (힌트) 문장의 수치에 해당하는 내용과 질문을 수식으로 표현해 보세요.
  - ✓ 정상인의 샘플 $n = 99000$ 개, 즉 $D_i = \{x_i | i = 1, \dots, n\}$ 를 가지고 MLE를 수행하자. Likelihood function 은 아래와 같이 정의한다.

$$p(D_i|\theta) = \prod_{i=1}^{n} \theta x_i e^{-\theta x_i}$$

  - ✓ 양변에 log를 위하여 log-likelihood 를 구하면 다음과 같다.

$$l(\theta) = n \log \theta + -\theta \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} \log x_i$$

  - ✓ 양변에 θ 에 대해 미분하여 그 값이 0이 되도록 θ 를 구하면

$$\frac{d}{d\theta} l(\theta) = \frac{n}{\theta} - \sum_{i=1}^{n} x_i = 0, \quad \rightarrow \quad \hat{\theta}_{MLE} = \frac{n}{\sum_{i=1}^{n} x_i}$$

  - ✓ 여기서 정상인의 경우 암표지자 평균이 0.01 이므로 정상인 분포의 $\hat{\theta}_1$은 $\hat{\theta}_1 = 100$ 이 되고 암환자의 경우는 암표지자 평균이 0.1이므로 암환자 분포의 $\hat{\theta}_2$ 는 $\hat{\theta}_2 = 10$이 된다.

Exercise

- 서울대학교병원에 암진단을 받으러 온 사람은 10만명이다. 그중 1000명이 암환자로 판명이 난다. 암표지자 확률 분포는 $p(x|\theta) = \theta x e^{-\theta x}, \quad for\ x > 0\ and\ \theta > 0$ 에서 환자의 검사결과로 부터 θ 를 추정하였더니 정상인인 경우$\hat{\theta}_1 = 100,$ 암환자의 경우$\hat{\theta}_2 = 10$ 으로 추정이 되었다. 암진단을 받으러 온 사람의 검사결과 $x = 0.06$ 으로 나왔다. 정상인을 암환자로 잘못 진단하였을 때 리스크를 1로 하고, 암환자를 정상인으로 잘 못 진단 하였을 때 리스크를 10으로 설정 하였다. 정확히 진단하였을 때 리스크는 0으로 한다. 이 리스크를 감안하여 암환자 여부를 진단하시오.

Exercise

- 서울대학교병원에 암진단을 받으러 온 사람은 10만명이다. 그중 1000명이 암환자로 판명이 난다. 암표지자 확률 분포는 $p(x|\theta) = \theta x e^{-\theta x}, \quad for\ x > 0\ and\ \theta > 0$ 에서 환자의 검사결과로 부터 θ 를 추정하였더니 정상인인 경우$\hat{\theta}_1 = 100$, 암환자의 경우$\hat{\theta}_2 = 10$ 으로 추정이 되었다. 암진단을 받으러 온 사람의 검사결과 $x = 0.06$ 으로 나왔다. 정상인을 암환자로 잘못 진단하였을 때 리스크를 1로 하고, 암환자를 정상인으로 잘 못 진단 하였을 때 리스크를 10으로 설정 하였다. 정확히 진단하였을 때 리스크는 0으로 한다. 이 리스크를 감안하여 암환자 여부를 진단하시오.

- Sol. (힌트) 문장의 수치에 해당하는 내용과 질문을 수식으로 표현해 보세요.
  - ✓ 정상인 확률은 $p(\hat{\theta}_1) = 0.99$ 이고 암환자의 확률은 $p(\hat{\theta}_2) = 0.01$ 이다.

    질문: $R(\alpha_1|x = 0.06) = \lambda_{11}p(\hat{\theta}_1|x = 0.06) + \lambda_{12}p(\hat{\theta}_2|x = 0.06) =?$

    $R(\alpha_2|x = 0.06) = \lambda_{21}p(\hat{\theta}_1|x = 0.06) + \lambda_{22}p(\hat{\theta}_2|x = 0.06) =?$

Exercise

- 서울대학교병원에 암진단을 받으러 온 사람은 10만명이다. 그중 1000명이 암환자로 판명이 난다. 암표지자 확률 분포는 $p(x|\theta) = \theta x e^{-\theta x}, \quad for\ x > 0\ and\ \theta > 0$ 에서 환자의 검사결과로 부터 θ 를 추정하였더니 정상인인 경우$\hat{\theta}_1 = 100,$ 암환자의 경우$\hat{\theta}_2 = 10$ 으로 추정이 되었다. 암진단을 받으러 온 사람의 검사결과 $x = 0.06$ 으로 나왔다. 정상인을 암환자로 잘못 진단하였을 때 리스크를 1로 하고, 암환자를 정상인으로 잘 못 진단 하였을 때 리스크를 10으로 설정 하였다. 정확히 진단하였을 때 리스크는 0으로 한다. 이 리스크를 감안하여 암환자 여부를 진단하시오.

- Sol. (힌트) 문장의 수치에 해당하는 내용과 질문을 수식으로 표현해 보세요.
  - ✓ 정상인 확률은 $p(\hat{\theta}_1) = 0.99$ 이고 암환자의 확률은 $p(\hat{\theta}_2) = 0.01$ 이다.

    질문: $R(\alpha_1|x = 0.06) = \lambda_{11} p(\hat{\theta}_1|x = 0.06) + \lambda_{12} p(\hat{\theta}_2|x = 0.06) = ?$

    $R(\alpha_2|x = 0.06) = \lambda_{21} p(\hat{\theta}_1|x = 0.06) + \lambda_{22} p(\hat{\theta}_2|x = 0.06) = ?$

  - ✓ $p(\hat{\theta}_1|x = 0.06) \propto p(x = 0.06|\hat{\theta}_1) p(\hat{\theta}_1) = 100 * 0.06 e^{-100*0.06} * 0.99 = 0.0147$
  - ✓ $p(\hat{\theta}_2|x = 0.06) \propto p(x = 0.06|\hat{\theta}_2) p(\hat{\theta}_2) = 10 * 0.06 e^{-10*0.06} * 0.01 = 0.00329$

Exercise

- 서울대학교병원에 암진단을 받으러 온 사람은 10만명이다. 그중 1000명이 암환자로 판명이 난다. 암표지자 확률 분포는 $p(x|\theta) = \theta x e^{-\theta x}, \quad for\ x > 0\ and\ \theta > 0$ 에서 환자의 검사결과로 부터 θ 를 추정하였더니 정상인인 경우 $\hat{\theta}_1 = 100,$ 암환자의 경우 $\hat{\theta}_2 = 10$ 으로 추정이 되었다. 암진단을 받으러 온 사람의 검사결과 $x = 0.06$ 으로 나왔다. 정상인을 암환자로 잘못 진단하였을 때 리스크를 1로 하고, 암환자를 정상인으로 잘 못 진단 하였을 때 리스크를 10으로 설정 하였다. 정확히 진단하였을 때 리스크는 0으로 한다. 이 리스크를 감안하여 암환자 여부를 진단하시오.

- Sol. (힌트) 문장의 수치에 해당하는 내용과 질문을 수식으로 표현해 보세요.
  - ✓ 정상인 확률은 $p(\hat{\theta}_1) = 0.99$ 이고 암환자의 확률은 $p(\hat{\theta}_2) = 0.01$ 이다.

    질문: $R(\alpha_1|x = 0.06) = \lambda_{11} p(\hat{\theta}_1|x = 0.06) + \lambda_{12} p(\hat{\theta}_2|x = 0.06) = ?$

    $R(\alpha_2|x = 0.06) = \lambda_{21} p(\hat{\theta}_1|x = 0.06) + \lambda_{22} p(\hat{\theta}_2|x = 0.06) = ?$
  - ✓ 정상인 확률은 $p(\hat{\theta}_1) = 0.99$ 이고 암환자의 확률은 $p(\hat{\theta}_2) = 0.01$ 이다.
  - ✓ $p(\hat{\theta}_1|x = 0.06) \propto p(x = 0.06|\hat{\theta}_1) p(\hat{\theta}_1) = 100 * 0.06 e^{-100*0.06} * 0.99 = 0.0147$
  - ✓ $p(\hat{\theta}_2|x = 0.06) \propto p(x = 0.06|\hat{\theta}_2) p(\hat{\theta}_2) = 10 * 0.06 e^{-10*0.06} * 0.01 = 0.00329$
  - ✓ $R(\alpha_1|x = 0.06) = 0 * 0.0147 + 10 * 0.00329 = 0.0329$

    $R(\alpha_2|x = 0.06) = 1 * 0.0147 + 0 * 0.00329 = 0.0147$

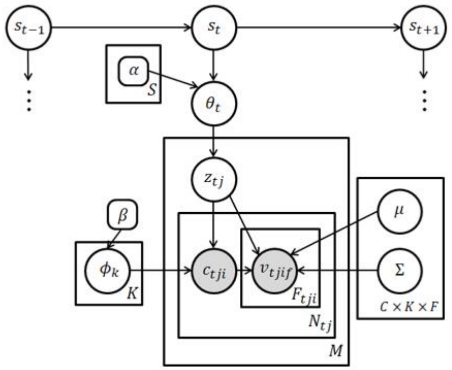
Bayesian Learning for Unsupervised Learning

Jin Young Choi

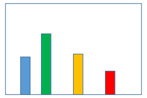
Seoul National University

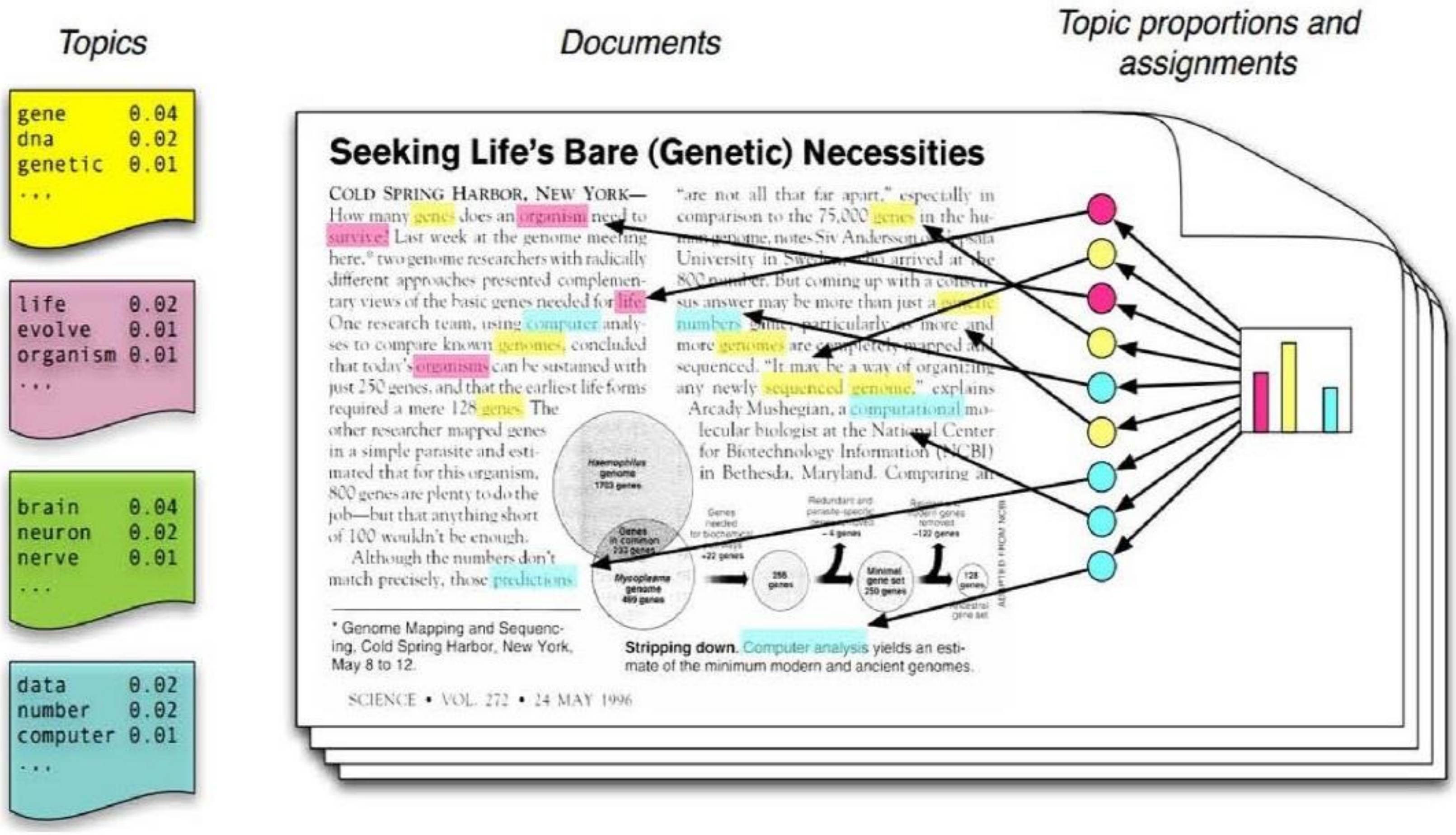
Surveillance in crowded scenes

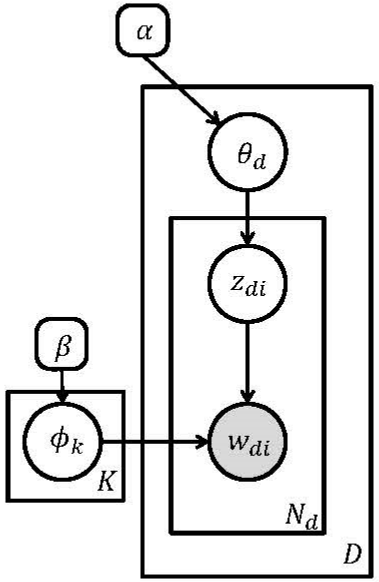
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$



Bayesian networks (Topic Modelling)



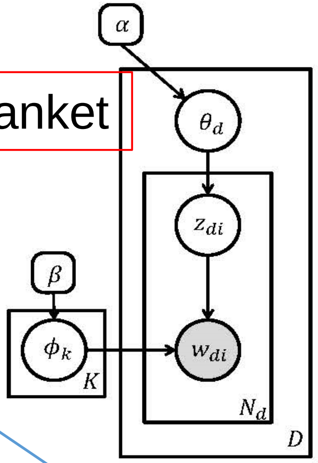




Bayesian Learning for Unsupervised Learning

- Markov Chain Monte Carlo (MCMC) framework

Markov Blanket



Posteriors

$$p(\theta_d|z,\alpha) = \frac{\overbrace{p(z|\theta_d)}^{Multinomial}\ \overbrace{p(\theta_d|\alpha)}^{Dirichlet}}{p(z|\alpha)}$$

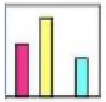
$$= Dir(\theta_d|\boxed{h_\theta(d,\cdot)} + \alpha),$$

$$p(\phi_k|z,w,\beta) = Dir(\phi_k|\boxed{h_\phi(k,\cdot)} + \beta).$$

$p(z|w,\phi_k,\theta_k)$

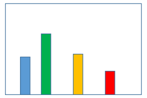
$$h_\theta(d,k) = \sum_{i=1}^{N_d} \delta\left[z_{di} - k\right]$$

$$h_\phi(k,v) = \sum_{d=1}^{D}\sum_{i=1}^{N_d} \delta\left[w_{di} - v\right]\delta\left[z_{di} - k\right];$$



$$\hat{\theta}_d(k) = E[\theta_d(k)|h_\theta(d,\cdot)+\alpha] = \frac{h_\theta(d,k)+\alpha(k)}{\sum_{k=1}^{K}\left[h_\theta(d,k)+\alpha(k)\right]},$$

$$\hat{\phi}_k(v) = E[\phi_k(v)|h_\phi(k,\cdot)+\beta] = \frac{h_\phi(k,v)+\beta(v)}{\sum_{v=1}^{V}\left[h_\phi(k,v)+\beta(v)\right]}.$$



w: 1, 3, 2, 3, 3, 5, 4, 1, 6
z : 1, 2, 2, 2, 1, 1, 2, 1, 2
h_θ(d,2): 5

w: 1, 3, 2, 3, 3, 5, 4, 1, 6
z : 1, 2, 2, 2, 1, 1, 2, 1, 2
h_ϕ(1,3): 1
h_ϕ(2,3): 2

Bayesian Learning

– Univariate Normal Distribution

- Let μ be the only unknown parameter

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

- Goal: to find posterior $p(\mu|D)$ from $i.i.d.\, D = \{x_1, \ldots, x_n\}$, where

$$p(\mu|D) = \frac{p(D|\mu)p(\mu)}{\int p(D|\mu)p(\mu)d\mu}$$

$$= \alpha \prod_{k=1}^{n} p(x_k|\mu)p(\mu).$$

- conjugate prior probability for μ is given by

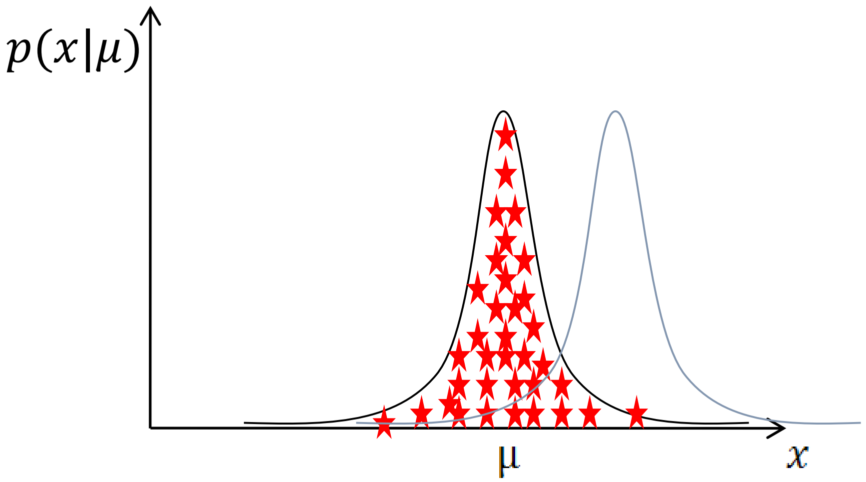
$$p(\mu) \sim N(\mu_0, \sigma_0^2) \rightarrow p(\mu|D) \sim N(\mu_n, \sigma_n^2)$$

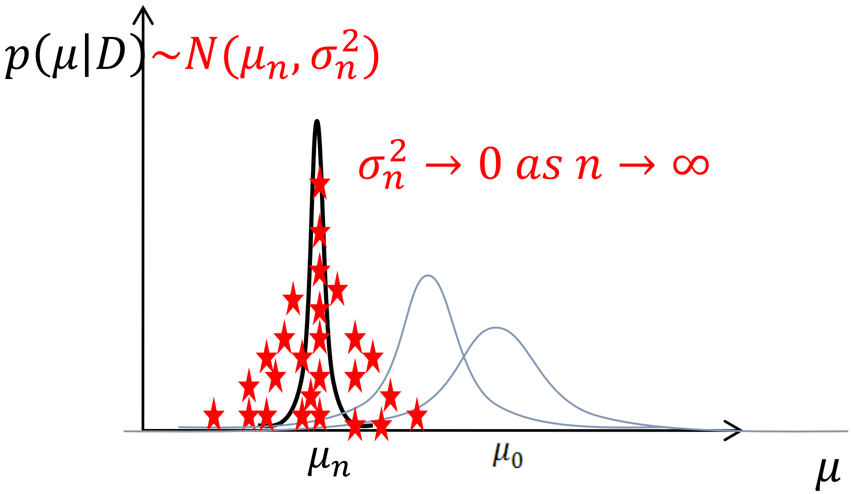
Maximum Likelihood vs Bayesian Learning

- Maximum likelihood estimation: find $\hat{\mu}(D)$ to maximize $p(x|D)$

$$p(x|D) \approx p(x|\hat{\mu}(D)), \hat{\mu}(D) = arg \max_{\mu} p(D|\mu)$$

- Bayesian learning: find $p(\theta|D)$.





Bayesian Learning

– Univariate Normal Distribution

$p(\mu|D) = \alpha \prod_{k=1}^{n} p(x_k|\mu)p(\mu)$

$p(\mu) \sim N(\mu_0, \sigma_0^2)$

- Computing the posterior distribution

$$p(\mu|D) = \alpha\, p(D|\mu)p(\mu)$$

$$= \alpha' \exp\left[-\frac{1}{2}\left(\sum_{k=1}^{n}\left(\frac{(x_k-\mu)}{\sigma}\right)^2 + \left(\frac{(\mu-\mu_0)}{\sigma_0}\right)^2\right)\right]$$

$$= \alpha'' \exp\left[-\frac{1}{2}\left(\left(\frac{n}{\sigma^2}+\frac{1}{{\sigma_0}^2}\right)\mu^2 - 2\left(\frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0^2}\right)\mu\right)\right]$$

$p(\mu|D) \sim N(\mu_n, \sigma_n^2)$

- Since $p(\mu|D)$ should be Gaussian

$$p(\mu|D) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2}\left(\frac{(\mu-\mu_n)}{\sigma_n}\right)^2\right] = \alpha''' \exp\left[-\frac{1}{2\sigma_n^2}\mu^2 - \frac{\mu_n}{\sigma_n}\mu\right]$$

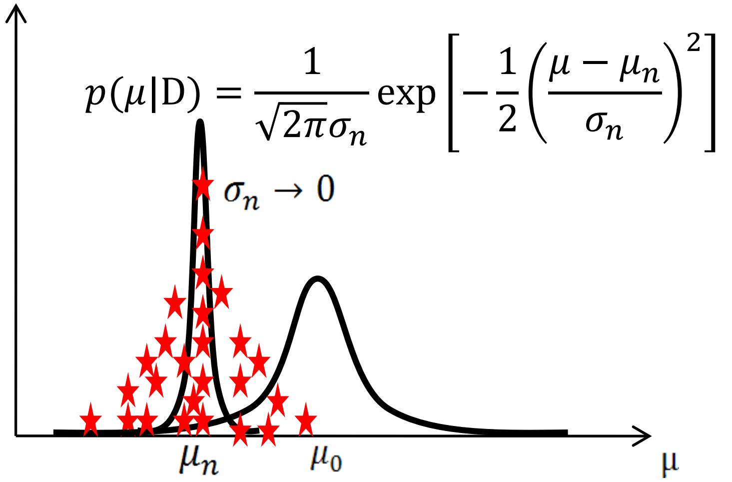
- $\mu_n = f(x_k, \mu_0, \sigma_0) =?, \quad \sigma_n = h(x_k, \mu_0, \sigma_0) =?$

Bayesian Learning

– Univariate Normal Distribution

- Solution: $\frac{1}{\sigma_n^2} = \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}$, $\frac{\mu_n}{\sigma_n^2} = \frac{n}{\sigma^2}\hat{\mu}_n + \frac{\mu_0}{\sigma_0^2}$

  where $\hat{\mu}_n = \frac{1}{n}\sum_{k=1}^{n} x_k$ is the sample mean.



- Solving explicitly for μ_n and σ_n^2 we obtain

$$\boxed{\sigma_n^2 = \frac{\sigma_0^2\sigma^2}{n\sigma_0^2+\sigma^2}, \qquad \mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2+\sigma^2}\right)\hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2+\sigma^2}\mu_0}$$

- μ_n represents our best guess for μ after observing n samples.
- σ_n^2 measures our uncertainty about this guess.
- σ_n^2 decreases monotonically with n

  (approaching $\sigma_n^2 \to \frac{\sigma^2}{n} \to 0$ as n approaches infinity)

Bayesian Learning

– Univariate Normal Distribution

- Likelihood Estimation:

$$p(\mathbf{x}|\omega) \approx p(x|D) = \int p(x|\mu)p(\mu|D)d\mu$$

$$= \int \frac{1}{2\pi\sigma\sigma_n} \exp\left[-\frac{1}{2}\left(\frac{(x-\mu)}{\sigma}\right)^2 - \frac{1}{2}\left(\frac{(\mu-\mu_n)}{\sigma_n}\right)^2\right] d\mu$$

$$= \frac{1}{2\pi\sigma\sigma_n} \exp\left[-\frac{1}{2}\frac{(x-\mu_n)^2}{\sigma^2+\sigma_n^2}\right] \frac{\sqrt{2\pi}\sigma\sigma_n}{\sqrt{\sigma^2+\sigma_n^2}}$$

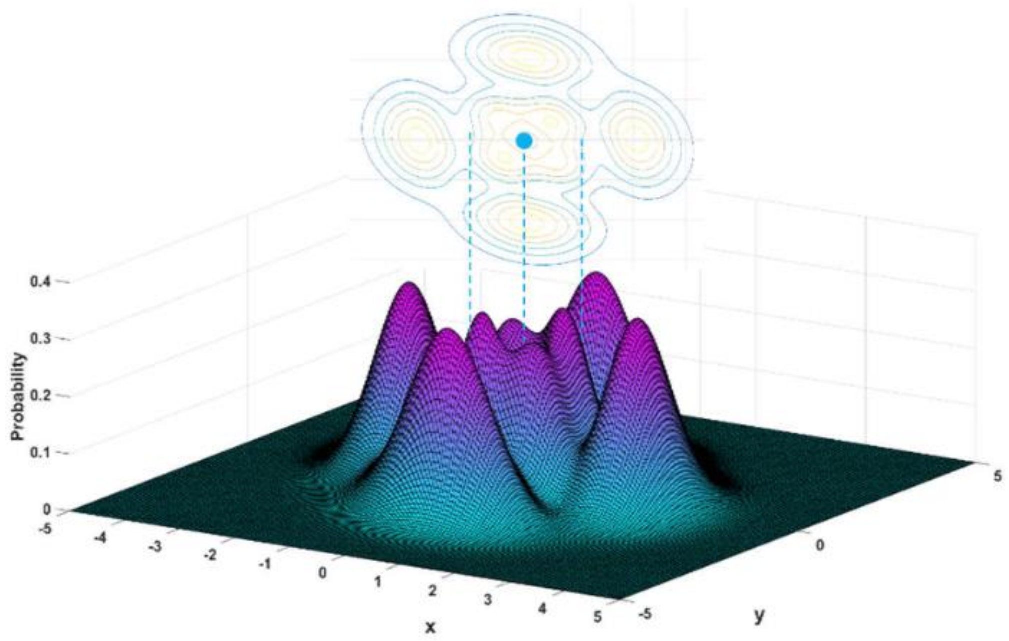
$$= \frac{1}{\sqrt{2\pi}\sqrt{\sigma^2+\sigma_n^2}} \exp\left[-\frac{1}{2}\frac{(x-\mu_n)^2}{\sigma^2+\sigma_n^2}\right]$$

$$\xrightarrow[n\to\infty]{} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right]$$

Nonparametric Density Estimation for Un/Semi/Self-supervised Learning, Generative Model Learning

Jin Young Choi

Seoul National University



Ciann-Dong Yang and Shiang-Yi Han, "Extending Quantum Probability from Real Axis to Complex Plane", Entropy. https://www.researchgate.net/publication/349120823_Extending_Quantum_Probability_from_Real_Axis_to_Complex_Plane/citation/download

Generative Model Learning

Setup:

- Assume we want to learn a generative model from a set of data points $\{\boldsymbol{x}_i\}$
  - $P_{data}(\boldsymbol{x})$ is the data distribution, which is never known to us, but we have sampled $\boldsymbol{x}_i \sim P_{data}(\boldsymbol{x})$
  - $P_{model}(\boldsymbol{x}|\boldsymbol{\theta})$ is the model, universally parametrized by $\boldsymbol{\theta}$, that we use to approximate $P_{data}(\boldsymbol{x})$

Goal:

- Make $P_{model}(\boldsymbol{x}|\boldsymbol{\theta})$ close to $P_{data}(\boldsymbol{x})$
- Make sure we can sample from $P_{model}(\boldsymbol{x}|\boldsymbol{\theta})$
  - We need to generate examples from $P_{model}(\boldsymbol{x}|\boldsymbol{\theta})$ by sampling

Generative Model Learning

Make $P_{model}(\boldsymbol{x}|\boldsymbol{\theta})$ close to $P_{data}(\boldsymbol{x})$

- Key Principle: Maximum Likelihood (Minimum Cross Entropy)
- Fundamental approach to modeling distributions

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{x}\sim P_{data}(\boldsymbol{x})} \log\, P_{model}(\boldsymbol{x}|\boldsymbol{\theta})$$

- Find parameters $\boldsymbol{\theta}^*$, such that for observed data points $\boldsymbol{x}_i \sim P_{data}(\boldsymbol{x})$

  $\sum_i \log\, P_{model}(\boldsymbol{x}_i|\boldsymbol{\theta})$ has the highest value, among all possible choices of $\boldsymbol{\theta}$
  - That is, find the model that is most likely to have generated the observed data $\boldsymbol{x}$

Generative Models

Sample from $P_{model}(\boldsymbol{x}|\boldsymbol{\theta})$

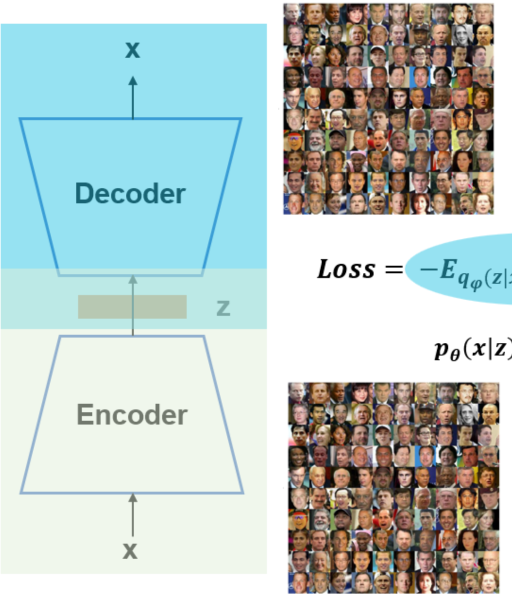
- **Goal:** Sample from a complex distribution
- The most common approach:
  – Sample from a simple noise distribution

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

  – Transform the noise z_i via $f(\cdot)$ to a sample $\boldsymbol{x}_i$

$$\boldsymbol{x}_i = f(z_i; \theta)$$

- **Q:** How to design $f(\cdot)$?
- **A:** Variation Auto-Encoder (VAE)



Reconstruction Loss

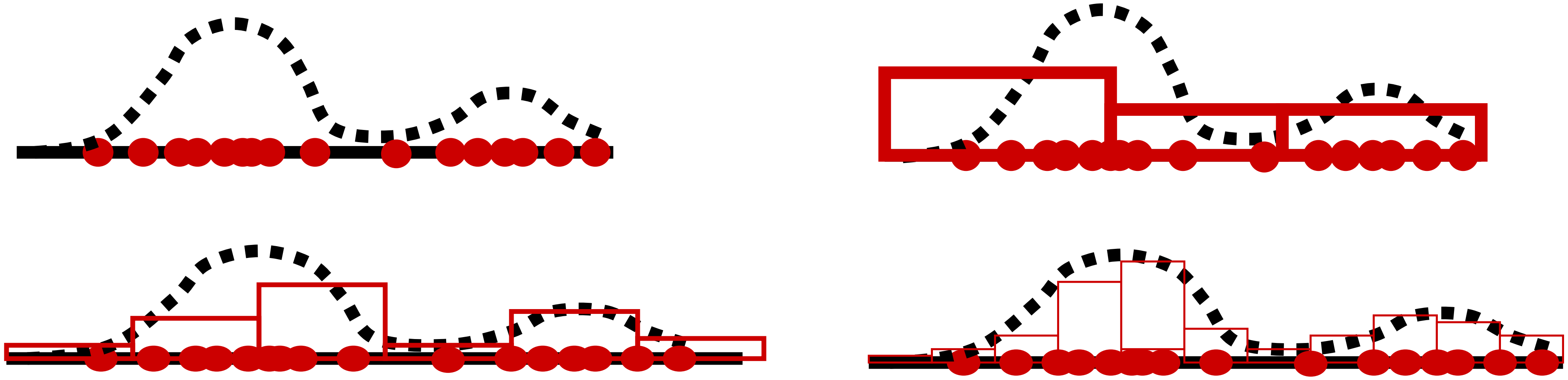
$$Loss = -E_{q_\varphi(z|x)} log P_\theta(\boldsymbol{x}|\boldsymbol{z}) + D_{KL}(q_\varphi(z|x)||P_\theta(z))$$

Variational Inference

$\boldsymbol{p}_\theta(\boldsymbol{x}|\boldsymbol{z})$: a multivariate Gaussian (real-valued data)
a Bernoulli (binary-valued data)

Nonparametric Density Estimation

- The form of $p(x|\omega_i)$ to be estimated is not assumed.
- Naïve approach is Histogram



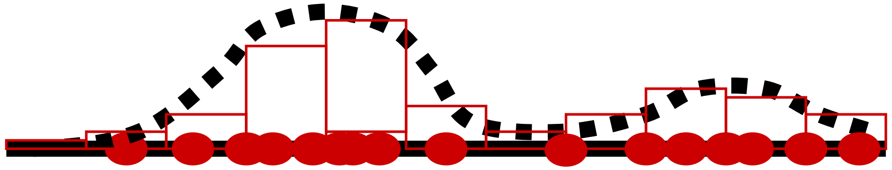
$$p = \int_R p(\mathbf{x}')d\,\mathbf{x}' \rightarrow P_k = \binom{n}{k} p^k (1-p)^{n-k} \rightarrow E[k] = np \rightarrow \mathrm{var}(k) = np(1-p)$$

$$E\left[\frac{k}{n}\right] = p, \qquad \mathrm{var}\left[\frac{k}{n}\right] = \frac{p(1-p)}{n} \qquad p = \int_R p(\mathbf{x}')d\,\mathbf{x}' \approx p(x)V \rightarrow p(x) = p/V \approx \frac{k/n}{V}$$

Three Conditions for Density Estimation

- Reducing the region by increasing the samples
- Let us take a growing sequence of samples $n = 1, 2, 3$ ...
- We take regions R_n with reduced volumes $V_1 > V_2 > V_3 > \cdots$
- Let k_n be the number of samples falling in R_n
- Let $p_n(x)$ be the n^{th} estimate for $p(x)$
- If $p_n(x)$ is to converge to $p(x)$, 3 conditions must be required:
  - $\lim_{n\to\infty} V_n = 0,$ resolution as big as possible (for smoothing)
  - $\lim_{n\to\infty} k_n = \infty,$ to preserve $\int p(x)d\,x = 1$
  - $\lim_{n\to\infty} \frac{k_n}{n} = 0$ to guarantee convergence of $p(x) \approx p_n(x) = \frac{k/n}{V}$

$$\int \hat{p}(\mathbf{x}|\omega_i)d\mathbf{x} = \sum_{j=1}^{m} \int_{b_j} \frac{k_j}{n_i V} d\mathbf{x} = \frac{1}{n_i} \sum_{j=1}^{m} k_j = 1$$



PARZEN WINDOW and KNN

- How to obtain the sequence $R_1, R_2, ..$?
- There are 2 common approaches of obtaining sequences of regions that satisfy the convergence conditions:
  - Specify k_n as some function of n, such as $k_n = \sqrt{n}$. Here the volume V_n is grown until it encloses k_n neighbors of x.
  - This is k_n –nearest-neighbor method.
  - Shrink an initial region by specifying the volume V_n as some function of n, such as $V_n = 1/\sqrt{n}$ and show that k_n and k_n/n behave properly i.e. $p_n(x)$ converges to $p(x)$.
  - This is Parzen-window (or kernel) method.

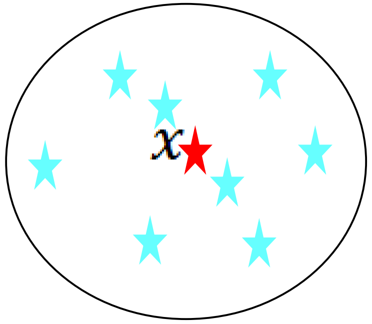
$$\lim_{n\to\infty} V_n = 0,$$
$$\lim_{n\to\infty} k_n = \infty,$$
$$\lim_{n\to\infty} \frac{k_n}{n} = 0$$

K_n-Nearest-Neighbor Estimation

- To estimate $p(x)$ from n training samples, we center a cell about x and let it grow until it captures k_n samples, where k_n is some specified function of n.
- These samples are the k_n nearest-neighbors of x.
- If the density is high near x, the cell will be relatively small good resolution.

$$\boxed{\lim_{n\to\infty} V_n = 0\,,\ \lim_{n\to\infty} k_n = \infty,\ \lim_{n\to\infty} k_n/n = 0} \to p(\mathbf{x}) \approx \frac{k_n/n}{V_n}$$

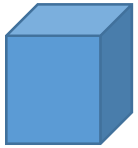
Let $k_n = \sqrt{n} \to V_n \approx 1/(\sqrt{n}p(x)) \to 0$



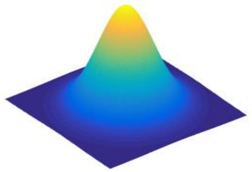
PARZEN WINDOWS

- Assume that the region R_n is a d −dimensional hypercube.
- If h_n is the length of an edge of that hypercube $\mathcal{H}(x)$ centered at x, then its volume is given by

$$V_n = h_n^d$$



$$V_n = 1/\sqrt{n}$$
$$h_n = 1/\sqrt[d]{n} \to 0$$



- Define the following window function:

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_j| \le 1/2\,; \quad j = 1, \ldots, d \\ 0 & \text{otherwise.} \end{cases}$$

- $\varphi(\mathbf{u})$ defines a unit hypercube centered at the origin.

$$\varphi((\mathbf{x} - \mathbf{x}_i)/h_n) = \begin{cases} 1 & \mathbf{x}_i \in \mathcal{H}(x) \\ 0 & \text{otherwise.} \end{cases}$$

$$\boxed{\begin{array}{c} -1/2 \le (\mathbf{x}_i - \mathbf{x})/h_n \le 1/2 \to -h_n/2 \le \mathbf{x}_i - \mathbf{x} \le h_n/2 \\ \to \mathbf{x} - h_n/2 \le \mathbf{x}_i \le \mathbf{x} + h_n/2 \end{array}}$$

- The number of samples in this hypercube is given by:

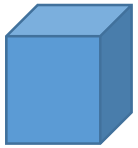
$$k_n = \sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) \to p_n(\mathbf{x}) = \frac{k_n}{nV_n} = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{V_n} \varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right).$$

$$\boxed{\lim_{n\to\infty} V_n = 0\,,\ \lim_{n\to\infty} k_n = \infty,\ \lim_{n\to\infty} k_n/n = 0} \quad \because k_n = nV_n p_n(\mathbf{x}) = \sqrt{n}\, p_n(\mathbf{x}), \quad k_n/n = p_n(\mathbf{x})/\sqrt{n}$$

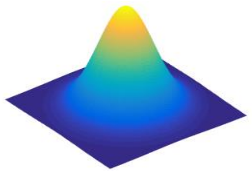
Gaussian Mixture Estimation

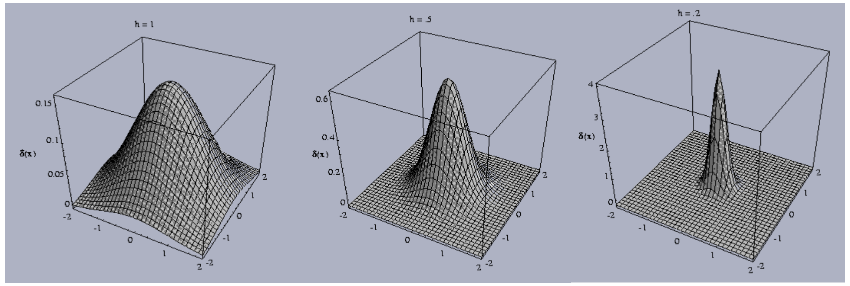
- Parzen Window

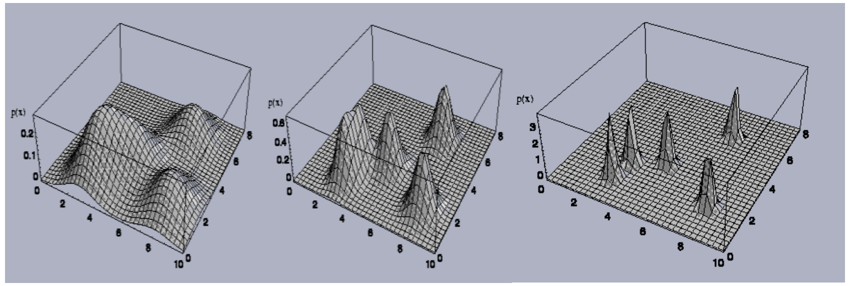
$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)$$



$$\varphi(\mathbf{u}) \geq 0 \quad \text{and} \quad \int \varphi(\mathbf{u})\, d\mathbf{u} = 1$$





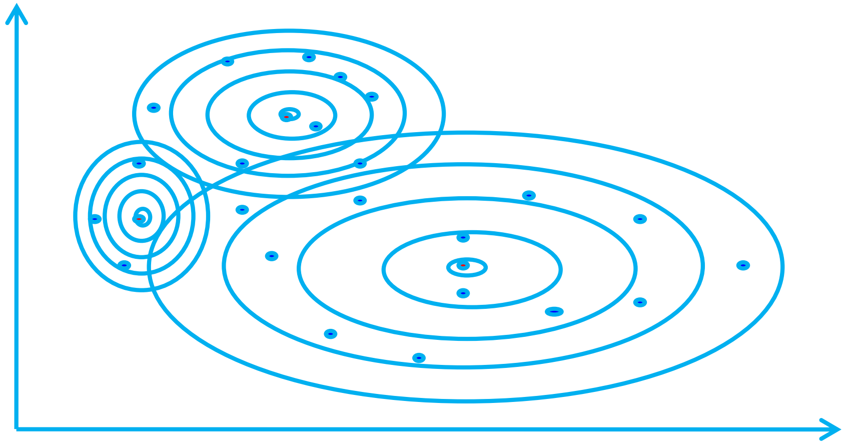


- Gaussian Mixture

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} w_k \varphi\left(\frac{\mathbf{x}-\mu_k}{\sigma_k}\right)$$

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} p(\mathbf{x}|\theta_k)p(\theta_k|\theta)$$

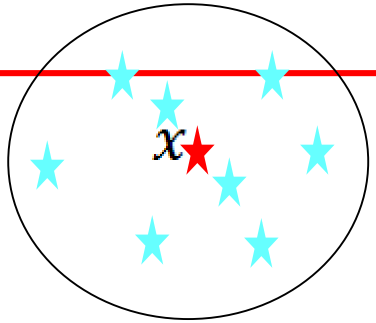
$$= \sum_{k=1}^{K} p(\mathbf{x}|k)p(k|\theta) = \sum_{k=1}^{K} p(\mathbf{x}, k|\theta)$$



K-means Clustering (naïve unsupervised learning)

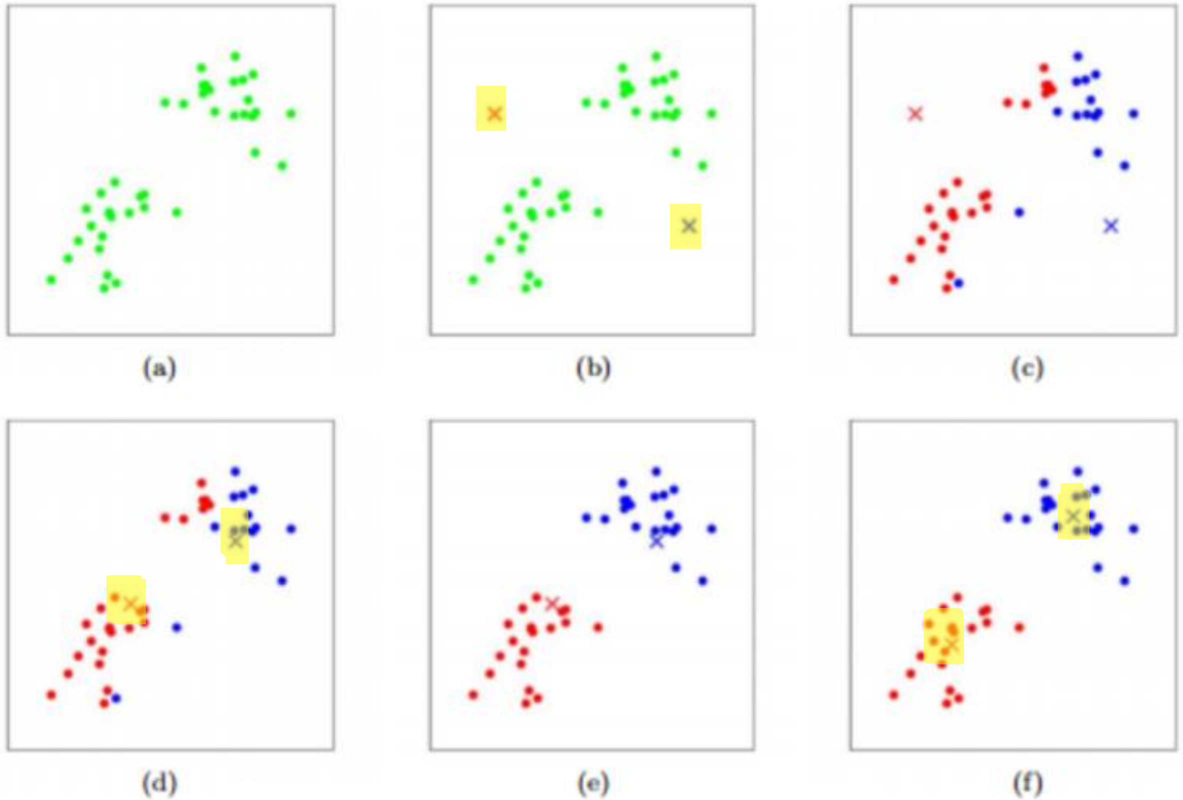
1. 일단 K개의 임의의 중심점(centroid)을 배치하고
2. 각 데이터들을 가장 가까운 중심점으로 할당한다. (일종의 군집을 형성한다.)
3. 군집으로 지정된 데이터들을 기반으로 해당 군집의 중심점을 업데이트한다.
4. 2번, 3번 단계를 그래서 수렴이 될 때까지, 즉 더 이상 중심점이 업데이트 되지 않을 때까지 반복한다.

K-Nearest-Neighbor



$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} w_k \varphi\left(\frac{\mathbf{x}-\mu_k}{\sigma_k}\right)$$

MLE to extimate w_k



Expectation-Maximization (EM)

- EM aims to find parameter values that maximize likelihood,

  $$L(\theta;\mathbf{x}) = p(\mathbf{x}|\theta) = \Sigma_k p(\mathbf{x},k|\theta) = \Sigma_k\ L(\theta\,;\mathbf{x}\,,k)$$

  where k is a latent variable.

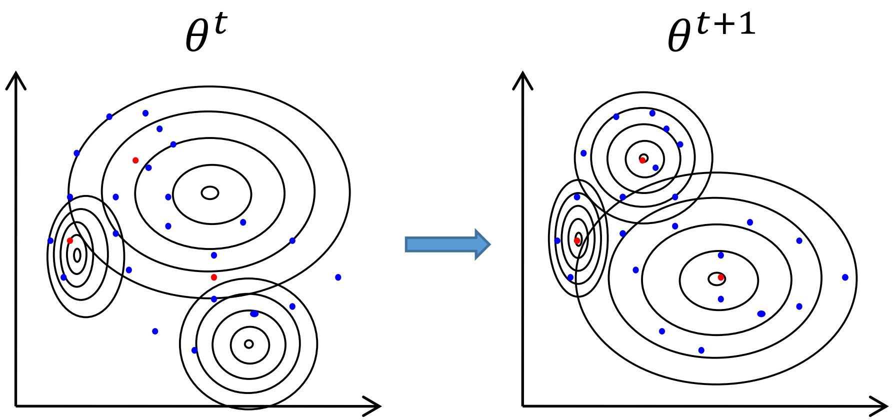
- **E-step:** For given $\theta^t, \mathbf{x}$, find expectation of the likelihood on the conditional distribution of k. θ^0 can be chosen by **K-means Clustering.**

  $$Q(\theta|\theta^t) = E_{k|\mathbf{x},\theta^t}[\log L\,(\theta\,;\mathbf{x}\,,k)] = \Sigma_k p(k|\mathbf{x},\theta^t)\log L\,(\theta\,;\mathbf{x}\,,k)$$

- **M-step:** Find θ^{t+1} maximizing Q.

  $$\theta^{t+1} = \underset{\theta}{\operatorname{argmax}} Q(\theta|\theta^t)$$

- Repeat E-step and M-step.



Expectation-Maximization (EM)

- **E-step**

$$Q(\theta|\theta^t) = E_{k|\mathbf{x},\theta^t}[\log L(\theta\,;\mathbf{x}\,,k)] = \Sigma_k p(k|\mathbf{x},\theta^t)\log L(\theta\,;\mathbf{x}\,,k)$$

$$T_{k,m}^t := p(k|\mathbf{x}=x_m,\theta^t) = \frac{p(x_m\,|\mu_k^t,\Sigma_k^t)\tau_k^t}{\Sigma_k\, p(x_m\,|;\mu_k^t,\Sigma_k^t)\tau_k^t}, \tau_k^t = p(k|\theta^t)$$

$$p(x_m\,|\mu_k^t,\Sigma_k^t) = \frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma_k^t|}}\exp\left(-\frac{(x_m-\mu_k^t)^T{\Sigma_k^t}^{-1}(x_m-\mu_k^t)}{2}\right)$$

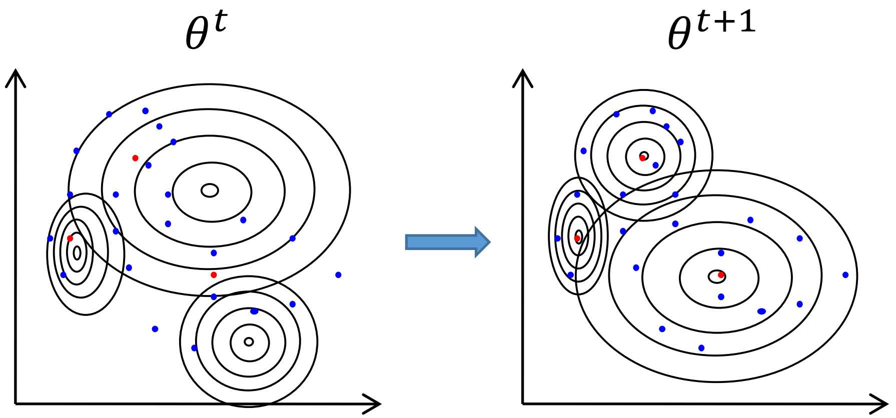
$$Q(\theta|\theta^t) = \Sigma_m\Sigma_k T_{k,m}^t\left(\log\tau_k - \frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x_m-\mu_k)^T\Sigma_k^{-1}(x_m-\mu_k)\right)$$

- **M-step**

$$\tau_k^{t+1} = \frac{\Sigma_m T_{k,m}^t}{\Sigma_k\Sigma_m T_{k,m}^t},$$

$$\mu_k^{t+1} = \frac{\Sigma_m T_{k,m}^t x_m}{\Sigma_k\Sigma_m T_{k,m}^t},$$

$$\Sigma_k^{t+1} = \frac{\Sigma_m T_{k,m}^t(x_m-\mu_k^{t+1})(x_m-\mu_k^{t+1})^T}{\Sigma_k\Sigma_m T_{k,m}^t}$$

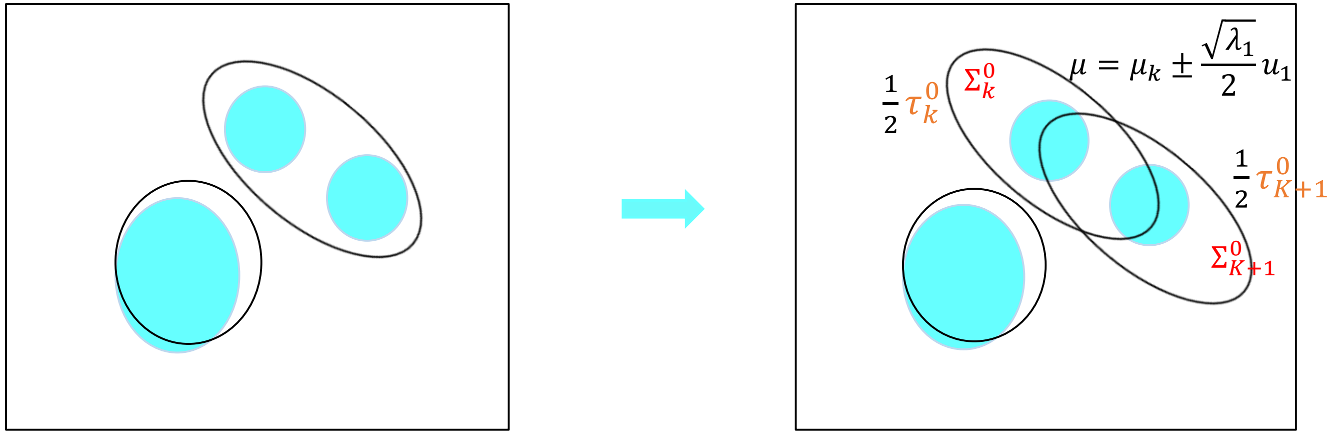


Automatic Model Order Selection

- For each component k, define a total responsibility $r(k)$ as

$$r(k) = \tau_k^T = p(k|\theta^T) = \sum_{m=1}^{M} p(k|\mathbf{x} = x_m, \theta^T) = \sum_{m=1}^{M} \frac{p(x_m; \mu_k, \Sigma_k)\tau_k}{\sum_k p(x_m; \mu_k, \Sigma_k)\tau_k}$$

- The cluster with the lowest $r(k)$ is splited.
- Covariance matrices equal to Σ_k
- New cluster center is set to $\mu = \mu_k \pm \frac{\sqrt{\lambda_1}}{2} u_1$, where λ_1 is the largest eigenvalue of Σ_k and u_1 is the corresponding eigenvector.



Automatic Model Order Selection

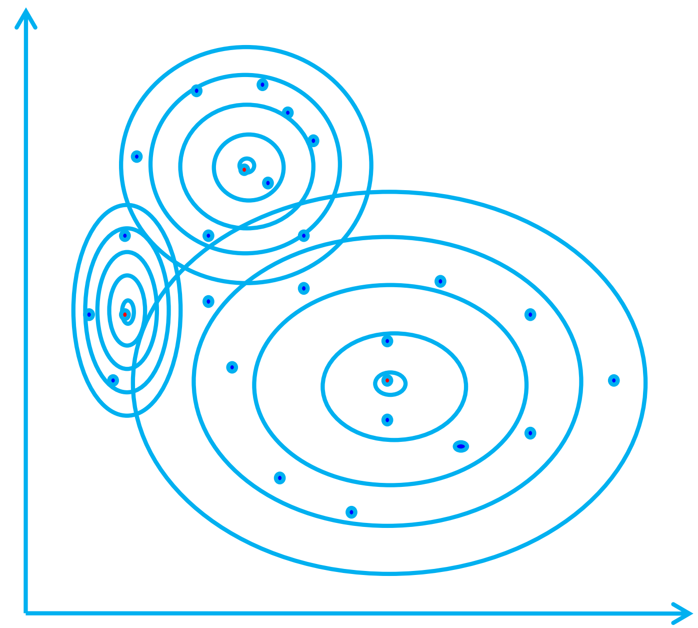
- Prior probabilities for the new components are set to $\frac{1}{2}p(k|\theta^T) = \frac{1}{2}\tau_k^0$
- K_i denotes the number of components in a model after i-th iteration
- L_i be the likelihood of the validation set given the model

1. Apply EM for model with K_i components.
2. Compute L_i for validation set
3. If $(L_i - L_{i-1} \leq \varepsilon)$, STOP.
4. Split the cluster k with the lowest total responsibility $r(k)$
5. Set $K_{i+1} = K_i + 1$ and $i = i + 1$
6. Go to 1.

Markov Chain Monte Carlo(MCMC)

- Monte Carlo : Sample from a distribution to estimate the distribution
- Markov Chain Monte Carlo (MCMC)
  - Applied to Clustering, Unsupervised Learning, Bayesian Inference
- Example: Estimation of Gaussian Mixture Model

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} p(\mathbf{x}|\theta_k) p(\theta_k|\theta)$$



- EM은 확률 모델에 기반하여 유도
- MCMC는 EM 보다 광범위하게 사용
- 계산량이 많음
- MCMC 대신 미분기반의 Variation Inference를 사용할 수 있음. 문제 별 유도 필요.

Monte Carlo Integration

- General problem: evaluating
$$\mathbb{E}_P[h(X)] = \int h(x)p(x)dx$$
can be difficult. ($\int |h(x)|p(x)dx < \infty$)

- If we can draw samples $x^{(s)} \sim p(x)$, then we can estimate
$$\mathbb{E}_P[h(X)] \approx \bar{h}_N = \frac{1}{N}\sum_{s=1}^{N} h\left(x^{(s)}\right).$$

- Monte Carlo integration is great if you can sample from the target distribution
  - But what if you can't sample from the target?
  - Importance sampling: Use of a simple distribution

Importance Sampling

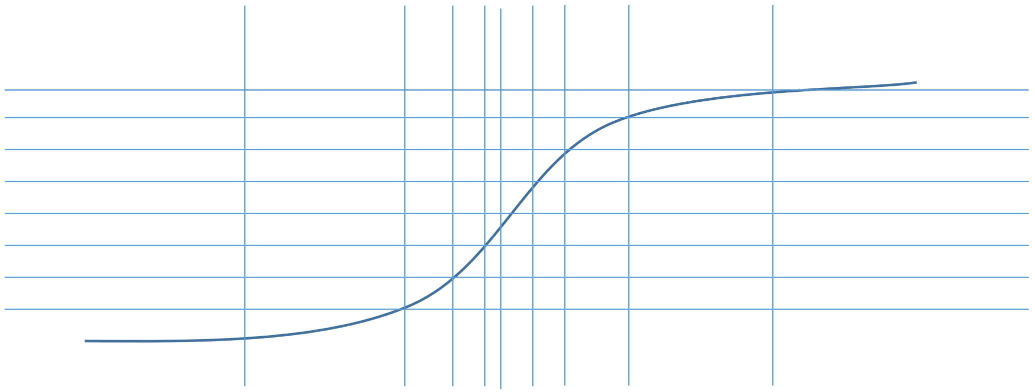
- Idea of importance sampling:

  Draw the sample from a proposal distribution $Q(\cdot)$ and re-weight by importance weights

$$\mathbb{E}_P[h(X)] = \int \frac{h(x)P(x)}{Q(x)} Q(x)dx = \mathbb{E}_Q\left[\frac{h(X)P(X)}{Q(X)}\right].$$

- Hence, given an iid sample $x^{(s)}$ from Q, our estimator becomes

$$E_Q\left[\frac{h(X)P(X)}{Q(X)}\right] = \frac{1}{N}\sum_{s=1}^{N} \frac{h\left(x^{(s)}\right)P\left(x^{(s)}\right)}{Q\left(x^{(s)}\right)}$$



Limitations of Monte Carlo

- Importance sampling
  - Do not work well if the proposal $Q(x)$ is very different from target $P(x)$
  - Yet constructing a $Q(x)$ similar to $P(x)$ can be difficult → Markov Chain

- Intuition: instead of a fixed proposal $Q(x)$, what if we could use an **adaptive** proposal?
  - X_{t+1} depends only on X_t, not on $X_0, X_1, \dots, X_{t-1}$
  - Markov Chain

Markov Chains: Notation & Terminology

- Countable (finite) state space Ω (e.g. **N**)
- Sequence of random variables $\{X_t\}$ on Ω for $t = 0,1,2,\ldots$

- Definition : $\{X_t\}$ is a Markov Chain if

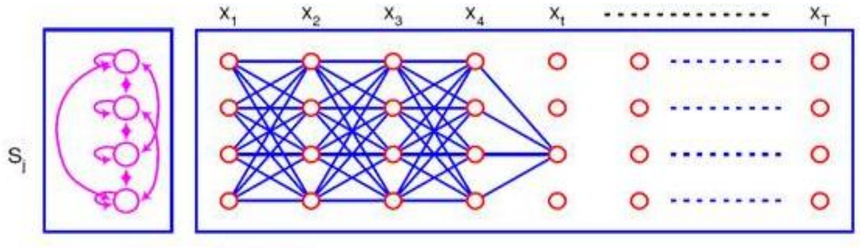
$$P(X_{t+1} = y \mid X_t = x_t, \ldots, X_0 = x_0) = P(X_{t+1} = y \mid X_t = x_t)$$

- Notation : $P(X_{t+1} = i \mid X_t = j) = p_{ji}$

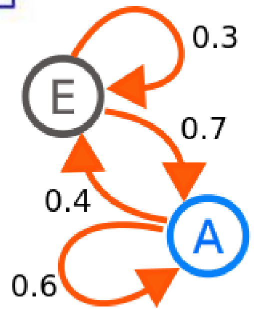
  - Random Works

- Stationary: $\pi = \pi P$

$$[\pi_1 \ \cdots \pi_n] = [\pi_1 \ \cdots \pi_n]\begin{bmatrix} p_{11} & \cdots & p_{11} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix}$$



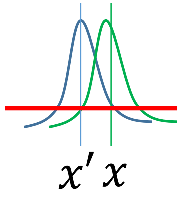
$$p_{AA} = P(X_{t+1} = A \mid X_t = A) = 0.6$$
$$p_{AE} = P(X_{t+1} = E \mid X_t = A) = 0.4$$
$$p_{EA} = P(X_{t+1} = A \mid X_t = E) = 0.7$$
$$p_{EE} = P(X_{t+1} = E \mid X_t = E) = 0.3$$

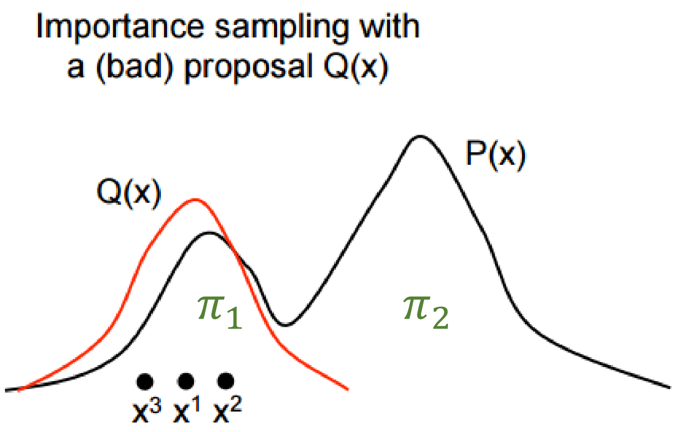


Markov Chain Monte Carlo

- MCMC algorithm feature adaptive proposals
  - Instead of $Q(x')$, they use $Q(x'|x)$ where x' is the new state being sampled, and x is the previous sample
  - As x changes, $Q(x'|x)$ can also change (as a function of x')
  - Stationary condition: the acceptance probability is $\text{set to } A(x'|x) = \min\left(1, \frac{P(x')/Q(x'|x)}{P(x)/Q(x|x')}\right)$ importance
  - No matter where we start, after some time, we will be in any state j with probability $\sim\pi_j$

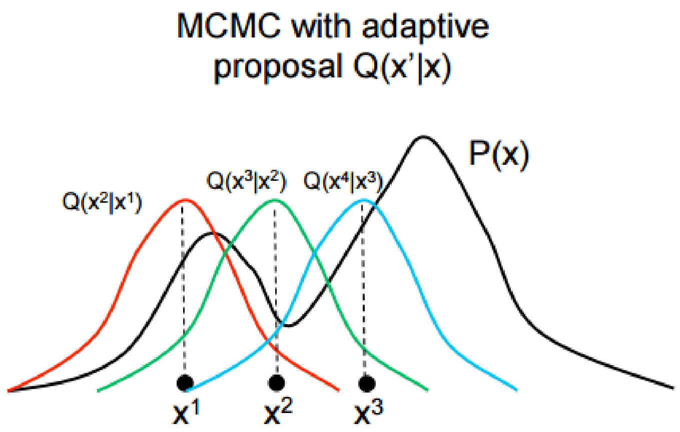
$Q(x'|x) = Q(x|x')$ for Gaussian Why?





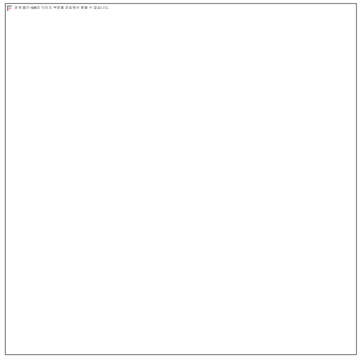
$p_{11} \quad \rightarrow p_{12} \quad p_{22}$

$\leftarrow p_{21}$



$p_{11} \quad \rightarrow p_{12} \quad p_{22}$

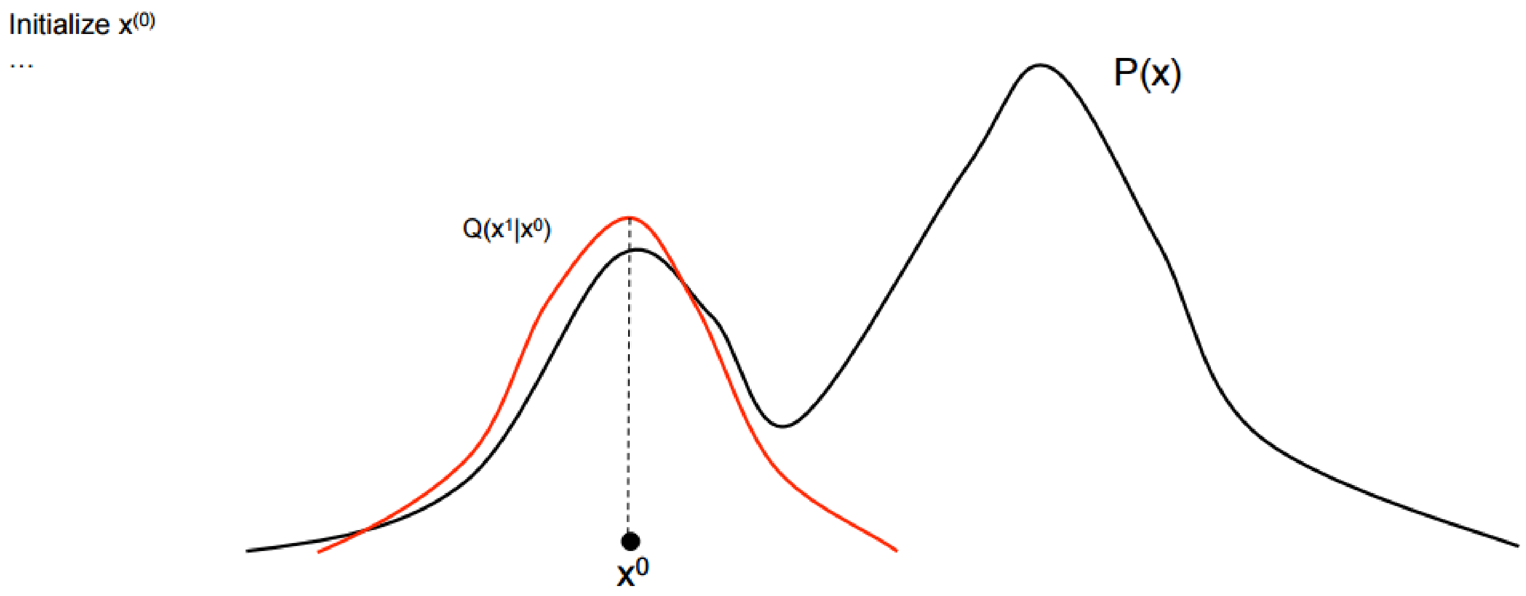
$\leftarrow p_{21}$



The MH Algorithm

- Example:
  - Let $Q(x'|x)$ be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution $P(x)$

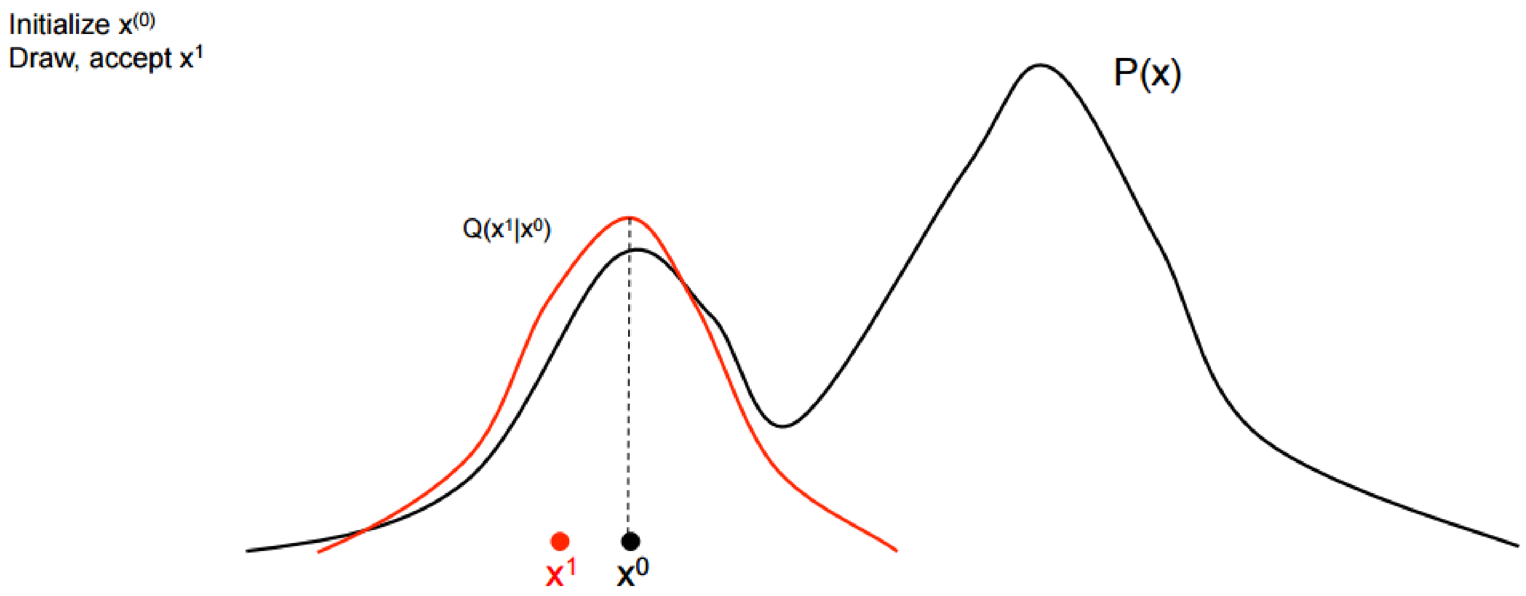
$$A(x'|x) = \min\left(1, \frac{P(x')/Q(x'|x)}{P(x)/Q(x|x')}\right)$$



The MH Algorithm

- Example:
  - Let $Q(x'|x)$ be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution $P(x)$

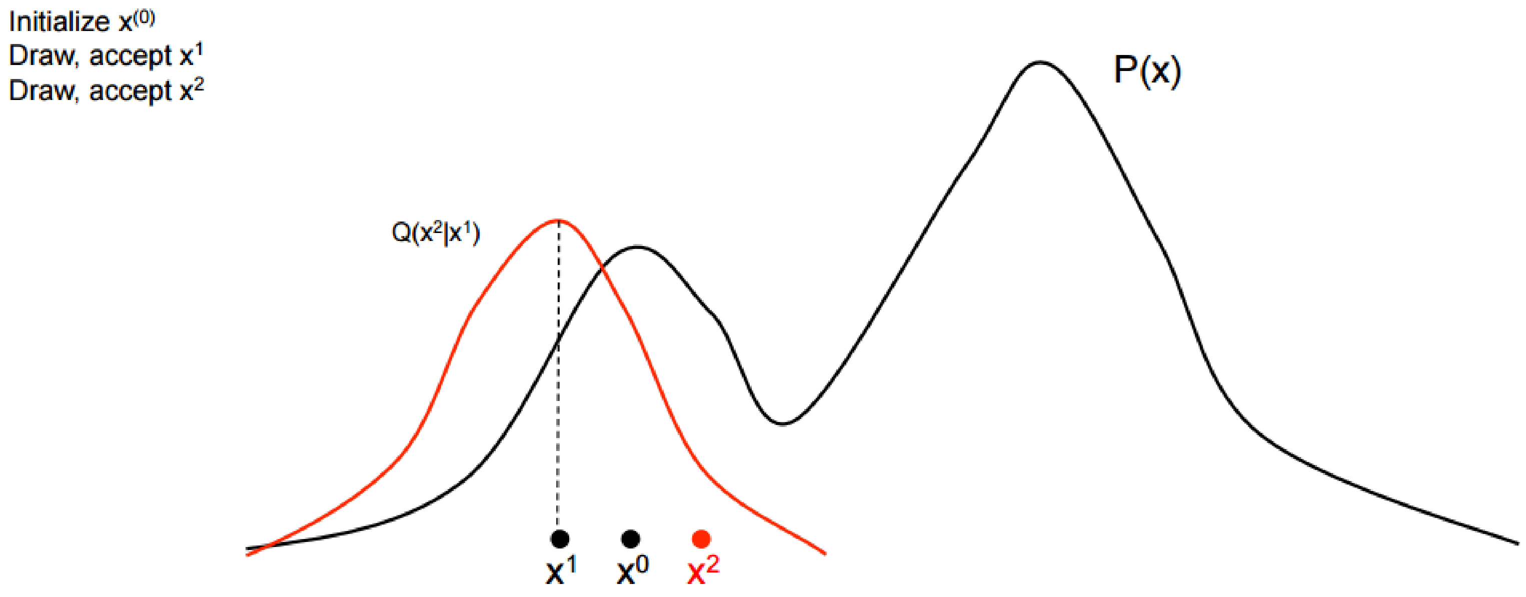
$$A(x'|x) = \min\left(1, \frac{P(x')/Q(x'|x)}{P(x)/Q(x|x')}\right)$$



The MH Algorithm

- Example:
  - Let $Q(x'|x)$ be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution $P(x)$

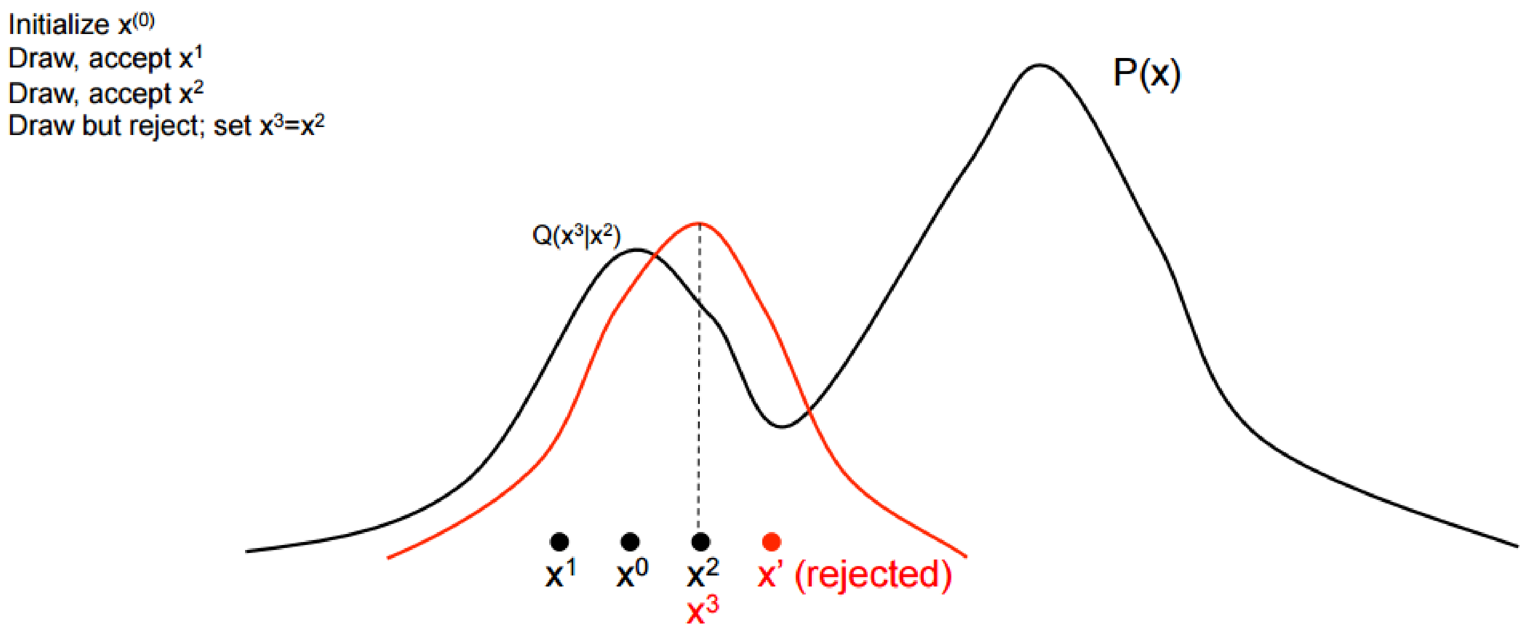
$$\boxed{A(x'|x) = \min\left(1, \frac{P(x')/Q(x'|x)}{P(x)/Q(x|x')}\right)}$$



The MH Algorithm

- Example:
  - Let $Q(x'|x)$ be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution $P(x)$

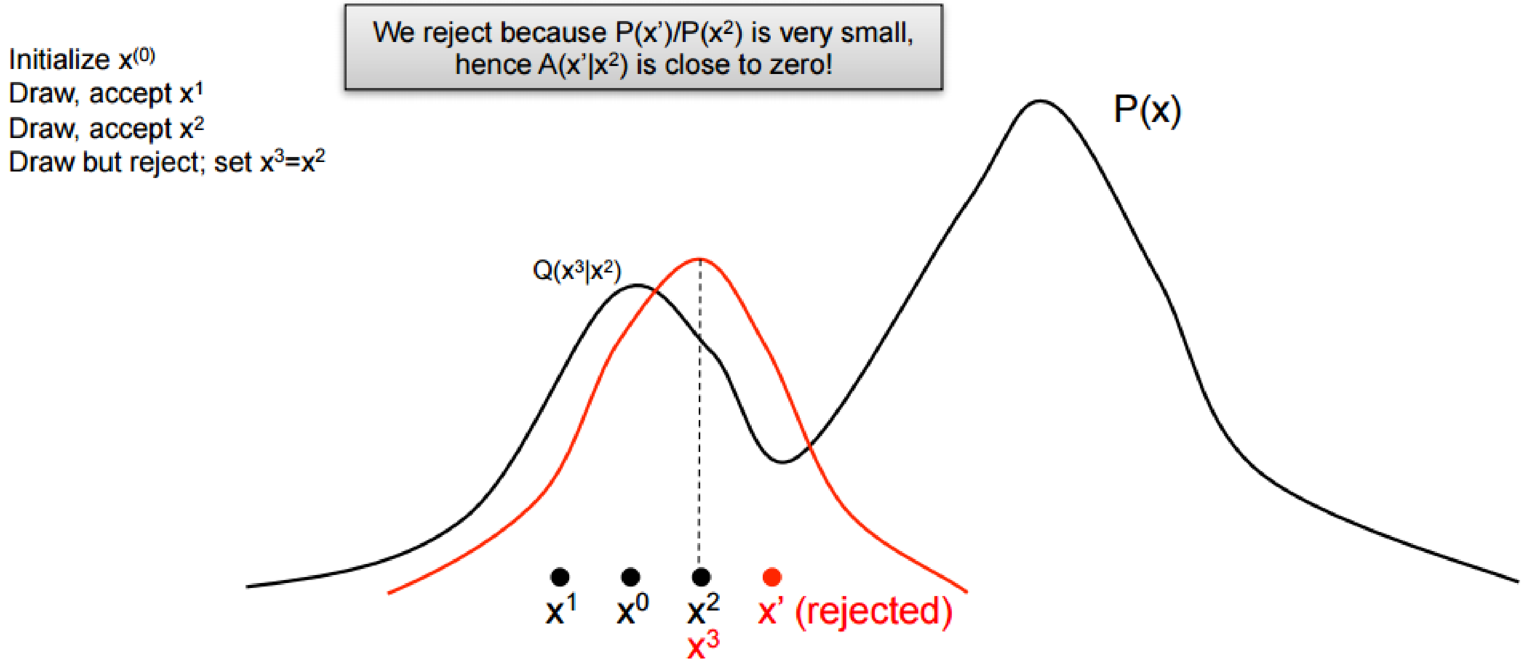
$$A(x'|x) = \min\left(1, \frac{P(x')/Q(x'|x)}{P(x)/Q(x|x')}\right)$$



The MH Algorithm

- Example:
  - Let $Q(x'|x)$ be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution $P(x)$

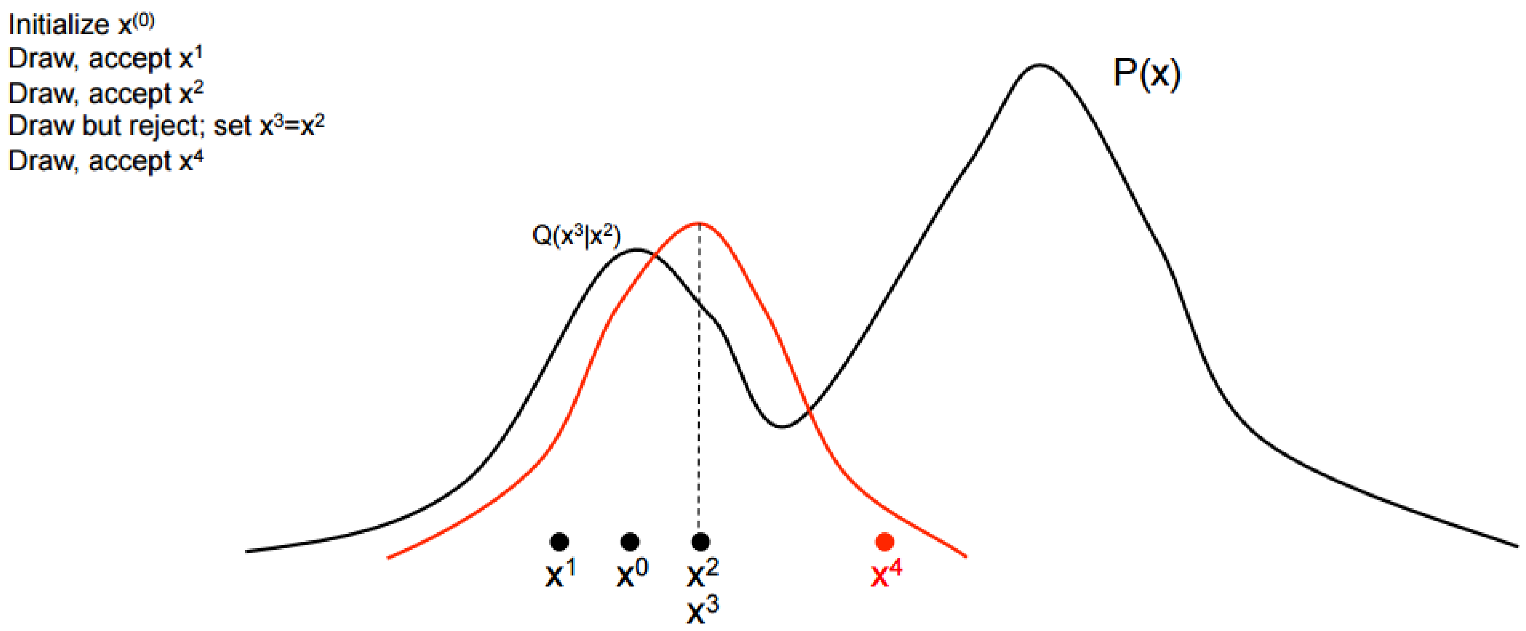
$$\boxed{A(x'|x) = \min\left(1, \frac{P(x')/Q(x'|x)}{P(x)/Q(x|x')}\right)}$$



The MH Algorithm

- Example:
  - Let $Q(x'|x)$ be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution $P(x)$

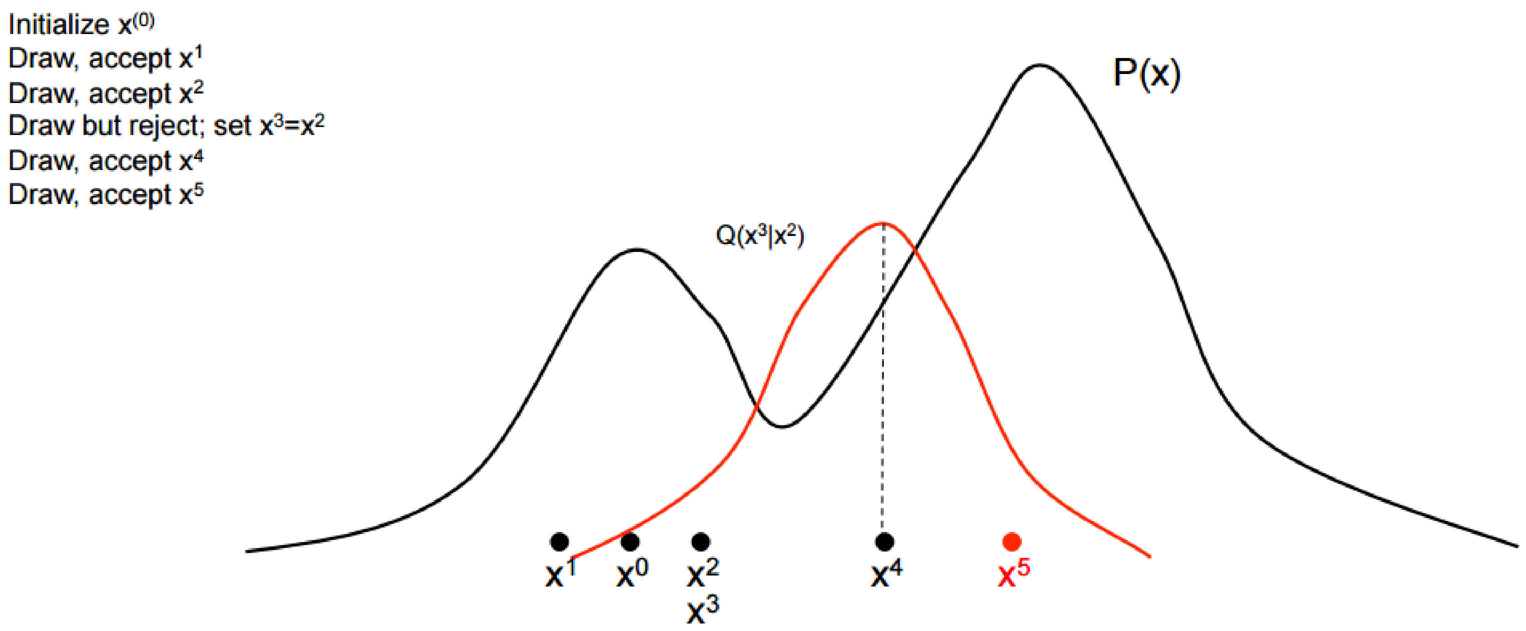
$$A(x'|x) = \min\left(1, \frac{P(x')/Q(x'|x)}{P(x)/Q(x|x')}\right)$$



The MH Algorithm

- Example:
  - Let $Q(x'|x)$ be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution $P(x)$

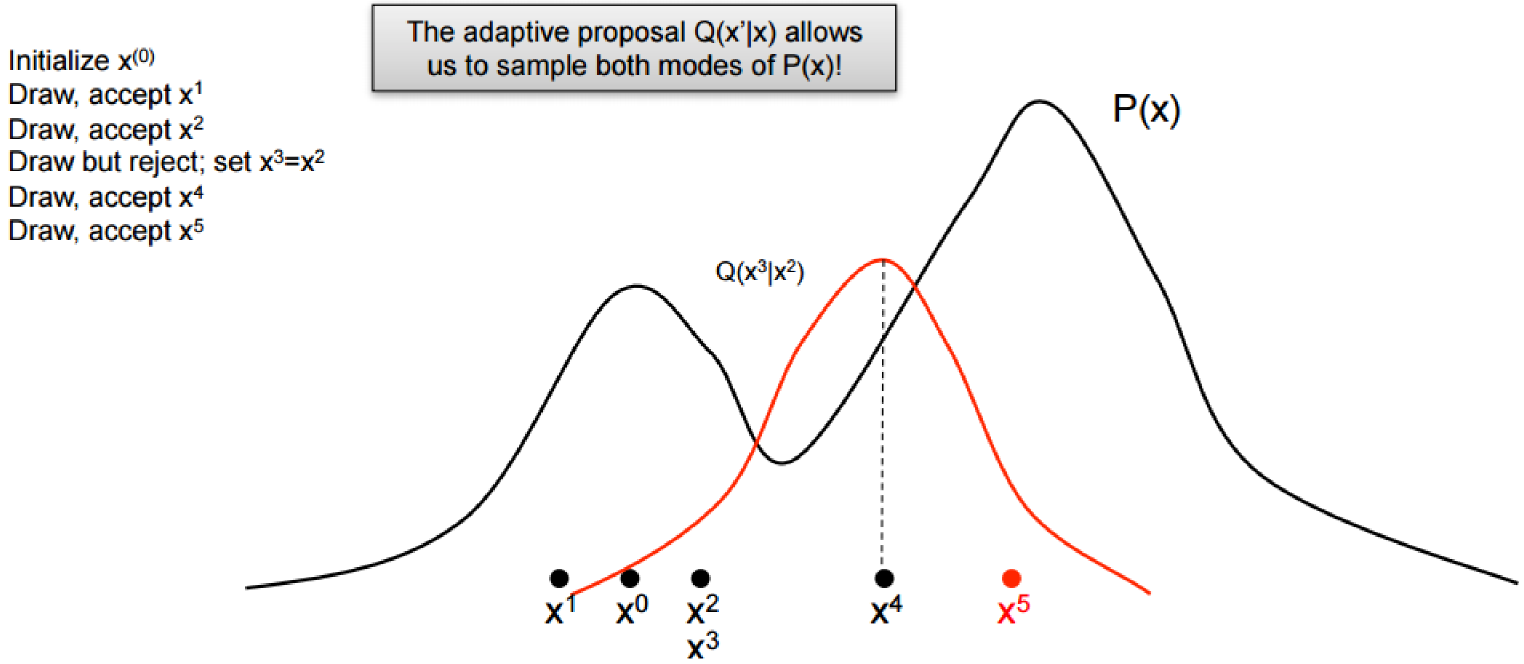
$$A(x'|x) = \min\left(1, \frac{P(x')/Q(x'|x)}{P(x)/Q(x|x')}\right)$$

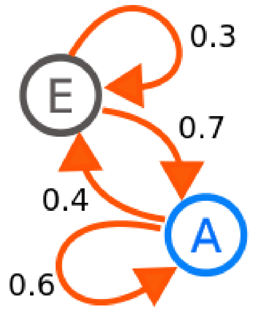


The MH Algorithm

- Example:
  - Let $Q(x'|x)$ be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution $P(x)$

$$A(x'|x) = \min\left(1, \frac{P(x')/Q(x'|x)}{P(x)/Q(x|x')}\right)$$





p_{11} $\rightarrow p_{12}$ p_{22}

$\leftarrow p_{21}$

The MH Algorithm

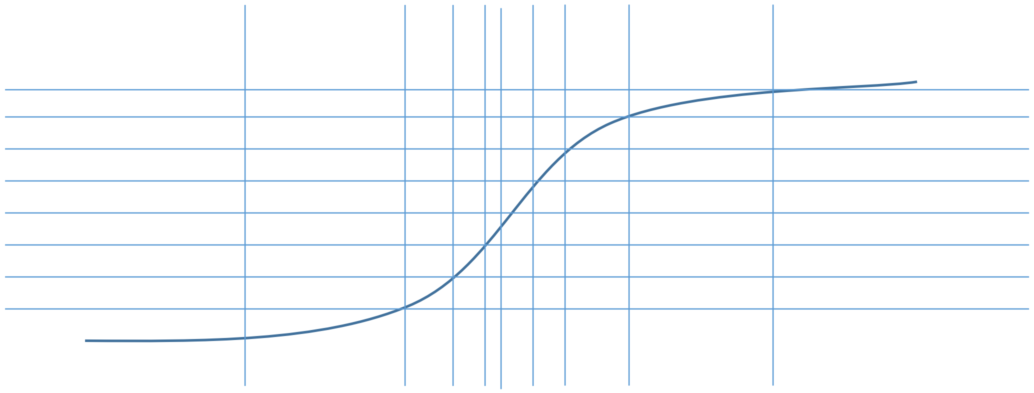
- Initialize starting state $x^{(0)}$,
- Burn-in: while samples have "not converged"
  - $x = x^{(t)}$
  - $t = t + 1$
  - Sample $x^* \sim Q(x^*|x)$ // draw from proposal
  - Sample $u \sim \text{Uniform}(0,1)$ // draw acceptance threshold
    - If $u < A(x^*|x) = \min\left(1, \frac{P(x^*)Q(x|x^*)}{P(x)Q(x^*|x)}\right)$, $x^{(t)} = x^*$ // transition
    - Else $x^{(t)} = x$ // stay in current state
  - Repeat until converging ($E_Q\left[\frac{h(X)P(X)}{Q(X)}\right] = \frac{1}{N}\sum_{s=1}^{N}\frac{h(x^{(s)})P(x^{(s)})}{Q(x^{(s)})}$)

Gibbs Sampling

- Direct (unconditional) sampling
  - Hard to get rare events in high-dimensional spaces → Gibbs sampling

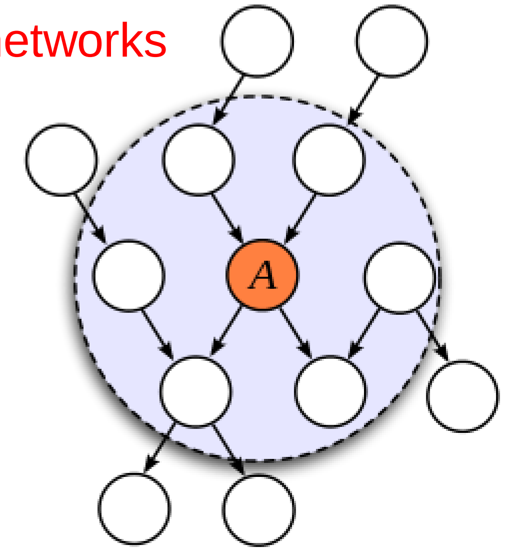
- Gibbs Sampling is an MCMC algorithm that is a special case of the MH algorithm

- Consider a factored state space
  - $x \in \Omega$ is a vector $x = (x_1, \ldots, x_m)$
  - Notation: $x_{-i} = \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m\}$



Gibbs Sampling

Bayesian networks



- The GS algorithm:
  1. Suppose the graphical model contains variables $x_1, \dots, x_n$
  2. Initialize starting values for $x_1, \dots, x_n$
  3. Do until convergence:
     1. Pick a component $i \in \{1, \dots, n\}$
     2. Sample value of $z \sim P(x_i | x_{-i})$, and update $x_i \leftarrow z$

- When we update x_i, we immediately use its new value for sampling other variables x_j

  $P(x_i | x_{-i})$ achieves the acceptance probability in MH algorithm.

$$A(x'|x) = \min\left(1, \frac{P(x')/Q(x'|x)}{P(x)/Q(x|x')}\right)$$

$$A(x_i', x_{-i} | x_i, x_{-i}) = \min\left(1, \frac{P(x_i', x_{-i})/P(x_i', x_{-i} | x_i, x_{-i})}{P(x_i, x_{-i})/P(x_i, x_{-i} | x_i', x_{-i})}\right)$$

$$= \min\left(1, \frac{P(x_i', x_{-i})/P(x_i', x_{-i})}{P(x_i, x_{-i})/P(x_i, x_{-i})}\right)$$

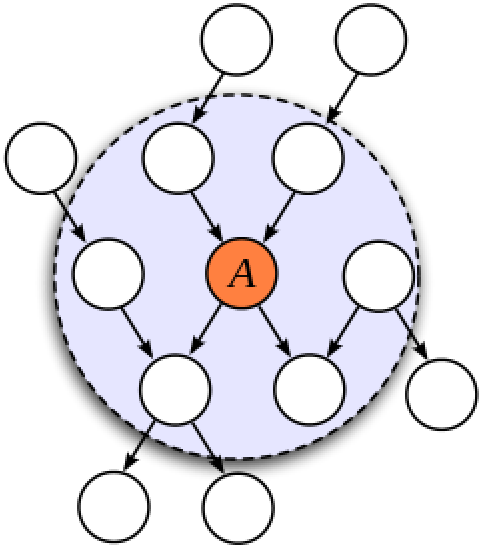
$\because x_i', x_i$ are independent

Markov Blankets

- The conditional $P(x_i|x_{-i})$ can be obtained using Markov Blanket
  - Let $MB(x_i)$ be the Markov Blanket of x_i, then

$$P(x_i \mid x_{-i}) = P\big(x_i|\mathrm{MB}(x_i)\big)$$

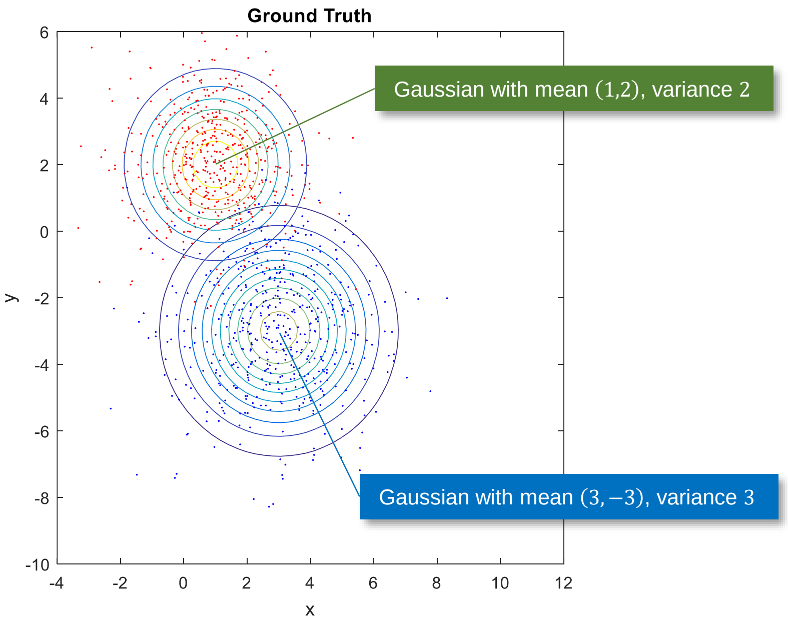
- For a Bayesian Network, the Markov Blanket of x_i is the set containing its parents, children, and co-parents

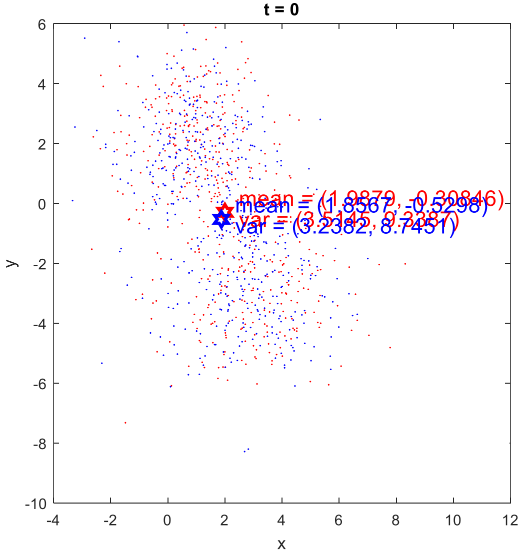


Gibbs Sampling: An Example

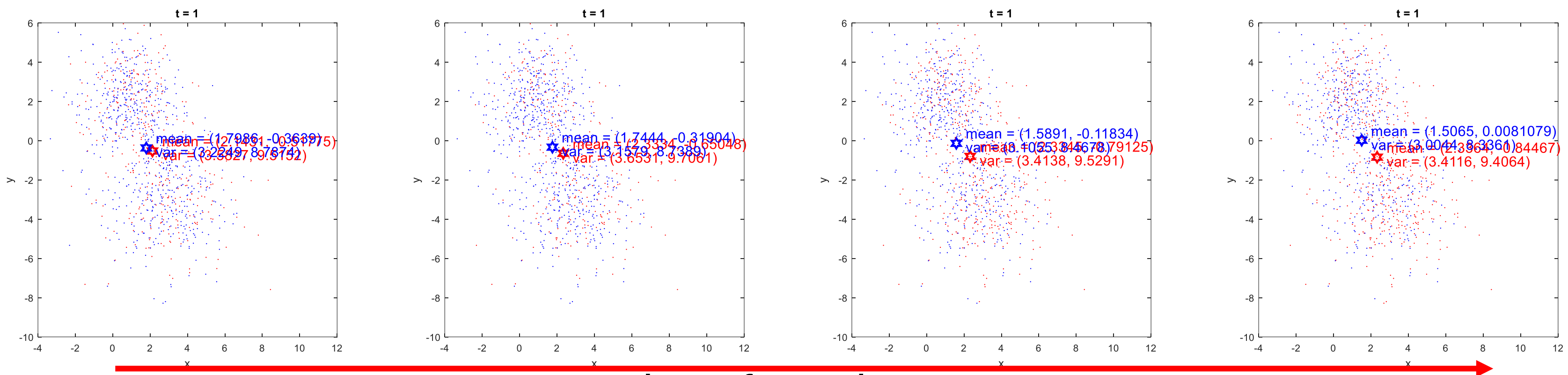
- Consider the GMM
  - The data x (position) are extracted from two Gaussian distribution
  - We do NOT know the class y of each data, and information of the Gaussian distribution
  - Initialize the class of each data at $t = 0$ to randomly

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} p(\mathbf{x}|\theta_k)p(\theta_k|\theta) = \sum_{k=1}^{K} p(\mathbf{x}|k)p(k|\theta) = \sum_{k=1}^{K} p(\mathbf{x}, k|\theta)$$





Gibbs Sampling: An Example

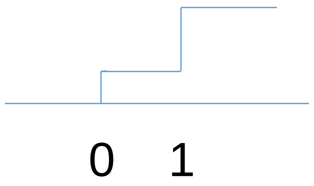


Iteration of i at the same t

Sampling $P(y_i \,|x_{-i}, y_{-i})$ at $t = 1$, we compute:

$$P(y_i = 0 \,|x_{-i}, y_{-i}) \propto \mathcal{N}\left(x_i|\mu_{x_{-i},0}, \sigma_{x_{-i},0}\right)$$

$$P(y_i = 1 \,|\, x_{-i}, y_{-i}) \propto \mathcal{N}\left(x_i|\mu_{x_{-i},1}, \sigma_{x_{-i},1}\right)$$



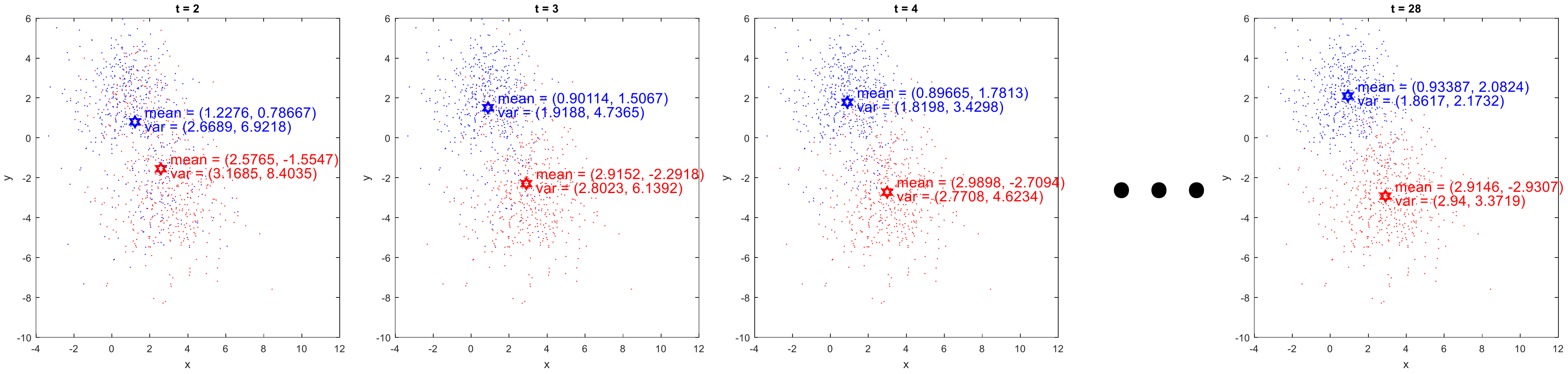
where

$$\mu_{x_{-i},K} = MEAN(X_{iK}), \sigma_{x_{-i},K} = VAR(X_{iK})$$

$$X_{iK} = \{x_j \,|\, x_j \in x_{-i}, y_j = K\}$$

And update y_i with $P(y_i \,|x_{-i}, y_{-i})$ and repeat for all data

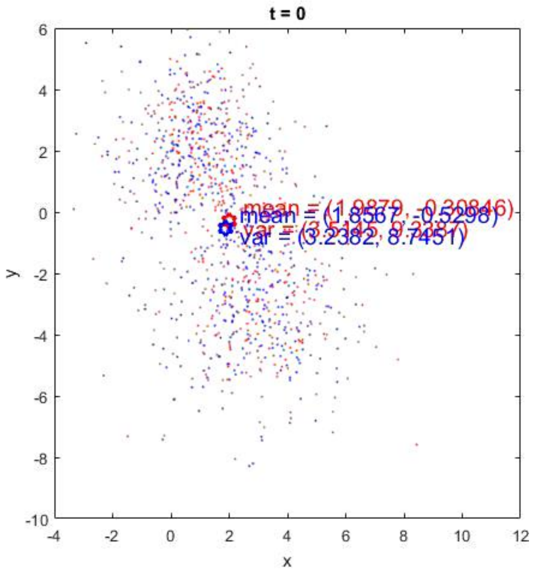
Gibbs Sampling: An Example

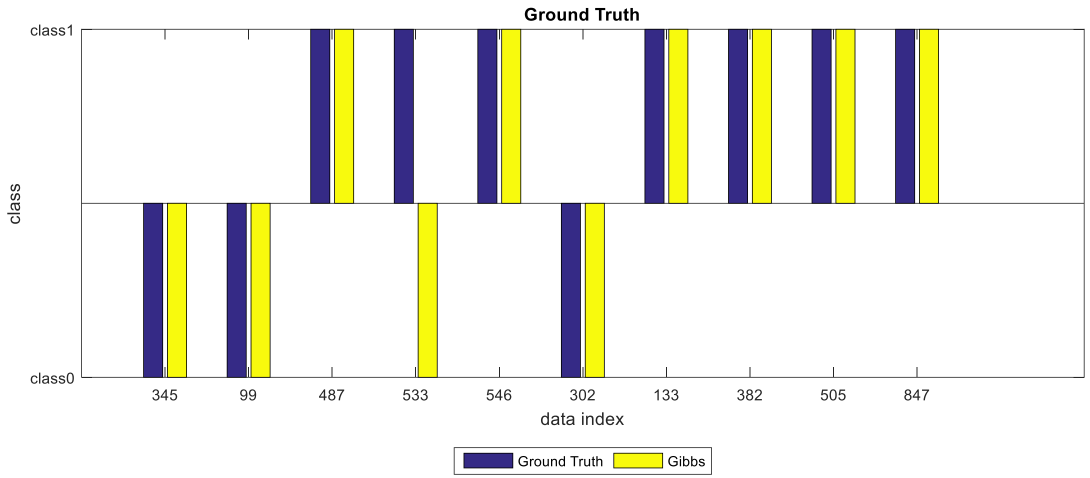


Now $t = 2$, and we repeat the procedure to sample new class of each data

And similarly for $t = 3, 4, \dots$

Gibbs Sampling: An Example





- Data i's class can be chosen with tendency of y_i
  - The classes of the data can be oscillated after the sufficient sequences
  - We can assume the class of datum as more frequently selected class

- In the simulation, the final class is correct with the probability of 94.9% at $t = 100$

Gaussian Process Regression

- K-means Clustering
- EM-Algorithm
- MCMC

data
tsp_dataset
Density Estimation.ipynb
GP Regression.ipynb
TSP_Climate_CNN.ipynb
TSP_Climate_data_preprocessing.ipynb
TSP_Climate_LSTM.ipynb
TSP_PLRegression.ipynb
TSP_WRLS.ipynb

(Un/Semi/Self-)Supervised Learning

- Supervised Learning
  - Labeled Deep learning
  - Labeled Density Estimation (Parametric)
- Un/Semi/Self-supervised Learning
  - Clustering
  - Bayesian Network Learning
  - Variational Auto-Encoder (VAE)
  - Active Learning (Uncertainty)
- Background Techniques
  - Entropy (Uncertainty)
  - Cross-Entropy, K-L Divergence
  - Bayesian Decision, Bayes Rule
  - Parametric Density Estimation (MLE, Bayesian Learning)
  - Non-parametric Density Estimation (EM, MCMC)