80250993 Machine Learning
@Tsinghua University

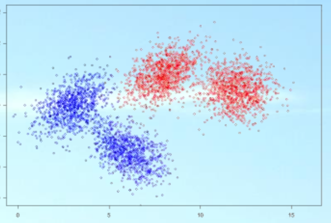
Lecture 14
Model-based Clustering

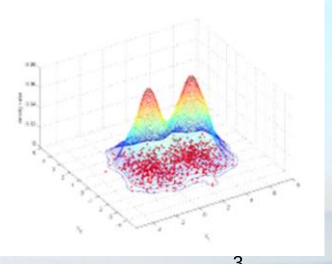
Xuegong Zhang
Nov. 18, 2021

14.1
Density Estimation as an Unsupervised Learning Task

Density Estimation
as a machine learning task

- Once we have learned the density
  - We can use the density for other learning tasks
  - We can infer relationships between features
  - We can infer properties of the population behind data
  - We can make predictions
  - …
  - We can do almost everything
- Learning the density of data is one of the tasks of **unsupervised learning**





Density Estimation: parametric estimation (revisit)

Recall: What is learning? —— Risk Minimization

Let the probability measure $F(z)$ be defined on the space Z. Consider the set of functions $Q(z,\alpha), \alpha \in \Lambda$. The goal is to find a function $Q(z,\alpha_0)$ that minimize the risk functional

$$R(\alpha) = \int Q(z,\alpha)dF(z), \ \ \alpha \in \Lambda,$$

where the probability measure $F(z)$ is unknown, but an i.i.d. sample

$$z_1, \cdots, z_l$$

is given.

Density Estimation as a learning problem

- Set of density functions $f(x,\alpha), \alpha \in \Lambda$
- Loss function $L(p(x,\alpha)) = -\log p\,(x,\alpha)$ or $\lambda(\hat{\theta},\theta) = \left(\theta - \hat{\theta}\right)^2$
- Risk: $R(\alpha) = \int L(f(x,\alpha))dF(x)$
- Density estimation:

To minimize the risk functional when the corresponding probability measure $F(x)$ is unknown, but i.i.d. data $x_1, \cdots, x_n$ is given.

- Likelihood Function $l(\theta) = p(\mathrm{X}|\theta) = p(x_1, x_2, \cdots, x_N|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$

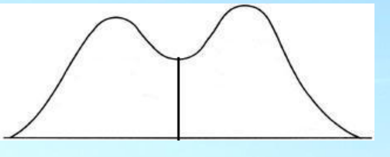
  $H(\theta) = \ln l\,(\theta)$
- Maximum Likelihood Estimation: $\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, H(\theta)$
- Bayesian Estimation:

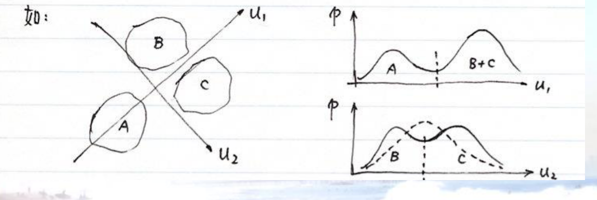
$$\hat{\theta} = \int_{\Theta} \theta p(\theta|\boldsymbol{X})d\theta, \ \ p(\theta|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\theta)p(\theta)}{\int_{\Theta} p(\boldsymbol{X}|\theta)p(\theta)d\theta}$$

14.2
Clustering by Density Partitioning

Clustering as Separation of Density Peaks

- Assumption
  - Each peak in the total density function represents a cluster
- Tasks:
  - How to estimate the density (in high-D)?
  - How to find the peaks?
- Projection Method:
  - Project samples onto a certain coordinate
  - Estimate the (marginal) density on the projected coordinate
  - Find peaks in the marginal density to divide the clusters





- Projection Method for Peak Clustering

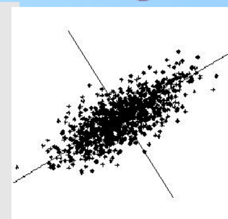
① Find the potentially most information coordinate u_j (via PCA etc.), project all samples onto $v_j = u_j^T x$

② Calculate the histogram (or other types of density estimation) $p(v_j)$ on $v_j = u_j^T x$

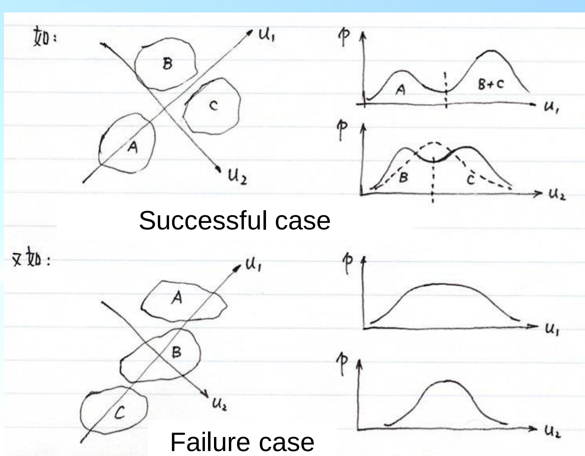
③ Find the minimums (valleys) in $p(v_j)$, and use the hyperplane orthogonal to u_j at the minimum points to divide the samples

④ If no valley found, repeat ② and ③ using another candidate coordinate

⑤ For samples in each division, do the above projects to find sub-divisions, until no valleys found in any coordinates



Examples:



Successful case

Failure case

5-second break

14.3
Mixture Models and the EM Algorithm

14.3.1
Estimation of Mixture Densities

Recall: Maximum Likelihood Estimation

- Set of the problem
  - ① Samples $\mathcal{X}_i, i = 1, \cdots, c$ are i.i.d. from density $p(x|\omega_i)$
  - ② $p(x|\omega_i)$ is of the density function form of $p(x|\theta_i)$, in which only the parameters θ_i are unknown
  - ③ Parameters θ_i are unknown deterministic vectors
  - **④ Samples only contain information of parameters of the same class.**

  Problem:

  To estimate θ_i from given samples

What if samples from c classes are mixed?

D

- Likelihood function
  - Mixture Density: $p(\boldsymbol{x}|\theta) = \sum_{i=1}^{c} p(\boldsymbol{x}|\omega_i, \theta_i)P(\omega_i)$
  - Component density: $p(\boldsymbol{x}|\omega_i, \theta_i)$
  - Mixing parameters: $P(\omega_i)$
  - Likelihood function

$$l(\theta) = p(\boldsymbol{\mathcal{X}}|\theta) = p(x_1, x_2, \cdots, x_N|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$$

$$H(\theta) = \ln l(\theta)$$

  - ML Estimation: $\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, H(\theta)$

→Everything is just the same.

→Even we can estimate $P(\omega_i)$ together.

So easy? Too good to be true!

- Requirement: Identifiability
  - PDF forms known (as in all parametric estimation)
  - $\theta \neq \theta'$ implies that there exist an $\boldsymbol{x}$ that $p(\boldsymbol{x}|\theta) \neq p(\boldsymbol{x}|\theta')$
- Identifiability is a property of the model
  - Most mixtures of continuous PDF models are identifiable
  - But mixtures of discrete distributions are not always

The Maximum Likelihood Estimation

- Solution is the same as in supervised cases

If $p(x|\theta)$ is differentiable to θ, let $\nabla_\theta H(\theta) = 0$ for

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

where $p(x|\theta) = \sum_{i=1}^{c} p(x|\omega_i, \theta_i) P(\omega_i)$,
we have

$$\nabla_{\theta_i} H(\theta) = \sum_{k=1}^{N} \frac{1}{p(x_k|\theta)} \nabla_{\theta_i} \left[\sum_{j=1}^{c} p(x_k|\omega_j, \theta_j) P(\omega_j)\right]$$

$$= \sum_{k=1}^{N} \frac{1}{p(x_k|\theta)} \nabla_{\theta_i} [p(x_k|\omega_i, \theta_i) P(\omega_i)]$$

Hint: assume θ_i and θ_j independent

$$= \sum_{k=1}^{N} P(\omega_i|x_k, \theta_i) \nabla_{\theta_i} \ln p(x_k|\omega_i, \theta_i)$$

$$\triangleq 0$$

Hint: $P(\omega_i|x_k, \theta_i) = \frac{p(x_k|\omega_i, \theta_i) P(\omega_i)}{p(x_k|\theta)}$

- If $P(\omega_i)$ unknown, we can also take them as parameters to be estimated, and define the task as

$$\max H(\theta)$$

$$\text{s.t. } P(\omega_i) > 0, \quad i = 1, \cdots, c \text{ and } \sum_{i=1}^{c} P(\omega_i) = 1$$

- Define the Lagrangian as $H' = H + \lambda[\sum_{i=1}^{c} P(\omega_i) - 1]$

we can have

$$\hat{P}(\omega_i) = \frac{1}{N} \sum_{k=1}^{N} \hat{P}(\omega_i|x_k, \hat{\theta}_i)$$

$$\sum_{k=1}^{N} \hat{P}(\omega_i|x_k, \hat{\theta}_i) \nabla_{\theta_i} \ln p(x_k|\omega_i, \hat{\theta}_i) = 0 \;, \quad i = 1, \cdots, c$$

where

$$\hat{P}(\omega_i|x_k, \hat{\theta}_i) = \frac{p(x_k|\omega_i, \hat{\theta}_i) \hat{P}(\omega_i)}{\sum_{j=1}^{c} p(x_k|\omega_j, \hat{\theta}_j) \hat{P}(\omega_j)}$$

which can be solved in an iterative manner.

14.3.2
Gaussian Mixture Models (GMMs)

The Gaussian Case

- Simplest scenario: only μ_i unknown, $\Sigma_i, P(\omega_i), c$ all known

The equations are

$$\sum_{k=1}^{N} \hat{P}(\omega_i|x_k,\hat{\theta}_i)\nabla_{\theta_i} \ln p\,(x_k|\omega_i,\hat{\theta}_i) = 0, \quad i = 1,\cdots,c$$

Put the Gaussian in, we have the equations

$$\sum_{k=1}^{N} P(\omega_i|x_k,\hat{\mu}_i)\,\Sigma_i^{-1}(x_k - \hat{\mu}_i) = 0, \qquad i = 1,\cdots,c$$

which give

$$\hat{\mu}_i = \frac{\sum_{k=1}^{N} P(\omega_i|x_k,\hat{\mu}_i)x_k}{\sum_{k=1}^{N} P(\omega_i|x_k,\hat{\mu}_i)}$$

and

$$\hat{P}(\omega_i|x_k,\hat{\mu}_i) = \frac{p(x_k|\omega_i,\hat{\mu}_i)P(\omega_i)}{\sum_{j=1}^{c} p(x_k|\omega_j,\hat{\mu}_j)P(\omega_j)}$$

which has a clear meaning but can only be solved in an iterative manner.

Iterative Solution:

– Initialize some $\hat{\mu}_i(0)$, and then repeat

$$\hat{P}(\omega_i|x_k,\hat{\mu}_i)=\frac{p(x_k|\omega_i,\hat{\mu}_i)P(\omega_i)}{\sum_{j=1}^{c}p\left(x_k\middle|\omega_j,\hat{\mu}_j\right)P\left(\omega_j\right)}$$

$$\hat{\mu}_i(j+1)=\frac{\sum_{k=1}^{N}P(\omega_i|x_k,\hat{\mu}_i(j))x_k}{\sum_{k=1}^{N}P(\omega_i|x_k,\hat{\mu}_i(j))}$$

—— Gradient decent method in nature, may not converge to global optimal, sensitive to initialization.

→ EM Algorithm

- If $\mu_i, \Sigma_i, P(\omega_i)$ are all unknown, with only c known,
  - The same approach can also be used for an iterative solution, with more complicated forms but also clear conceptual explanation.

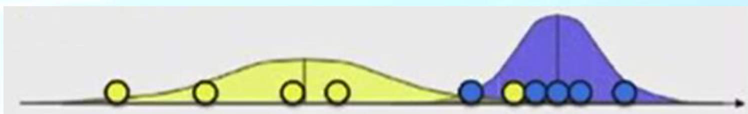
Gaussian Mixture Models (GMMs)

Revisit the estimation of mixture densities

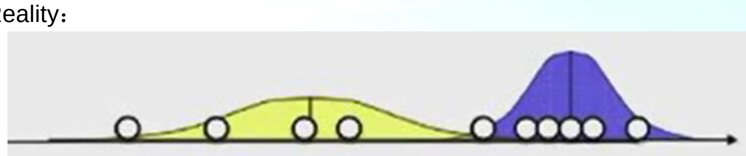
$$P(\boldsymbol{x}|\theta) = \sum_{i=1}^{c} p(\boldsymbol{x}|\omega_i, \theta_i) P(\omega_i)$$

- Suppose data in each class follow a 1D Gaussian model

Dream：

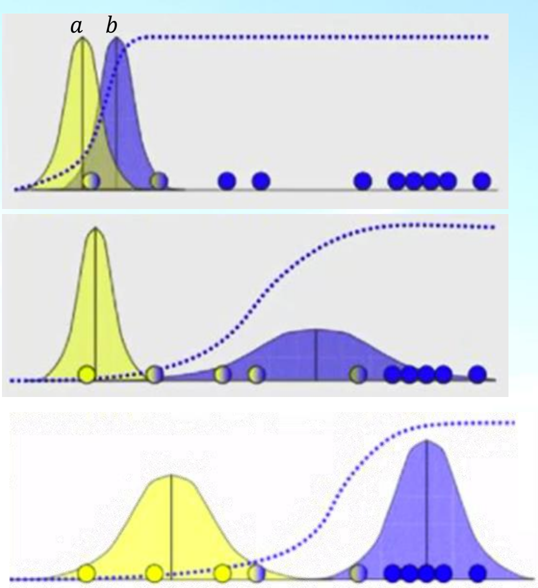


Reality：



1D Illustration of EM for Mixture Models

Self-study



$$P(x_i|b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x_i-\mu_b)^2}{2\sigma_b^2}\right)$$

$$b_i = P(b|x_i) = \frac{P(x_i|b)P(b)}{P(x_i|b)P(b)+P(x_i|a)P(a)}$$

$$a_i = P(a|x_i) = 1 - b_i$$

$$\mu_b = \frac{b_1x_1+b_2x_2+\cdots+b_nx_n}{b_1+b_2+\cdots+b_n}$$

$$\sigma_b^2 = \frac{b_1(x_1-\mu_b)^2+b_2(x_2-\mu_b)^2+\cdots+b_n(x_n-\mu_b)^2}{b_1+b_2+\cdots+b_n}$$

$$\mu_a = \frac{ax_1+a_2x_2+\cdots+a_nx_n}{a_1+a_2+\cdots+a_n}$$

$$\sigma_a^2 = \frac{a_1(x_1-\mu_a)^2+a_2(x_2-\mu_a)^2+\cdots+a_n(x_n-\mu_a)^2}{a_1+a_2+\cdots+a}$$

清华大学

10-second break

清华大学

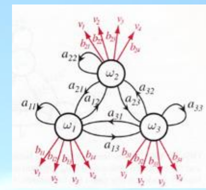
14.3.3
The Expectation-Maximization (EM) Algorithm

W

Parameter estimation for HMMs
(the *learning* problem)

- The framework:
  - A set of example sequences (training sequences)
    $$x^1, \dots, x^n$$
    which are independent.
  - Maximum Likelihood Method:
    - The joint probability of all the sequences given a particular assignment of parameter
  - Log likelihood of the sequences given the model:
    $$l(x^1, \cdots, x^n|\theta) = \log P\,(x^1, \cdots, x^n|\theta) = \sum_{j=1}^{n} \log P\,(x^j|\theta)$$

Xuegong Zhang 36



$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}}$$

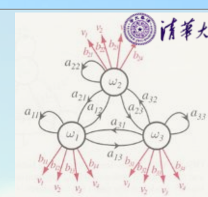
$$e_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')}$$

$$P(x,\pi) = a_{0\pi_1} \prod_{i=1}^{L} e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}}$$

Xuegong Zhang 27

When paths are unknown

Baum-Welch Algorithm ---- an EM algorithm



- Initialization: pick arbitrary model parameters
- Recurrence:
  - Set all *A* and *E* variables to their *pseudocount* values *r* (or 0)
  - For each sequence *j*=1,…,*n*:
    - Calculate $f_k(i)$ for sequence *j* using the forward algorithm
    - Calculate $b_k(i)$ for sequence *j* using the backward algorithm
    - Add the contribution of sequence *j* to *A* and *E*

The probability that a_{kl} is used (all positions, all sequences)

$$A_{kl} = \sum_j \frac{1}{P(x^j)} \sum_i f_k^j(i) a_{kl} e_l(x_{i+1}^j) b_l^j(i+1)$$

\# of times the letter *b* appears in state *k*

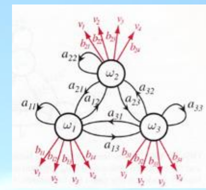
$$E_k(b) = \sum_j \frac{1}{P(x^j)} \sum_{\{i|x_i^j=b\}} f_k^j(i) b_k^j(i)$$

  - Calculate the new model parameters
    $$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}}, \quad e_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')}$$
  - Calculate the new log likelihood of the model
- Termination:
  - Stop if the change in log likelihood is less than some predefined threshold, or the maximum number of iterations is exceeded.

Xuegong Zhang 28

When paths are unknown

$$P(x,\pi) = a_{0\pi_1}\prod_{i=1}^{L} e_{\pi_i}(x_i)a_{\pi_i\pi_{i+1}}$$



- Model with missing parameters
  - Unobserved latent variables
- Basic idea of the Baum-Welch Algorithm
  - First estimates the A_{kl} and $E_k(b)$ by considering probable paths for the training sequences using the current model — Expectation
  - Then derive new values of the parameters — Maximization
  - Iterating until some stopping criterion is reached

The general idea of EM algorithms

- A statistical model for X with unobserved latent data or missing values Z and a vector of unknown parameters θ.

$$p(X,Z;\theta)$$

- Likelihood of the observed data

$$L(\theta;X) = p(X|\theta) = \int p(X,Z|\theta)dZ$$

- However, we don't know Z and enumerating all possibilities is often infeasible.
- The idea is to make the estimation in two iterative steps:
  - Get the **expected** value of the log likelihood of θ based on some estimate of Z given X with the current estimation $\theta^{(t)}$ — Expectation (E-step)
  - Find the next estimation $\theta^{(t+1)}$ that **maximizes** this expected log likelihood — Maximization (M-step)

The general idea of EM algorithms

- Given: $X = \{x_1, \cdots, x_n\}$
- Model: $p(X, Z; \theta)$
- Goal: MLE (Maximum Likelihood Estimation) $\theta_{\text{MLE}} \in \underset{\theta}{\operatorname{argmax}}\, L(\theta; X)$

$$L(\theta; X) = p(X|\theta) = \int p(X, Z|\theta)\, dZ$$

- E-M algorithm:
  - Initialize $\theta_0 \in \Theta$
  - For $t = 0, 1, 2 \cdots$ (until convergence)
    - E-step: $Q(\theta, \theta_t) = \mathbb{E}_{Z_{\theta_t}}(\log p(X, Z; \theta)|X)$
    - M-step: $\theta_{t+1} \in \underset{\theta}{\operatorname{argmax}}\, Q(\theta, \theta_t)$

D

Gaussian Mixture Model

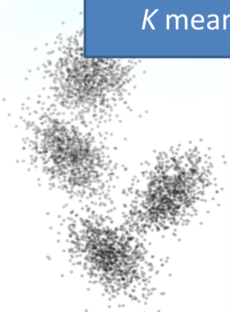
- The generative model:
  - Data generated from a K Gaussian models of unknown parameters

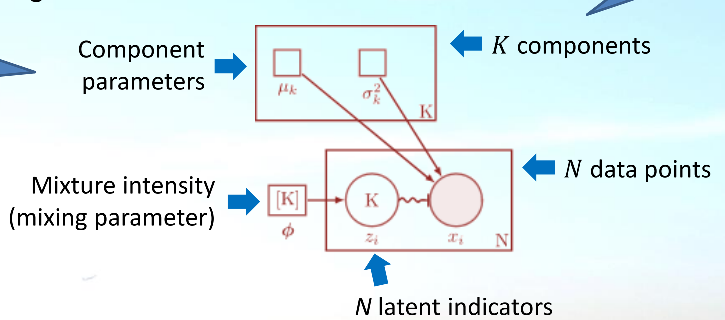
$$p(x) = \sum_{k=1}^{K} \phi_k\, \varphi(x|\mu_k, \sigma_k^2)$$

- The learning task: to learn the model from data

K clusters

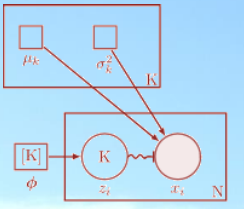
K means





EM Learning for GMM

- EM algorithm (K known)
  1. Initialization
  2. **E-step**: Calculate the probability a data point belonging to a component
  3. **M-step**: Estimate parameters associated with the components
  4. Until converge



Model selection: Deciding on the number of clusters

- When K unknown
  - Try different K, select the one minimize BIC (Bayesian Information Criterion) or AIC (Akaike Information Criterion).

Self-study

$$BIC = \ln(n)\,k - 2\ln(\hat{L})$$

$$AIC = 2k - 2\ln(\hat{L})$$

where $\hat{L} = p(x|\hat{\theta}, M)$ is the likelihood of the model, x are the observed data of number n, k is the number of parameters.

EM algorithm for Gaussian Mixtures

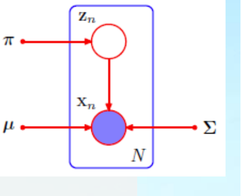
Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (the means μ_k, covariances Σ_k of the components and mixing coefficients π_k).

1. Initialize μ_k, Σ_k and π_k, and evaluate the initial value of the log likelihood
2. **E step**. Evaluate the *responsibilities* (posterior) using the current parameters

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

Responsibility:
$\gamma(z_{nk}) \equiv p(z_k = 1|x_n)$

3. **M step**. Re-estimate the parameters using the current responsibilities



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

$$\Sigma_k^{new} = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$$

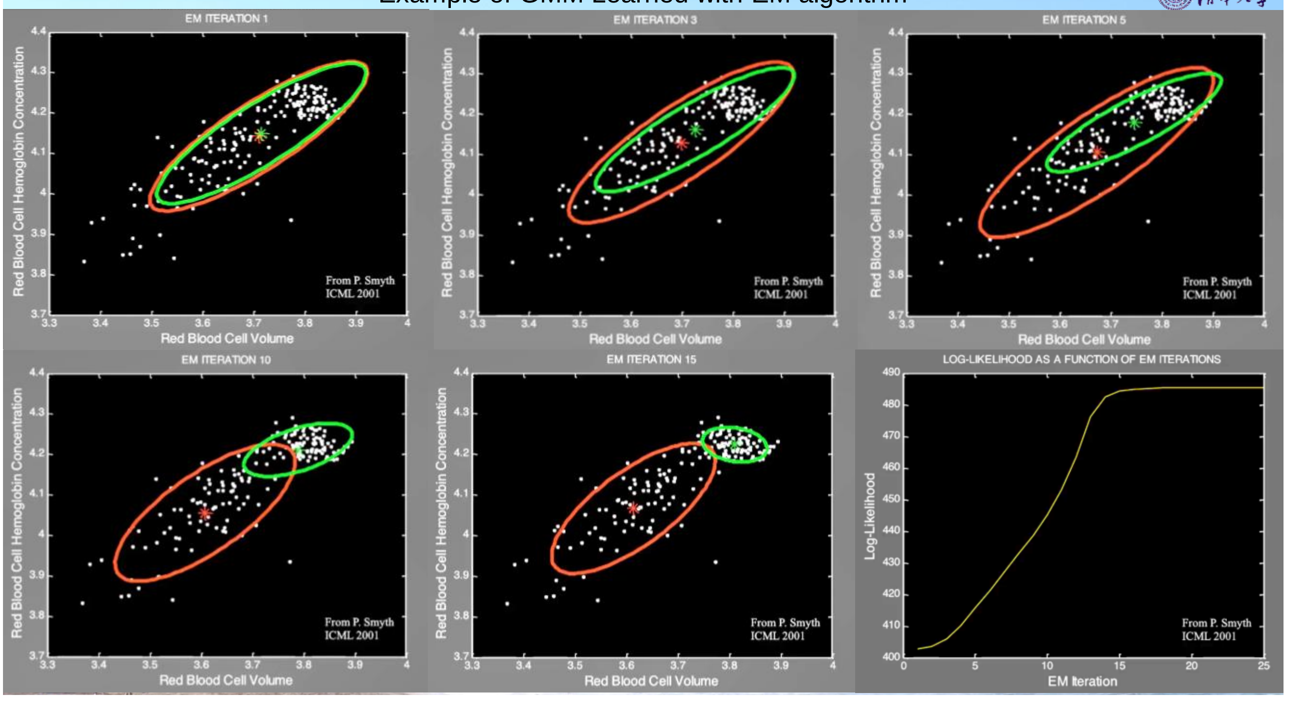
$$\pi_k^{new} = N_k/N$$

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

4. Evaluate the log likelihood $\ln p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{N} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)\}$ and check for convergence of either the parameters or the log likelihood. If the convergence is not satisfied, return to step 2.

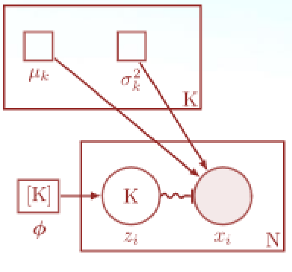
Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006, pp.438-439

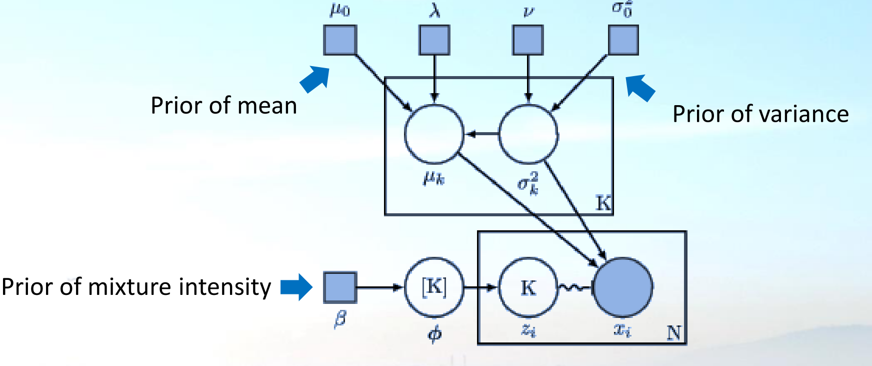
Example of GMM Learned with EM algorithm

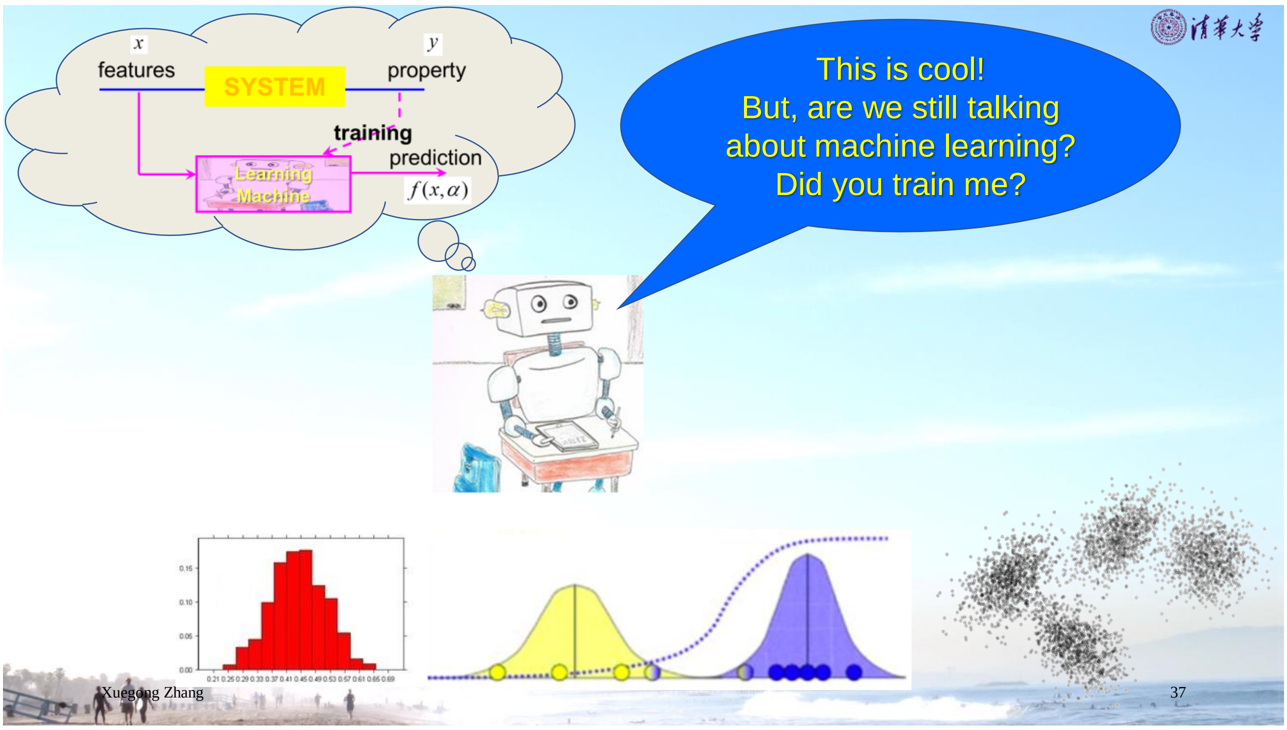


Bayesian treatment

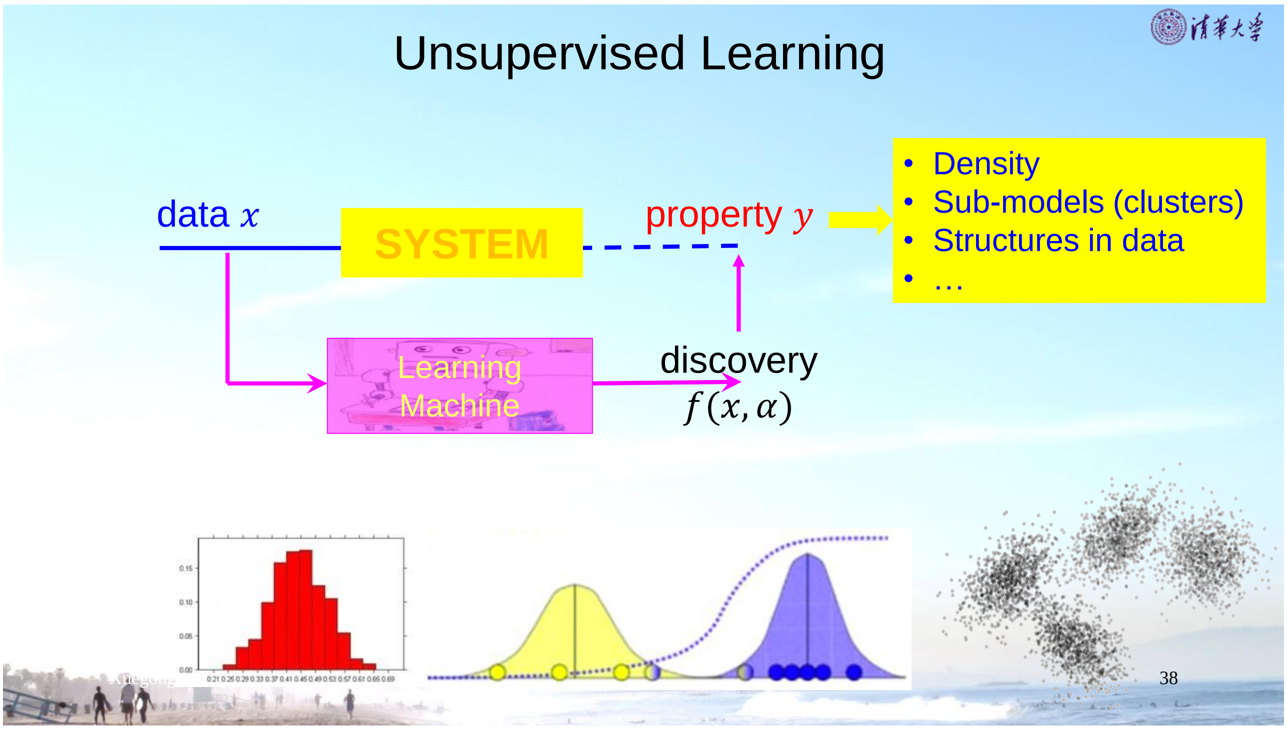
- Assume common prior distributions for different components
- Consider full distributions instead of maximum likelihoods
- Obtain Bayesian estimate via Markov chain Monte Carlo (MCMC) or Gibbs sampling







x
features
SYSTEM
y
property
training
prediction
Learning Machine
f(x,α)
This is cool!
But, are we still talking about machine learning?
Did you train me?
Xuegong Zhang
37



Unsupervised Learning
data x
SYSTEM
property y
• Density
• Sub-models (clusters)
• Structures in data
• …
Learning Machine
discovery
f(x,α)
38

Homework

- Problem Set (Pr.7)
  - Derivation of EM algorithm for GMM.
- Deadline:
  - Nov. 24 (Wednesday), 23:00

D

See you next week

for

Dimensionality Reduction and Embedding