UCLA Samueli Computer Science

CS M146 Discussion: Week 9

Naive Bayes, Clustering (K-Means, Gaussian Mixture Model)

Junheng Hao

Friday, 03/05/2021

UCLA
Engineer Change.

Roadmap

- Announcement
- Naive Bayes
- K-Means
- Gaussian Mixture Model

UCLA
Engineer Change.

Announcements

- **5:00 pm PST, Mar 5 (Friday):** Weekly Quiz 9 released on Gradescope.
- **11:59 pm PST, Mar 7 (Sunday):** Weekly Quiz 9 closed on Gradescope!
  - Start the quiz before **11:00 pm PST, Mar 7** to have the full 60-minute time
- **Grading update:** Lowest two quiz scores are dropped. The rest 7 quizzes are counted into final grading.
- **Problem set 4** released on CCLE, submission on Gradescope.
  - Please assign pages of your submission with corresponding problem set outline items on GradeScope.
  - You need to submit code and the results required by the problem set
  - Due on **next Friday, 11:59pm PST, Mar 12 (Friday)**

Late Submission of PS will NOT be accepted!

UCLA
Engineer Change.

About Quiz 9

- Quiz release date and time: **Mar 5, 2021 (Friday) 05:00 PM PST**
- Quiz due/close date and time: **Mar 7, 2021 (Sunday) 11:59 PM PST**
- You will have up to **60 minutes** to take this exam. → Start before **11:00 PM** Sunday
- You can find the exam entry named "Week 9 Quiz" on GradeScope.
- Topics: **Naive Bayes, Clustering**
- Question Types
  - True/false, multiple choices
  - Some questions may include several subquestions.
- Some light calculations are expected. Some scratch paper and one scientific calculator (physical or online) are recommended for preparation.
- **Note: This is the last quiz in this quarter. Highest 7 quiz scores are counted for final grading.**

Prof. Sankararaman's post on updated quiz grading:
https://campuswire.com/c/GB5E561C3/feed/438

UCLA
Engineer Change.

Updates: Final Exam

- Open book and open notes, on GradeScope: “quiz”-like exam
- Start attempting the exam from **8:00 am PST on March 15**; Submit your exam before **8:00am PST March 16** (No extensions). → 24h time window
- Exam duration: **3 hours** (time limit after start the exam)
- Type: True/false and multiple choice questions (free text boxes are given for justification)
- The instructors will be available to provide clarifications on CampusWire (visible for everyone) from 8:00am-11:00am on March 15. Later questions on Campuswire may not be answered.
- Some calculations are expected.

MUST READ: Official post about final exam on Campuswire:
https://campuswire.com/c/GB5E561C3/feed/437

Naive Bayes: Model Summary

- Defines a joint distribution

$$P(X = x, Y = c) = P(Y = c) \prod_{d=1}^{D} P(X_d = x_d | Y = c)$$
$$= P(Y = c) \prod_{k} P(k | Y = c)^{z_k} = \pi_c \prod_{k} \theta_{ck}^{z_k}$$

- Learning problem formulation

Training data $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N} \rightarrow \mathcal{D} = \{(\{z_{nk}\}_{k=1}^{K}, y_n)\}_{n=1}^{N}$

Constraints $\sum_c \pi_c = 1 \quad \sum_k \theta_{ck} = \sum_k P(k|Y = c) = 1$

Objective

$$\mathcal{L} = \log P(\mathcal{D}) = \log \prod_{n=1}^{N} \pi_{y_n} P(x_n | y_n) \longrightarrow (\pi_c^*, \theta_{ck}^*) = \arg\max \sum_n \log \pi_{y_n} + \sum_{n,k} z_{nk} \log \theta_{y_n k}$$

- Solution

$$\pi_c^* = \frac{\#\text{of data points labeled as c}}{\text{N}}$$

$$\theta_{ck}^* = \frac{\#\text{of times word k shows up in data points labeled as c}}{\#\text{data points labeled as c}}$$

Naive Bayes: Derivation

A short derivation of the maximum likelihood estimation

To maximize

$$\sum_{n:y_n=c,k} z_{nk} \log \theta_{ck}$$

We use the Lagrangian multiplier

$$\sum_{n:y_n=c,k} z_{nk} \log \theta_{ck} + \lambda \left(\sum_k \theta_{ck} - 1 \right)$$

Taking derivatives with respect to θ_{ck} and then find the stationary point

$$\left(\sum_{n:y_n=c} \frac{z_{nk}}{\theta_{ck}} \right) + \lambda = 0 \rightarrow \theta_{ck} = -\frac{1}{\lambda} \sum_{n:y_n=c} z_{nk}$$

Apply constraint $\sum_k \theta_{ck} = 1$, plug in expression above for θ_{ck}, solve for λ, and plug back into expression for θ_{ck}:

$$\theta_{ck} = \frac{\sum_{n:y_n=c} z_{nk}}{\sum_k \sum_{n:y_n=c} z_{nk}}$$

UCLA
Engineer Change.

Naive Bayes: Example

- #docs in Class 1: 25, #docs in Class 2: 75

Index	Word	Count in Class 1	Count in Class 2
1	I	1	9
2	like	2	1
3	machine	3	9
4	learning	4	1

Naive Bayes: Learning (Step 1)

- How to obtain the parameters in Naive Bayes classifier? *(shown in class)*

Index	Word	# in C1	# in C2
1	I	1	9
2	like	2	1
3	machine	3	9
4	learning	4	1

UCLA
Engineer Change.

Naive Bayes: Predicting

- Predicting new document: {I:3, like:1, machine:5, learning:1} *(shown in class)*

Index	Word	# in C1	# in C2
1	I	1	9
2	like	2	1
3	machine	3	9
4	learning	4	1

- One further question: How to predict new document: {I:3, like:1, machine:5, learning:1, love:2} → Label Smoothing

UCLA
Engineer Change.

Naive Bayes: Summary, Pros and Cons

- A linear classifier (same as logistic regression)
- Generative model, modeling joint distribution (probabilities) → *What is the model assumption?*
- Pros:
  - Fast and simple compared to other complicated algorithms, easy training
  - Works well with high-dimension data such as text classification
- Cons:
  - Strong assumptions (feature independency)
  - Not fit to regression
  - Smoothing is somewhat required for generalization

UCLA
Engineer Change.

Generative vs Discriminative Models

- Training classifiers involve estimating *f: X → Y*, or P(Y|X)
- Generative classifiers → *"distribution"*
  - Assume some functional form for P(Y), P(X|Y)
  - Estimate parameters of P(X|Y), P(Y) directly from training data
  - Use Bayes rule to calculate P(Y|X)
  - Actually learn the underlying structure of the data
- Discriminative Classifiers → *"boundary"*
  - Assume some functional form for P(Y|X)
  - Estimate parameters of P(Y|X) directly from training data.
  - Learn the mappings directly from the points to the classes

- Generative models
  - Naive Bayes
  - HMM
- Discriminative models
  - Logistic Regression
  - Neural Network / Perceptron
  - SVM

Q: With the aim of classification only, which type of models may less expensive?

Credit: https://stats.stackexchange.com/questions/12421/generative-vs-discriminative

UCLA
Engineer Change.

Naive Bayes vs Logistic Regression

- Compare the learning and prediction procedure on Naive Bayes and Logistic Regression in spam classification example

MLE vs MAP

- From Bayes rule:

Likelihood
How probable is the evidence given that our hypothesis is true?

Prior
How probable was our hypothesis *before* observing the evidence?

$$P(H \mid e) = \frac{P(e \mid H)\, P(H)}{P(e)}$$

Posterior
How probable is our hypothesis given the observed evidence? (Not directly computable)

Marginal
How probable is the new evidence under all possible hypotheses?
$P(e) = \sum P(e \mid H_i)\, P(H_i)$

- Comparing MAP and MLE: [Link]

$$\begin{aligned}\theta_{MLE} &= \arg\max_{\theta} \log P(X|\theta) \\ &= \arg\max_{\theta} \log \prod_i P(x_i|\theta) \\ &= \arg\max_{\theta} \sum_i \log P(x_i|\theta)\end{aligned}$$

$$\begin{aligned}\theta_{MAP} &= \arg\max_{\theta} P(X|\theta)P(\theta) \\ &= \arg\max_{\theta} \log P(X|\theta) + \log P(\theta) \\ &= \arg\max_{\theta} \log \prod_i P(x_i|\theta) + \log P(\theta) \\ &= \arg\max_{\theta} \sum_i \log P(x_i|\theta) + \log P(\theta)\end{aligned}$$

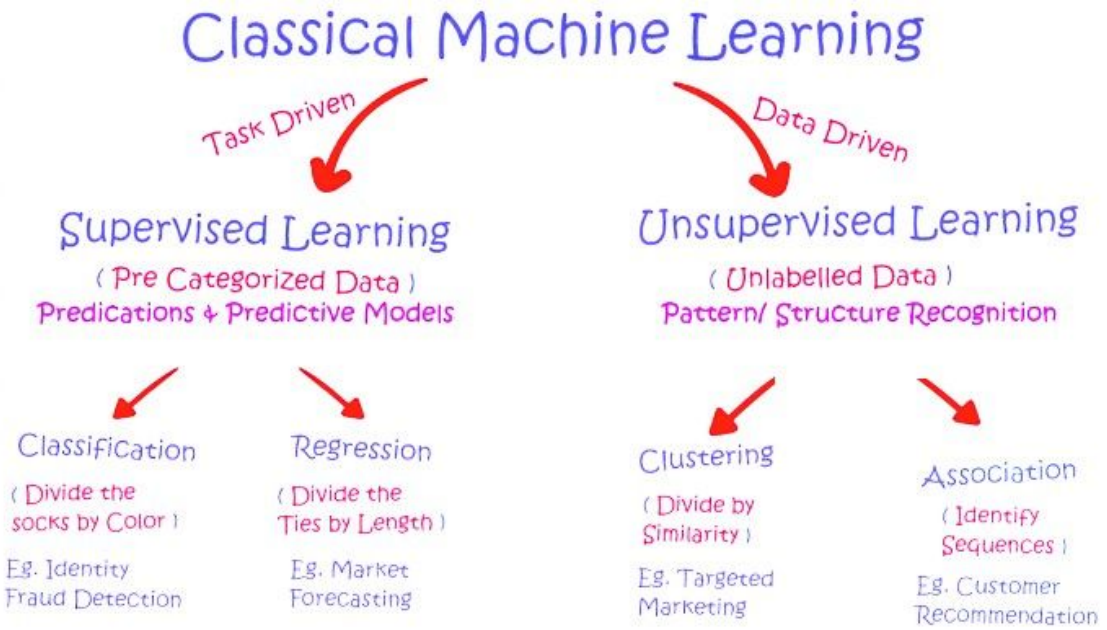
UCLA
Engineer Change.

Clustering

- Clustering: Input / Output / Goal of clustering analysis
  - Large amount of unlabeled data in real life
- Supervised learning v.s. unsupervised learning
- Unsupervised learning cases: Clustering and dimension reduction
- Clustering algorithm examples in this course:
  - K-means
  - Gaussian Mixture models
- Dimension reduction algorithm examples in this course:
  - PCA

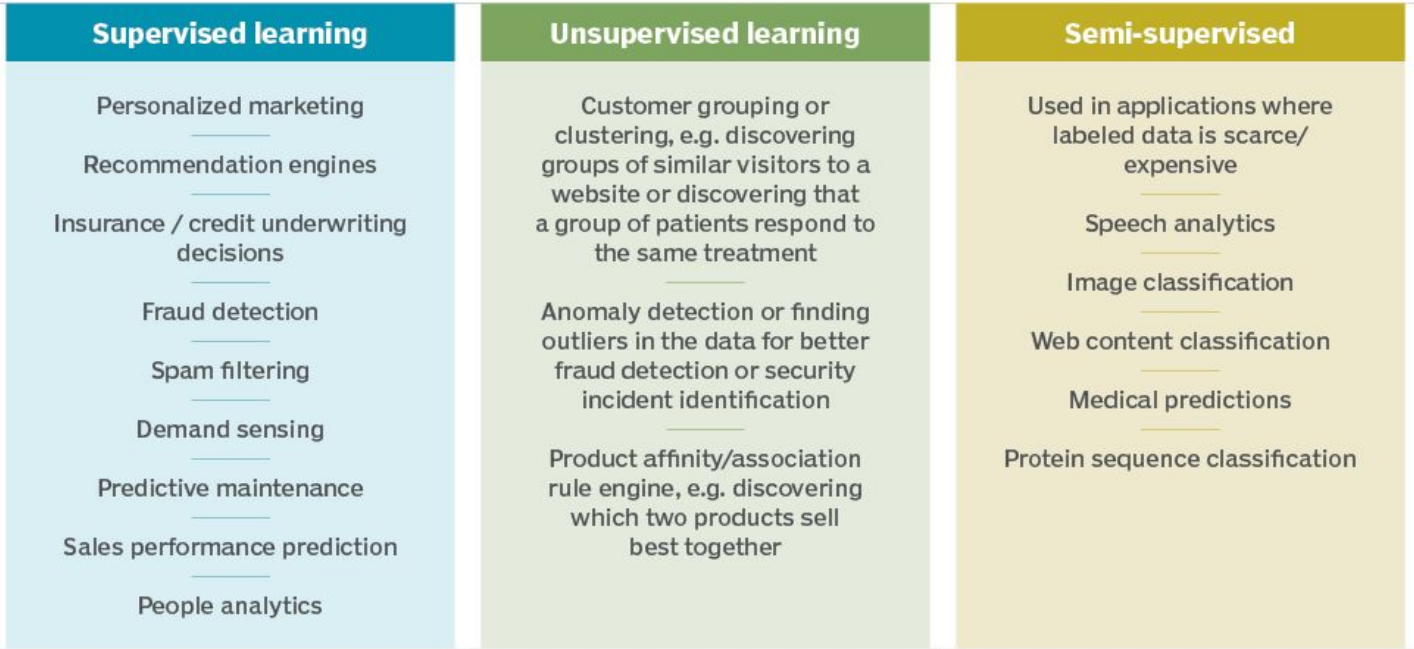
Supervised learning / Unsupervised learning

	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction



UCLA
Engineer Change.

Applications of ML Categories

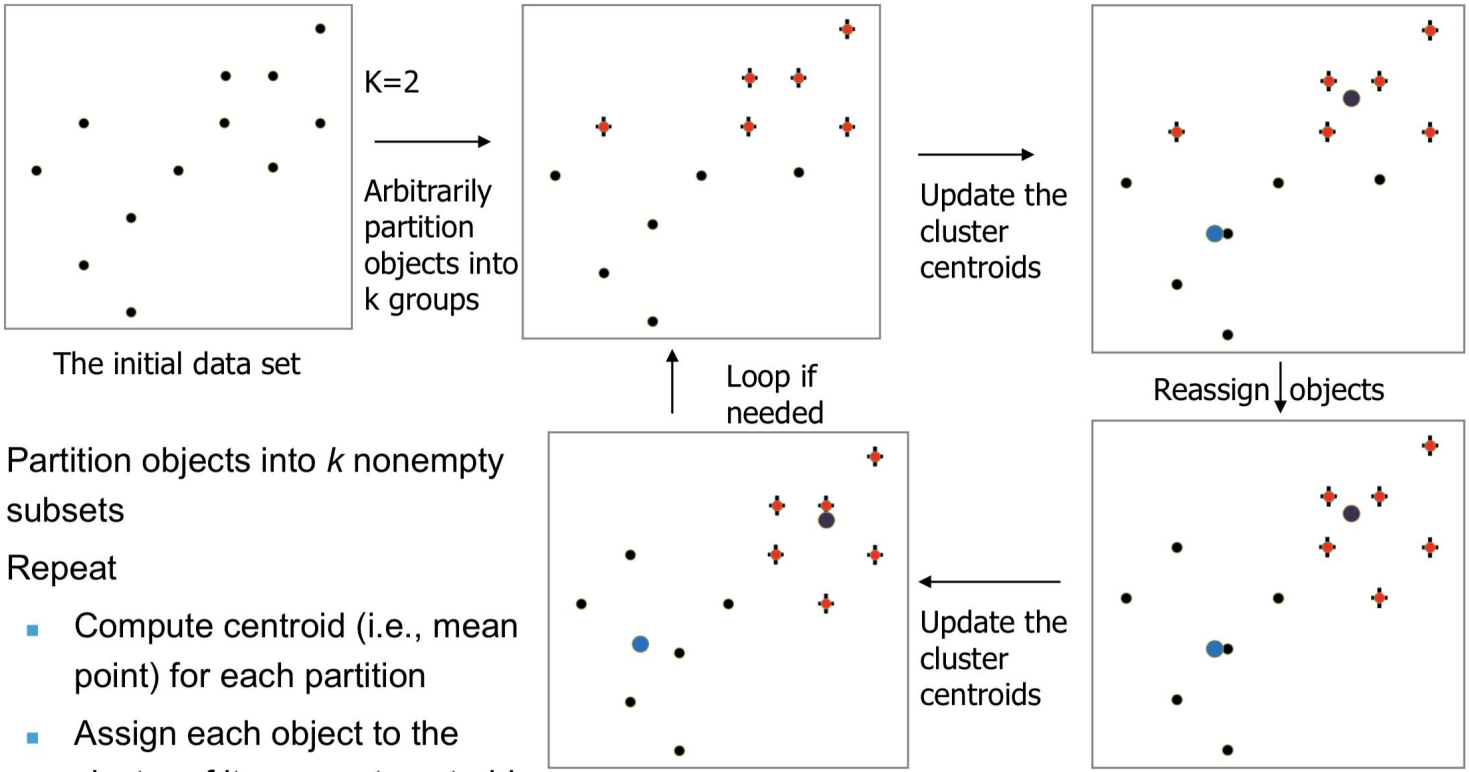


Other “learning”: Self-supervised learning, reinforcement learning

K-means

- Demo 1: http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html
- Demo 2: https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

K-means: Steps



- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

K-means

- Distortion measure

$$J(\{r_{nk}\}, \{\boldsymbol{\mu}_k\}) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|_2^2$$

- Key idea of K-means algorithms:
  - Step 1: Partition into k non-empty subsets (select K points as initial centroids)
  - Step 2: Iteration: Update mean point and assign object to cluster again
  - Step 3: Stop when converge
- Partition-based clustering methods
- Can be considered as a special case of Gaussian Mixture Model (GMM)

K-means

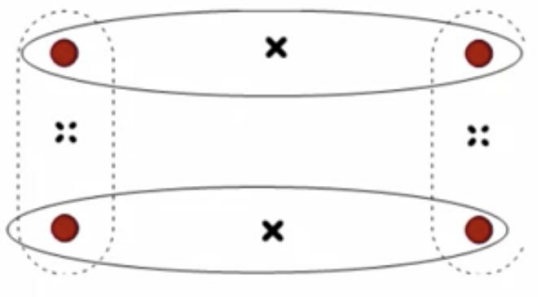
- Q1: Will K-means converge?
- Q2: Will different initialization of K-means generate different clustering results?

K-means

- Q1: Will K-means converge?
- A1: Yes.

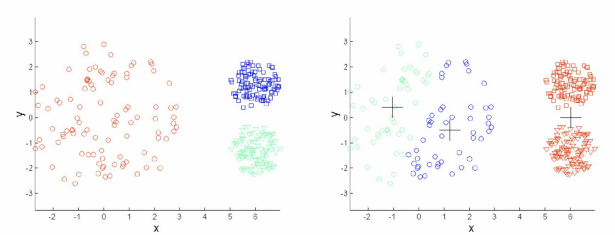
$$J = \sum_{j=1}^{k} \sum_{C(i)=j} d(x_i, c_j)^2$$

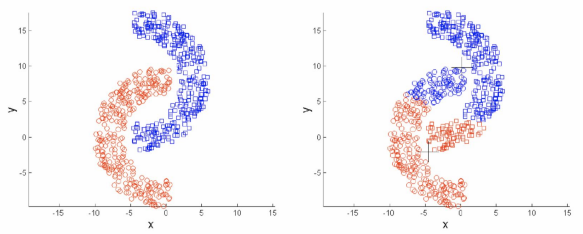
- Q2: Will different initialization of K-means generate different clustering results?
- A2: Yes. Initialization matters!



K-means: Discussion

- Efficiency: **O(tkn)** normally k,t are much smaller than n → efficient
- Can terminate at a local optimum
- Need to specify **k** (or take time to find best **k**)
- Sensitive to noisy data and outliers → K-medoids
- Different sizes and variances
- Not suitable to discover clusters with non-convex shapes
- Many variants of K-means:
  - K-means++, Genetics K-means, etc.



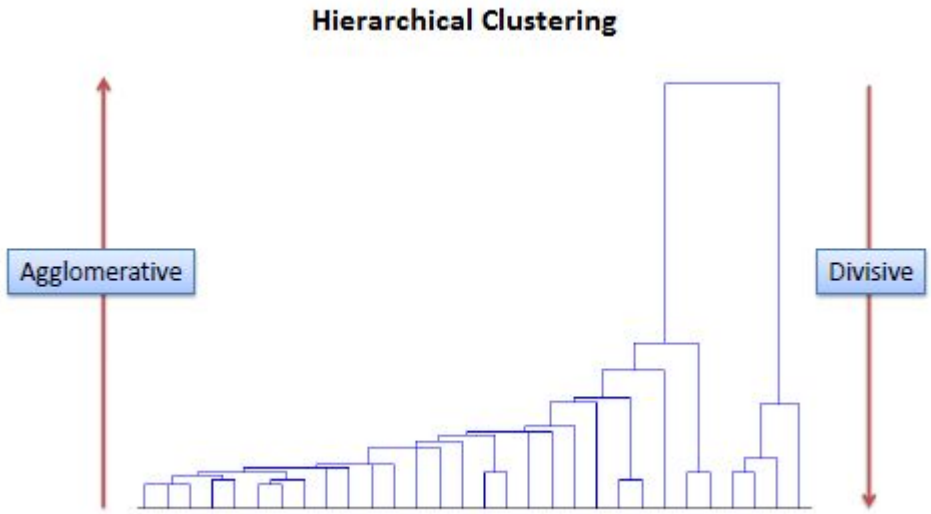


UCLA
Engineer Change.

*Hierarchical Clustering

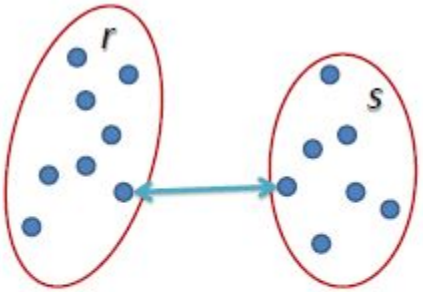
- Method
  - Divisive (Top-down)
  - Agglomerative (Bottom-up)

- Distance metrics
  - Single linkage
  - Complete linkage
  - Average linkage
  - Centroid
  - Medoid



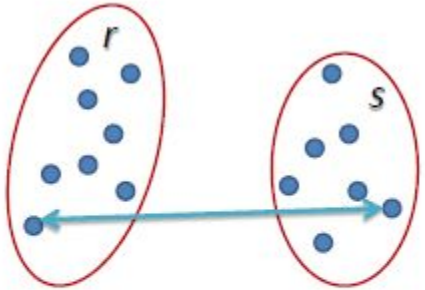
*Hierarchical Clustering

- Single Linkage



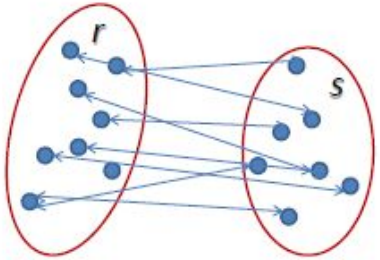
$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

- Complete Linkage



$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

- Average Linkage



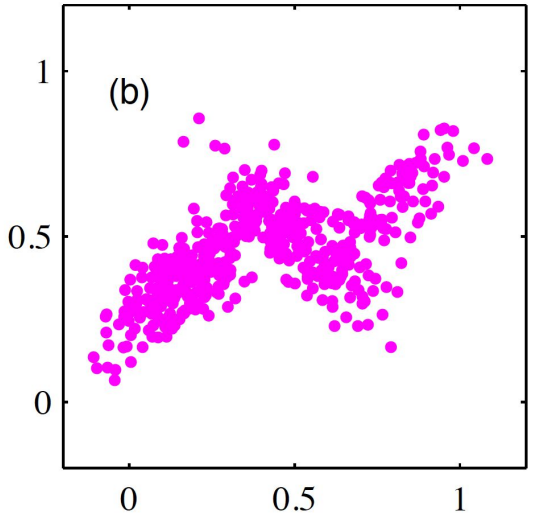
$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

Gaussian Mixture Model

Probabilistic interpretation of clustering?

We can impose a probabilistic interpretation of our intuition that points stay close to their cluster centers

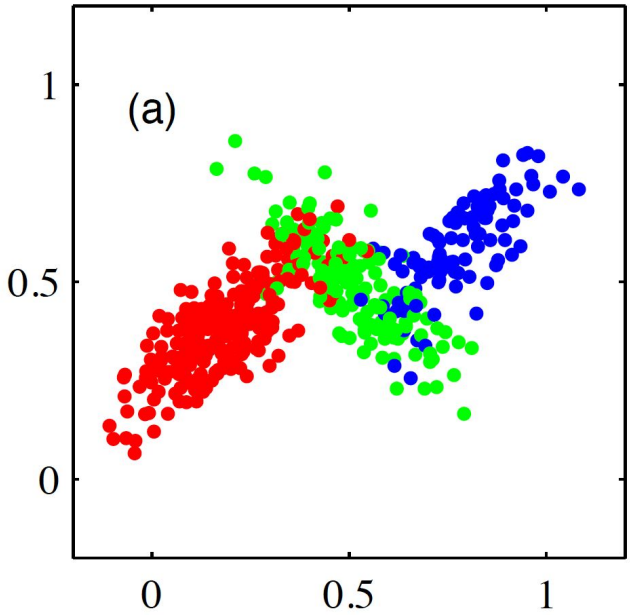
How can we model p(x) to reflect this?



Gaussian Mixture Model

Intuition

- We can model each region with a distinct distribution
- Common to use Gaussians, i.e.,
- Gaussian mixture models (GMMs) or mixture of Gaussians (MoGs).
- We don't know cluster assignments (label) or parameters of Gaussians or mixture components



Gaussian Mixture Model

Gaussian mixture models: formal definition

A Gaussian mixture model has the following density function for $\boldsymbol{x}$

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

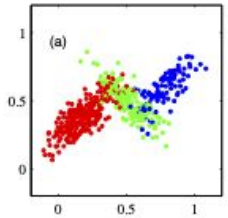
- K: the number of Gaussians — they are called (mixture) components
- $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$: mean and covariance matrix of the k-th component
- ω_k: mixture weights – they represent how much each component contributes to the final distribution. It satisfies two properties:

$$\forall\ k,\ \omega_k > 0, \quad \text{and} \quad \sum_k \omega_k = 1$$

The properties ensure $p(\boldsymbol{x})$ is a properly normalized probability density function.

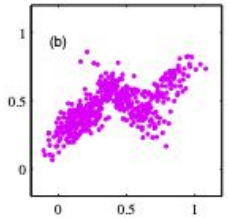
Gaussian Mixture Model

GMMs: example



The conditional distribution between $\boldsymbol{x}$ and z (representing color) are

$$p(\boldsymbol{x}|z = red) = N(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$
$$p(\boldsymbol{x}|z = blue) = N(\boldsymbol{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$
$$p(\boldsymbol{x}|z = green) = N(\boldsymbol{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$



The marginal distribution is thus

$$p(\boldsymbol{x}) = p(red)N(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + p(blue)N(\boldsymbol{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) + p(green)N(\boldsymbol{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

GMM: Incomplete Data and EM algorithm

Parameter estimation for GMMs: Incomplete data

GMM Parameters

$$\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$$

Incomplete Data

Our data contains observed and unobserved random variables, and hence is incomplete

- Observed: $\mathcal{D} = \{\boldsymbol{x}_n\}$
- Unobserved (hidden): $\{\boldsymbol{z}_n\}$

Goal Obtain the maximum likelihood estimate of $\boldsymbol{\theta}$:

$$\widehat{\boldsymbol{\theta}} = \arg\max \ell(\boldsymbol{\theta}) = \arg\max \log P(\mathcal{D}) = \arg\max \sum_n \log p(\boldsymbol{x}_n|\boldsymbol{\theta})$$

$$= \arg\max \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})$$

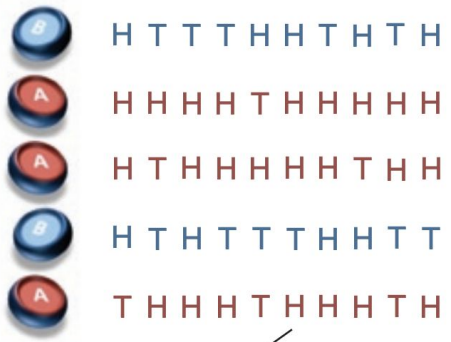
The objective function $\ell(\boldsymbol{\theta})$ is called the *incomplete* log-likelihood.

Typical EM iterations

1. Initialize θ with some values (random or otherwise)
2. Repeat
   a. **E-Step:** Compute γnk using the current θ
   b. **M-Step:** Update θ using the γnk we just computed
3. Until Convergence

EM Algorithms: Coin example (only M-step)

a Maximum likelihood

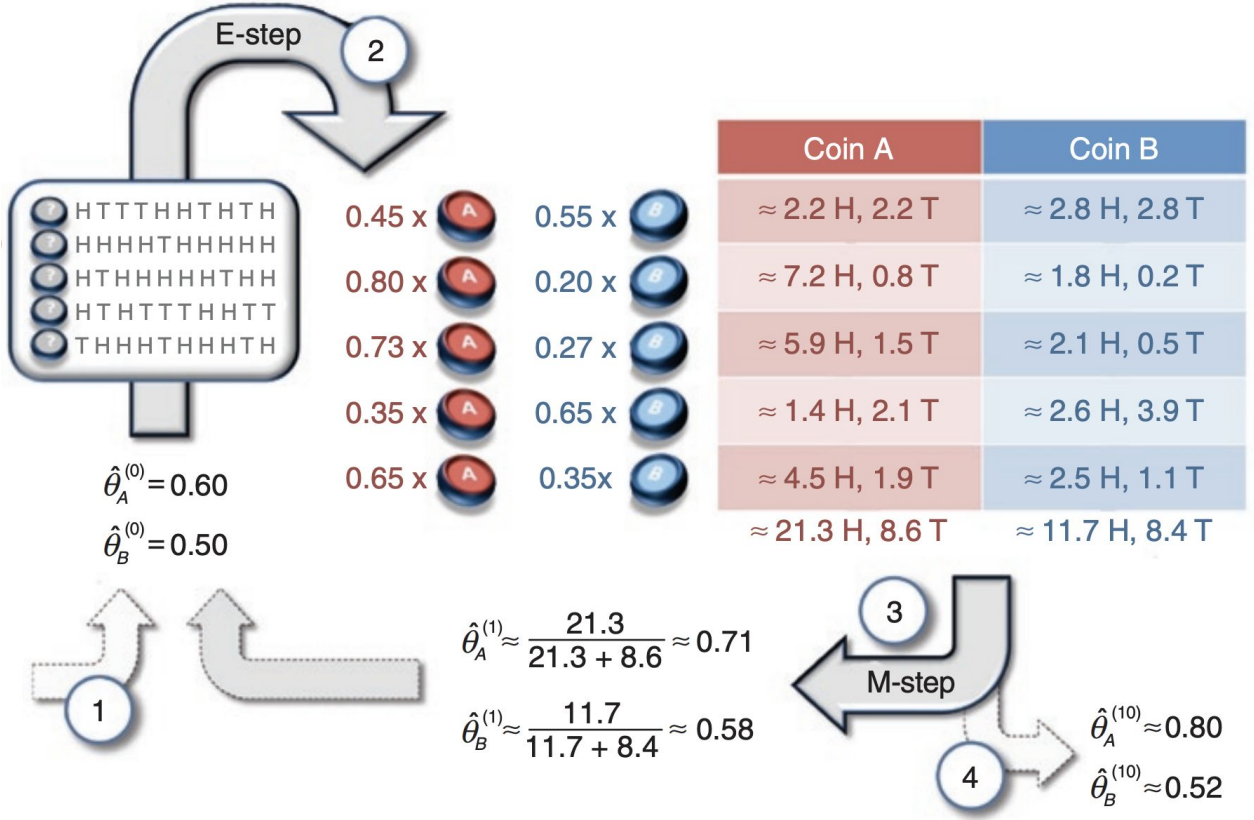


Coin A	Coin B
	5 H, 5 T
9 H, 1 T	
8 H, 2 T	
	4 H, 6 T
7 H, 3 T	
24 H, 6 T	9 H, 11 T

$$\hat{\theta}_A = \frac{24}{24+6} = 0.80$$

$$\hat{\theta}_B = \frac{9}{9+11} = 0.45$$

EM Algorithms: Coin example (EM)



GMM: E-Step

E-step: Soft cluster assignments

We define γ_{nk} as $p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\theta})$

- This is the posterior distribution of z_n given $\boldsymbol{x}_n$ and $\boldsymbol{\theta}$
- Recall that in complete data setting γ_{nk} was binary
- Now it's a "soft" assignment of $\boldsymbol{x}_n$ to k-th component, with $\boldsymbol{x}_n$ assigned to each component with some probability

Given $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$, we can compute γ_{nk} using Bayes theorem:

$$\begin{aligned}
\gamma_{nk} &= p(z_n = k|\boldsymbol{x}_n) \\
&= \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{p(\boldsymbol{x}_n)} \\
&= \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{\sum_{k'=1}^K p(\boldsymbol{x}_n|z_n = k')p(z_n = k')} \quad = \frac{\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\omega_k}{\sum_{k'=1}^K \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})\omega_{k'}}
\end{aligned}$$

GMM: M-Step

M-step: Maximimize complete likelihood

Recall definition of complete likelihood from earlier:

$$\sum_n \log p(\boldsymbol{x}_n, z_n) = \sum_k \sum_n \gamma_{nk} \log \omega_k + \sum_k \left\{ \sum_n \gamma_{nk} \log \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Previously γ_{nk} was binary, but now we define $\gamma_{nk} = p(z_n = k | \boldsymbol{x}_n)$ (E-step)

We get the same simple expression for the MLE as before!

$$\omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}, \quad \boldsymbol{\mu}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \boldsymbol{x}_n$$

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

Intuition: Each point now contributes some fractional component to each of the parameters, with weights determined by γ_{nk}

UCLA
Engineer Change.

GMM: Example (M-Step calculation)

Consider clustering ID data with a mixture of **2 gaussian**. You're given the 1-D data points **x = [1 2 20 40]**. Suppose the E step is the following matrix :

$$\begin{bmatrix} 0.5 & 0.5 \\ 0.2 & 0.8 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

→ What's the mixing weights after M-step?

→ What's the new values of means after M-step?

UCLA
Engineer Change.

GMM: Cheatsheet

Expectation (E) Step:

Calculate $\forall i, k$

$$\hat{\gamma}_{ik} = \frac{\hat{\phi}_k \mathcal{N}(x_i \mid \hat{\mu}_k, \hat{\sigma}_k)}{\sum_{j=1}^{K} \hat{\phi}_j \mathcal{N}(x_i \mid \hat{\mu}_j, \hat{\sigma}_j)},$$

where $\hat{\gamma}_{ik}$ is the probability that x_i is generated by component C_k. Thus, $\hat{\gamma}_{ik} = p(C_k|x_i, \hat{\phi}, \hat{\mu}, \hat{\sigma})$.

Maximization (M) Step:

Using the $\hat{\gamma}_{ik}$ calculated in the expectation step, calculate the following in that order $\forall k$:

- $\hat{\phi}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N}$
- $\hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} x_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$
- $\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} (x_i - \hat{\mu}_k)^2}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}.$

UCLA
Engineer Change.

GMM: Questions

- How does GMM relate to K-means? What are the similarities and differences?
- Will the GMM optimization process converge? (connected to K-means)

UCLA
Engineer Change.

General EM Algorithms

- The EM algorithm is used to find **(local) maximum likelihood parameters** of a statistical model in cases where the equations cannot be solved directly.
- Typically these models involve latent variables in addition to unknown parameters and known data observations.
- Example applied cases: K-Means, GMM
- Reading: http://ai.stanford.edu/~chuongdo/papers/em_tutorial.pdf

UCLA Samueli Computer Science

Thank you!

Q & A

UCLA Samueli Computer Science

Evaluation of Instruction

Reminder: You have until **Saturday, March 13 8:00 AM PST** to complete confidential evaluations for CSM146 and Dis 1C (Junheng).