18-661 Introduction to Machine Learning

Clustering, Part II

Spring 2025

ECE – Carnegie Mellon University

Announcements

- Homework 4 is due on Wednesday, April 16.
- Mini-exam 3 will be held on April 16.
  - The mini-exam will cover transformers, distributed learning, clustering (k-means and Gaussian mixture models) and dimensionality reduction.
  - Similar format to the last two mini-exams, with true/false, multiple choice, and descriptive questions.

Outline

1. Review: Clustering and k-means

2. Gaussian Mixture Models

3. EM Algorithm

Review: Clustering and k-means

Supervised versus Unsupervised Learning

Supervised Learning: labeled observations $\{(\boldsymbol{x}_1, y_1), \ldots (\boldsymbol{x}_n, y_n)\}$

- Labels 'teach' algorithm to learn mapping from observations to labels
- Examples: Classification (Logistic Reg., SVMs, Neural Nets, Nearest Neighbors, Decision Trees), Regression (Linear Reg., Neural Nets)

Unsupervised Learning: unlabeled observations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$

- Learning algorithm must find patterns from features alone
- Can be goal in itself (discover hidden patterns, exploratory analysis)
- Can be means to an end (pre-processing for supervised task)
- Examples:
  - K-means clustering, Gaussian Mixture Models
  - Dimensionality Reduction: Transform an initial feature representation into a more concise representation

K-means Clustering: Details

Intuition: Data points assigned to cluster k should be near prototype $\boldsymbol{\mu}_k$

Distortion measure: (clustering objective function, cost function)

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 = \sum_{k=1}^{K} \underbrace{\sum_{n:A(\boldsymbol{x}_n)=k} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2}_{\text{spread within the } k\text{th cluster}}$$

where $r_{nk} \in \{0, 1\}$ is an indicator variable

$$r_{nk} = 1 \quad \text{if and only if} \quad A(\boldsymbol{x}_n) = k$$

How to measure distortion?

- Distance measure: $\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$ calculates how far $\boldsymbol{x}_n$ is from the cluster center $\boldsymbol{\mu}_k$
- Canonical example is the 2-norm, i.e., $\|\cdot\|_2^2$, but could be some other distance measure!

Optimization Algorithm

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

- What are the variables that we need to optimize? $\{r_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$
- Difficult to jointly optimize both
- Solution: Alternative optimization between $\{r_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$

- **Step 0** Initialize $\{\boldsymbol{\mu}_k\}$ to some values
- **Step 1** Fix $\{\boldsymbol{\mu}_k\}$ and minimize over $\{r_{nk}\}$, to get this assignment:

$$r_{nk} = \begin{cases} 1 & \text{if } k = \text{argmin}_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise} \end{cases}$$

- **Step 2** Fix $\{r_{nk}\}$ and minimize over $\{\boldsymbol{\mu}_k\}$ to get this update:

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \boldsymbol{x}_n}{\sum_n r_{nk}}$$

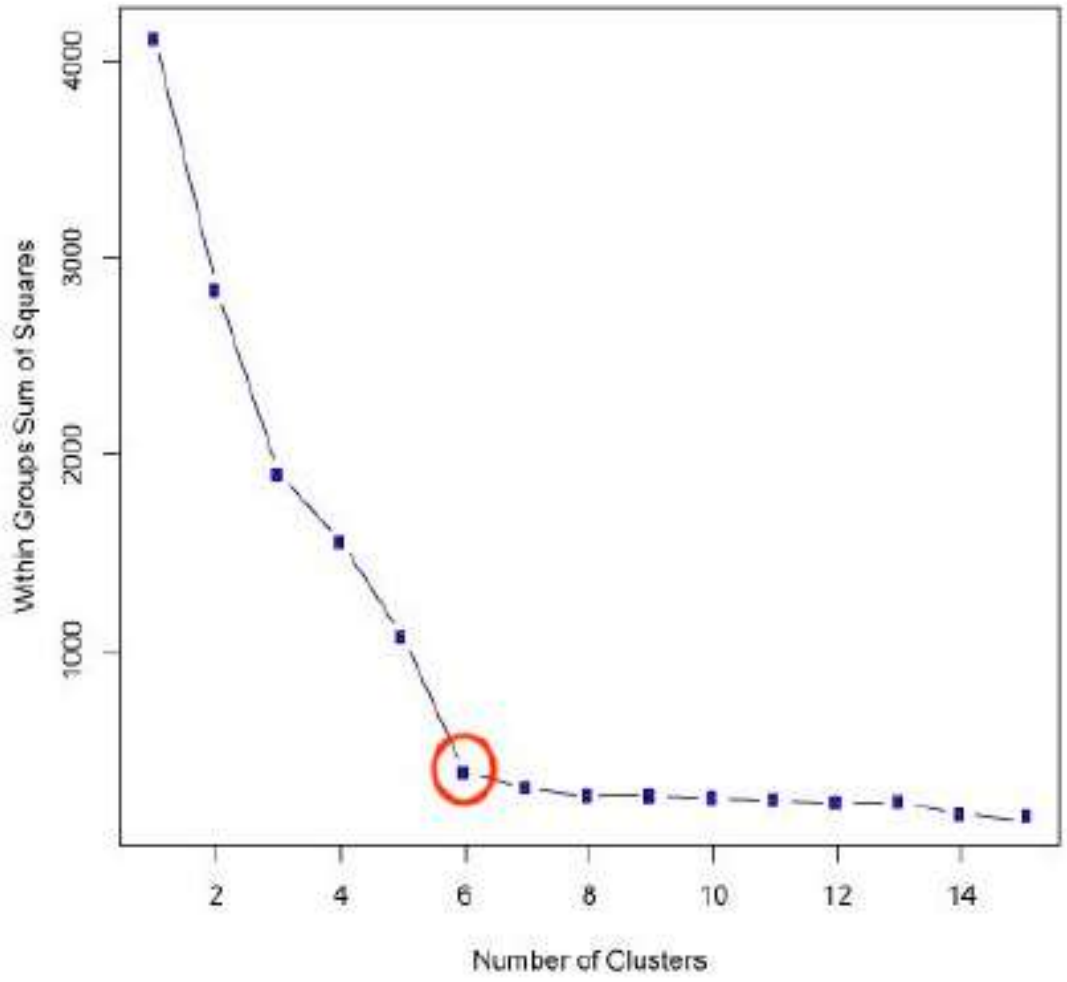
- **Step 3** Return to Step 1 unless stopping criterion is met

Practical Issues with K-means

- How to select k?
  - Prior knowledge
  - Heuristics (e.g., elbow method)
- How to select the distance measure?
  - Often requires some knowledge of problem
  - Some examples: Euclidean distance (for images), Hamming distance (distance between two strings), shared key words (for websites)
- How to initialize cluster centers?
  - The final clustering can depend significantly on the initial points you pick!

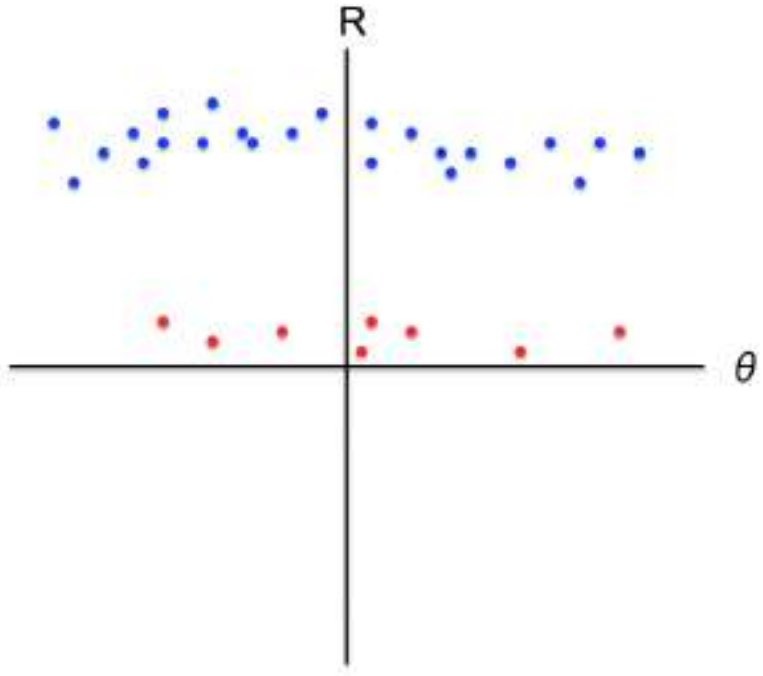
Selecting k: the Elbow Method

Select a small value of k such that adding a new cluster doesn't reduce the within-cluster distances much



Selecting a Distance Measure

Changing features (distance measure) can help



If the cluster i mean is $(\mu_{i,x}, \mu_{i,y})$, the distance of (x_n, y_n) from it can be defined as $|\sqrt{\mu_{i,x}^2 + \mu_{i,y}^2} - \sqrt{x_n^2 + y_n^2}|$

Initializing Clusters: K-means++

Key idea: Run K-means, but with a better initialization

- Choose center μ_1 at random
- For $j = 2, \ldots, k$
  - Choose μ_j among $x_1, \ldots, x_n$ with probability:

$$P(\mu_j = x_i) \propto min_{j' < j} \|x_i - \mu_{j'}\|^2$$

This means that if x_i is close to one of the already chosen cluster means $\mu_1, \ldots \mu_{j-1}$, then we assign a lower probability of selecting it as the next cluster mean.

Initialization helps to get good coverage of the space

Theorem: K-means++ always obtains a $O(\log k)$ approximation to the optimal solution in expectation.

Running K-means after this initialization can only improve on the result

Outline

1. Review: Clustering and k-means

2. Gaussian Mixture Models

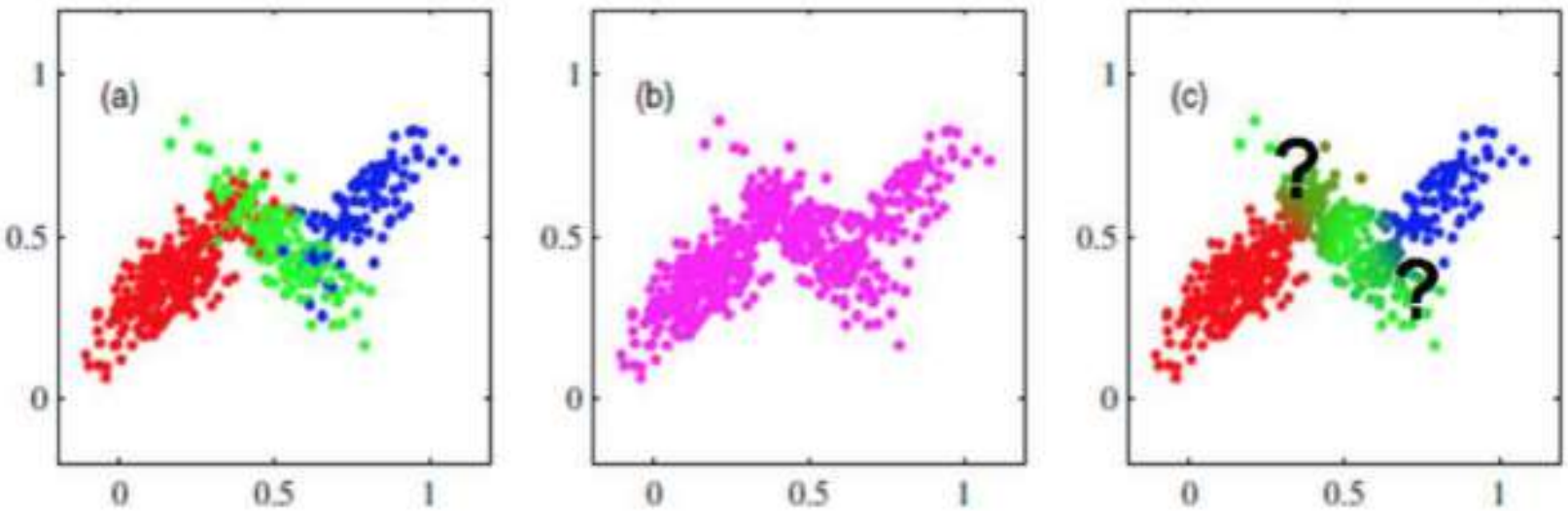
3. EM Algorithm

Gaussian Mixture Models

Potential Issues with k-Means . . .

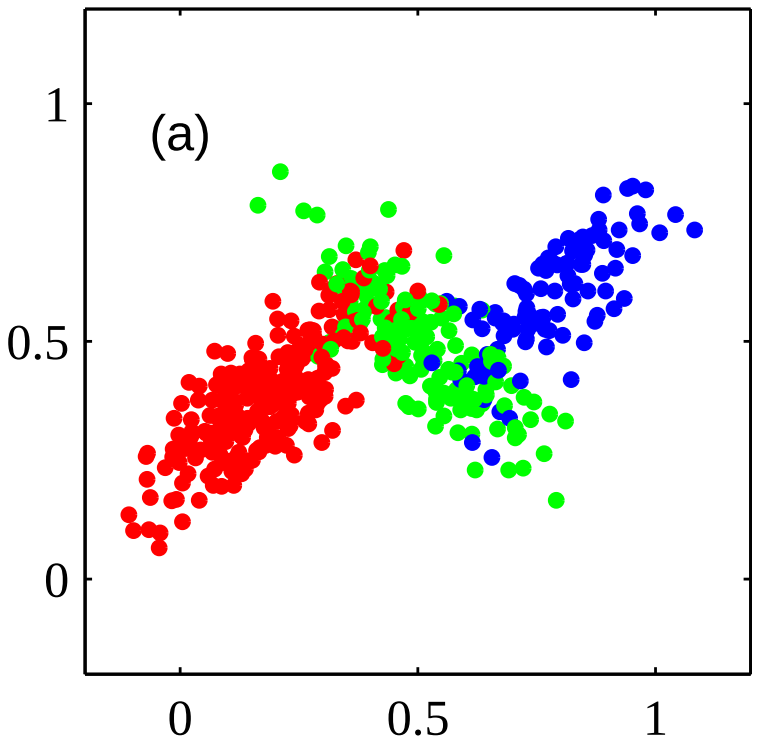
Data points are assigned *deterministically* to one (and only one) cluster

In reality, clusters may overlap, and it may be better to identify the *probability* that a point belongs to each cluster



Also, distances are measured in a homogeneous manner. In reality, some clusters may be more spread out than others

Gaussian Mixture Models: Intuition



- Key idea: Model *each* region (cluster) with a distinct distribution
- Can use Gaussians — Gaussian mixture models (GMMs)
- *However*, we don't know *cluster assignments* (label), *parameters* of Gaussians, or *mixture components*!
- Must learn from *unlabeled* data $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^{N}$

Gaussian Mixture Models: Formal Definition

GMM 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: number of Gaussians — they are called mixture components
- $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$: mean and covariance matrix of k-th component
- ω_k: mixture weights (or priors) represent how much each component contributes to final distribution. They satisfy 2 properties:

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

These properties ensure that $p(\boldsymbol{x})$ is a probability density function

GMM as the Marginal Distribution of a Joint Distribution

Consider the following joint distribution

$$p(\boldsymbol{x}, z) = p(z)p(\boldsymbol{x}|z)$$

where z is a discrete random variable taking values between 1 and K.

Denote

$$\omega_k = p(z = k)$$

Now, assume the conditional distributions are Gaussian distributions

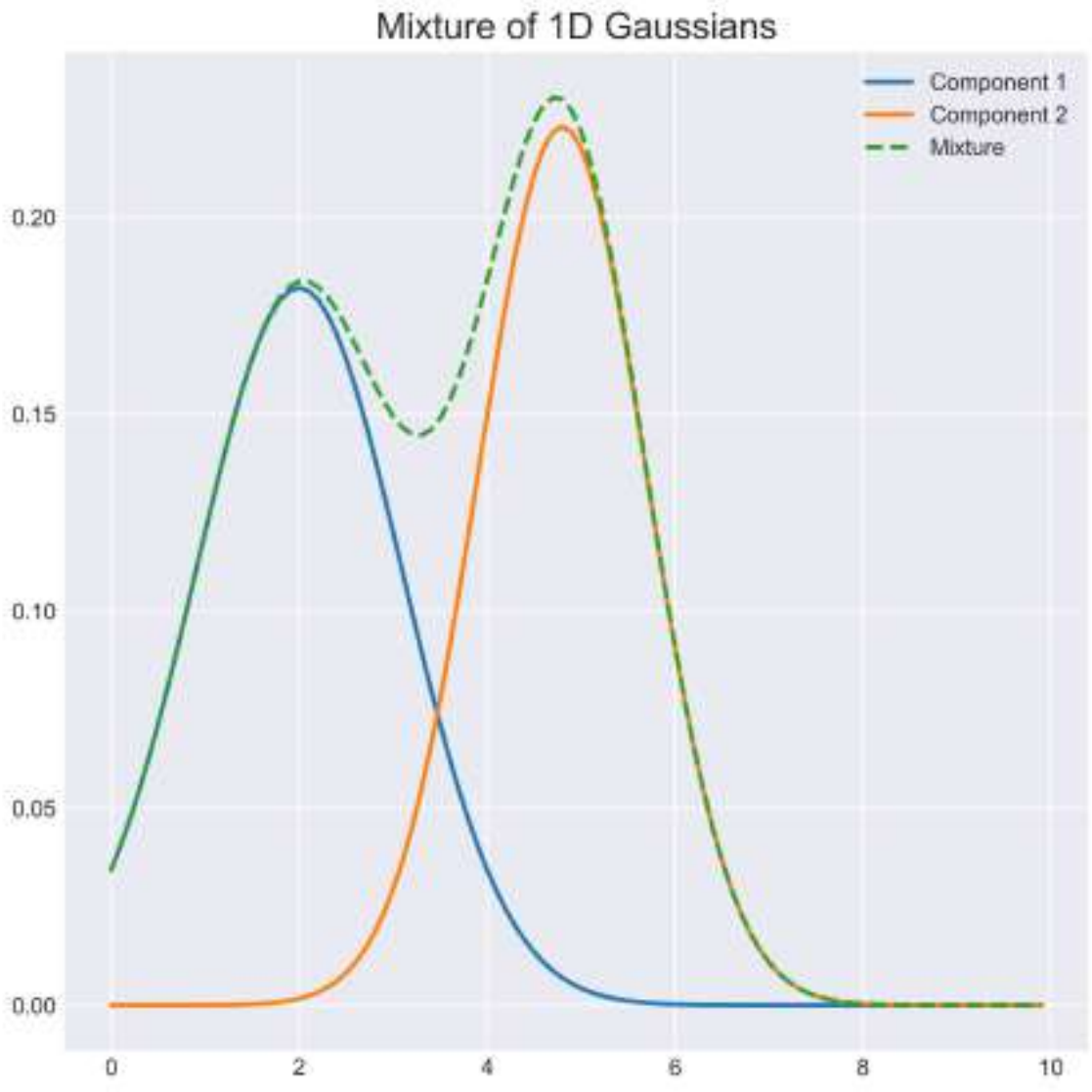
$$p(\boldsymbol{x}|z = k) = N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Then, the marginal distribution of $\boldsymbol{x}$ is

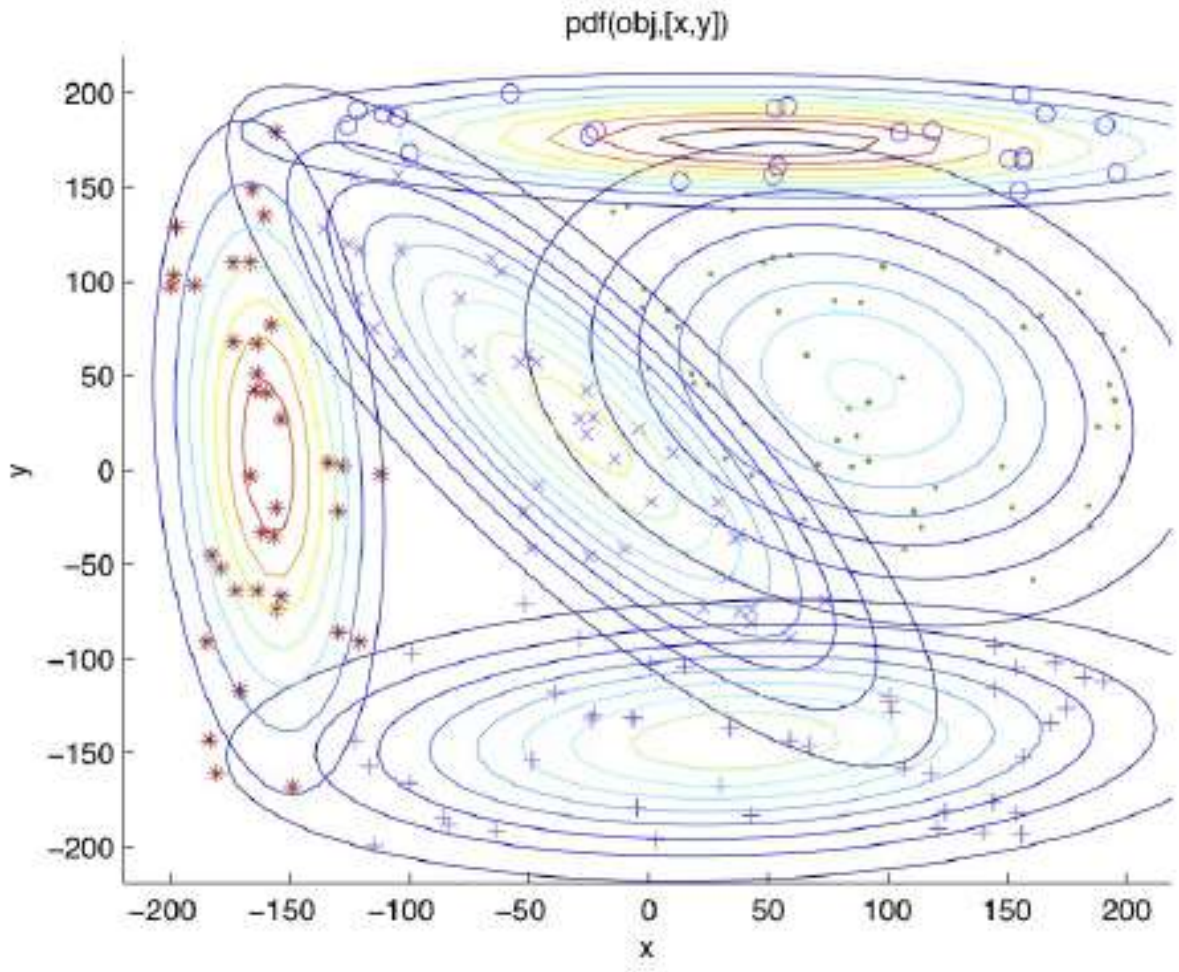
$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Namely, the Gaussian mixture model!

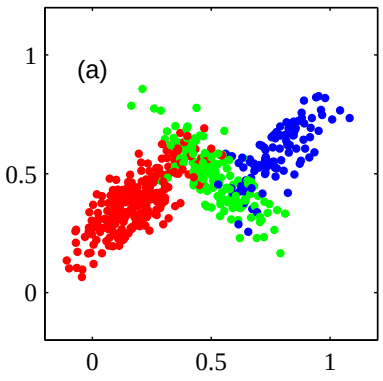
Gaussian Mixtures in 1D

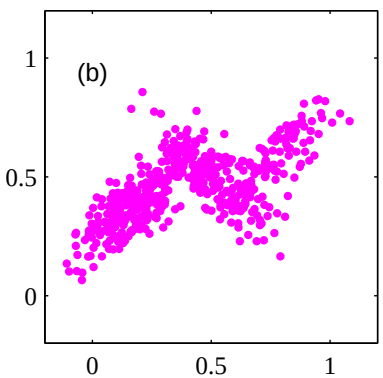


Gaussian Mixture Model for Clustering



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

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

Parameter Estimation for Gaussian Mixture Models

The parameters in GMMs are $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$

Let's first consider the simple/unrealistic case where *we have labels z*

Define $\mathcal{D}' = \{\boldsymbol{x}_n, z_n\}_{n=1}^N$, $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$

- $\mathcal{D}'$ is the complete data
- $\mathcal{D}$ the incomplete data

How can we learn our parameters?

Given $\mathcal{D}'$, the maximum likelihood estimation of the $\boldsymbol{\theta}$ is given by

$$\boldsymbol{\theta} = \arg\max \log \mathcal{D}' = \sum_n \log p(\boldsymbol{x}_n, z_n)$$

Parameter Estimation for GMMs: Complete Data

The complete likelihood is decomposable

$$\sum_n \log p(\boldsymbol{x}_n, z_n) = \sum_n \log p(z_n) p(\boldsymbol{x}_n | z_n) = \sum_k \sum_{n:z_n=k} \log p(z_n) p(\boldsymbol{x}_n | z_n)$$

where we have grouped data by cluster labels z_n.

Let $r_{nk} \in \{0, 1\}$ be a binary variable that indicates whether $z_n = k$:

$$\begin{aligned} \sum_n \log p(\boldsymbol{x}_n, z_n) &= \sum_k \sum_n r_{nk} \log p(z = k) p(\boldsymbol{x}_n | z = k) \\ &= \sum_k \sum_n r_{nk} \left[\log \omega_k + \log N(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right] \end{aligned}$$

Note: in the complete setting the r_{nk} are binary, but later we will 'relax' these variables and allow them to take on fractional values

Parameter Estimation for GMMs: Complete Data

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

Regrouping, we have

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

The term inside the braces depends on k-th component's parameters. It is now easy to show (left as an exercise) that the MLE is:

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

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n r_{nk}} \sum_n r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top$$

What's the intuition?

Intuition

Since r_{nk} is binary, the previous solution is nothing but:

- ω_k: fraction of total data points whose cluster label z_n is k
  - note that $\sum_k \sum_n r_{nk} = N$
- $\boldsymbol{\mu}_k$: mean of all data points whose z_n is k
- $\boldsymbol{\Sigma}_k$: co-variance of all data points whose z_n is k

We use the knowledge of true cluster labels z_n (which imply the r_{nk}) to estimate $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$.

What do we do when we *do not* know z_n (incomplete data)?

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 data, and hence is incomplete

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

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

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

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

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

Parameter Estimation for GMMs: Incomplete Data

When z_n is not given, we can guess it via the posterior probability (recall: Bayes' rule!)

$$p(z_n = k|\boldsymbol{x}_n) = \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{p(\boldsymbol{x}_n)} = \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{\sum_{k'=1}^{K} p(\boldsymbol{x}_n|z_n = k')p(z_n = k')}$$
$$= \frac{N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \times \omega_k}{\sum_{k'=1}^{K} N(\boldsymbol{x}_n|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'}) \times \omega_{k'}}$$

To compute the posterior probability, we need to know the parameters $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$

Idea: Let's pretend we know these parameters so we can compute the posterior probability.

How is that going to help us?

Estimation with Soft r_{nk}

We define $r_{nk} = p(z_n = k|\boldsymbol{x}_n)$

- Recall that r_{nk} was previously binary
- Now it's a "soft" assignment of $\boldsymbol{x}_n$ to k-th component
- Each $\boldsymbol{x}_n$ is assigned to a component fractionally according to $p(z_n = k|\boldsymbol{x}_n)$

If we solve for the MLE of $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ given soft r_{nk}s, we get the same expressions as before!

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

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n r_{nk}} \sum_n r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top$$

But remember, we're 'cheating' by using $\boldsymbol{\theta}$ to compute r_{nk}!

Iterative Procedure

Alternate between estimating r_{nk} and computing parameters

- Step 0: initialize $\boldsymbol{\theta}$ with some values (random or otherwise)
- Step 1 (E-Step): set $r_{nk} = p(z_n = k|\boldsymbol{x}_n)$ for current $\boldsymbol{\theta}$ using Bayes Rule
- Step 2 (M-Step): update $\boldsymbol{\theta}$ using these r_{nk}s using MLE
- Step 3: go back to Step 1

At the end convert r_{nk} back to binary by setting the largest r_{nk} for point x_n to 1 and others to 0.

This is an example of the EM algorithm — a powerful procedure for model estimation with hidden/latent variables

GMMs and K-means

GMMs provide probabilistic interpretation for K-means.

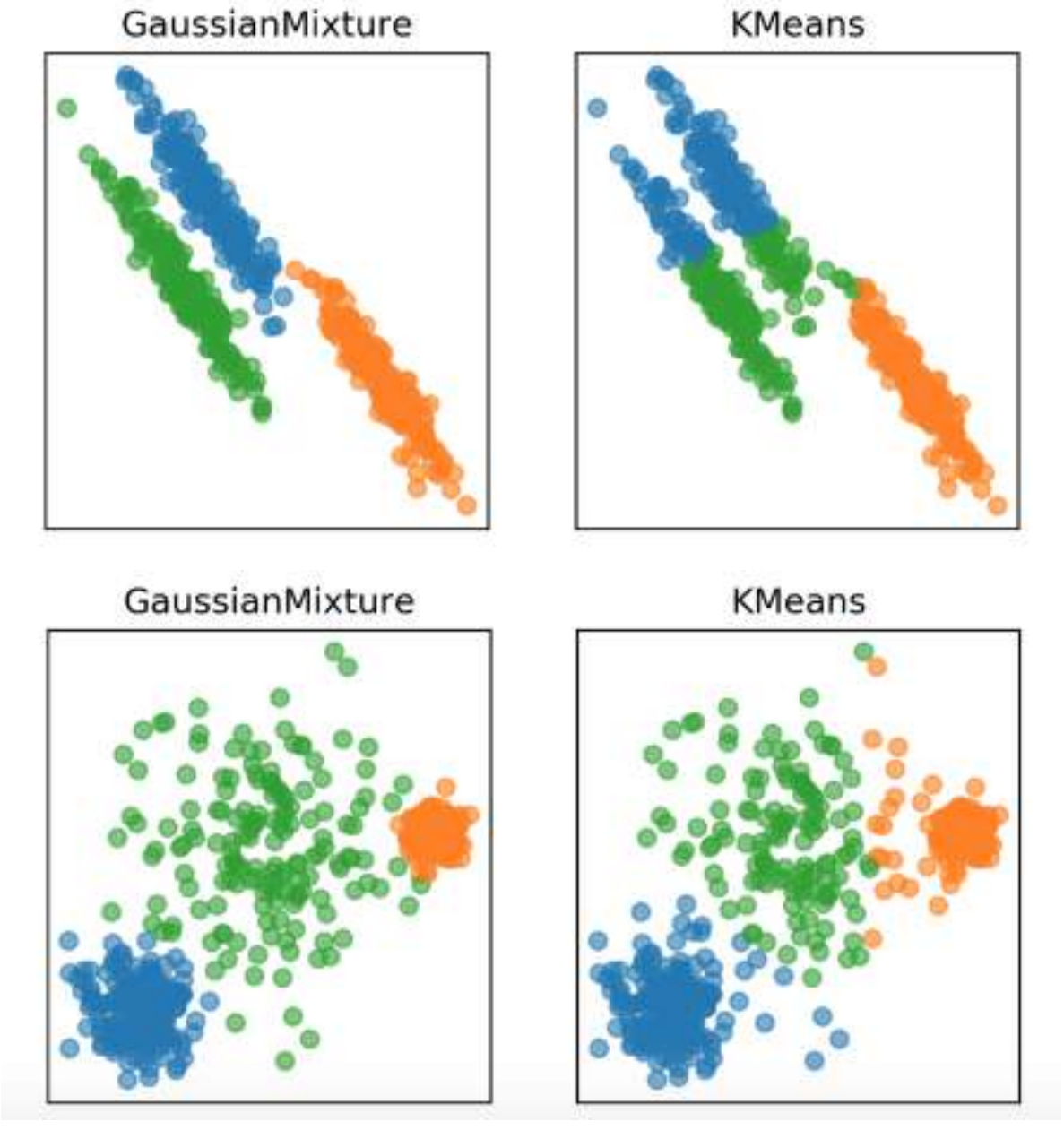
GMMs reduce to K-means under the following assumptions (in which case EM for GMM parameter estimation simplifies to K-means):

- Assume all Gaussians have $\sigma^2 \boldsymbol{I}$ covariance matrices
- Further assume $\sigma \to 0$, so we only need to estimate $\boldsymbol{\mu}_k$, i.e., means

K-means is often called "hard" GMM or GMMs is called "soft" K-means

The posterior r_{nk} provides a probabilistic assignment for $\boldsymbol{x}_n$ to cluster k

GMMs vs. K-Means



GMMs vs. K-Means

Pros/Cons

- k-means is a simpler, more straightforward method, but might not be as accurate because of deterministic clustering
- GMMs can be more accurate, as they model more information (soft clustering, variance), but can be more expensive to compute
- Both methods have a similar set of practical issues (having to select k, pre-process features, and the initialization)

EM Algorithm

EM Algorithm: Motivation

EM is a general procedure to estimate parameters for probabilistic models with hidden/latent variables.

Maximum Likelihood from Incomplete Data via the *EM* Algorithm

By A. P. Dempster, N. M. Laird and D. B. Rubin

Harvard University and Educational Testing Service

[Read before the Royal Statistical Society at a meeting organized by the Research Section on Wednesday, December 8th, 1976, Professor S. D. Silvey in the Chair]

Summary

A broadly applicable algorithm for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the algorithm is derived. Many examples are sketched, including missing value situations, applications to grouped, censored or truncated data, finite mixture models, variance component estimation, hyperparameter estimation, iteratively reweighted least squares and factor analysis.

Maximum Likelihood from Incomplete Data Via the *EM* Algorithm

AP Dempster, NM Laird... - Journal of the Royal ..., 1977 - Wiley Online Library

A broadly applicable algorithm for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the algorithm is derived. Many examples are sketched, including missing value situations, applications to grouped, censored or truncated data, finite mixture models, variance component estimation, hyperparameter estimation, iteratively reweighted least squares and factor analysis.

☆ ▽▽ Cited by 56239 Related articles All 59 versions Web of Science: 25101

EM Algorithm: Setup

- Suppose the model is given by a joint distribution

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})$$

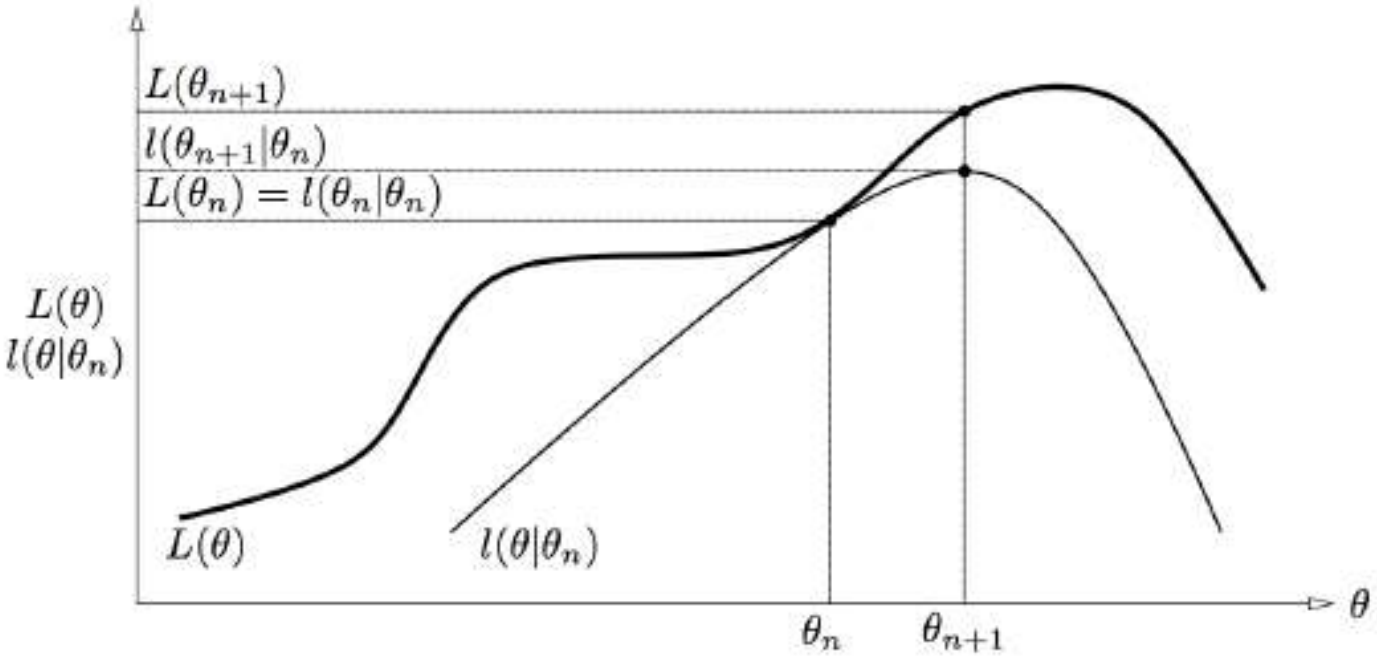
- Given incomplete data $\mathcal{D} = \{\boldsymbol{x}_n\}$ our goal is to compute MLE of $\boldsymbol{\theta}$:

$$\begin{aligned}\boldsymbol{\theta} = \arg\max \ell(\boldsymbol{\theta}) = \arg\max \log p(\mathcal{D}) &= \arg\max \sum_n \log p(\boldsymbol{x}_n|\boldsymbol{\theta}) \\ &= \arg\max \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})\end{aligned}$$

- The objective function $\ell(\boldsymbol{\theta})$ is called *incomplete* log-likelihood
- log-sum form of incomplete log-likelihood is difficult to work with

EM: Principle

- EM: construct lower bound on $\ell(\boldsymbol{\theta})$ (E-step) and optimize it (M-step)
- "Majorization-minimization (MM)"
- Optimizing the lower bound will hopefully optimize $\ell(\boldsymbol{\theta})$ too.



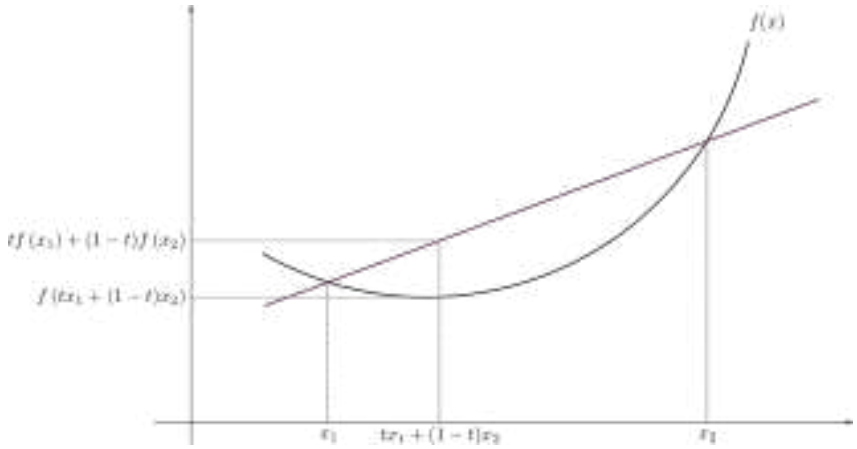
(Figure from tutorial by Sean Borman)

Detour: Jensen's Inequality

Jensen's inequality: if $f(\cdot)$ is a *convex* function, then for any random variable X

$$f(\mathbb{E}X) \leq \mathbb{E}f(X)$$

where equality holds when $f(\cdot)$ is a constant function.



- Example: for $f(x) = x^2$ which is convex:

$$(\mathbb{E}X)^2 \leq \mathbb{E}X^2 \Longrightarrow \text{Var}(X) = \mathbb{E}X^2 - (\mathbb{E}X)^2 \geq 0$$

- Example: for $f(x) = \log x$ which is *concave*:

$$\log(\mathbb{E}X) \geq \mathbb{E}\log(X)$$

Constructing a Lower Bound

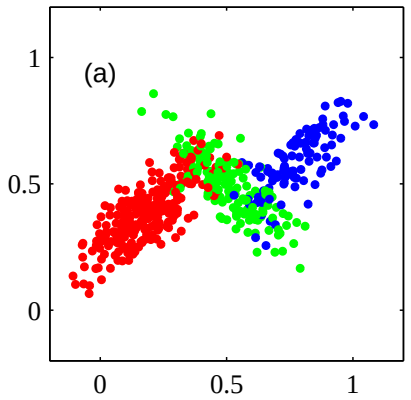
If we define $q(\boldsymbol{z})$ as a distribution over $\boldsymbol{z}$, then

$$\begin{aligned}\ell(\boldsymbol{\theta}) &= \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}) \\ &= \sum_n \underbrace{\log \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}}_{f\left(\mathbb{E}_{q(\boldsymbol{z}_n)}\left[\frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}\right]\right)}\end{aligned}$$

Apply Jensen's inequality to each term, i.e., $f(\mathbb{E}X) \geq \mathbb{E}f(X)$, for concave function $f(\cdot) = \log(\cdot)$. We take the expectation of $X = \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$, a random variable depending on $\boldsymbol{z}_n$, over the probability distribution $q(\boldsymbol{z}_n)$.

$$\ell(\boldsymbol{\theta}) \geq \sum_n \underbrace{\sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}}_{\mathbb{E}_{q(\boldsymbol{z}_n)}\left[f\left(\frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}\right)\right]}$$

Which $q(z)$ Should We Choose?



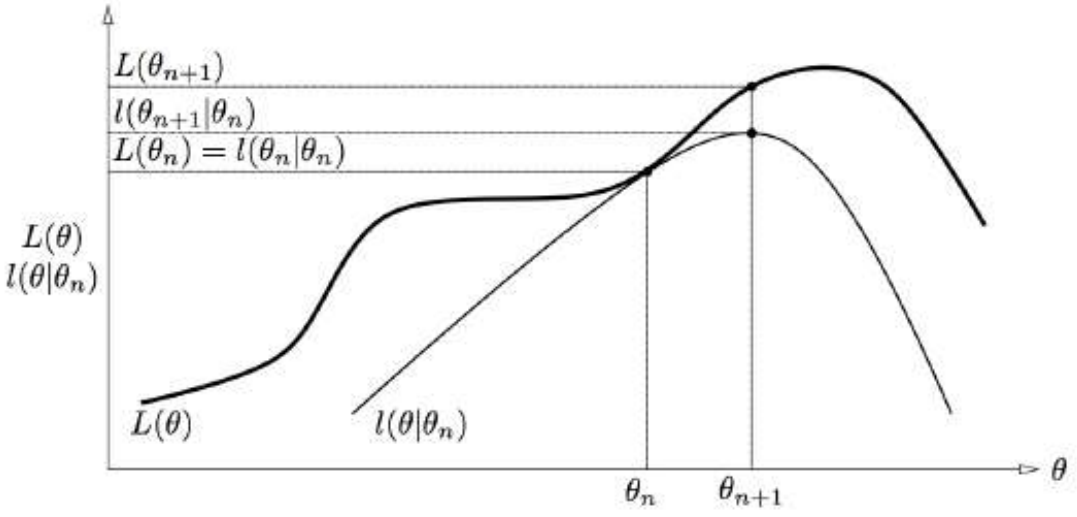
- Consider the previous model where $\boldsymbol{x}$ could be from 3 regions
- We can choose $q(\boldsymbol{z})$ as any valid distribution
- e.g., $q(z = k) = 1/3$ for any of 3 colors
- e.g., $q(z = k) = 1/2$ for red and blue, 0 for green

Which $q(\boldsymbol{z})$ should we choose?

Which $q(z)$ to Choose?

$$\ell(\boldsymbol{\theta}) \geq \sum_n \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$

- The lower bound we derived for $\ell(\boldsymbol{\theta})$ holds for all choices of $q(\cdot)$
- We want a tight lower bound, so given some current estimate $\boldsymbol{\theta}^t$, we will pick $q_t(\cdot)$ such that our lower bound holds with equality at $\boldsymbol{\theta}^t$.
- We will choose a *different* $q_t(\cdot)$ *for each iteration* t.



(Figure from tutorial by Sean Borman)

Pick $q_t(z_n)$

Pick $q_t(\boldsymbol{z}_n)$ so that

$$\ell(\boldsymbol{\theta}^t) = \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^t) = \sum_n \sum_{\boldsymbol{z}_n} q_t(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^t)}{q_t(\boldsymbol{z}_n)}$$

- Pick the distribution where the equality in Jensen's inequality holds.
- Choose $q_t(\boldsymbol{z}_n) \propto p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^t)$!
- Since $q_t(\cdot)$ is a distribution, we have

$$q_t(\boldsymbol{z}_n) = \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^t)}{\sum_k p(\boldsymbol{x}_n, z_n = k | \boldsymbol{\theta}^t)} = \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^t)}{p(\boldsymbol{x}_n | \boldsymbol{\theta}^t)} = p(\boldsymbol{z}_n | \boldsymbol{x}_n; \boldsymbol{\theta}^t)$$

- This is the posterior distribution of $\boldsymbol{z}_n$ given $\boldsymbol{x}_n$ and $\boldsymbol{\theta}^t$

E-Step in GMMs

GMM Parameters

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

Incomplete Data

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

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

Guess the distribution of $\boldsymbol{z}_n$ with the posterior probabilities, given estimates of $\boldsymbol{\theta}^t$:

$$q_t(\boldsymbol{z}_n) = p(\boldsymbol{z}_n = k|\boldsymbol{x}_n; \boldsymbol{\theta}^t) = \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{\sum_{k'=1}^K p(\boldsymbol{x}_n|z_n = k')p(z_n = k')}$$

$$= \frac{N(\boldsymbol{x}_n|\boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t) \times \omega_k^t}{\sum_{k'=1}^K N(\boldsymbol{x}_n|\boldsymbol{\mu}_{k'}^t, \boldsymbol{\Sigma}_{k'}^t) \times \omega_{k'}^t}$$

E- and M-Steps

Our lower bound for the log-likelihood:

$$\ell(\boldsymbol{\theta}) \geq \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log \frac{p(\boldsymbol{x}_n,\boldsymbol{z}_n|\boldsymbol{\theta})}{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)}$$

$$\geq \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log p(\boldsymbol{x}_n,\boldsymbol{z}_n|\boldsymbol{\theta}) - \underbrace{\sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)}_{\text{does not depend on } \boldsymbol{\theta}}$$

Therefore, it suffices to maximize the first term.

Why is this called the E-Step? Because we can view it as computing the *expected (complete) log-likelihood*:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t) = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \underbrace{\log p(\boldsymbol{x}_n,\boldsymbol{z}_n|\boldsymbol{\theta})}_{\text{complete log-likelihood}} = \mathbb{E}_{q_t} \sum_n \log p(\boldsymbol{x}_n,\boldsymbol{z}_n|\boldsymbol{\theta})$$

M-Step: Maximize $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t)$, i.e., $\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t)$

Example: Applying EM to GMMs

What is the E-Step in GMM?

$$r_{nk} = p(z = k|\boldsymbol{x}_n; \boldsymbol{\theta}^t)$$

What is the M-Step in GMM? The Q-function is

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \sum_n \sum_k p(z = k|\boldsymbol{x}_n; \boldsymbol{\theta}^t) \log p(\boldsymbol{x}_n, z = k|\boldsymbol{\theta}) \\ &= \sum_n \sum_k r_{nk} \log p(\boldsymbol{x}_n, z = k|\boldsymbol{\theta}) \\ &= \sum_k \sum_n r_{nk} \log p(z = k) p(\boldsymbol{x}_n|z = k) \\ &= \sum_k \sum_n r_{nk} \left[\log \omega_k + \log N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right] \end{aligned}$$

We have recovered the parameter estimation algorithm for GMMs that we previously discussed!

Iterative and Monotonic Improvement

$$\ell(\boldsymbol{\theta}) \geq \underbrace{\sum_{n} \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)}}_{Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t)}$$

- We can show that $\ell(\boldsymbol{\theta}^{t+1}) \geq \ell(\boldsymbol{\theta}^t)$.
- Recall that we chose $q_t(\cdot)$ in the E-Step such that:

$$\ell(\boldsymbol{\theta}^t) = Q(\boldsymbol{\theta}^t|\boldsymbol{\theta}^t)$$

- However, in the M-step, $\boldsymbol{\theta}^{t+1}$ is chosen to maximize $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t)$, thus

$$\ell(\boldsymbol{\theta}^{t+1}) \geq Q(\boldsymbol{\theta}^{t+1}|\boldsymbol{\theta}^t) = \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t) \geq Q(\boldsymbol{\theta}^t|\boldsymbol{\theta}^t) = \ell(\boldsymbol{\theta}^t)$$

- Note: the EM procedure converges but only to a local optimum

Example: Estimating Height Distributions

Suppose the heights of men and women follow two normal distributions with different parameters:

$$\text{Men} : N(\mu_1, \sigma_1^2) \quad \text{Women} : N(\mu_2, \sigma_2^2)$$

Our data, x_n, $n = 1, 2, \ldots, 5$ is the heights of five people: 179, 165, 175, 185, 158 (in cm).

We are missing the labels z_n = each person's gender. Let π equal the fraction of males in the population.

How do we estimate $\mu_1, \sigma_1, \mu_2, \sigma_2, \pi$?

Example taken from: `http://web1.sph.emory.edu/users/hwu30/teaching/statcomp/Notes/Lecture3_EM.pdf`

Example: E-Step

Initialize $\mu_1^0 = 175$, $\mu_2^0 = 165$, $\sigma_1^0 = \sigma_2^0 = 10$, $\pi^0 = 0.6$.

E-Step: Estimate the probability of each person being male or female.

$$p(z_n = \text{male}|\mu_1, \sigma_1, \mu_2, \sigma_2, \pi) = \frac{0.6 \exp\left[\frac{-(x_n-175)^2}{200}\right]}{0.6 \exp\left[\frac{-(x_n-175)^2}{200}\right] + 0.4 \exp\left[\frac{-(x_n-165)^2}{200}\right]}$$

$$p(z_n = \text{female}|\mu_1, \sigma_1, \mu_2, \sigma_2, \pi) = 1 - p(z_n = \text{male}|\mu_1, \sigma_1, \mu_2, \sigma_2, \pi)$$

Person	1	2	3	4	5
x_i: height	179	165	175	185	158
Prob. male	0.79	0.48	0.71	0.87	0.31

We can use these probabilities to find $Q(\boldsymbol{\theta}^1|\boldsymbol{\theta}^0)$.

Example: M-Step

M-Step: Find $\mu_1^1, \mu_2^1, \sigma_1^1, \sigma_2^1, \pi^1$ that maximize $Q(\boldsymbol{\theta}^1|\boldsymbol{\theta}^0)$:

$$\mu_1^1 = \frac{\sum_{n=1}^5 p(z_n = \text{male}|\boldsymbol{\theta}^0) x_n}{\sum_{n=1}^5 p(z_n = \text{male}|\boldsymbol{\theta}^0)}, \; \mu_2^1 = \frac{\sum_{n=1}^5 p(z_n = \text{female}|\boldsymbol{\theta}^0) x_n}{\sum_{n=1}^5 p(z_n = \text{female}|\boldsymbol{\theta}^0)},$$

$$\sigma_1^1 = \frac{\sum_{n=1}^5 p(z_n = \text{male}|\boldsymbol{\theta}^0)(x_n - \mu_1^1)^2}{\sum_{n=1}^5 p(z_n = \text{male}|\boldsymbol{\theta}^0)},$$

$$\sigma_2^1 = \frac{\sum_{n=1}^5 p(z_n = \text{female}|\boldsymbol{\theta}^0)(x_n - \mu_2^1)^2}{\sum_{n=1}^5 p(z_n = \text{female}|\boldsymbol{\theta}^0)},$$

$$\pi^1 = \frac{1}{5} \sum_{n=1}^5 p(z_n = \text{male}|\boldsymbol{\theta}^0)$$

Here we are using the MLE solution that we derived earlier for GMMs.

Numerically, $\mu_1^1 = 176$, $\mu_2^1 = 167$, $\sigma_1^1 = 8.7$, $\sigma_2^1 = 9.2$, $\pi^1 = 0.63$.

Example: After 15 iterations...

After iteration 1:

Person	1	2	3	4	5
x_1: height	179	165	175	185	158
Prob. male	0.79	0.48	0.71	0.87	0.31

Parameter estimates: $\mu_1^1 = 176$, $\mu_2^1 = 167$, $\sigma_1^1 = 8.7$, $\sigma_2^1 = 9.2$, $\pi^1 = 0.63$.

After iteration 15:

Person	1	2	3	4	5
x_1: height	179	165	175	185	158
Prob. male	0.999997	0.0004009	0.9991	1	2.44e-06

Final estimates: $\mu_1 = 179.6$, $\mu_2 = 161.5$, $\sigma_1 = 4.1$, $\sigma_2 = 3.5$, $\pi = 0.6$.

Applications of EM

EM is a general method to deal with hidden data; we have studied it in the context of hidden *labels* (unsupervised learning). Common applications include:

- Filling in missing data in a sample
- Discovering the value of latent model variables
- Estimating parameters of finite mixture models
- As an alternative to direct maximum likelihood estimation

You Should Know

- How GMMs differ from k-means (and why we care)
- The difference between complete, incomplete data/likelihood
- How to learn the parameters in a GMM
- EM is a general procedure for maximizing a likelihood with *latent (unobserved) variables*
- The two steps of EM:
  - (1) Estimating unobserved data from observed data and current parameters
  - (2) Using this "complete" data to find the maximum likelihood parameter estimates
- Pros: Guaranteed to converge, no parameters to tune (e.g., compared to gradient methods)
- Cons: Can get stuck in local optima, can be expensive

Another Example: Multinomial Distributions

Suppose we are trying to model the number of people who will vote for one of four candidates. We know that the probability of a single person voting for each candidate is:

$$(p_1, p_2, p_3, p_4) = \left(\frac{1}{2} + \frac{\theta}{4}, \frac{1-\theta}{4}, \frac{1-\theta}{4}, \frac{\theta}{4}\right).$$

Let $Y = (y_1, y_2, y_3, y_4)$ denote our observation of the number of votes for each candidate. How do we estimate θ?

- Option 1: MLE. But it is "hard" to optimize the log-likelihood...

$$\log p(Y|\theta) = y_1 \log\left(\frac{1}{2} + \frac{\theta}{4}\right) + (y_2 + y_3)\log(1-\theta) + y_4 \log\theta$$

- Option 2: EM. We will introduce two *unobserved/latent variables* x_0 and x_1, where $x_0 + x_1 = y_1$. In other words, y_1 combines the counts of two categories, whose individual counts are given by x_0 and x_1. We have a probability $\frac{1}{2}$ of picking the x_0 category, and a probability $\frac{\theta}{4}$ of picking the x_1 category.

Applying EM to Multinomial Distributions

Complete likelihood function:

$$\ell(\theta) = \log p(X, Y|\theta) = (x_1 + y_4)\log\theta + (y_2 + y_3)\log(1-\theta)$$

E-Step: Find $Q(\theta|\theta^t) = \mathbb{E}_{q_t}[\ell(\theta)]$, where q_t is the posterior distribution of x_1 given $y_1, y_2, y_3, y_4, \theta$. To do this, we need to find $\mathbb{E}_{q_t}[x_1]$:

$$x_1^{t+1} = \mathbb{E}_{q_t}[x_1] = y_1 \frac{\frac{\theta^t}{4}}{\frac{1}{2} + \frac{\theta^t}{4}}.$$

M-Step: Find θ that maximizes

$$Q(\theta|\theta^t) = \left(y_1 \frac{\frac{\theta^t}{4}}{\frac{1}{2}+\frac{\theta^t}{4}} + y_4\right)\log\theta + (y_2 + y_3)\log(1-\theta).$$

$$\theta^{t+1} = \frac{x_1^{t+1} + y_4}{x_1^{t+1} + y_4 + y_2 + y_3}$$

What Does This Look Like in Practice?

Observe $Y = (125, 18, 20, 34)$ and initialize $\theta^0 = 0.5$.

k	Parameter update $\theta^{(k)}$	Convergence to $\hat{\theta}$ $\theta^{(k)} - \hat{\theta}$	Convergence rate $(\theta^{(k)} - \hat{\theta})/(\theta^{(k-1)} - \hat{\theta})$
0	.500000000	.126821498	
1	.608247423	.018574075	.1465
2	.624321051	.002500447	.1346
3	.626488879	.000332619	.1330
4	.626777323	.000044176	.1328
5	.626815632	.000005866	.1328
6	.626820719	.000000779	.1328
7	.626821395	.000000104	
8	.626821484	.000000014	
$\hat{\theta}$	.626821498	Stop	