CSC 580 Principles of Machine Learning

11 PGM: Gaussian mixture models; Expectation-Maximization (EM) algorithms

Chicheng Zhang

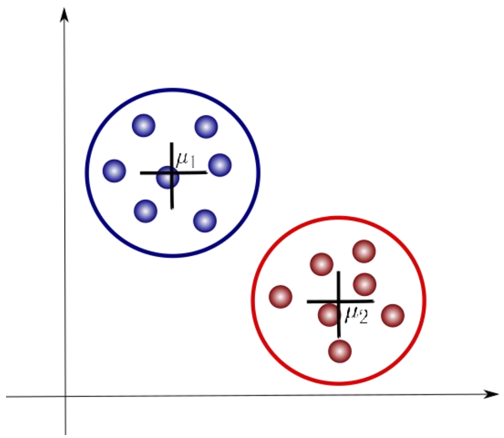
Department of Computer Science

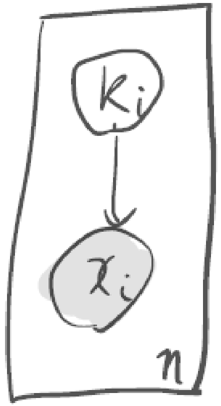
THE UNIVERSITY OF ARIZONA

*slides credit: built upon CSC 580 Fall 2021 lecture slides by Kwang-Sung Jun

Gaussian mixture model (GMM) for clustering

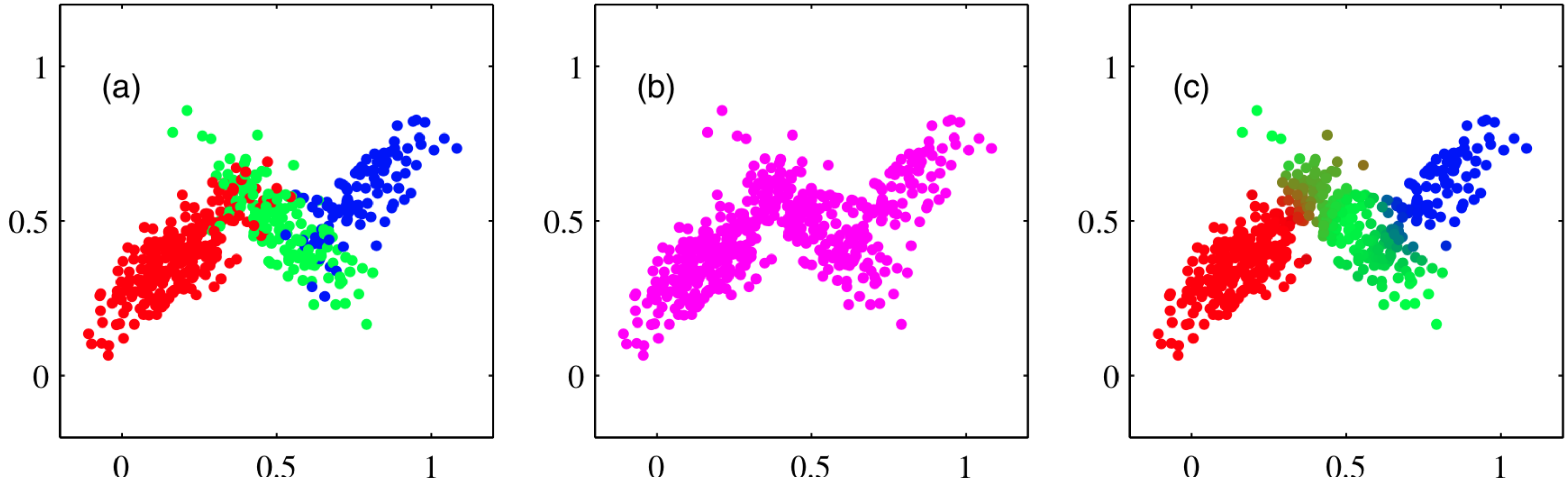
- Clustering
- Data: $S = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$
- Given: K - the number of clusters.
- Generative story:
    - $k \sim \text{Categorical}(\pi)$ (*hidden)*
    - $x \mid k \sim N(\mu_k, \Sigma_k)$





- Maximum likelihood estimation: $\underset{\pi, \{\mu_k, \Sigma_k\}_{k=1}^{K}}{\text{argmax}} \; \sum_i \log(\sum_{k=1}^{K} \pi_k \, p(x_i; \mu_k, \Sigma_k))$
    - How to solve it?
    - How do we get the cluster assignments?

Illustration



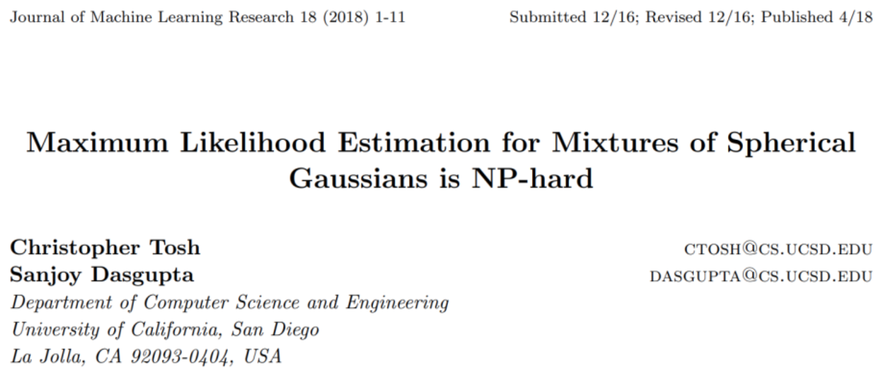
- Mixture of 3 Gaussians
- (a) is ground truth (we don't know this).
- (b) is what we see, (c) is what the algorithm can recover.

GMM for clustering: algorithms

- Maximum likelihood estimation

$$\underset{\pi,\{\mu_k,\Sigma_k\}_{k=1}^K}{\operatorname{argmax}} \sum_i \log\left(\sum_{k=1}^K \pi_k\, p(x_i; \mu_k, \Sigma_k)\right)$$

  is (1) computationally hard (2) ill-posed (see later slides)



Journal of Machine Learning Research 18 (2018) 1-11 Submitted 12/16; Revised 12/16; Published 4/18

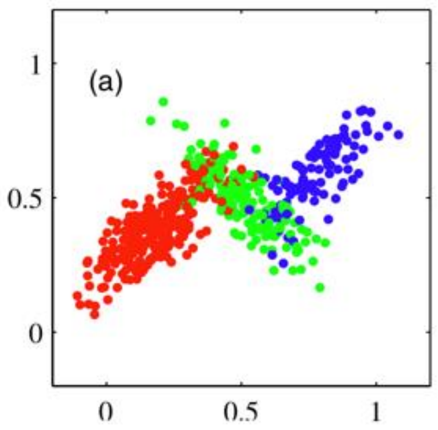
Maximum Likelihood Estimation for Mixtures of Spherical Gaussians is NP-hard

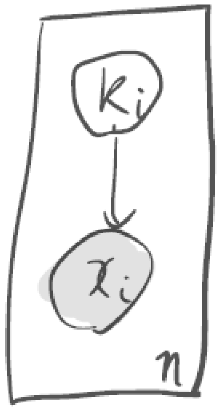
Christopher Tosh CTOSH@CS.UCSD.EDU
Sanjoy Dasgupta DASGUPTA@CS.UCSD.EDU
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0404, USA

- How to design computationally efficient algorithms that can reasonably maximize the log-likelihood function?

- Observation: if for each data point i,

  we not only have x_i but also have k_i, then MLE is easy to calculate



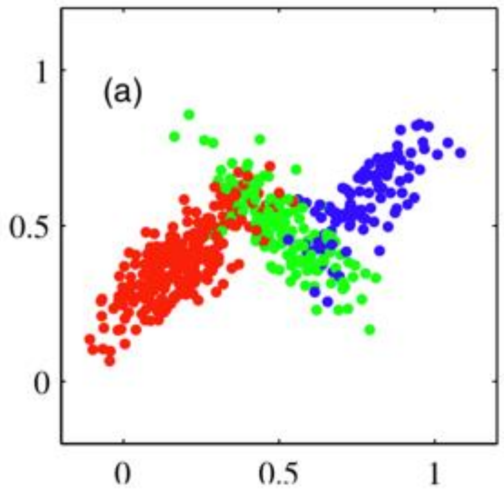


Warmup: MLE for GMM with known cluster membership

- Maximize likelihood ⇔ maximize log-likelihood
- $\max_{\pi,\{\mu,\Sigma\}} L(\pi,\{\mu,\Sigma\}) = \max_{\pi,\{\mu,\Sigma\}} \sum_i \log P(x_i, k_i; \pi, \{\mu,\Sigma\})$

$$= \max_{\pi,\{\mu,\Sigma\}} \left(\sum_i \log P(\, x_i \mid k_i; \{\mu,\Sigma\}\,) + \sum_i \log P(k_i;\pi) \right)$$

$$= \max_{\{\mu,\Sigma\}} \sum_i \log P(x_i \mid k_i; \{\mu,\Sigma\}) \; + \max_{\pi} \sum_i \log P(k_i;\pi)$$

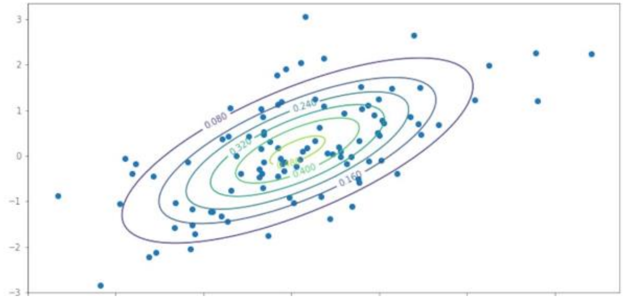


- $\underset{\pi}{\text{maximize}} \sum_i \log P(k_i;\pi) = \sum_{k=1}^{K} n_k \ln \pi_k$, where $n_k = \#\{i:\ k_i = k\}$

  $\Rightarrow \pi_k = \frac{n_k}{n}$

- $\max_{\{\mu,\Sigma\}} \sum_i \log P(\, x_i \mid k_i; \{\mu,\Sigma\}\,) = \sum_k \max_{\mu_k,\Sigma_k} \sum_{i:k_i=k} \log P(\, x_i \mid k_i = k; \mu_k, \Sigma_k \,)$

Warmup: MLE for GMM with known cluster membership (cont'd)



- $\max_{\mu_k, \Sigma_k} \sum_{i: k_i = k} \ln P(\, x_i \mid k_i = k; \mu_k, \Sigma_k \,)$

- Simplified problem: $\max_{\mu, \Sigma} \sum_i \ln N(x_i; \mu, \Sigma)$, where N here denotes Gaussian pdf

$$N(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1} (x-\mu)\right)$$

- Observation 1: for any fixed Σ, the optimal μ is $\mu = \frac{1}{n} \sum_i x_i$ (Exercise)
- Observation 2: for any fixed μ, the optimal Σ is such that $\Lambda = \Sigma^{-1}$ equals

$$\operatorname*{argmax}_{\Lambda} f(\Lambda) \coloneqq \frac{1}{2} \sum_i \ln|\Lambda| - \frac{1}{2} (x_i - \mu)^\top \Lambda (x_i - \mu)$$

  - Fact: f is concave in Λ
  - $\nabla f(\Lambda) = 0 \Rightarrow n\Lambda^{-1} - \sum_i (x_i - \mu)(x_i - \mu)^\top = 0 \Rightarrow \Sigma = \frac{1}{n} \sum_i (x_i - \mu)(x_i - \mu)^\top$

https://www.youtube.com/watch?v=jAyTgkiaBbY

Warmup: MLE for GMM with known cluster membership (cont'd)

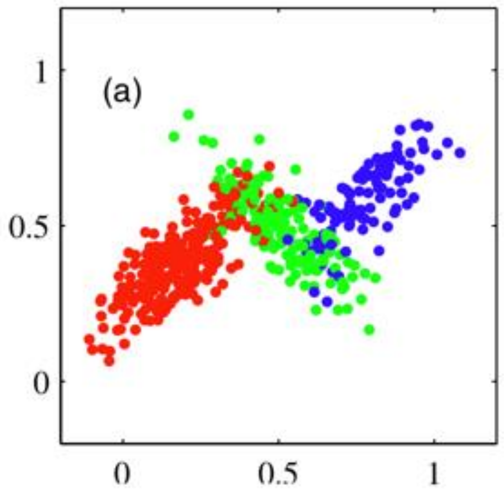
- In summary, for every k, the solution of

$$\max_{\mu_k, \Sigma_k} \sum_{i:k_i=k} \ln P(\, x_i \mid k_i = k; \mu_k, \Sigma_k \,)$$

is given by:

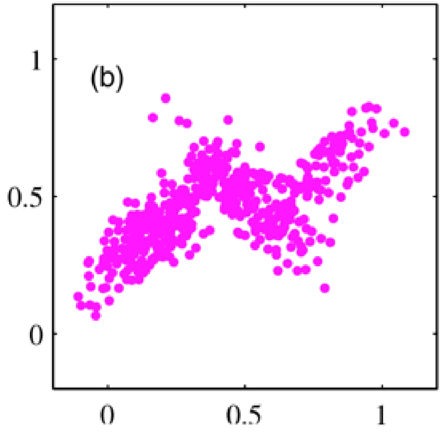
$$\mu_k = \frac{1}{n_k} \sum_{i:k_i=k} x_i$$

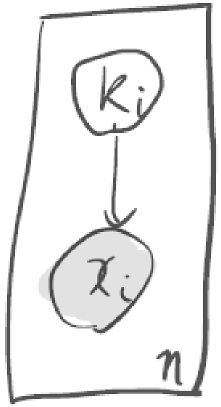
$$\Sigma_k = \frac{1}{n_k} \sum_{i:k_i=k} (x_i - \mu_k)(x_i - \mu_k)^\top$$



- Also, recall that for every k, the optimal $\pi_k = \frac{n_k}{n}$

GMM for clustering: algorithms





- What is the cluster memberships are unknown?
- This is generally known as the *latent variable* issue
- Expectation-Maximization (EM) algorithm (Dempster et al, 1977) provides a *general* approach for approximate MLE for probabilistic models with latent variables
  - Has wide applications well-beyond GMMs
- High-level idea: *reduce* to MLE for fully-observed probabilistic models

EM algorithm: high-level idea

- Given: a probabilistic model $P(x, z; \theta)$,

  with x being the observed part, z being the latent part

- Would like to maximize the log-likelihood on the observed data: $\ln P(x; \theta) = \ln \sum_z P(x, z; \theta)$

- Maximizing $\ln \sum_z P(x, z; \theta)$ is intractable => instead, maximize a lower bound of it

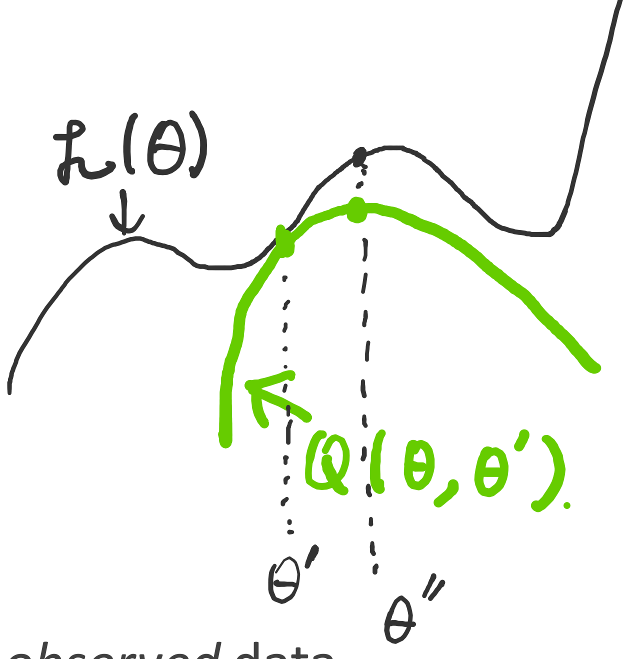
$$\ln P(x; \theta) = \ln \sum_z P(x, z; \theta) = \ln \sum_z P(z \mid x; \theta') \cdot \frac{P(x,z;\theta)}{P(z|x;\theta')}$$

$$\geq \sum_z P(z \mid x; \theta') \ln \frac{P(x,z;\theta)}{P(z|x;\theta')} \quad \text{(Jensen's inequality \& concavity of } \ln x\text{)}$$

- With n iid samples

$$\underbrace{\sum_{i=1}^n \ln P(x_i; \theta)}_{\mathcal{L}(\theta)} \geq \underbrace{\sum_{i=1}^n \sum_z P(z \mid x_i; \theta') \ln \frac{P(x_i,z;\theta)}{P(z|x_i;\theta')}}_{Q(\theta;\theta')}$$

EM algorithm: high-level idea



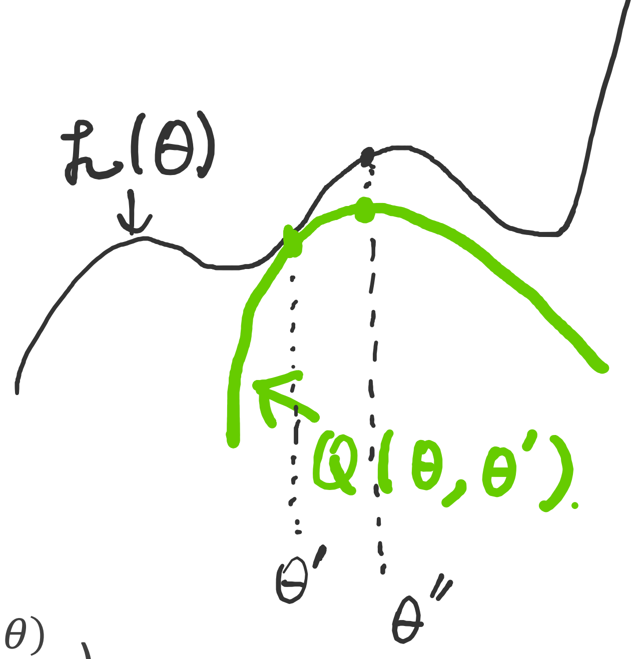
- $$\underbrace{\sum_{i=1}^{n} \ln P(x_i;\theta)}_{\mathcal{L}(\theta)} \geq \underbrace{\sum_{i=1}^{n} \sum_z P(z \mid x_i;\theta') \ln \frac{P(x_i,z;\theta)}{P(z \mid x_i;\theta')}}_{Q(\theta;\theta')}$$

- Why optimizing $Q(\theta;\theta')$?
  - Can be viewed as the log-likelihood of model θ on a “soft” set of *fully-observed* data

- The lower bound approximate $Q(\theta;\theta')$ is sometimes tight
  - At $\theta = \theta'$, $Q(\theta';\theta') = \mathcal{L}(\theta')$
  - For general θ, $\mathcal{L}(\theta) - Q(\theta;\theta') = \sum_{i=1}^{n} \mathrm{KL}\big(P(z \mid x_i;\theta'), P(z \mid x_i;\theta)\big) \geq 0$

- Kullback-Leibler (KL) divergence: $\mathrm{KL}(p,q) = \mathrm{E}_{z\sim p} \ln \frac{p(z)}{q(z)}$
- Properties: $\mathrm{KL}(p||q) \geq 0$, for all p,q; $\mathrm{KL}(q||q) = 0$, for all q

EM algorithm: the procedure



1. Initialize parameters $\theta^{(1)}$
2. For $n = 1,2, \ldots$:
   - E-step: for each example i, evaluate $P\left(z \mid x_i; \theta^{(n)} \right)$

     (This is for calculating $Q\left(\theta; \theta^{(n)}\right) = \sum_{i=1}^{n} \sum_z P\left(z \mid x_i; \theta^{(n)} \right) \ln \frac{P(x_i, z; \theta)}{P(z|x_i; \theta^{(n)})}$)

   - M-step: $\theta^{(n+1)} \leftarrow \mathrm{argmax}_\theta \, Q(\theta; \theta^{(n)})$

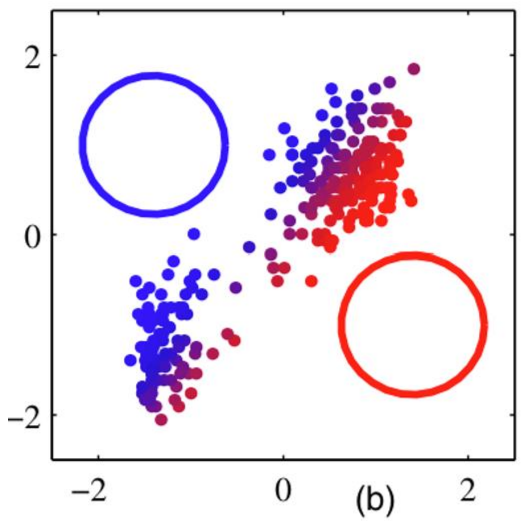
   - Check convergence of either log-likelihood or parameters; if yes, return

- Monotone improvement guarantee: $\mathcal{L}\left(\theta^{(n)}\right) = Q\left(\theta^{(n)}, \theta^{(n)}\right) \leq Q\left(\theta^{(n+1)}, \theta^{(n)}\right) \leq \mathcal{L}\left(\theta^{(n+1)}\right)$

EM algorithm: application to GMMs

- Recall: latent variable k (cluster membership), parameters $\theta = (\pi, \{\mu, \Sigma\})$

- The E-step:
  - for each example i, evaluate $P(\, k_i \mid x_i; \theta \,)$ for $\theta = \theta^{(n)}$
  - $P(\, k_i = k \mid x_i; \theta \,) = \frac{P(k_i = k,\, x_i; \theta)}{P(x_i; \theta)} = \frac{\pi_k N(x_i; \mu_k, \Sigma_k)}{\sum_{c=1}^{K} \pi_c N(x_i; \mu_c, \Sigma_c)} =: \gamma_{ik}$
  - γ_{ik}: the *responsibility* component k has for generating x_i



Conceptually, γ_{ik} can be thought of as

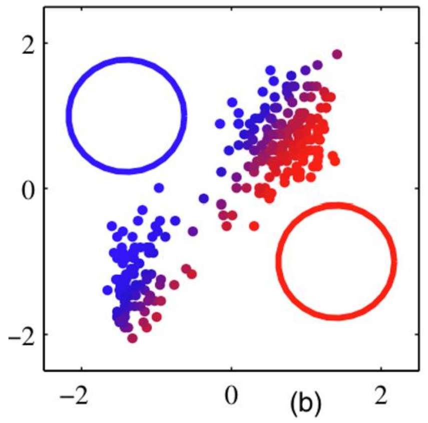
- soft cluster membership
- “pseudo-count” of data point (x_i, k)

EM algorithm: application to GMMs (cont'd)

- The M-step:

$$\theta^{(n+1)} \leftarrow \text{argmax}_\theta\, Q(\theta; \theta^{(n)}),$$

where $Q\left(\theta; \theta^{(n)}\right) = \sum_{i=1}^{n} \sum_k P\left(k_i = k \mid x_i; \theta^{(n)} \right) \ln \frac{P(x_i, k; \theta)}{P(k|x_i; \theta^{(n)})}$



This is equivalent to $\text{argmax}_\theta \sum_{i=1}^{n} \sum_k \gamma_{ik} \ln P(x_i, k_i = k; \theta)$

- Can view the above as the log-likelihood of weighted dataset $\{(x_i, k), \gamma_{ik}\}_{i \in [n], k \in [K]}$

- Using MLE for GMM with fully-observed data (recall slide 7), we have:

$$\pi_k = \frac{n_k}{n}, \text{ where } n_k = \sum_{i=1}^{n} \gamma_{ik}$$

EM algorithm: application to GMMs (cont'd)

- M-step

$$\text{argmax}_\theta \sum_{i=1}^{n} \sum_k \gamma_{ik} \ln P(x_i, k_i = k; \theta)$$

What about optimal $\{\mu, \Sigma\}$?

(Previously)

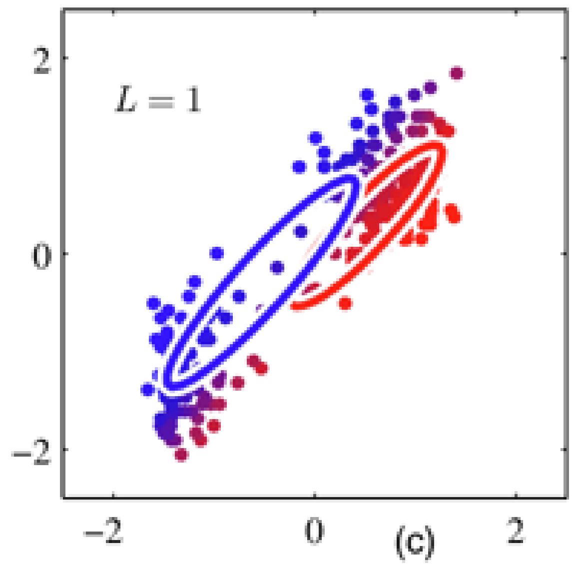
$$\mu_k = \frac{1}{n_k} \sum_{i:k_i=k} x_i$$

$$\Sigma_k = \frac{1}{n_k} \sum_{i:k_i=k} (x_i - \mu_k)(x_i - \mu_k)^\top$$

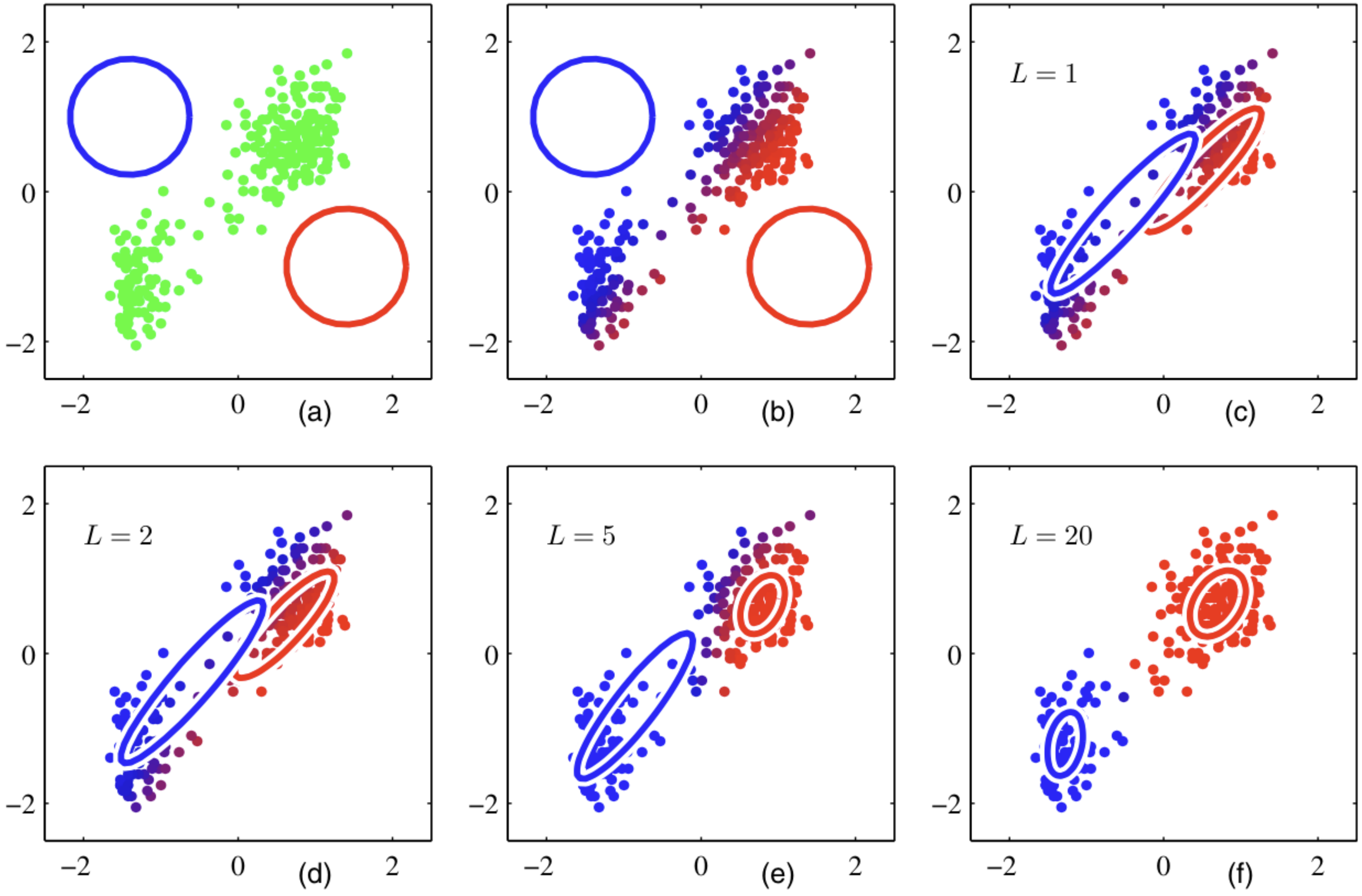
(Now, for optimizing $Q\left(\theta; \theta^{(n)}\right)$)

$$\mu_k = \frac{\sum_i \gamma_{ik} x_i}{\sum_i \gamma_{ik}}$$

$$\Sigma_k = \frac{\sum_i \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^\top}{\sum_i \gamma_{ik}}$$



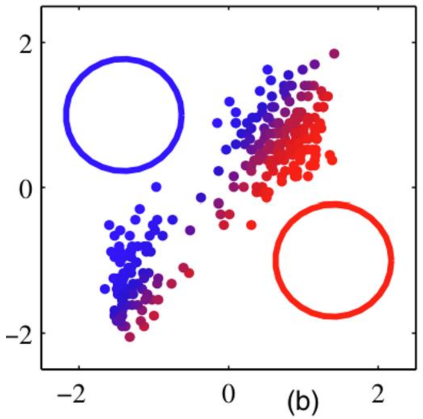
EM in action

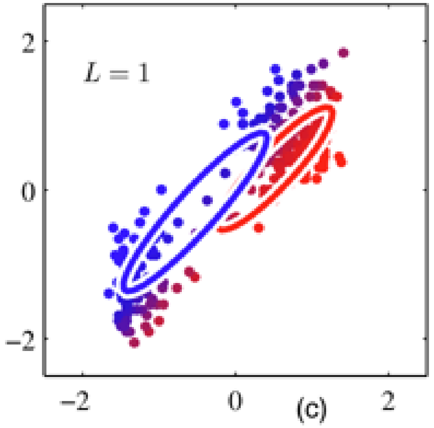


EM for GMM: 1-slide summary

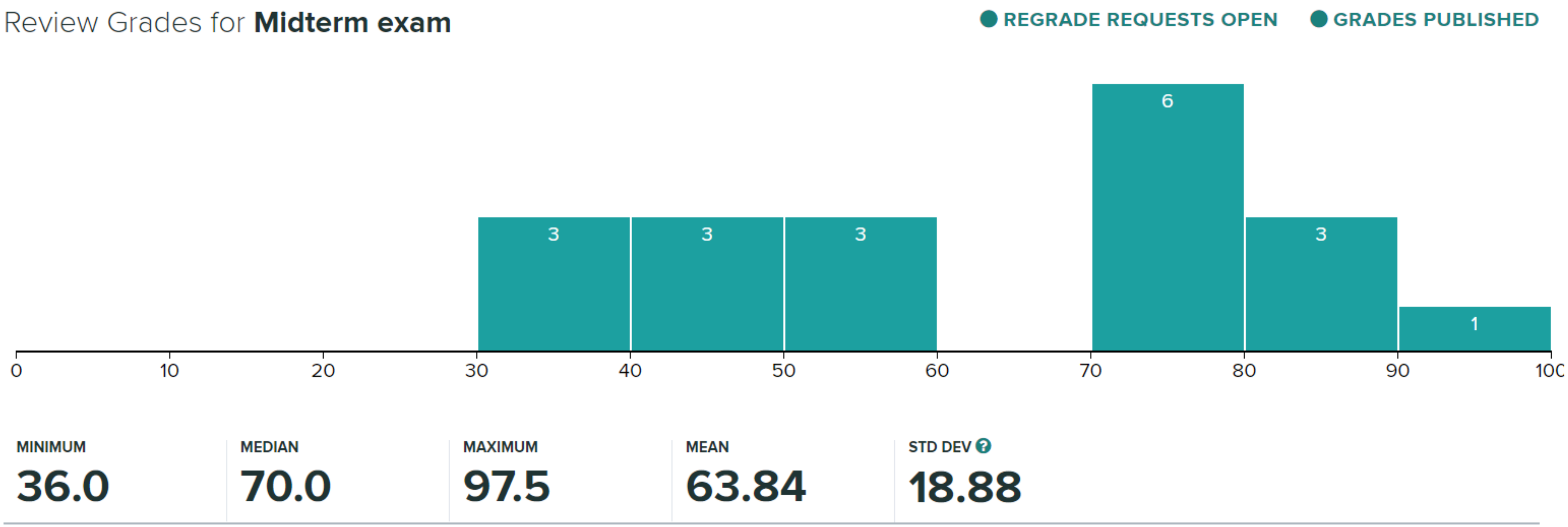
- Initialize: $\pi \in \Delta^K$, $\{\mu_k \in \mathbb{R}^d, \Sigma_k \in \mathbb{R}^{d\times d}\}_{k=1}^K$

- (E)xpectation step: for every i, k:
  - $\gamma_{ik} = \frac{\pi_k \, p(x_i \mid z_i = k)}{\sum_{k'=1}^K \pi_{k'} \, p(x_i \mid z_i = k')}$ responsibility
  - Let $n_k = \sum_{i=1}^n \gamma_{ik}$ soft counts
- (M)aximization step: for every k:
  - $\mu_k' = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ik} x_i$
  - $\Sigma_k' = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ik} (x_i - \mu_k')(x_i - \mu_k')^\top$ note we use μ_k' rather than μ_k
  - $\pi_k' = \frac{n_k}{n}$
  - Set $\mu_k \leftarrow \mu_k'$, $\Sigma_k \leftarrow \Sigma_k'$, $\pi_k \leftarrow \pi_k'$,
- Stop when: the log likelihood does not increase much or the parameters do not change much.





Midterm exam: summary



- My suggestion: demonstrating clarity on basic concepts / definitions >> calculations
- If you are on a right track to solve a question, I am usually generous in giving partial credits

Tips

- Stopping criteria:
  - Likelihood-based: $\frac{|\mathcal{L}(\theta') - \mathcal{L}(\theta)|}{|\mathcal{L}(\theta)|} \leq \epsilon$
  - Parameter-based: $\|\mu_k - \mu_k'\| + \|\Sigma_k - \Sigma_k'\|_F + \|\pi_k - \pi_k'\| \leq \epsilon$

- Initialization of $\pi, \{\mu, \Sigma\}$
  - E.g. $\pi \leftarrow \left(\frac{1}{K}, \dots, \frac{1}{K}\right)$, $\mu \leftarrow$cluster centers of Lloyd's algorithm, $\Sigma = \mathrm{I}$

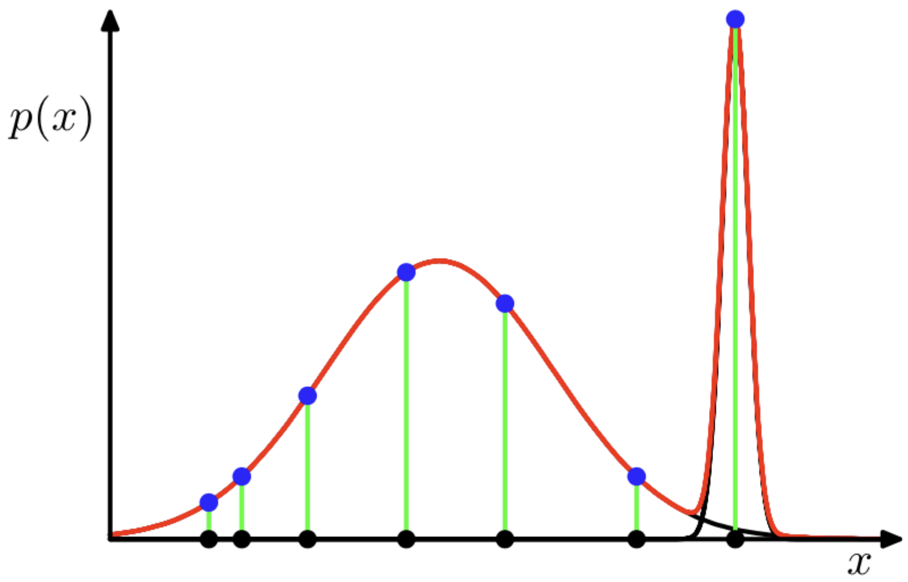
- Beware of pitfalls

Pitfalls

- Maximum likelihood of GMM can result in severe overfitting
- In the log-likelihood expression $\sum_{i=1}^{n} \ln P(x_i;\theta)$,

  it is possible to set θ so that:

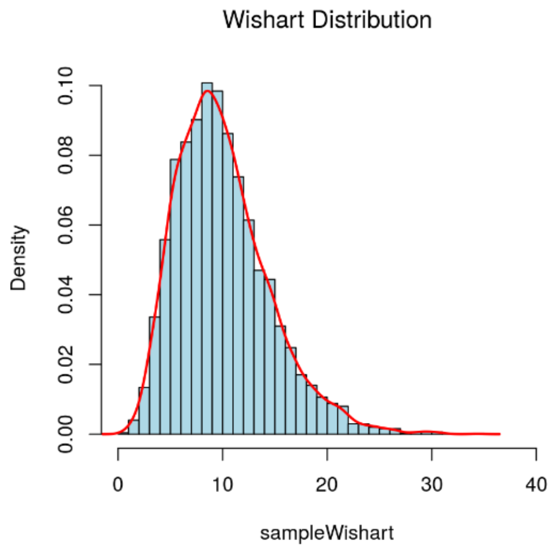
  for one example i, $\ln P(x_i;\theta)$ is arbitrarily large



- Imagine Gaussian MLE on one data point:

$$\max_{\mu,\sigma^2} \ln N(x_1;\mu,\sigma^2) = \max_{\mu,\sigma^2} \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\right)$$

- Solution:
  - Bayesian approach: instead of MLE
    - Put a prior on Σ, e.g. Wishart distribution
    - Compute maximum-a-posteriori (MAP) estimate
  - Detect overly small Σ_k and restart EM



https://www2.karlin.mff.cuni.cz/~maciak/NMST539/cvicenie2018_4.html

Lloyd's algorithm is EM in the limit

- Suppose we use EM for $\underset{\pi,\{\mu,\Sigma\}}{\text{maximize}}\ L(\pi,\{\mu,\Sigma\})$, subject to:

  for every k,

  $\Sigma_k = \epsilon \cdot I \in \mathbb{R}^{d\times d}$ for some $\epsilon > 0$

  $\pi_k = \frac{1}{K}$

- Running the EM algorithm: (fix Σ_k, π throughout -- do not update them)
- E-step:
  - $p(x\ |\mu_k, \Sigma_k) \propto \exp\left(-\frac{1}{2\epsilon}\ \|x-\mu_k\|_2^2\right)$
  - $\gamma_{ik} = \dfrac{\pi_k \exp\left(-\frac{\|x_i-\mu_k\|^2}{2\epsilon}\right)}{\sum_{k'=1}^{K} \pi_{k'} \exp\left(-\frac{\|x_i-\mu_{k'}\|^2}{2\epsilon}\right)}$

- Imagine $K = 2$

Lloyd's algorithm is EM in the limit

- Initialize: $\pi \in \Delta^K$, $\{\mu_k \in \mathbb{R}^d, \Sigma_k \in \mathbb{R}^{d\times d}\}_{k=1}^K$

  Imagine π = Uniform, $\Sigma_k = \frac{1}{\epsilon} I$ with a very small ϵ

- (E)xpectation step:
  - $\gamma_{ik} = \frac{\pi_k \, p(x_i \mid z_i = k)}{\sum_{k'=1}^K \pi_{k'} \, p(x_i \mid z_i = k')}$ — $\gamma_{ik} = 1$ if μ_k is the cluster center closest to x_i; 0 otherwise
  - Let $n_k = \sum_{i=1}^n \gamma_{ik}$ — count how many points assigned to the centroid μ_k
- (M)aximization step:
  - $\mu_k = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ik} x_i$ — update centroid μ_k as the mean of the points assigned to cluster k
  - $\Sigma_k = \frac{1}{n_k} \sum_{i=1}^n \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^\top$
  - $\pi_k = \frac{n_k}{n}$
- Stop when: the log likelihood does not increase much or parameter does not change much.

Gaussian Mixture Models: additional remarks

- EM is not the only method that maximizes likelihood in GMMs
  - E.g. can just gradient ascent on the likelihood function

Gradient-Based Training of Gaussian Mixture Models for High-Dimensional Streaming Data

Alexander Gepperth[1] · Benedikt Pfülb[1]

Accepted: 15 July 2021 / Published online: 17 August 2021

- Another popular approach: spectral methods
  - Key idea: use *Method of Moments* to estimate model parameters
  - Has provable guarantees when the model is ``well-specified''
  - Can be combined with EM

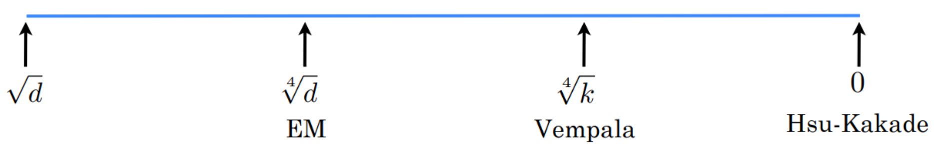
Spectral Methods meet EM: A Provably Optimal Algorithm for Crowdsourcing

Yuchen Zhang, Xi Chen, Dengyong Zhou, Michael I. Jordan

- Generally, stronger assumption on data generating process

  => easier to learn

Algorithms that assume a certain amount of separation:



http://www.phillong.info/stoc13/stoc13_ml_sanjoy_dasgupta.pdf

EM as a generic tool: additional remarks

- **EM is universal**: any situation where you have latent variables.
  - E-step: compute the posterior probability (=responsibilities) for the latent variables
  - M-step: use the responsibilities as 'soft membership', and find parameters that maximize $\sum_j q(z=j) \cdot \ln(p(x, z=j|\theta))$. I.e., weighted joint likelihood.

- Other popular examples:
  - Semi-supervised learning
    - Some labels are unobserved – the hidden labels are the z_i's!

- Missing data
  - Some features are often missing for various reason. (e.g., for survey, they just did not fill out)
  - "Grading an example without an answer key" – CIML Sec 16.1
  - Once you provide a generative model, you know how to apply EM

Recap

- GMM: a generative model.
- Difference from supervised learning: we must infer the latent, unobserved variable.
- Connection to k-means and Lloyd's algorithm

- The power of graphical models: specify reasonable generative model, and what you should do, ideally, is already well-defined.
  - The pain is in the computational complexity
  - EM is one way to get around.

- Additional reading: Bishop, "Pattern Recognition and Machine Learning", Chap. 9