CSC 480/580 Principles of Machine Learning

10 Probabilistic ML: Gaussian mixture models; Expectation-Maximization (EM)

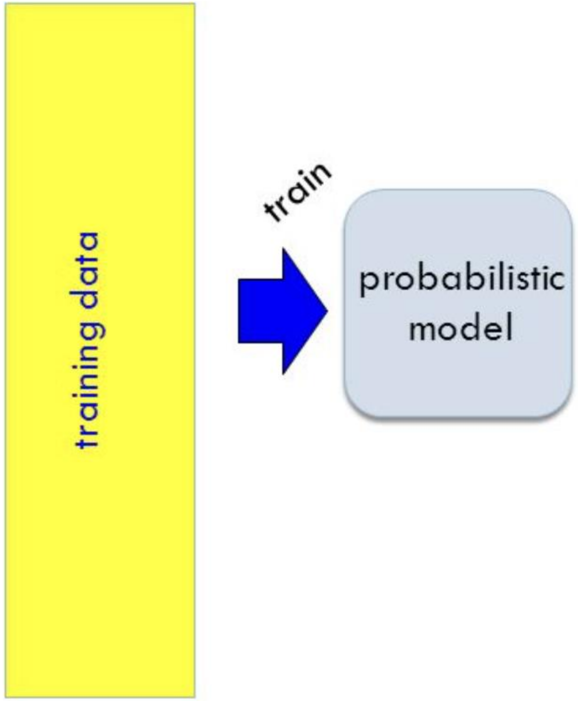
Chicheng Zhang

Department of Computer Science

THE UNIVERSITY OF ARIZONA®

Probabilistic modeling: systematic approach for ML

- The recipe:
  1. Model how the data is generated by probabilistic models, but with parameters unspecified (modeling assumption / generative story)
  2. (Training) Learn the model parameter $\hat{\theta}$
  3. (Test) Make prediction / decision based on the learned model $P(z; \hat{\theta})$



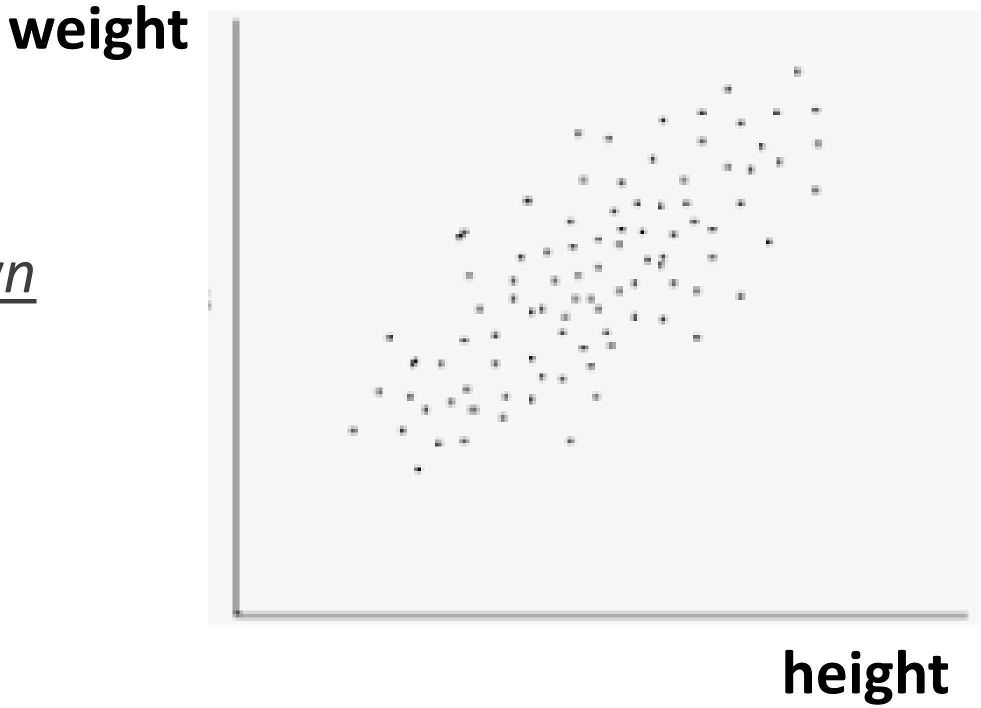
http://slideplayer.com/slide/4527958/

Warm-up Example: estimate population height &weight

- Suppose we have collected a sample of UA students height & weight data $(x_1(1), x_1(2)), \dots, (x_n(1), x_n(2))$

  height weight

- Model it using a 2-d Gaussian distribution with *unknown mean & variance*
  - Train the model using maximum-likelihood
  - What does the log-likelihood function look like?

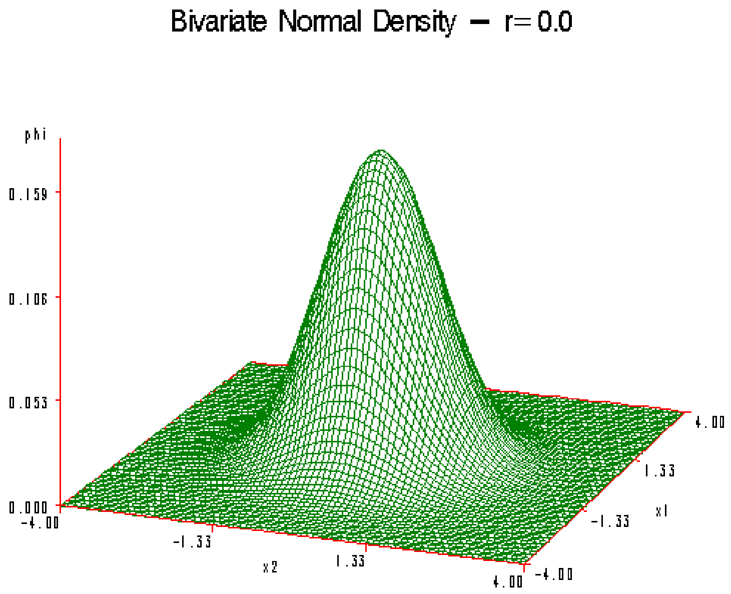


Probability review: multivariate Gaussian

Multivariate Gaussian For RV $X \in \mathbb{R}^d$ with mean μ and positive semidefinite covariance matrix Σ, its probability density function (PDF) is ,

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

$|A|$: matrix determinant of A



Interpretation

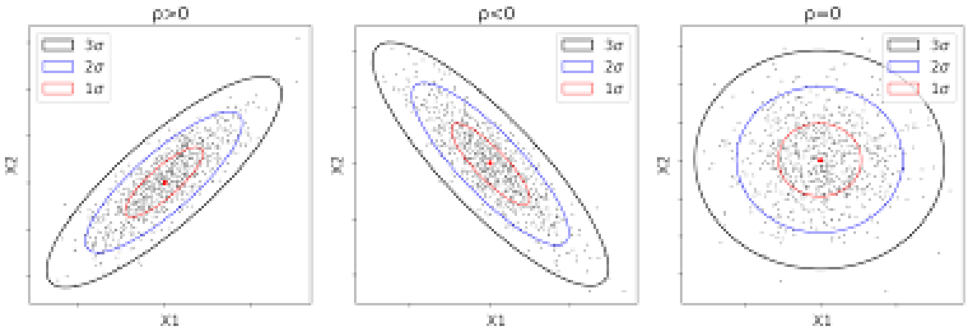
μ: peak location of the PDF (mode)

Σ: the covariance matrix; specifically when $d = 2$:

$$\Sigma = \begin{pmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{YX} & \sigma_Y^2 \end{pmatrix}$$

-diagonal entries: variance of each coordinate

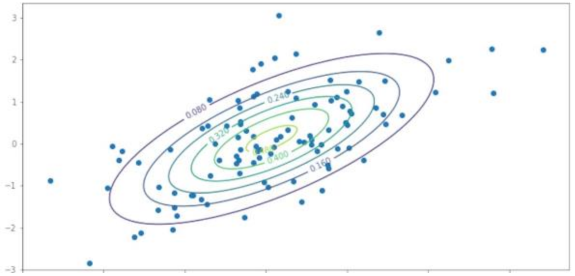
-off diagonal entries: correlation b/w coordinates



Warm-up Example: estimate population height & weight

- MLE: solve $\max_{\mu,\Sigma} \sum_i \ln P(x_i; \mu, \Sigma)$, where

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



Sample mean

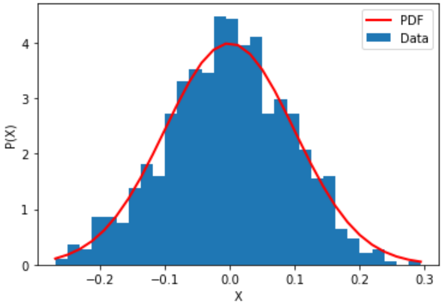
- Observation 1: for any fixed Σ, the optimal μ is $\mu = \frac{1}{n}\sum_i x_i$ (Exercise)
- Observation 2: for any fixed μ, the optimal Σ is such that $\Lambda = \Sigma^{-1}$ equals

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

  - Fact: f is concave in Λ

Sample covariance matrix

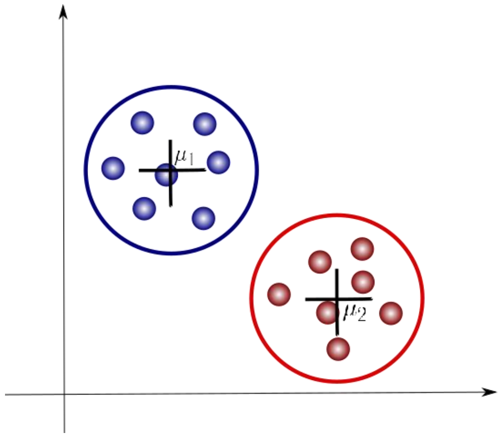
  - $\nabla f(\Lambda) = 0 \Rightarrow n\Lambda^{-1} - \sum_i (x_i - \mu)(x_i - \mu)^\top = 0 \Rightarrow \Sigma = \frac{1}{n}\sum_i (x_i - \mu)(x_i - \mu)^\top$

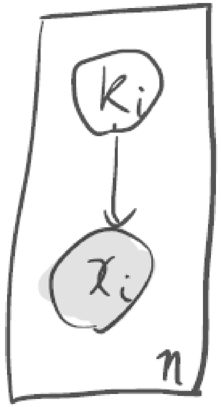


- Quick Q1: can you simplify the expressions when $d = 1$?
- Quick Q2: what if the data is importance-weighted?

Probabilistic clustering: Gaussian mixture model (GMM)

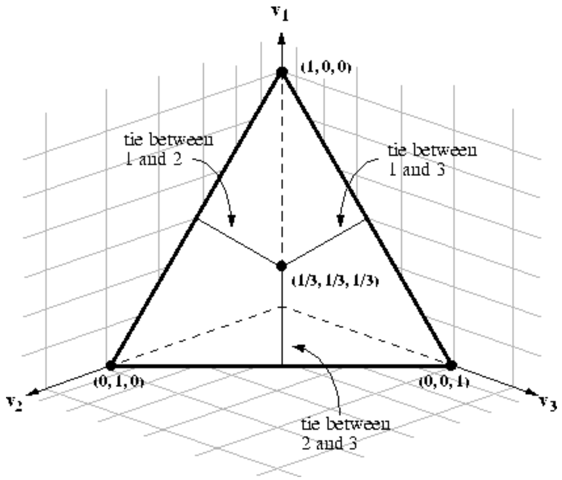
- Data: $S = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$
- Given: K - the number of clusters.
- Generative story:
  - $k \sim \text{Categorical}(\pi)$ (*hidden – latent variable)*
  - $x \mid k \sim N(\mu_k, \Sigma_k)$

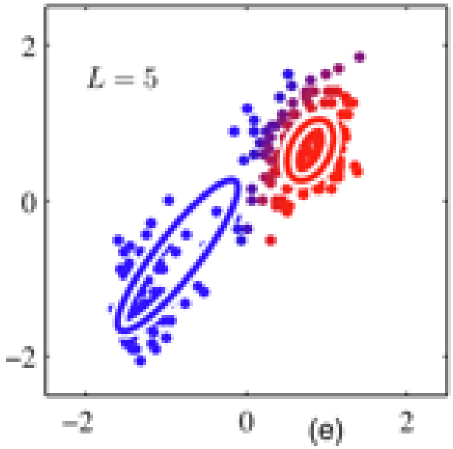




Parameters to learn:

- Cluster weight $\pi = (\pi_1, \dots, \pi_K) \in \Delta^{K-1}$
- Cluster location $\mu = (\mu_1, \dots, \mu_K)$
- Cluster shape (covariance matrix) $\Sigma = (\Sigma_1, \dots, \Sigma_K)$





Marginal Likelihood

More often, we have a joint distribution with observations x, latent variables k, and parameters θ

$$p(k, x \mid \theta) = p(k \mid \theta) p(x \mid k, \theta)$$

Need to *marginalize* out latent variables, hence the name *marginal likelihood:*

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

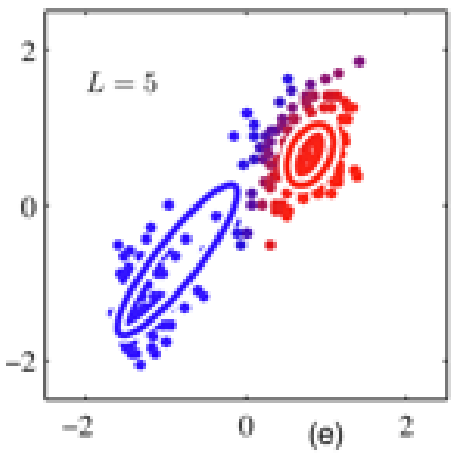
In GMM: $\theta = (\pi, \mu, \Sigma)$

- Observation x, latent variable k
- $p(k \mid \theta) = \pi_k,\ p(x \mid k, \theta) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp\left(-\frac{1}{2}(x - \mu_k)^\top \Sigma_k^{-1} (x - \mu_k)\right) =: N(x; \mu_k, \Sigma_k)$
- $p(x \mid \theta) = \sum_{k=1}^{K} \pi_k\, N(x; \mu_k, \Sigma_k)$

Maximum likelihood estimation for GMM

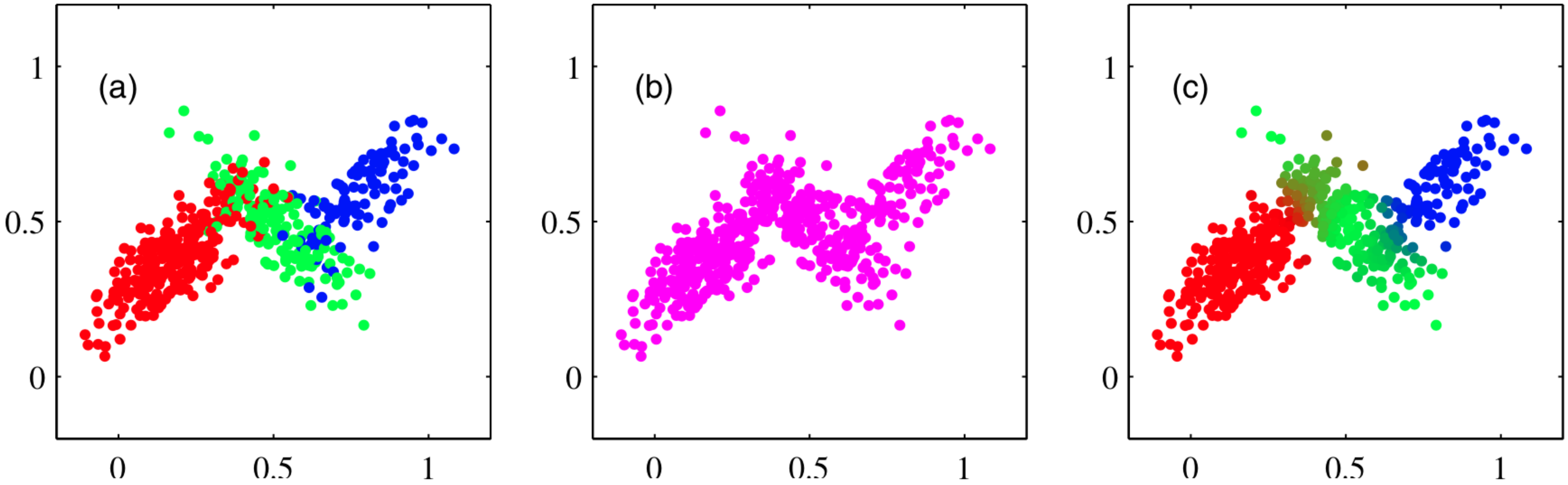
- Maximum likelihood estimation:

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



- How to solve it?
- How do we get the cluster assignments?

Illustration



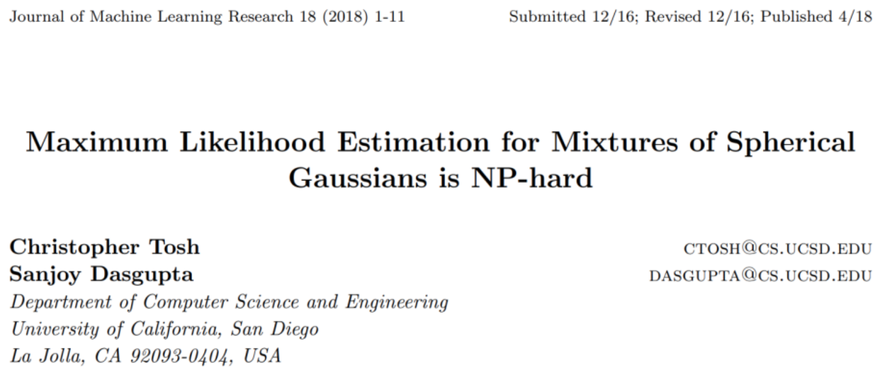
- Mixture of 3 Gaussians
- (a) is ground truth (we don’t know this -- the k_i (color) for each example x_i are hidden).
- (b) is what we see, (c) is what the algorithm can recover.

GMM for clustering: algorithms

- Maximum likelihood estimation

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

  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

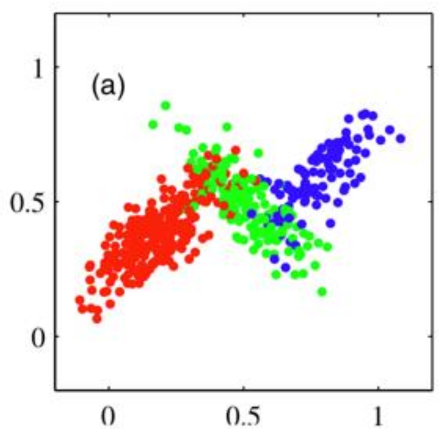
Christopher Tosh CTOSH@CS.UCSD.EDU
Sanjoy Dasgupta DASGUPTA@CS.UCSD.EDU
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0404, USA

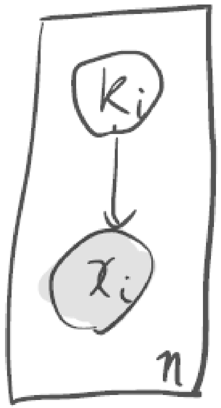
- How to design computationally efficient algorithms that can approximately maximize the log-likelihood function?

- Observation: if for each data point i, we not only have x_i *but also have k_i, (supervised learning setting)*

  then MLE is easy to obtain

- Let's see why & why this is useful..





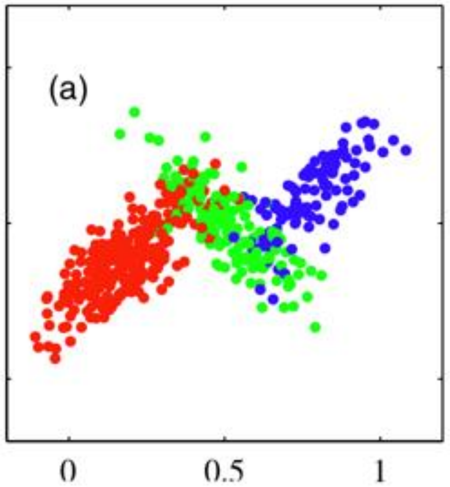
Warmup: MLE for GMM with known cluster membership

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

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

Only related to π

Only related to μ, Σ



$\max\limits_{\pi} \sum_i \log P(k_i;\pi) = \sum_{k=1}^{K} n_k \ln \pi_k$, where $n_k = \#\{i\colon\ k_i = k\}$

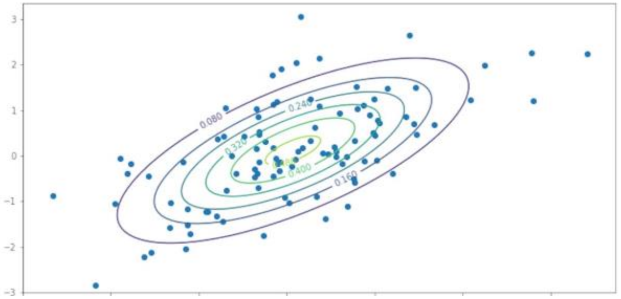
$=> \pi_k = \dfrac{n_k}{n}$

Only related to μ_k, Σ_k

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

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

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



- Equivalent to Gaussian MLE problem

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

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

- From slide 5, we know its solution is:

$\mu_k =$ [sample mean for examples from class k] $= \frac{1}{n_k} \sum_{i:k_i=k} x_i$

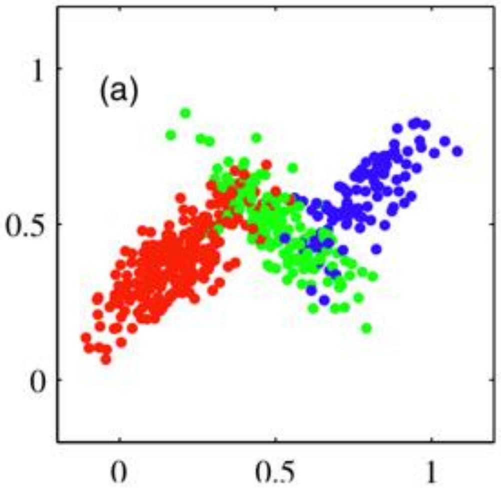
$\Sigma_k =$ [sample covariance matrix for examples from class k] $= \frac{1}{n_k} \sum_{i:k_i=k} (x_i - \mu_k)(x_i - \mu_k)^\top$

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

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

- In summary, the MLE for GMM with known-cluster membership data (x_i, k_i)'s is given by:
- For every k:

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

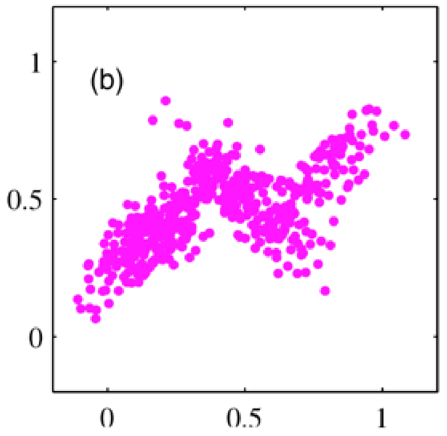


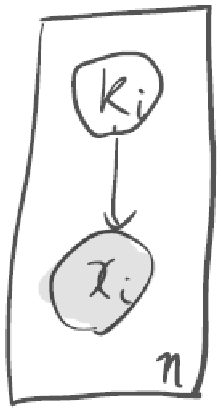
- What if the dataset is importance weighted: $((x_i, k_i), w_i), i = 1, \dots, n$?
- The weighted MLE solution is: for every k:

$$\mu_k = \frac{1}{W_k} \sum_{i:k_i=k} w_i \, x_i$$
$$\Sigma_k = \frac{1}{W_k} \sum_{i:k_i=k} w_i (x_i - \mu_k)(x_i - \mu_k)^\top$$
$$\pi_k = \frac{W_k}{W}$$

Here, $W_k = \sum_{i:k_i=k} w_i \, , W = \sum_i w_i$

GMM for clustering: algorithms





- Coming back to the original question..
- What if the cluster memberships are unknown?
- $\underset{\pi,\mu,\Sigma}{\operatorname{argmax}} \sum_i \log(\sum_{k=1}^{K} \pi_k \, N(x_i; \mu_k, \Sigma_k))$

- 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: the 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

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

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

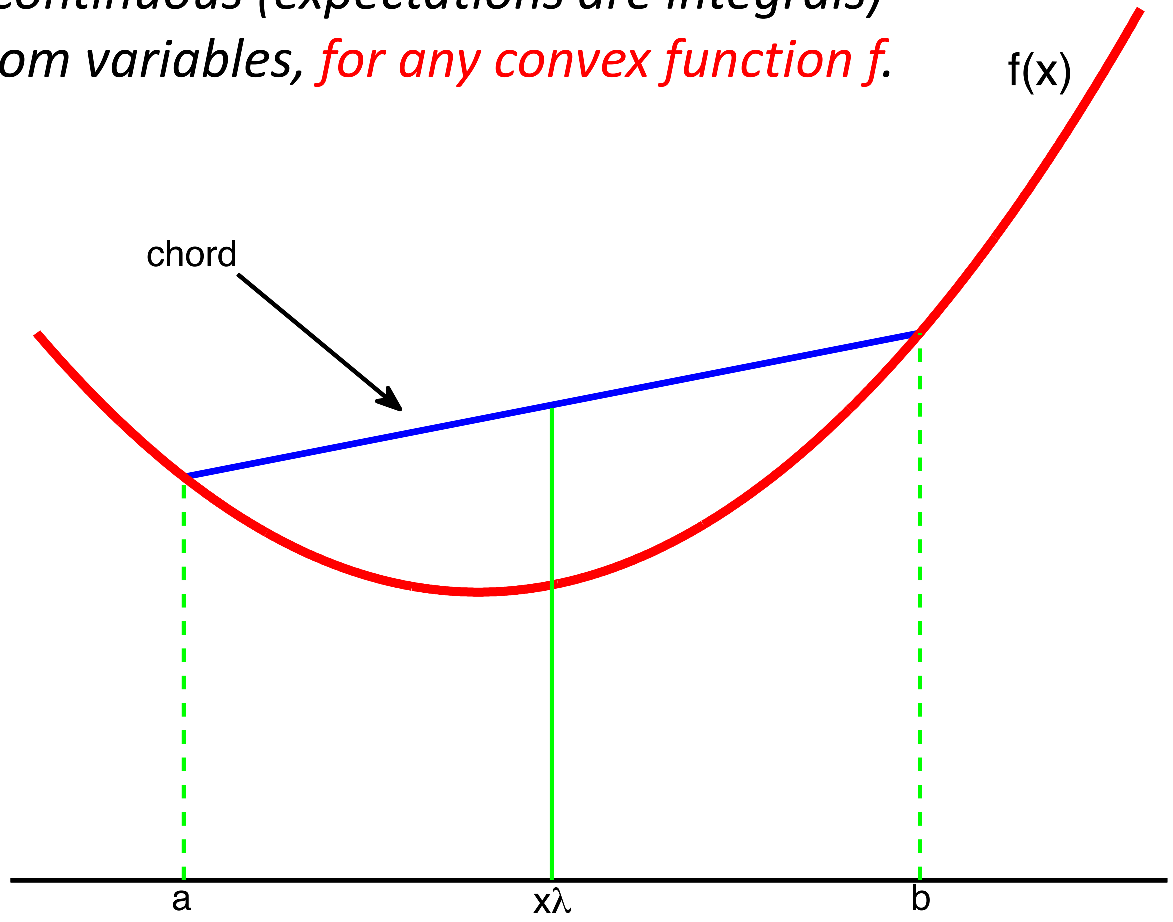
- With n iid examples,

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

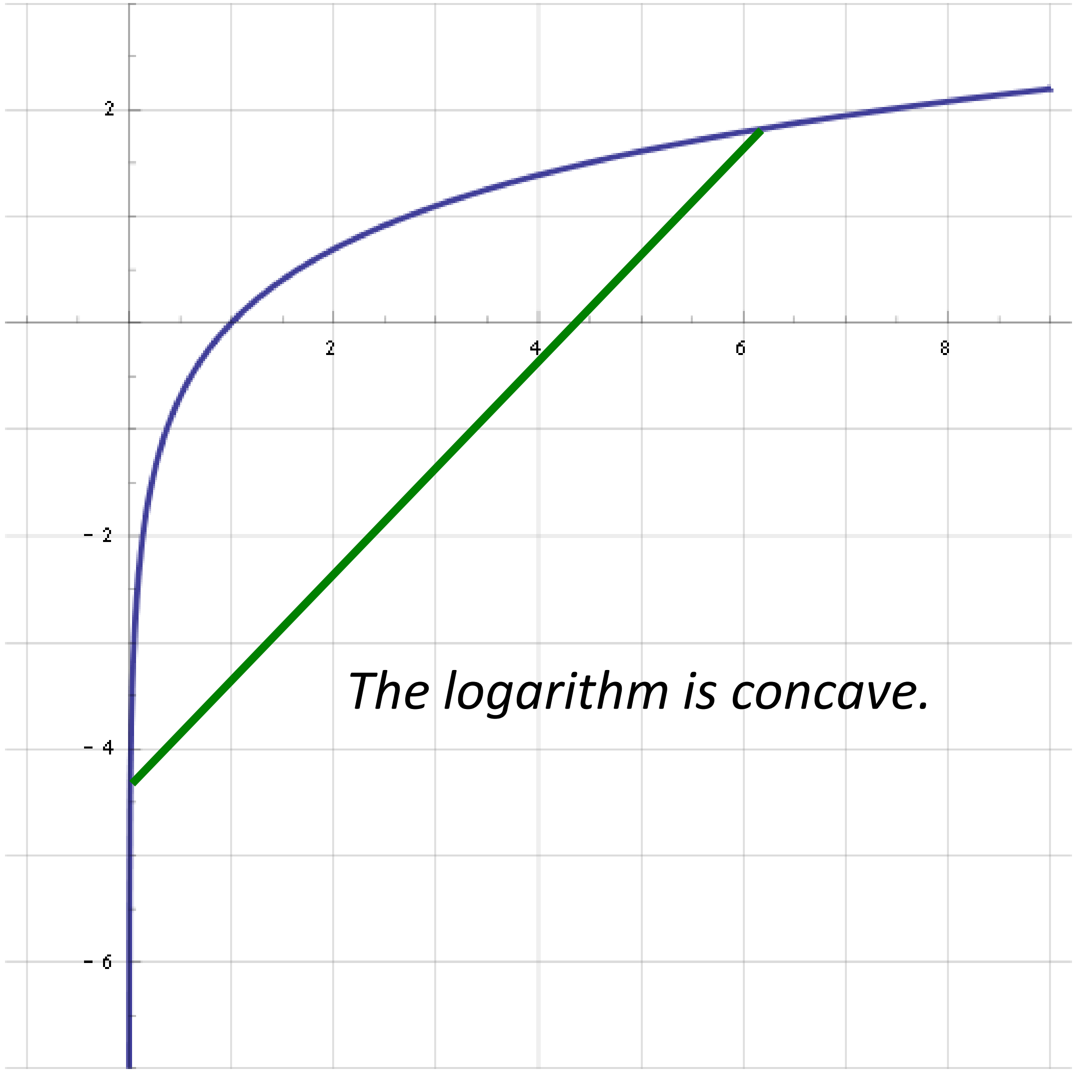
Jensen's Inequality

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

Valid for both discrete (expectations are sums) and continuous (expectations are integrals) random variables, for any convex function f.

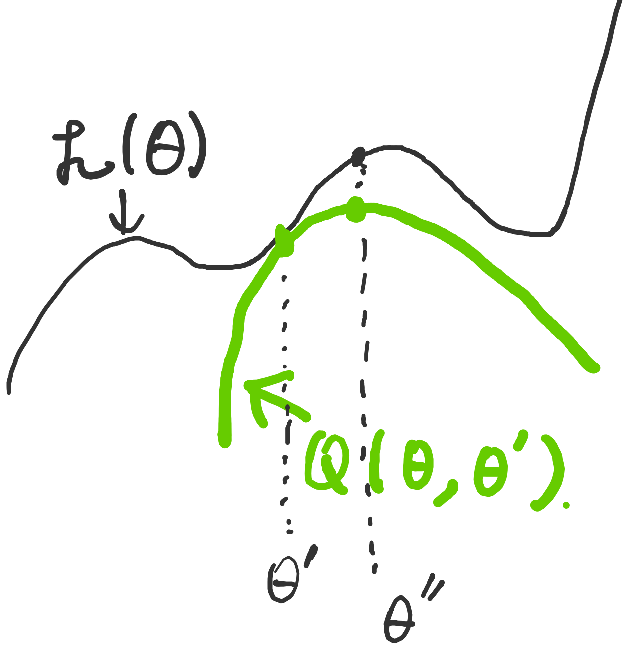


$$\ln(\mathbb{E}[X]) \geq \mathbb{E}[\ln(X)]$$



The logarithm is concave.

EM algorithm: the idea



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

- Why optimizing $Q(\theta;\theta')$?
  - $Q(\theta;\theta') = \sum_{i=1}^{n} \sum_z P(z \mid x_i;\theta') \ln P(x_i,z;\theta) + \boxed{g(\theta')}$

    Irrelevant to θ
  - Maximizing $Q(\theta;\theta')$ $\Leftrightarrow$ maximizing the log-likelihood of model θ on an *importance-weighted* set of *fully-observed* data
  - Example:

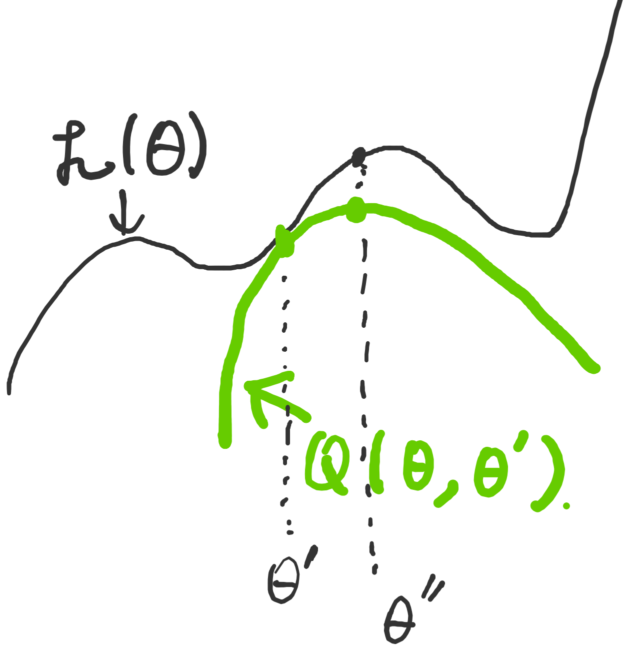
	Value	$P(z = \mathbf{1} \mid x_i;\theta')$	$P(z = \mathbf{2} \mid x_i;\theta')$
x_1	(4.2, -7.1)	0.2	0.8
x_2	(0.05, -1.2)	0.98	0.02

⟹

(x,z) **value**	**weight**
(4.2, -7.1), 1	0.2
(4.2, -7.1), 2	0.8
(0.05, -1.2), 1	0.98
(0.05, -1.2), 2	0.02

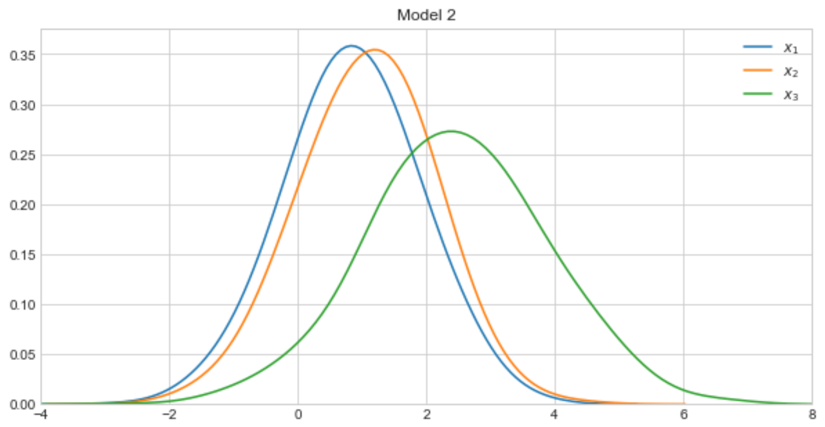
EM algorithm: the idea

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



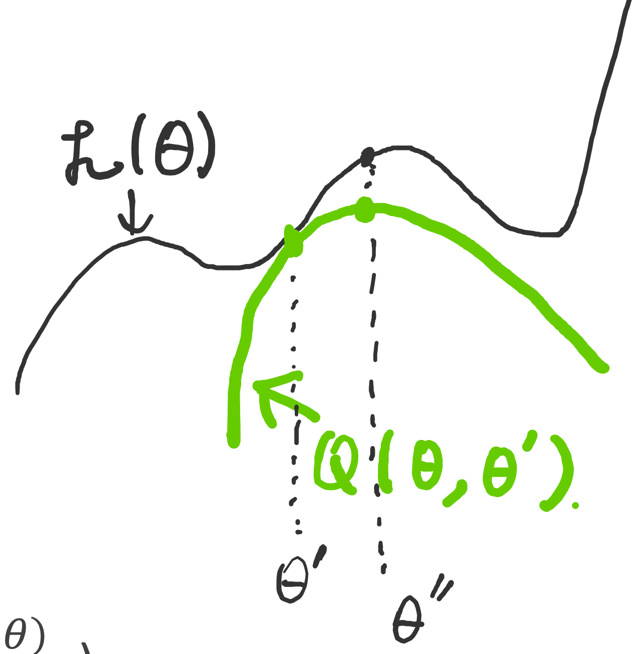
- The lower bound approximate $Q(\theta;\theta')$ is sometimes tight
  - At $\theta = \theta'$, $Q(\theta';\theta') = \mathcal{L}(\theta')$
  - For general θ, $\mathcal{L}(\theta) - Q(\theta;\theta') = \sum_{i=1}^{n} \mathrm{KL}\big(P(\, z \mid x_i;\theta'\,), P(\, z \mid x_i;\theta\,)\big) \geq 0$

- Kullback-Leibler (KL) divergence: $\mathrm{KL}(p,q) = \mathrm{E}_{z\sim p}\left[\ln\frac{p(z)}{q(z)}\right]$
- Measures difference between distributions
- Properties:
  - $\mathrm{KL}(p||q) \geq 0$, for all p, q;
  - $\mathrm{KL}(q||q) = 0$, for all q



https://datascience.oneoffcoder.com/kullback-leibler-divergence.html

EM algorithm: the procedure



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

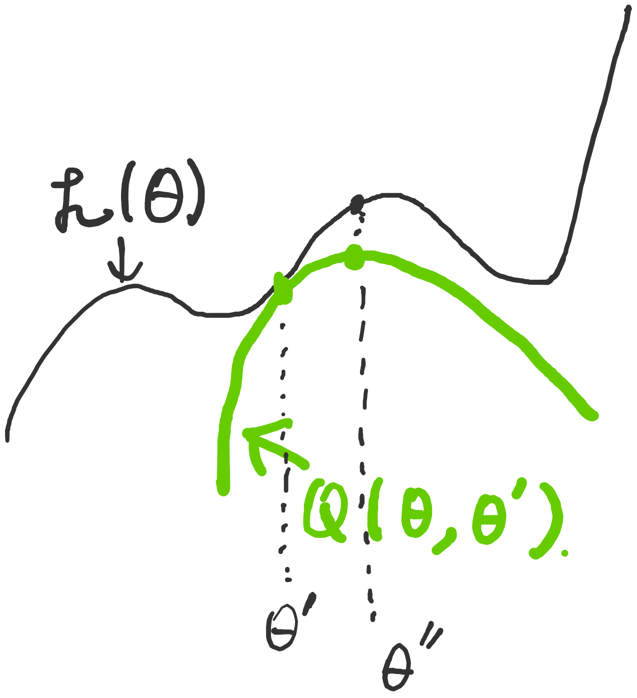
     (This is for calculating $Q\left(\theta; \theta^{(n)}\right) = \sum_{i=1}^{n} \sum_z P\left(z \mid x_i; \theta^{(n)} \right) \ln \frac{P(x_i, z; \theta)}{P(z|x_i; \theta^{(n)})}$)

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

     (Performing MLE over an importance-weighted dataset of fully observed data)

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

EM algorithm: convergence guarantee

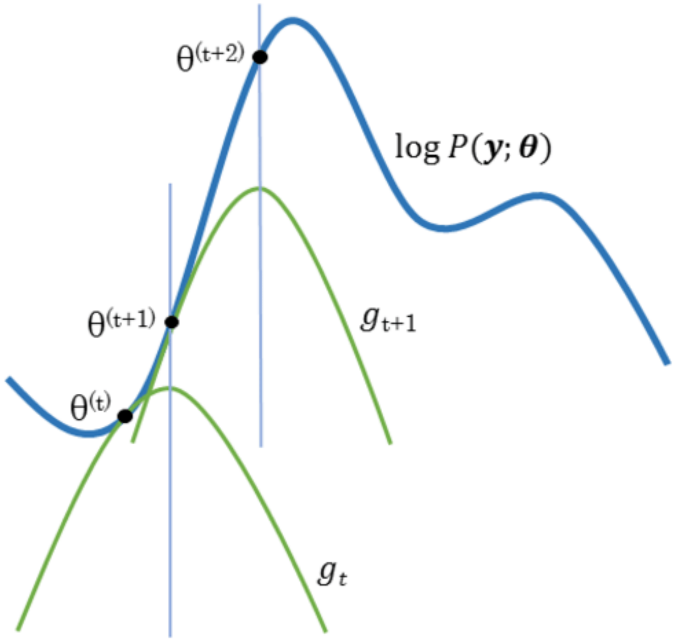


- Monotone improvement of likelihood function
- Illustration:

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

- Therefore,

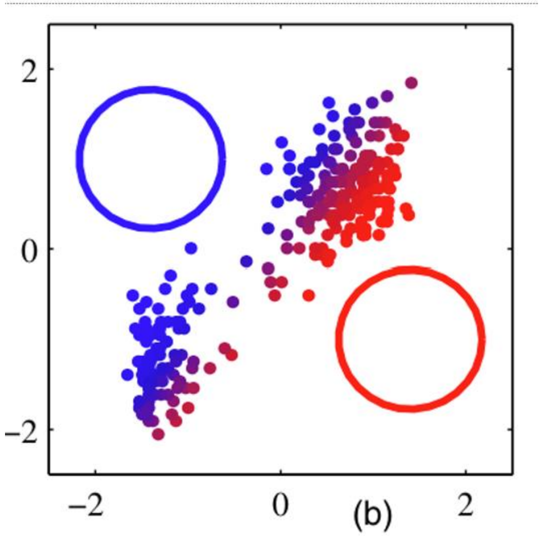
$$\begin{aligned}\mathcal{L}(\theta^{(n)}) &= Q(\theta^{(n)}, \theta^{(n)}) \\ &\leq Q(\theta^{(n+1)}, \theta^{(n)}) \\ &\leq \mathcal{L}(\theta^{(n+1)}) \\ &\leq \mathcal{L}(\theta^{(n+2)}) \\ &\leq \cdots\end{aligned}$$



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 of example i (e.g. cluster 1 = blue, γ_{i1} larger => bluer) based on current belief

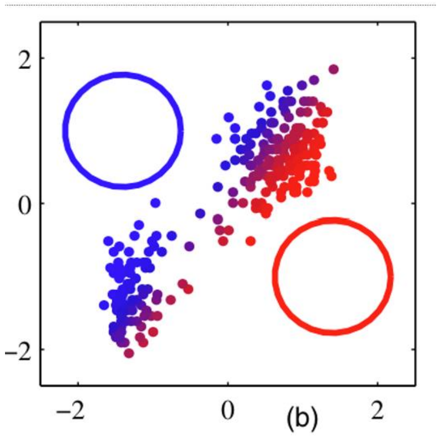
EM algorithm: application to GMMs (cont'd)

- The M-step:

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

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



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

EM algorithm: application to GMMs (cont'd)

- How to solve

$$\max_{\theta=(\pi,\mu,\Sigma)} \sum_{i=1}^{n} \sum_{k} \gamma_{ik} \ln P(x_i, k_i = k; \theta)?$$

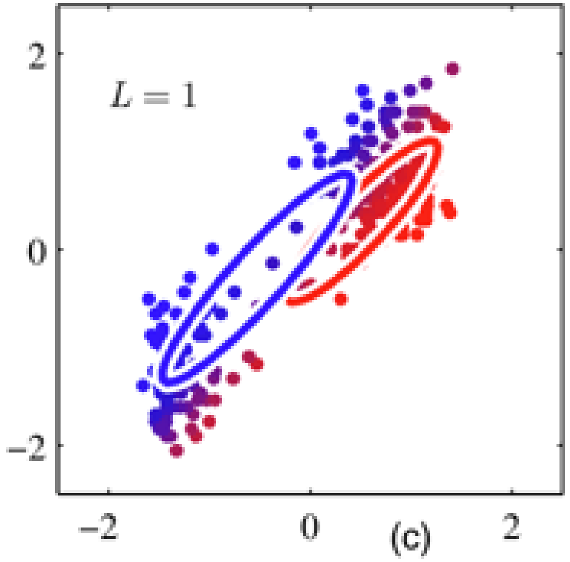
- This is MLE with fully-observed data with nK importance-weighted examples $\{(x_i, k), \gamma_{ik}\}_{i\in[n],k\in[K]}$

- We have seen its solution before:

$$\pi_k = \frac{\Gamma_k}{\Gamma}$$

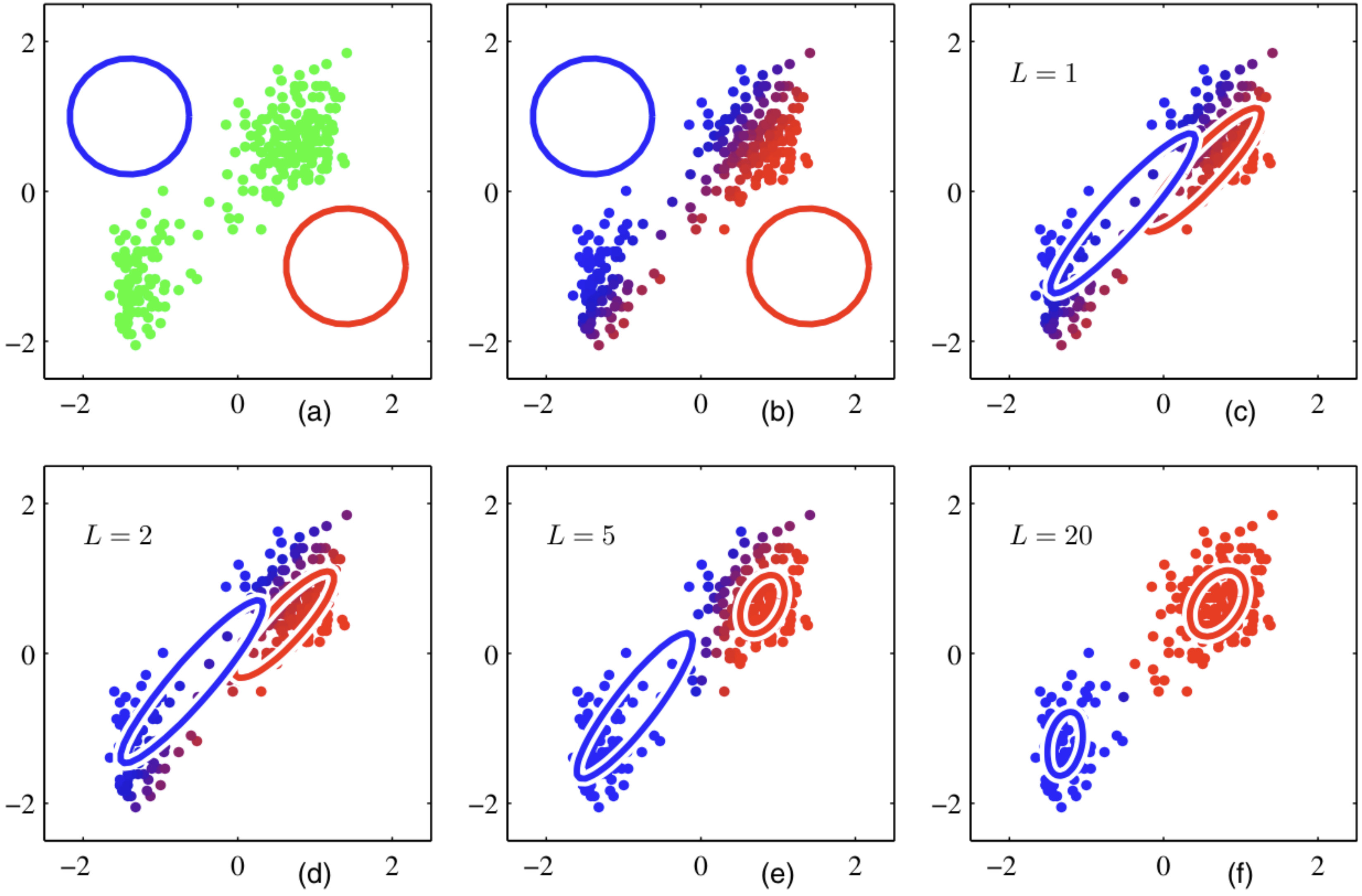
$$\mu_k = \frac{\sum_i \gamma_{ik}\, x_i}{\Gamma_k}$$

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



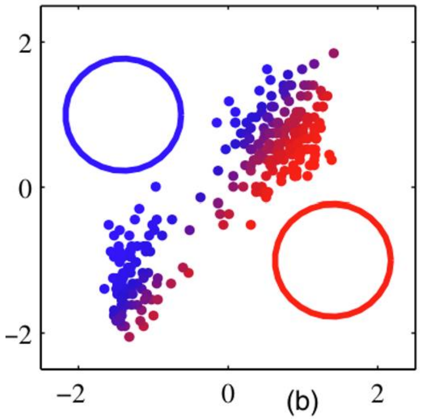
- Here $\Gamma_k = \sum_{i=1}^{n} \gamma_{ik}$, $\Gamma = \sum_{i,k} \gamma_{ik} = n$

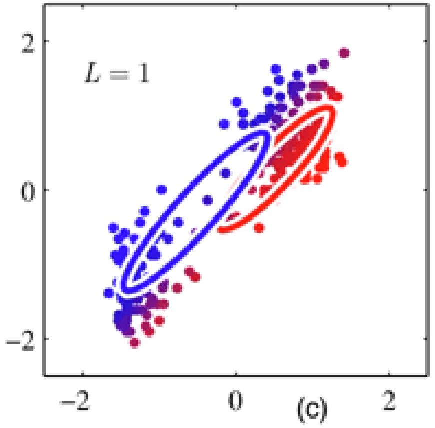
EM in action



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:
  - $\gamma_{ik} = \frac{\pi_k \, N(x_i; \mu_k, \Sigma_k)}{\sum_{k'=1}^K \pi_{k'} \, N(x_i; \mu_{k'}, \Sigma_{k'})}$ responsibility
  - Let $\Gamma_k = \sum_{i=1}^n \gamma_{ik}$ soft counts
- (M)aximization step: for every k:
  - $\mu_k' = \frac{1}{\Gamma_k} \sum_{i=1}^n \gamma_{ik} x_i$
  - $\Sigma_k' = \frac{1}{\Gamma_k} \sum_{i=1}^n \gamma_{ik} (x_i - \mu_k')(x_i - \mu_k')^\top$ note we use μ_k' rather than μ_k
  - $\pi_k' = \frac{\Gamma_k}{n}$
  - 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.





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. $\pi \leftarrow \left(\frac{1}{K}, \dots, \frac{1}{K}\right)$, μ ←cluster centers of Lloyd's algorithm, $\Sigma = \mathrm{I}$

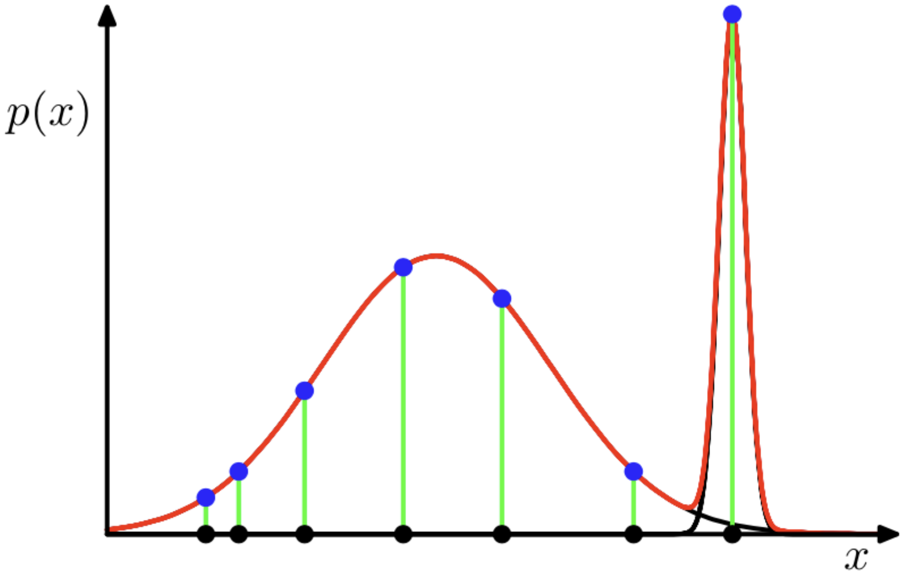
- Beware of pitfalls

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:

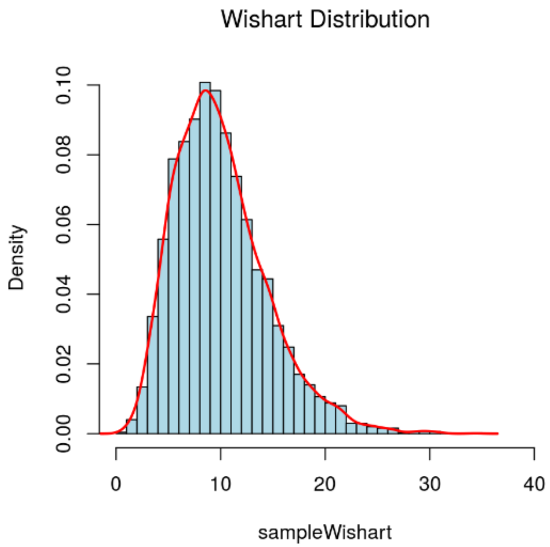
  for one example i, $\ln P(x_i;\theta)$ is arbitrarily large



- Imagine Gaussian MLE on one data point:

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

- To address this:
  - Regularization: penalize overly small Σ_k
  - Detect overly small Σ_k and restart EM
  - Bayesian treatment: impose a prior on Σ_k's



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,

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

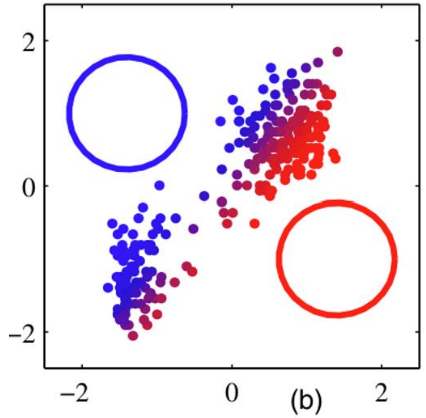
  (fix Σ_k, π throughout -- do not update them)

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

- Running the EM algorithm:
- E-step:
  - $p(x \mid \mu_k, \Sigma_k) \propto \exp\left(-\frac{1}{2\epsilon}\|x-\mu_k\|_2^2\right)$
  - $\gamma_{ik} = \dfrac{\pi_k \exp\left(-\frac{\|x_i-\mu_k\|^2}{2\epsilon}\right)}{\sum_{k'=1}^{K}\pi_{k'}\exp\left(-\frac{\|x_i-\mu_{k'}\|^2}{2\epsilon}\right)}$

When $\epsilon \to 0$:
$\gamma_{ik} = 1$ if μ_k is the cluster center closest to x_i; 0 otherwise

- Imagine $K = 2$



Lloyd’s algorithm is EM in the limit

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

Imagine $\pi = \text{Uniform}, \Sigma_k = \epsilon I$ with a very small ϵ

- (E)xpectation step:
  - $\gamma_{ik} =$ ~~$\frac{\pi_k\, p(x_i \mid z_i=k)}{\sum_{k'=1}^K \pi_{k'}\, p(x_i \mid z_i=k')}$~~

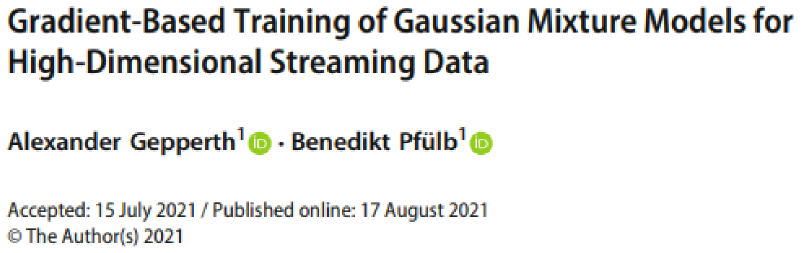
    $\gamma_{ik} = 1$ if μ_k is the cluster center closest to x_i; 0 otherwise
  - Let $n_k = \sum_{i=1}^n \gamma_{ik}$

    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 can maximizes likelihood in GMMs
  - E.g. can just do gradient ascent on the likelihood function

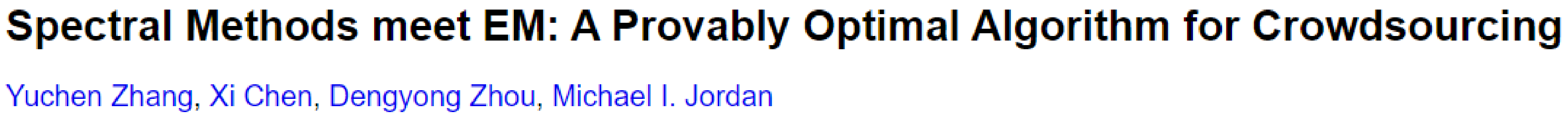


Gradient-Based Training of Gaussian Mixture Models for High-Dimensional Streaming Data

Alexander Gepperth[1] · Benedikt Pfülb[1]

Accepted: 15 July 2021 / Published online: 17 August 2021
© The Author(s) 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

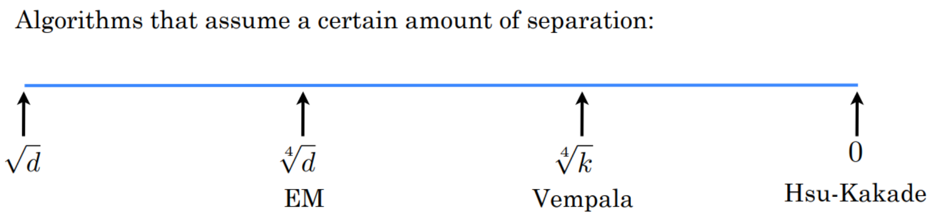


Spectral Methods meet EM: A Provably Optimal Algorithm for Crowdsourcing

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 $Q\left(\theta, \theta^{(n)}\right)$ -- log-likelihood on an importance-weighted, fully-observed dataset

- 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

Backup

Marginal Likelihood

More often, we have a joint distribution with observations x, unknown variables k, and parameters θ

$$p(k, x \mid \theta) = p(k \mid \theta) p(x \mid k, \theta)$$

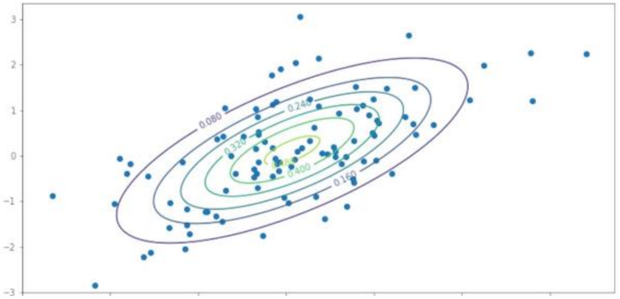
Need to *marginalize* out latent variables, hence the name *marginal likelihood:*

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

In the GMM:

- $\theta = (\pi, \mu, \Sigma)$
- $p(k \mid \theta) = \pi_k$
-

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



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

- Conceptually the same as the Gaussian MLE problem $\max\limits_{\mu,\Sigma} \sum_i \ln N(x_i; \mu, \Sigma)$, where

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

- From slide 5, we know its solution is

μ_k = sample mean for examples from class $k = \frac{1}{n_k} \sum_{i:k_i=k} x_i$

Σ_k = sample mean for examples from class $k = \frac{1}{n_k} \sum_{i:k_i=k} (x_i - \mu_k)(x_i - \mu_k)^\top$

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

EM algorithm: application to GMMs (cont'd)

- How to compute

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

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

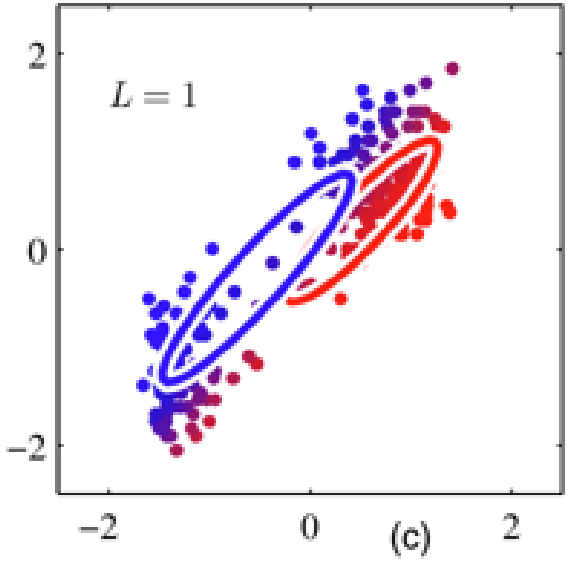
$$\mu_k = \frac{1}{n_k} \sum_{i:k_i=k} x_i$$

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

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

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

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

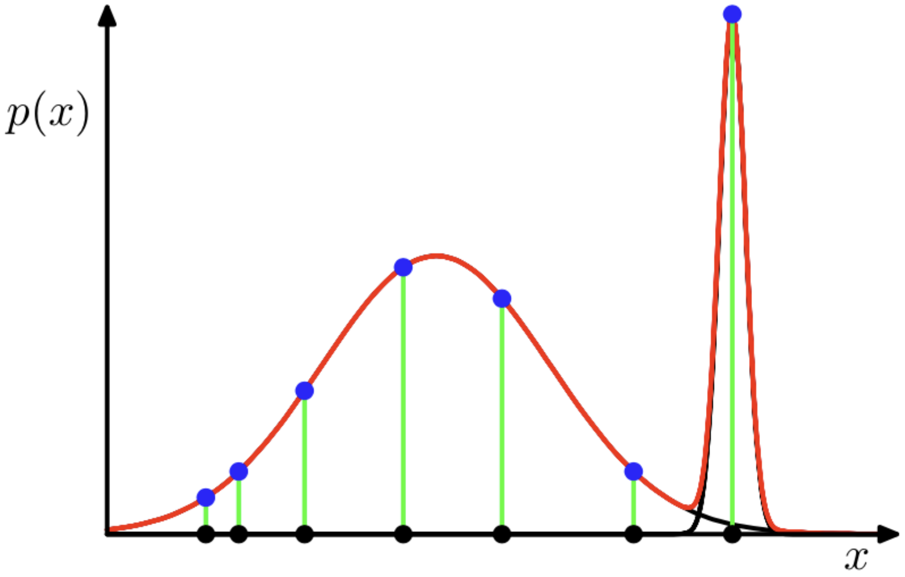


Pitfalls

- 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:

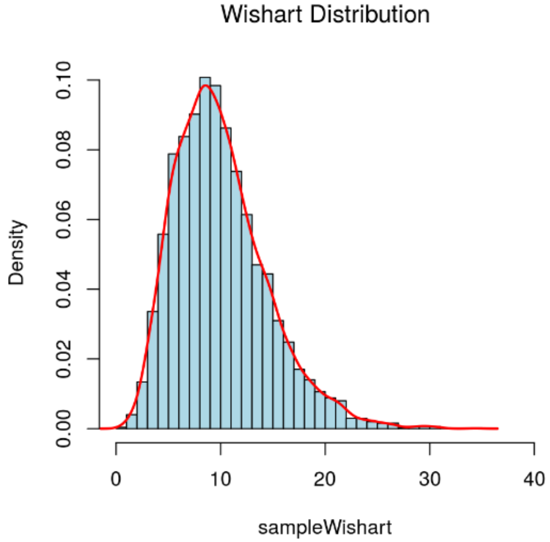
  for one example i, $\ln P(x_i; \theta)$ is arbitrarily large



- 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:
  - Regularization: penalize overly small Σ_k
  - Detect overly small Σ_k and restart EM



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