Expectation Maximization

COS 424/524, SML 302: Fundamentals of Machine Learning

Professor Engelhardt

COS 424/524, SML 302

Lecture 14

The Expectation Maximization algorithm

In lecture 12, we learned about Gaussian mixture models.

We discussed, intuitively, how to fit these models to data using a probabilistic version of the K-means algorithm.

Today, we will discuss this algorithm, the Expectation-Maximization (EM) algorithm, more formally in the context of latent variable models.

The Expectation Maximization algorithm

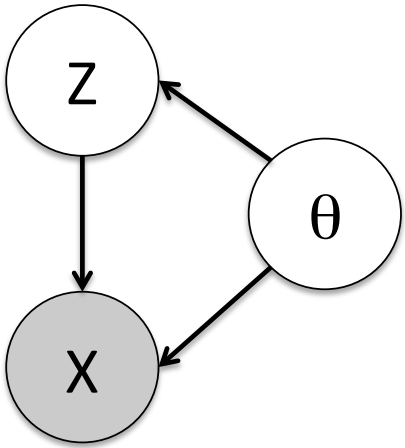
EM maximizes the likelihood of the observed data. For a generic model:

- The *latent variables* are z.
- The *observations* are x.
- The *parameters* are θ.
- The *joint distribution* is

$$p(z, x \mid \theta) = p(z \mid \theta)p(x \mid z, \theta).$$

- The observed data *likelihood* is

$$p(x \mid \theta) = \sum_{z \in \mathcal{Z}} p(z \mid \theta)p(x \mid z, \theta).$$



The Expectation Maximization algorithm

For a generic model:

- The *log likelihood* is

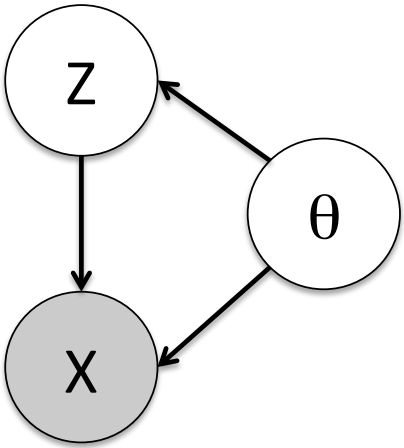
$$p(x \mid \theta) = \log \sum_{z \in \mathcal{Z}} p(z \mid \theta) p(x \mid z, \theta).$$

- The *complete log likelihood* is

$$\ell_c(\theta; x, z) = \log p(z, x \mid \theta)$$

- The *expected complete log likelihood* is

$$\mathrm{E}_\theta[\log p(Z, x \mid \theta)].$$

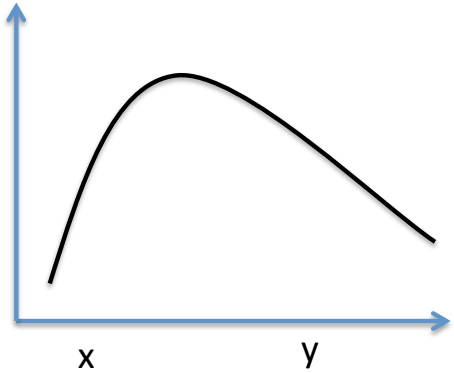


Jensen's inequality

Jensen's inequality:

For a concave function and $\lambda \in (0, 1)$,

$$f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y)$$

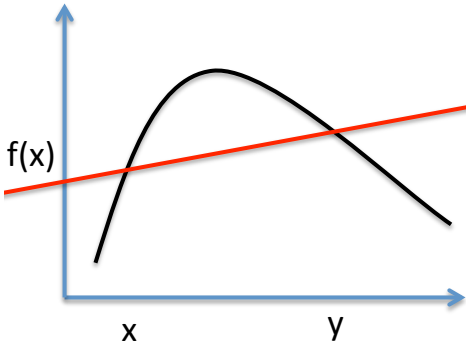


Jensen's inequality

Jensen's inequality:

For a concave function and $\lambda \in (0, 1)$,

$$f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y)$$

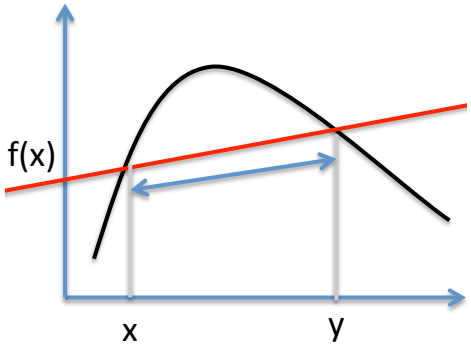


Jensen's inequality

Jensen's inequality:

For a concave function and $\lambda \in (0, 1)$,

$$f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y)$$



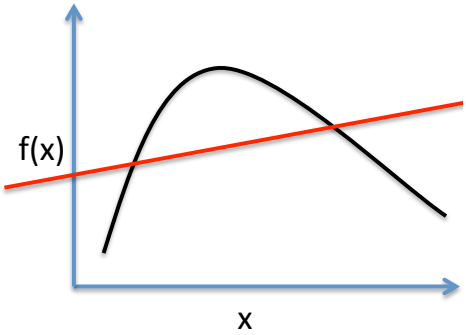
Jensen's inequality and expected values

Jensen's inequality generalizes to probability distributions

For concave function $f(\cdot)$,

$$f(\mathrm{E}[X]) \geq \mathrm{E}[f(X)].$$

.



Why?

Lower bound on the log likelihood

Use Jensen's inequality to derive lower bound on log likelihood, $\mathcal{L}(\theta, q)$:

$$\begin{aligned}
\ell(\theta; x) &= \log p(x \mid \theta) \\
&= \log \sum_{z \in \mathcal{Z}} p(x, z \mid \theta) \\
&= \log \sum_{z \in \mathcal{Z}} q(z \mid x) \frac{p(x, z \mid \theta)}{q(z \mid x)} \\
&\geq \sum_{z \in \mathcal{Z}} q(z \mid x) \log \frac{p(x, z \mid \theta)}{q(z \mid x)} \\
&= \mathcal{L}(q, \theta).
\end{aligned}$$

This *lower bound* on the log likelihood depends on a distribution over the latent variables $q(z \mid x)$ and on the parameters θ.

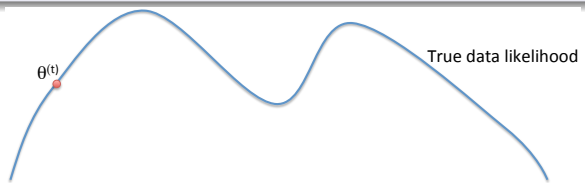
EM as coordinate ascent

EM is a *coordinate ascent algorithm* with distribution q and parameters θ.

We optimize the log likelihood lower bound $\mathcal{L}$ with respect to each argument, holding the other argument fixed.

EM algorithm

$$
\begin{aligned}
q^{(t+1)}(z \mid x) &= \arg\max_{q} \quad \mathcal{L}(q, \theta^{(t)}) \\
\theta^{(t+1)} &= \arg\max_{\theta} \quad \mathcal{L}(q^{(t+1)}, \theta)
\end{aligned}
$$



E-step of EM

E-step

$$q^{(t+1)}(z \mid x) = \arg\max_{q} \quad \mathcal{L}(q, \theta^{(t)})$$

In the E-step, we find $q^*(z \mid x)$, which, to maximize the data log likelihood lower bound, we set equal to the posterior $p(z \mid x, \theta)$,

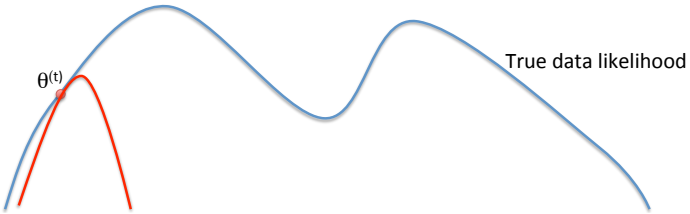
$$\begin{aligned}
\mathcal{L}(q^*(z \mid x), \theta) &= \sum_{z \in \mathcal{Z}} q(z \mid x) \log \frac{p(x, z \mid \theta)}{q(z \mid x)} \\
&= \sum_{z} p(z \mid x, \theta) \log \frac{p(x, z \mid \theta)}{p(z \mid x, \theta)} \\
&= \sum_{z} p(z \mid x, \theta) \log p(x \mid \theta) \\
&= \log p(x \mid \theta) \\
&= \ell(\theta; x)
\end{aligned}$$

E-step of EM

E-step

$$q^{(t+1)}(z \mid x) = \arg\max_{q} \quad \mathcal{L}(q, \theta^{(t)})$$

In words: we find the $q^*(z \mid x)$ that minimizes the distance from the lower bound $\mathcal{L}$ to the true data likelihood at point $\theta^{(t)}$.



M-step of EM

M-step

$$\theta^{(t+1)} = \arg\max_{\theta} \quad \mathcal{L}(q^{(t+1)}, \theta)$$

In the M-step, we consider the expected complete log likelihood only:

$$\mathcal{L}(q, \theta) \quad = \quad \sum_z q(z \mid x) \log p(x, z \mid \theta) - \sum_z q(z \mid x) \log q(z \mid x)$$

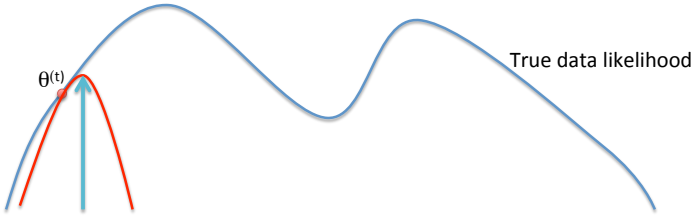
The second term captures our uncertainty in z, and is constant with respect to θ. The first term is the *expected complete log likelihood*.

$$\mathrm{E}_\theta[\log p(Z, x \mid \theta)] \quad = \quad \sum_z p(z \mid x, \theta) \log p(x, z \mid \theta)$$

Aside: what is another interpretation of this representation of $\mathcal{L}(q, \theta)$?

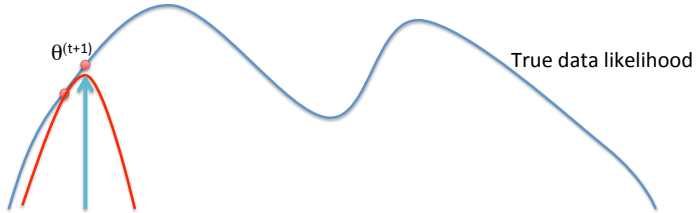
M-step of EM

In words, we optimize the expected complete log likelihood with respect to parameters θ to get the new value of $\theta^{(t+1)}$:



This is often a simple MLE or MAP estimate of the parameters. But, you can use whatever inference tools you have at your disposal to estimate the parameters.

M-step of EM

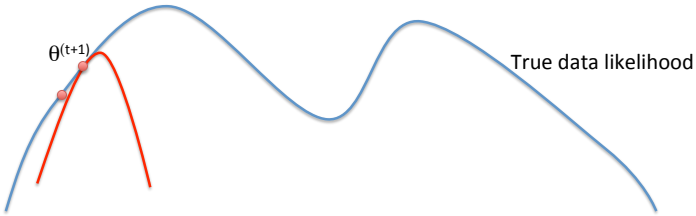


The expected complete log likelihood

- decomposes as if it were the fully observed log likelihood
- has posterior expectations replacing the latent variables.

Iterate again: E-step of EM

E-step again finds the value of $q^*(z \mid x)$ that minimizes the loss, or the distance from lower bound $\mathcal{L}$ to data log likelihood, which is $p(z|x, \theta^{(t+1)})$



Note that $\mathcal{L}$ and the log likelihood meet at $\theta^{(t+1)}$, where they share the same gradient.

Does EM converge?

Details and assumptions of EM

- EM iterates between optimizing lower bound, tightening lower bound
- Coordinate ascent in this space finds a *local optimum*
- Convergence of EM is guaranteed because coordinate ascent is monotone increasing with respect to lower bound on log likelihood.
- EM is sensitive to initialization
- Simple test for convergence: After E-step, compute likelihood at $\theta^{(t)}$; stop when likelihood increases less than ϵ.

Big picture: fitting latent variable models

- With EM, we can fit all kinds of latent variable models with maximum likelihood or maximum a posteriori estimates
- EM has impacted statistics, machine learning, signal processing, computer vision, natural language processing, speech recognition, genomics, and other fields that use latent variable models.
- Latent variables are no longer a block. Powerful idea:
  - If I observe everything, then I can (generally) fit the MLE or MAP
  - But I cannot observe everything
  - With EM, fit the MLE with latent variables and reach a local optimum

Examples of EM in practice

Uses of EM

- Used to "fill in" missing pixels in tomography.
- Modeling nonresponses in sample surveys (PCA).
- Inferring ancestral population from a set of genotypes.
- Time series models and latent states.
- Financial models with latent states
- Collaborative filtering (movies, purchases)
- ... and most models used in unsupervised learning.

Other methods for parameter estimation

EM is closely related to

- Variational methods for approximate posterior inference
- Gibbs sampling
- General methods for sampling

For some models, either the E-step or the M-step is difficult to compute.

Key point: EM algorithm is a general approach that allows any method (approximate, exact, or sample-based) to be used for either step.

Recall: Gaussian mixture model

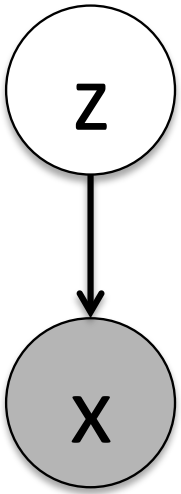
Let $\mathcal{D} = \{x_1, \ldots, x_n\}$ be n samples with K mixture components. Let z_i be multinomial vectors that probabilistically assign a sample to K components.

Density of 1D Gaussian mixture (unit variance)

$$p(x \mid \pi, \mu_{1:K}) = \sum_{z=1}^{K} \pi_z \mathcal{N}(x; \mu_z, 1)$$

Here, $\mathcal{N}(x; \mu_z, 1)$ is the Gaussian density for cluster k with mean μ_z and unit variance

π_z is the mixture proportion for component $z = k$.



EM for Gaussian mixture model

EM for Gaussian mixture model

Repeat until convergence:

1. **E-step**: For each sample i, compute $\hat{z}_i = p(z_i \mid x_i, \pi^{(t)}, \mu_{1:K}^{(t)})$
2. **M-step**: For each cluster k's distribution

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^n \hat{z}_{i,k} x_i}{\sum_{i=1}^n \hat{z}_{i,k}}$$

and for the cluster proportions

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^n \hat{z}_{i,k}}{n}$$

Let's revisit each step in light of our discussion today.

EM for Gaussian mixture model

E-step

For each sample i, compute $\hat{z}_i = p(z_i \mid x_i, \pi^{(t)}, \mu_{1:K}^{(t)})$

The E-step computes the conditional expectation of latent variables z conditioned on x and the current parameter values $\mu^{(t)}$.

For multinomial random variable $z_i \mid x_i, \theta^{(t)}$, this is just:

$$\begin{aligned}
\hat{z}_i^k &= p(z_i = k \mid x_i, \pi_k^{(t)}, \mu_k^{(t)}) \\
&= \frac{p(x_i \mid z_i = k, \mu_k^{(t)}) p(z_i = k \mid \pi_k^{(t)})}{\sum_{\ell=1}^K p(x_i \mid z_i = \ell, \mu_\ell^{(t)}) p(z_i = \ell \mid \pi_\ell^{(t)})} \\
&= \frac{\pi_k^{(t)} \mathcal{N}(x_i, \mu_k^{(t)})}{\sum_{\ell=1}^K \pi_\ell^{(t)} \mathcal{N}(x_i, \mu_\ell^{(t)})}.
\end{aligned}$$

Why (first to second line)?

EM for mixture model

M-step

For each cluster k:

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^n \hat{z}_{i,k} x_i}{\sum_{i=1}^n \hat{z}_{i,k}}$$

The M-step computes the model parameters that optimize the expected complete log likelihood.

For a mixture model, the expected complete log likelihood:

$$\mathrm{E}_\theta[\log p(z, x \mid \theta^{(t)})] = \mathrm{E}\left[\log \prod_{i=1}^n p(x_i, z_i \mid \theta^{(t)})\right].$$

EM for mixture model

For a mixture model, the expected complete log likelihood:

$$
\begin{aligned}
\mathrm{E}_\theta[\log p(z, x \mid \theta^{(t)})] &= \mathrm{E}\left[\sum_{i=1}^{n} \log p(x_i, z_i \mid \theta^{(t)})\right] \\
&= \sum_{i=1}^{n} \mathrm{E}\left[\log\left[\prod_{k=1}^{K} (\pi_k^{(t)} p(x_i \mid \theta_k^{(t)}))^{z_i^k}\right]\right] \\
&= \sum_{i=1}^{n} \mathrm{E}\left[\sum_{k=1}^{K} z_i^k \log(\pi_k^{(t)} p(x_i \mid \theta_k^{(t)}))\right] \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathrm{E}[z_i^k] \log(\pi_k^{(t)}) + \mathrm{E}[z_i^k] \log p(x_i \mid \theta_k^{(t)}) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} \hat{z}_i^k \log(\pi_k^{(t)}) + \hat{z}_i^k \log p(x_i \mid \theta_k^{(t)}).
\end{aligned}
$$

For another model, where might this equation become intractable?

EM for mixture model

M-step

For each cluster k:

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^n \hat{z}_{i,k} x_i}{\sum_{i=1}^n \hat{z}_{i,k}}$$

For a Gaussian mixture model, find the MLE estimates of μ_k with respect to the expected complete log likelihood:

$$\frac{\partial \mathrm{E}_\theta[\log p(z, x \mid \theta^{(t)})]}{\partial \mu_k} = \frac{\partial \sum_{i=1}^n \hat{z}_i^k \log \mathcal{N}(x_i \mid \mu_k^{(t)})}{\partial \mu_k}.$$

Set to zero, solve for μ_k:

$$\mu_k = \frac{\sum_{i=1}^n \hat{z}_i^k x_i}{\sum_{i=1}^n \hat{z}_i^k}.$$

Exercise: show that this is the case.

EM for mixture model

M-step

For each cluster k:

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{i,k}}{n}$$

For a Gaussian mixture model, find the MLE estimates of π_k with respect to the expected complete log likelihood with respect to π_k:

$$\pi_k = \frac{\sum_{i=1}^{n} \hat{z}_i^k}{n}.$$

Exercise: show that this is the case.

EM for Gaussian mixture model

EM for Gaussian mixture model

Repeat until convergence:

1. **E-step**: For each sample i, compute

$$\hat{z}_i = \frac{\pi_k^{(t)} \mathcal{N}(x_i, \mu_k^{(t)})}{\sum_{\ell=1}^{K} \pi_\ell^{(t)} \mathcal{N}(x_i, \mu_\ell^{(t)})}$$

2. **M-step**: For each cluster k's distribution

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{i,k} x_i}{\sum_{i=1}^{n} \hat{z}_{i,k}}$$

and for the cluster proportions

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{i,k}}{n}$$

- E-step is a simple expectation of a multinomial variable;
- M-step is MLE parameter updates for the fully observed model.

EM for Poisson mixture model

How hard is it to adapt this to Poisson mixture models?

EM for Poisson mixture model

Repeat until convergence:

1. **E-step**: For each sample i, compute

$$\hat{z}_i = \frac{\pi_k^{(t)} Pois(x_i \mid \mu_k^{(t)})}{\sum_{\ell=1}^{K} \pi_\ell^{(t)} Pois(x_i \mid \mu_\ell^{(t)})}$$

2. **M-step**: For each cluster k's distribution

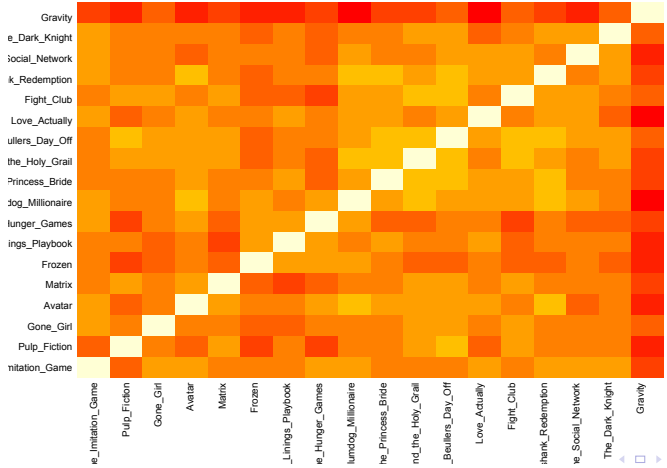
$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{i,k} x_i}{\sum_{i=1}^{n} \hat{z}_{i,k}}$$

and for the cluster proportions

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{i,k}}{n}$$

EM for Gaussian vs Poisson mixture models

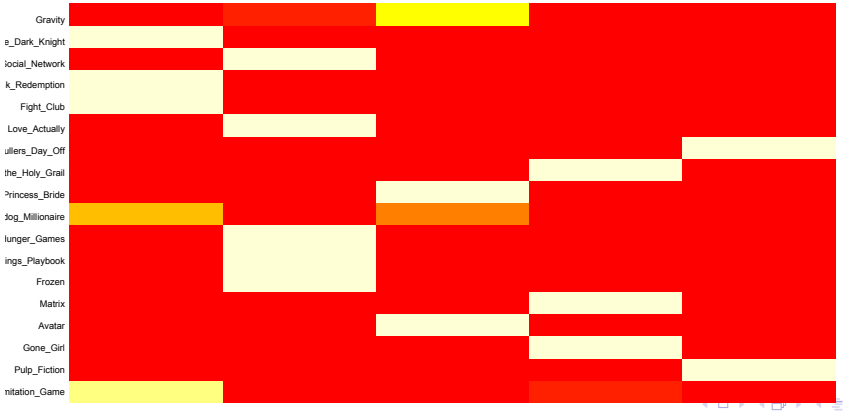
Let's look at some examples of Gaussian and Poisson mixture models to cluster movies based on movie rating data.



EM for Poisson mixture models

The Poisson mixture model has a Poisson mean parameter for each cluster K and each dimension p.

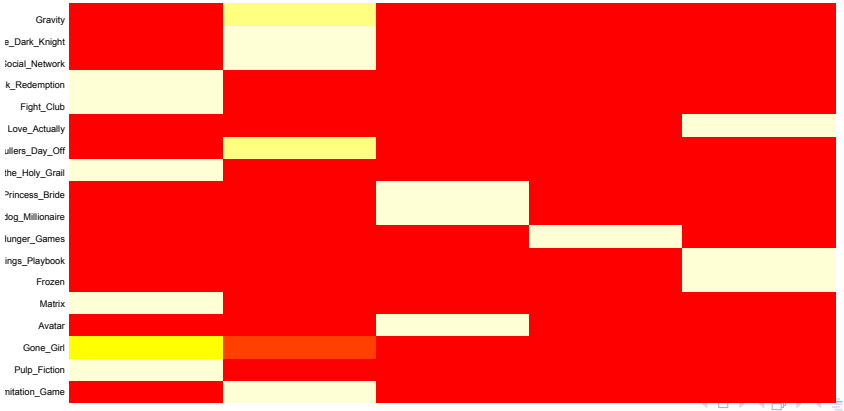
p is very large here (number of users) with respect to n or cluster size.



EM for Poisson mixture models

The Poisson mixture model has a Poisson mean parameter for each cluster K and each dimension p.

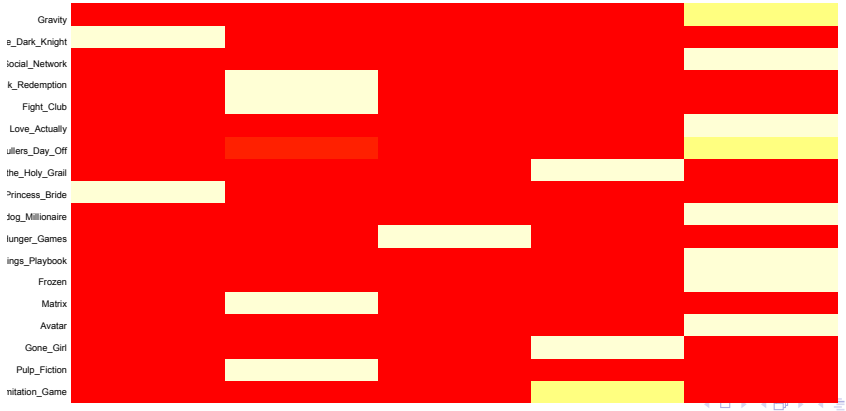
p is very large here (number of users) with respect to n or cluster size.



EM for Poisson mixture models

The Poisson mixture model has a Poisson mean parameter for each cluster K and each dimension p.

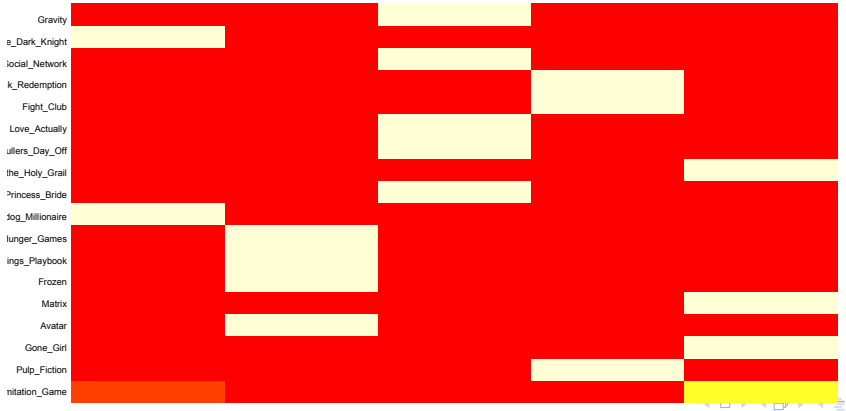
p is very large here (number of users) with respect to n or cluster size.



EM for Poisson mixture models

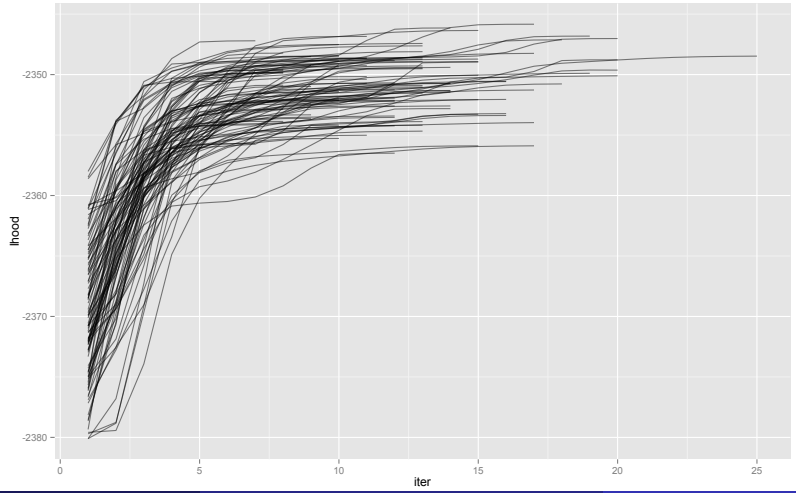
The Poisson mixture model has a Poisson mean parameter for each cluster K and each dimension p.

p is very large here (number of users) with respect to n or cluster size.



EM for Poisson mixture models

The Poisson mixture model has a Poisson mean parameter for each cluster K and each dimension p.



Concerns when applying the EM algorithm for PMM

Let's look more carefully at these parameters:

- The number of latent variables z in the PMM: $n \times K$
- The number of mean parameters μ in the PMM: $p \times K$
- When K is large, and $p \gg n$, there will be many local optima, and EM will be sensitive to initialization

How is a Gaussian mixture model the same or different?

Possibilities for applying the EM algorithm with MM

Possible ways to address these problems:

- Initialize using K-means: cluster centroids are initial cluster means
- Run multiple times from many starting points
- Look carefully at log likelihood over different runs
- Feature selection: reduce the number of features p
- Parameter sharing: share parameters across features p

Possibilities for applying the EM algorithm to mixture models

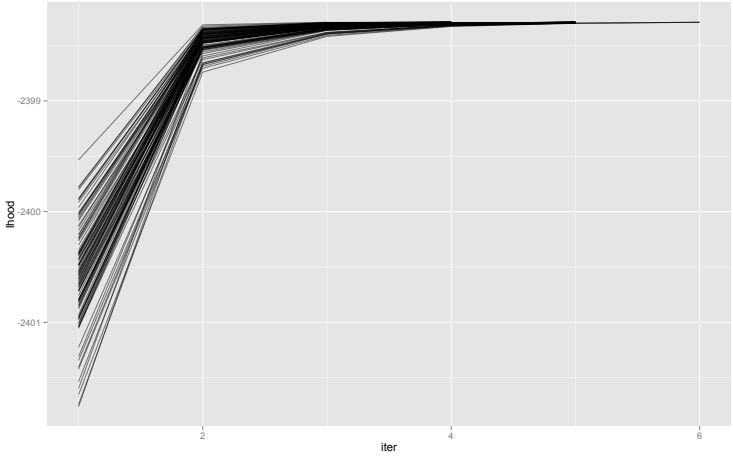
How can the probabilistic approach address limitations? Compare to k-means.

- Try different distributions for the mixture components.
- Make simplifying assumptions about mixture distributions (e.g., variance of Gaussian always 1)
- Add prior distribution on mixture model parameters:
  - Do I expect clearly differentiated clusters, or cluster overlap?
  - Do I know where or how large the variance is for my clusters?

EM for Poisson mixture models: cluster viewers

The Poisson mixture model has a mean parameter for each cluster K, dimension p.

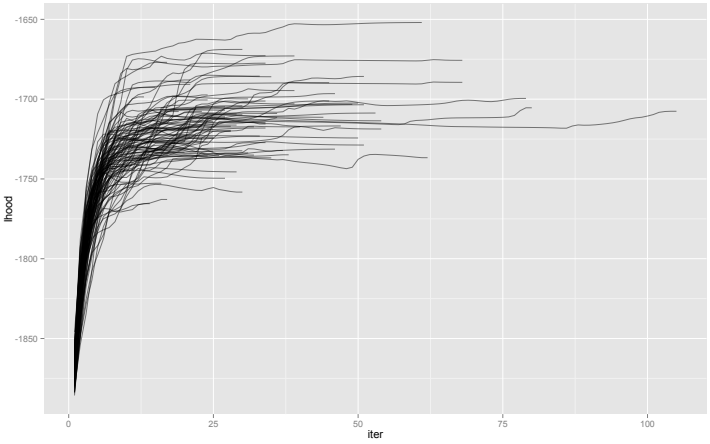
n is large (number of users) with respect to p or cluster size. This is good.



EM for Gaussian mixture models

The Gaussian mixture model has Gaussian parameter(s) for each cluster K, dimension p.

Nice when n is large with respect to p or cluster size.



EM for Gaussian mixture models

We can go back and look at the mean parameters for these models to understand the clusters.

Movie	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
The Imitation Game	4.131629	3.837014	4.589708	3.856768	4.149015
Pulp Fiction	4.476511	4.235793	3.590697	4.093246	4.382440
Gone Girl	3.882391	3.630153	4.288486	3.917859	3.984796
Avatar	3.386513	2.980243	4.056991	3.004667	4.133431
Matrix	4.682168	3.836441	4.033798	4.095153	4.403759
Frozen	3.528985	3.228648	4.460985	2.989737	3.660469
Silver Linings Playbook	3.485819	3.099168	4.266431	3.093523	3.636253
The Hunger Games	3.280261	2.810679	3.927428	3.311962	3.347376
Slumdog Millionaire	3.928292	3.398019	4.206427	3.658538	3.921264
The Princess Bride	3.718751	3.431397	4.219120	3.311042	4.427990
Monty Python and the Holy Grail	4.001472	4.012837	4.298896	4.232885	4.829485
Ferris Bueller's Day Off	4.054426	3.486423	3.846174	3.092836	3.941625
Love Actually	3.577531	3.179729	3.994402	2.969437	3.248977
Fight Club	4.671836	4.010403	4.279520	3.864173	4.806511
Shawshank Redemption	4.624158	4.109242	4.697616	4.409711	4.624788
The Social Network	3.802247	3.499471	3.977943	3.005592	3.564456
The Dark Knight	4.461302	3.710459	4.426439	3.387820	4.472258
Gravity	3.877861	3.696139	3.961985	3.544130	3.865516

EM for mixture models

Mixture model is a latent variable model that assumes each sample is generated from one cluster.

EM for mixture models applied to real data are very sensitive to starting points, choice of K, and often gets stuck in local minima.

Clustering in real data is rarely easy to interpret or validate, even with EM.

History of the EM algorithm

- Dempster et al. (1977): "Maximum likelihood from incomplete data via the EM algorithm"
- Hartley: "I feel like the old minstrel who has been singing his song for 18 years and now finds, with considerable satisfaction, that his folklore is the theme of an overpowering symphony."
- Hartley (1958) "Maximum likelihood procedures for incomplete data"
- McKendrik (1926) "Applications of mathematics to medical problems"

Additional Resources

- MLAPA: Chapter 11
- *Pattern Recognition and Machine Learning*, Chapter 9
- Dempster et al. (1977): "Maximum likelihood from incomplete data via the EM algorithm"
- Metacademy: *Expectation-Maximization algorithm*
- (video) Mathematical Monk: *Expectation-Maximization (EM)*