



Lecture 15 - EM Algorithm

Final exam



- When: **Friday May 6th** at 10am.
- What: 5-7 minute presentation each. (Can join with 1 or 2 others if you prefer.)
- Where: Zoom (room details will be on Blackboard under ZoomPro tab).
- I will be strict with timing so that we are not there all day (with apologies).
- Each person will share their screen when presenting
- Please have fun with it!



The Expectation-Maximization (EM) algorithm

The expectation maximization algorithm is a natural generalization of maximum likelihood estimation to the incomplete data case. – Chuong B Do & Serafim Batzoglou. What is the expectation maximization algorithm? Nature Biotechnology. 2008.

Overall



- A general framework, introduced by Laird and Rubin (1977) to encompass a variety of problems:
 - Filling in missing data
 - Inferring latent variables
 - Estimating HMM parameters
 - **Estimating parameters for mixture models**
 - Unsupervised cluster learning



EM had been around for a while before
it was formally 'christened'...

Newcomb (1887)

McKendrick (1926)

Hartley (1958)

Baum et al. (1970) [The HMM Baum-Welch
algorithm is an EM algorithm!]

Canonical example



- Have a mixture distribution.
- Don't know which points belong to which components of the mixture.
- Wish to estimate parameters of underlying mixture (or assign points probabilistically to the mixture components).

Motivating example



- Imagine we have two coins, for each of which the prob. of getting a “Heads”, p_1 , p_2 , is unknown.
- Suppose we repeat the following n times:
 - Pick a coin, uniformly at random, and toss it m times, recording the outcome.
- We can construct the maximum likelihood estimators of p_1 & p_2 as
 - MLE of p_1 = proportion of tosses of coin 1 that resulted in a “Heads”.
- *But what if we don't know which coin was chosen for each of the n trials?*

Motivating example



- Possible approach:
 1. Choose initial values for p_1 & p_2 .
 2. Given those values, find the prob. that each of the n trials used coin 1. (Bayes' theorem). Call these numbers π_1, \dots, π_n
 3. For each i , if $\pi_i > 0.5$ assign the i^{th} trial to coin 1; otherwise assign it to coin 2.
 4. Re-estimate p_1 & p_2 assuming the assignments in step 3 were correct.
 5. Repeat steps 2-4 until the estimates for p_1 & p_2 converge.

[This is essentially a version of k-means clustering.]

Motivating example

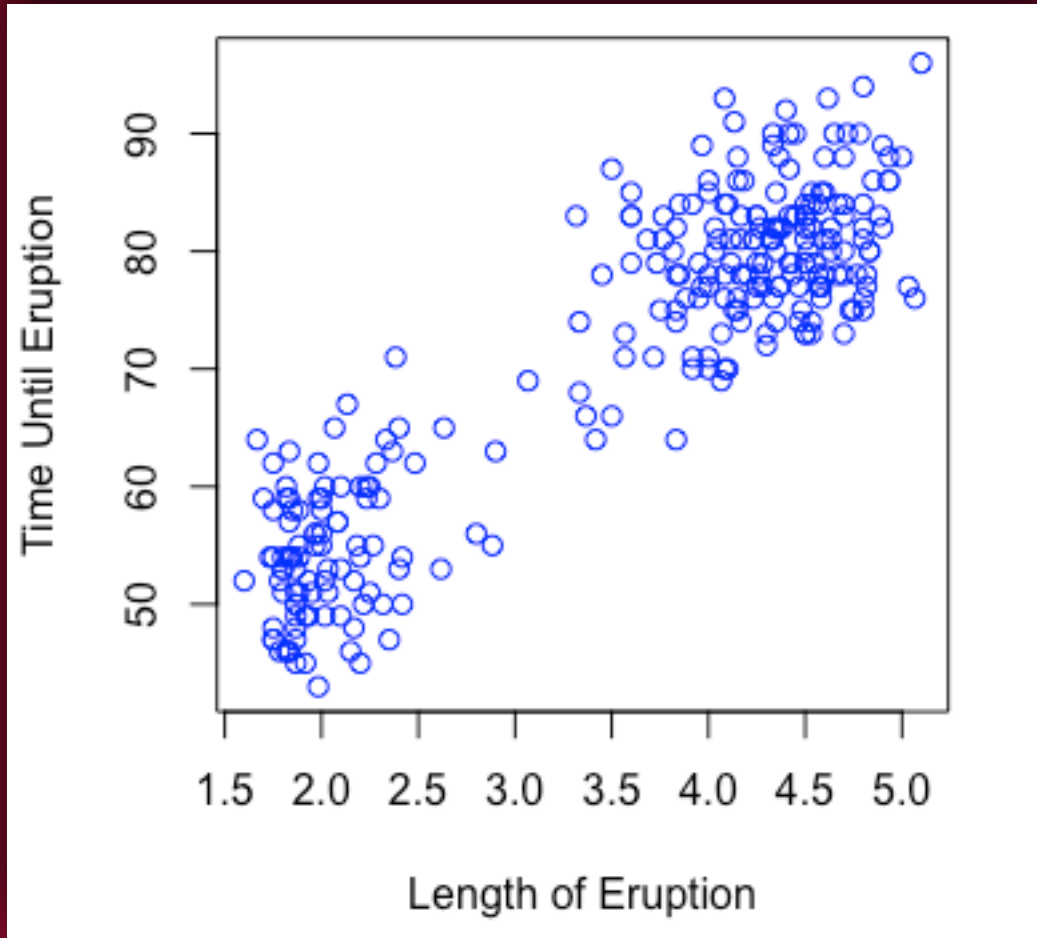


- The EM algorithm does the same thing, but with “soft” assignments.
- So for each trial, i , we would say it has probability π_i of being generated by coin 1 and $(1-\pi_i)$ of being generated by coin 2.
- We then repeat the algorithm on the previous page, but maximizing the likelihood for p_1 and p_2 using these *soft* assignments. So we get terms like

$$P(\text{HTTHH} \mid \text{trial } i) = \pi_i p_1^3 (1-p_1)^2 + (1-\pi_i) p_2^3 (1-p_2)^2$$

in the likelihood term that we are maximizing with respect to p_1 and p_2 .

Old Faithful

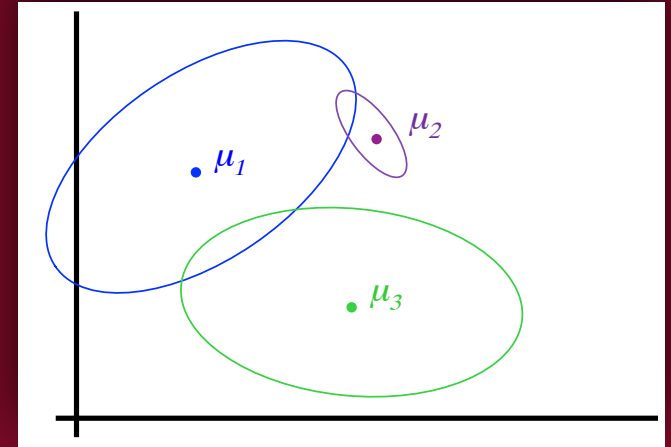


Gaussian Mixture Model example



Underlying assumptions:

1. There are k components
2. Each point is generated by:
 - Sampling a component Y with prob. $P(Y)$
 - Sampling from $N(\mu_Y, \sigma_Y^2)$





Probability model

- So, in general, for each datapoint, \mathbf{x} , we will have:

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \underbrace{\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{\text{Component}}$$

↑
Mixing coefficient

$$\forall k : \pi_k \geq 0 \quad \sum_{k=1}^K \pi_k = 1$$



Maximum likelihood

- The EM algorithm, seeks to maximize the marginal likelihood:

$$\operatorname{argmax}_{\theta} \prod_j P(x_j) = \operatorname{argmax}_{\theta} \prod_j \sum_{k=1}^K P(Y_j=k, x_j)$$

where Y_j is the cluster to which x_j is probabilistically assigned.

E-M algorithm



- The algorithm alternates between two steps:
 - Expectation - compute conditional probs. to **probabilistically** fill in missing (unobserved) values conditional on the current parameters.
 - Maximization - re-estimate the parameters conditional on the current **probabilistic** assignments (using maximum likelihood).

EM algorithm - mixture models



- E step: Calculate $P(Y_j=k \mid x_j, \theta)$.
- M step:
Set $\theta = \operatorname{argmax}_{\theta} \sum_j \sum_k P(Y_j=k \mid x_j, \theta) \log P(Y_j=k, x_j \mid \theta)$

x_j = the datapoints

Y_j = cluster to which x_j is assigned (probabilistic!)

θ = parameters (Normal means and variances here)

K = cluster index

EM algorithm - mixture models



- E-step:
 - Compute 'expected' clusters of all datapoints

$$P(Y_j = k | x_j, \mu_1 \dots \mu_K) \propto \exp\left(-\frac{1}{2\sigma^2} \|x_j - \mu_k\|^2\right) P(Y_j = k)$$

EM algorithm - mixture models



- M-step:
 - Compute most likely cluster means given current assignments:

$$\mu_k = \frac{\sum_{j=1}^m P(Y_j = k | x_j) x_j}{\sum_{j=1}^m P(Y_j = k | x_j)}$$



EM Algorithm

- See examples in EM_Algorithm repo from week 15.
- **Note:** convergence is only guaranteed to a local maximum, so run the algorithm multiple times from different start points.
- Further reading: “What is the expectation maximization algorithm?” [Chuong B Do & Serafim Batzoglou](#), [Nature Biotechnology](#), volume 26, pages 897–899 (2008)



END