*I529: Machine Learning in Bioinformatics* (Spring 2018)

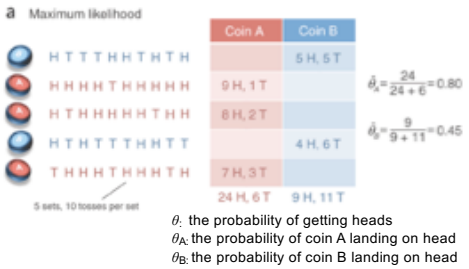
# Expectation-Maximization (EM) algorithm

Yuzhen Ye
School of Informatics and Computing
Indiana University, Bloomington
Spring 2018

## Contents



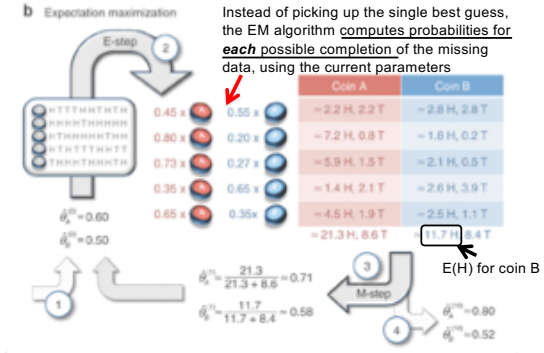
## A coin-flipping experiment




$\theta$: the probability of getting heads
$\theta_A$: the probability of coin A landing on head
$\theta_B$: the probability of coin B landing on head

Ref: What is the expectation maximization algorithm?
Nature Biotechnology 26, 897 - 899 (2008)

## When the identities of the coins are unknown




Instead of picking up the single best guess, the EM algorithm computes probabilities for ***each*** possible completion of the missing data, using the current parameters

E(H) for coin B

## Main applications of the EM algorithm

- When the data indeed has missing values, due to problems with or limitations of the observation process
- When optimizing the likelihood function is analytically intractable but it can be simplified by assuming the existence of and values for additional but *missing (or hidden)* parameters.

## The EM algorithm handles hidden data

Consider a model where, for observed data $x$ and model parameters $\theta$:

$$p(x|\theta) = \sum_z p(x,z|\theta).$$

**$z$ is the "hidden" variable that is marginalized out**

Finding $\theta^*$ which maximizes $\sum_z p(\boldsymbol{x},\boldsymbol{z}|\theta)$ is hard!

The EM algorithm reduces the difficult task of optimizing log $P(x; \theta)$ into a sequence of simpler optimization subproblems.

In each iteration, The EM algorithm receives parameters $\theta^{(t)}$, and returns new parameters $\theta^{(t+1)}$, s.t. $p(x|\theta^{(t+1)}) > p(x|\theta^{(t)})$.

## The EM algorithm

In each iteration the EM algorithm does the following.

E step: Calculate

$$Q_t(\theta) = \sum_z P(z|x;\hat{\theta}^{(t)}) log P(x,z;\theta)$$

M step: Find $\hat{\theta}^{(t+1)}$ which maximizes the $Q$ function

(Next iteration sets $\theta^{(t)} \leftarrow \hat{\theta}^{(t+1)}$ and repeats).

The EM update rule:

$$\hat{\theta}^{(t+1)} = \arg\max_\theta \sum_z P(z|x;\hat{\theta}^{(t)}) log P(x,z;\theta)$$

## Convergence of the EM algorithm

Compare the Q function and the g function

$$Q_t(\theta) = \sum_z P(z|x;\hat{\theta}^{(t)}) log P(x,z;\theta)$$

$$g_t(\theta) = \sum_z P(z|x;\hat{\theta}^{(t)}) log \frac{P(x,z;\theta)}{P(z|x;\hat{\theta}^{(t)})}$$

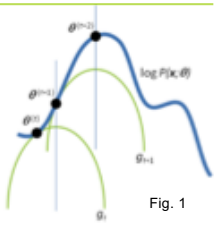



Fig. 1

Fig 1 demonstrates the convergence of the EM algorithm. Starting from initial parameters $\theta^{(t)}$, the E-step of the EM algorithm constructs a function $g_t$ that lower-bounds the objective function $log P(x;\theta)$ (i.e., $g_t \le log P(x;\theta)$; and $g_t(\hat{\theta}^{(t)}) = log P(x;\hat{\theta}^{(t)})$. In the M-step, $\theta^{(t+1)}$ is computed as the maximum of $g_t$. In the next E-step, a new lower-bound $g_{t+1}$ is constructed; maximization of $g_{t+1}$ in the next M-step gives $\theta^{(t+2)}$, etc.

As the value of the lower-bound $g_t$ matches the objective function at $\hat{\theta}^{(t)}$, if follows that

$$log P(x;\hat{\theta}^{(t)}) = g_t(\hat{\theta}^{(t)}) \le g_t(\hat{\theta}^{(t+1)}) = log P(x;\hat{\theta}^{(t+1)}) \quad (2)$$

So the objective function monotonically increases during each iteration of expectation maximization!

Ref: Nature Biotechnology 26, 897 - 899 (2008)

## The EM update rule

$$\hat{\theta}^{(t+1)} = \arg\max_\theta \sum_z P(z|x;\hat{\theta}^{(t)}) log P(x,z;\theta)$$

The EM update rule maximizes the log likelihood of a dataset expanded to contain all possible completions of the unobserved variables, where each completion is weighted by the posterior probability!

## Coin toss with **missing** data

- Given a coin with two possible outcomes: $H$ (head) and $T$ (tail), with probabilities $\theta$ and 1-$\theta$, respectively.
- The coin is tossed twice, **but only the 1st outcome, *T*, is seen. So the data is *x* = (*T*,*) (with incomplete data!)**
- We wish to apply the EM algorithm to get parameters that increase the likelihood of the data.
- Let the initial parameters be $\theta$ = ¼.

## The EM algorithm at work

$$Q_t(\theta) = \sum_z P(z|x;\theta^t) log P(x,z;\theta)$$

$$= P(z1|x;\theta^t) log P(x,z1;\theta) + P(z2|x;\theta^t) log P(x,z2;\theta)$$

$$= P(z1|x;\theta^t) log P(z1;\theta) + P(z2|x;\theta^t) log P(z2;\theta)$$

$$= P(z1|x;\theta^t) log[\theta^{n_H(z1)} \times (1-\theta)^{n_T(z1)}] + P(z2|x;\theta^t) log[\theta^{n_H(z2)} \times (1-\theta)^{n_T(z2)}]$$

$$= P(z1|x;\theta^t)[n_H(z1) log\theta + n_T(z1) log(1-\theta)] + P(z2|x;\theta^t)[n_H(z2) log\theta + n_T(z2) log(1-\theta)]$$

$$= [P(z1|x;\theta^t) n_H(z1) + P(z2|x;\theta^t) n_H(z2)] log\theta + [P(z1|x;\theta^t) n_T(z1) + P(z2|x;\theta^t) n_T(z2)] log(1-\theta)$$

Inputs:
Observation: x=(T,*)
Hidden data: z1=(T,T) z2=(T,H)
Initial guess: $\theta^t$ = ¼

$n_H = P(z1|x;\theta^t) n_H(z1) + P(z2|x;\theta^t) n_H(z2)$ $\quad$ $n_T = P(z1|x;\theta^t) n_T(z1) + P(z2|x;\theta^t) n_T(z2)$

$n_H log\theta + n_T log(1-\theta)$ is maximized when $\theta = \frac{n_H}{n_H + n_T}$

$P(x;\theta^t) = P(z1;\theta^t) + P(z2;\theta^t) = (1-\theta^t)^2 + (1-\theta^t)\theta = 3/4$

$P(z1|x;\theta) = P(x,z1;\theta^t)/P(x;\theta^t) = (1-\theta^t)^2 / P(x;\theta^t) = \frac{3/4 \times 3/4}{3/4} = 3/4$

$P(z2|x;\theta^t) = 1 - P(z1|x;\theta^t) = 1/4$

$n_H(z1) = 0$, $n_T(z1) = 2$, $n_H(z2) = 1$, and $n_T(z2) = 1$

$n_H = 1/4 \times 1 = 1/4$, $n_T = 3/4 \times 2 + 1/4 \times 1 = 7/4$, $\theta = \frac{n_H}{n_H + n_T} = \frac{1/4}{1/4 + 7/4} = 1/8$

## The EM algorithm at work: continue

- Initial guess $\theta$ = ¼
- After one iteration $\theta$ = 1/8
- …
- **The optimal parameter $\theta$ will never be reached by the EM algorithm!**

## Coin toss with **hidden** data

Two coins A and B, with parameters θ={$\theta_A$, $\theta_B$}; compute θ that maximizes the log likelihood of the observed data x={$x_1$,$x_2$,..$x_5$}

E.g., initial parameter $\theta$: $\theta_A$=0.60, $\theta_B$=0.50

$P(z_1 = A|x;\theta^t) = P(z_1 = A|x_1;\theta^t)$ ($x_1, x_2, ..x_5$ are independent observations)

$$= \frac{P(z_1 = A, x_1;\theta^t)}{P(z_1 = A, x_1;\theta^t) + P(z_1 = B, x_1;\theta^t)}$$

$$= \frac{0.6^5 \times 0.4^5}{0.6^5 \times 0.4^5 + 0.5^5 \times 0.5^5} = 0.58$$

| observation | $n_H$ | $n_T$ | P(A) | P(B) | Coin A $n_H$ | Coin A $n_T$ | B $n_H$ | B $n_T$ |
|---|---|---|---|---|---|---|---|---|
| x1: HTTTHHTHTH | 5 | 5 | 0.58 | 0.42 | 2.9 | 2.9 | 2.1 | 2.1 |
| x2: HHHHTHHHHH | 9 | 1 | 0.84 | 0.16 | 7.6 | 0.8 | 1.4 | 0.2 |
| x3: HTHHHHHTHH | 8 | 2 | 0.81 | 0.19 | 6.4 | 1.6 | 1.6 | 0.4 |
| x4: HTHTTTHHTT | 4 | 6 | 0.25 | 0.75 | 1.0 | 1.5 | 3.0 | 4.5 |
| x5: THHHTHHHTH | 8 | 2 | 0.81 | 0.19 | 6.4 | 1.6 | 1.6 | 0.4 |
| | | | | | 24.3H | 8.4T | 9.7H | 7.6T |

New parameter: $\theta_A$=24.3/(24.3+8.4)=0.74, $\theta_B$=9.7/(9.7+7.6)=0.56

## Motif finding problem

- Motif finding problem is not that different from the coin toss problem!
- Probabilistic approaches to motif finding
  - EM
  - Gibbs sampling (a generalized EM algorithm)
- There are also combinatorial approaches

## Motif finding problem

- Given a set of DNA sequences:

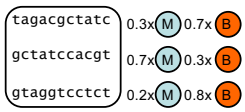
```
cctgatagacgctatctggctatccacgtacgtaggtcctctgtgcgaatctatgcgtttccaaccat
agtactggtgtacatttgatacgtacgtacaccggcaacctgaaacaaacgctcagaaccagaagtgc
aaacgtacgtgcaccctctttcttcgtggctctggccaacgagggctgatgtataagacgaaaatttt
agcctccgatgtaagtcatagctgtaactattacctgccacccctattacatcttacgtacgtataca
ctgttatacaacgcgtcatggcggggtatgcgttttggtcgtcgtacgctcgatcgttaacgtacgtc
```

- Find the motif in each of the individual sequences

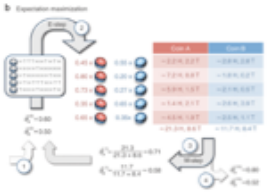
## The MEME algorithm

- Collect all substrings with the same length *w* from the input sequences: X = ($X_1$,…,$X_n$)
- Treat sequences as bags of subsequences: a bag for motif, and a bag for background
- Need to figure out two models (one for motif, and one for the background), and assign each of the subsequences to one of the bags, such that the likelihood of the data (subsequences) is maximized
  - Difficult problem
  - Solved by the EM algorithm

## Motif finding vs coin toss




Probability of a subsequence: P(x|M), or P(x|B)



$\theta$: the probability of getting heads
$\theta_A$: P(head) for coin A
$\theta_B$: P(head) for coin B

Probability of an observation sequence: $P(x|\theta)=\theta^{\#(heads)}(1-\theta)^{\#(tails)}$

## Fitting a mixture model by EM

- A finite mixture model:
  - data X = ($X_1$,…,$X_n$) arises from two or more groups with *g* models $\theta$ = ($\theta_1$, …, $\theta_g$).
- Indicator vectors Z = ($Z_1$,…,$Z_n$), where $Z_i$ = ($Z_{i1}$,…,$Z_{ig}$), and $Z_{ij}$ = 1 if $X_i$ is from group j, and = 0 otherwise.
- $P(Z_{ij}= 1|\theta_j) = \lambda_j$ . For any given i, all $Z_{ij}$ are 0 except one j;
- g=2: class 1 (the motif) and class 2 (the background) are given by position specific and a general multinomial distribution

## The E- and M-step

- E-step: Since the log likelihood is the sum of over i and j of terms multiplying $Z_{ij}$, and these are independent across i, we need only consider the expectation of one such, given $X_i$. Using initial parameter values $\theta'$ and $\lambda'$, and the fact that the $Z_{ij}$ are binary, we get

  $E(Z_{ij} | X, \theta', \lambda') = \lambda'_j P(X_i|\theta'_j) / \sum_k \lambda'_k P(X_i|\theta'_k) = Z'_{ij}$

- M-step: The maximization over $\lambda$ is independent of the rest and is readily achieved with

  $\lambda_j'' = \sum_i Z'_{ij} / n.$

## Baum-Welch algorithm for HMM parameter estimation

$$A_{kl} = \sum_{j=1}^{n} \frac{1}{p(x^j)} \sum_{i=1}^{L} p(s_{i-1}=k, s_i=l, x^j | \theta)$$

$$A_{kl} = \sum_{j=1}^{n} \frac{1}{p(x^j)} \sum_{i=1}^{L} f_k^j(i-1) a_{kl} e_l(x_i) b_l^j(i)$$

$$E_k(b) = \sum_{j=1}^{n} \frac{1}{p(x^j)} \sum_{i:x_i^j=b} f_k^j(i) f_k^j(i)$$

During each iteration, compute the expected transitions between any pair of states, and expected emissions from any state, using averaging process (E-step), which are then used to compute new parameters (M-step).

## Application of EM algorithms in metagenomics: Binning

- AbundanceBin
  - Binning of short reads into bins (species)
  - A Novel Abundance-Based Algorithm for Binning Metagenomic Sequences Using l-tuples ( JCB 2011, 18(3): 523-534. RECOMB 2010)
- MaxBin/MaxBin2
  - Binning of assembled metagenomic scaffolds using an EM algorithm (Microbiome, 2014 doi: 10.1186/2049-2618-2-26)

## Pros and Cons

- Cons
  - Slow convergence
  - Converge to local optima
- Pros
  - The E-step and M-step are often easy to implement for many problems, thanks to the nice form of the complete-data likelihood function
  - Solutions to the M-steps often exist in the closed form
- Ref
  - On the convergence properties of the EM algorithm. CFJ WU, 1983
  - A gentle tutorial of the EM algorithm and its applications to parameter estimation for Gaussian mixture and hidden Markov models, JA Bilmes, 1998
  - What is the expectation maximization algorithm? 2008