Workshop 3 – Advanced Clustering

Advanced Analytics and Applications [AAA]

Information Systems for Sustainable Society (is3) | WiSo Faculty | Univ.-Prof. Dr. Wolfgang Ketter | 4/27/23

I Multiple Choice Questions

II Expectation Maximization Algorithm

III Programming

Question 1.1: Soft vs. Hard Clustering

What are the differences of soft clustering compared to hard clustering?

i. Soft clustering is faster than hard clustering.
ii. Both approaches assign a data point to every cluster with a certain probability.
iii. Only soft clustering assigns an item to each and every cluster with a certain probability.

Question 1.1: Soft vs. Hard Clustering

What are the differences of soft clustering compared to hard clustering?

i. Soft clustering is faster than hard clustering.
ii. Both approaches assign a data point to every cluster with a certain probability.
iii. Only soft clustering assigns an item to each and every cluster with a certain probability.

Question 1.2: True or False? GMM superb?

True or False? Of the clustering algorithms covered in class, Gaussian Mixture Models used for clustering always outperforms k-means and single link clustering

i. True
ii. False

Question 1.2: True or False? GMM superb?

True or False? Of the clustering algorithms covered in class, Gaussian Mixture Models used for clustering always outperforms k-means and single link clustering

i. True

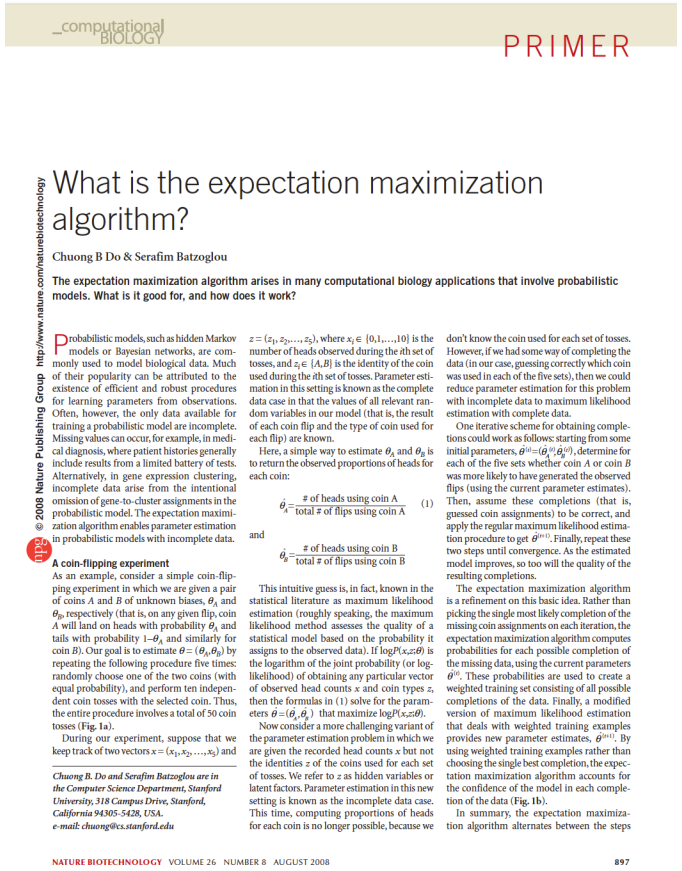
ii. False

I Multiple Choice Questions

II Expectation Maximization Algorithm

III Programming

Expectation Maximization



_computational BIOLOGY

PRIMER

What is the expectation maximization algorithm?

Chuong B Do & Serafim Batzoglou

The expectation maximization algorithm arises in many computational biology applications that involve probabilistic models. What is it good for, and how does it work?

Probabilistic models, such as hidden Markov models or Bayesian networks, are commonly used to model biological data. Much of their popularity can be attributed to the existence of efficient and robust procedures for learning parameters from observations. Often, however, the only data available for training a probabilistic model are incomplete. Missing values can occur, for example, in medical diagnosis, where patient histories generally include results from a limited battery of tests. Alternatively, in gene expression clustering, incomplete data arise from the intentional omission of gene-to-cluster assignments in the probabilistic model. The expectation maximization algorithm enables parameter estimation in probabilistic models with incomplete data.

A coin-flipping experiment

As an example, consider a simple coin-flipping experiment in which we are given a pair of coins A and B of unknown biases, θ_A and θ_B, respectively (that is, on any given flip, coin A will land on heads with probability θ_A and tails with probability $1-\theta_A$ and similarly for coin B). Our goal is to estimate $\theta = (\theta_A, \theta_B)$ by repeating the following procedure five times: randomly choose one of the two coins (with equal probability), and perform ten independent coin tosses with the selected coin. Thus, the entire procedure involves a total of 50 coin tosses (**Fig. 1a**).

During our experiment, suppose that we keep track of two vectors $x = (x_1, x_2, \ldots, x_5)$ and $z = (z_1, z_2, \ldots, z_5)$, where $x_i \in \{0,1,\ldots,10\}$ is the number of heads observed during the ith set of tosses, and $z_i \in \{A,B\}$ is the identity of the coin used during the ith set of tosses. Parameter estimation in this setting is known as the complete data case in that the values of all relevant random variables in our model (that is, the result of each coin flip and the type of coin used for each flip) are known.

Here, a simple way to estimate θ_A and θ_B is to return the observed proportions of heads for each coin:

$$\hat{\theta}_A = \frac{\#\text{ of heads using coin A}}{\text{total }\#\text{ of flips using coin A}} \quad (1)$$

and

$$\hat{\theta}_B = \frac{\#\text{ of heads using coin B}}{\text{total }\#\text{ of flips using coin B}}$$

This intuitive guess is, in fact, known in the statistical literature as maximum likelihood estimation (roughly speaking, the maximum likelihood method assesses the quality of a statistical model based on the probability it assigns to the observed data). If $\log P(x,z;\theta)$ is the logarithm of the joint probability (or log-likelihood) of obtaining any particular vector of observed head counts x and coin types z, then the formulas in (1) solve for the parameters $\hat{\theta} = (\hat{\theta}_A, \hat{\theta}_B)$ that maximize $\log P(x,z;\theta)$.

Now consider a more challenging variant of the parameter estimation problem in which we are given the recorded head counts x but not the identities z of the coins used for each set of tosses. We refer to z as hidden variables or latent factors. Parameter estimation in this new setting is known as the incomplete data case. This time, computing proportions of heads for each coin is no longer possible, because we don't know the coin used for each set of tosses. However, if we had some way of completing the data (in our case, guessing correctly which coin was used in each of the five sets), then we could reduce parameter estimation for this problem with incomplete data to maximum likelihood estimation with complete data.

One iterative scheme for obtaining completions could work as follows: starting from some initial parameters, $\theta^{(t)} = (\theta_A^{(t)}, \theta_B^{(t)})$, determine for each of the five sets whether coin A or coin B was more likely to have generated the observed flips (using the current parameter estimates). Then, assume these completions (that is, guessed coin assignments) to be correct, and apply the regular maximum likelihood estimation procedure to get $\theta^{(t+1)}$. Finally, repeat these two steps until convergence. As the estimated model improves, so too will the quality of the resulting completions.

The expectation maximization algorithm is a refinement on this basic idea. Rather than picking the single most likely completion of the missing coin assignments on each iteration, the expectation maximization algorithm computes probabilities for each possible completion of the missing data, using the current parameters $\theta^{(t)}$. These probabilities are used to create a weighted training set consisting of all possible completions of the data. Finally, a modified version of maximum likelihood estimation that deals with weighted training examples provides new parameter estimates, $\theta^{(t+1)}$. By using weighted training examples rather than choosing the single best completion, the expectation maximization algorithm accounts for the confidence of the model in each completion of the data (**Fig. 1b**).

In summary, the expectation maximization algorithm alternates between the steps

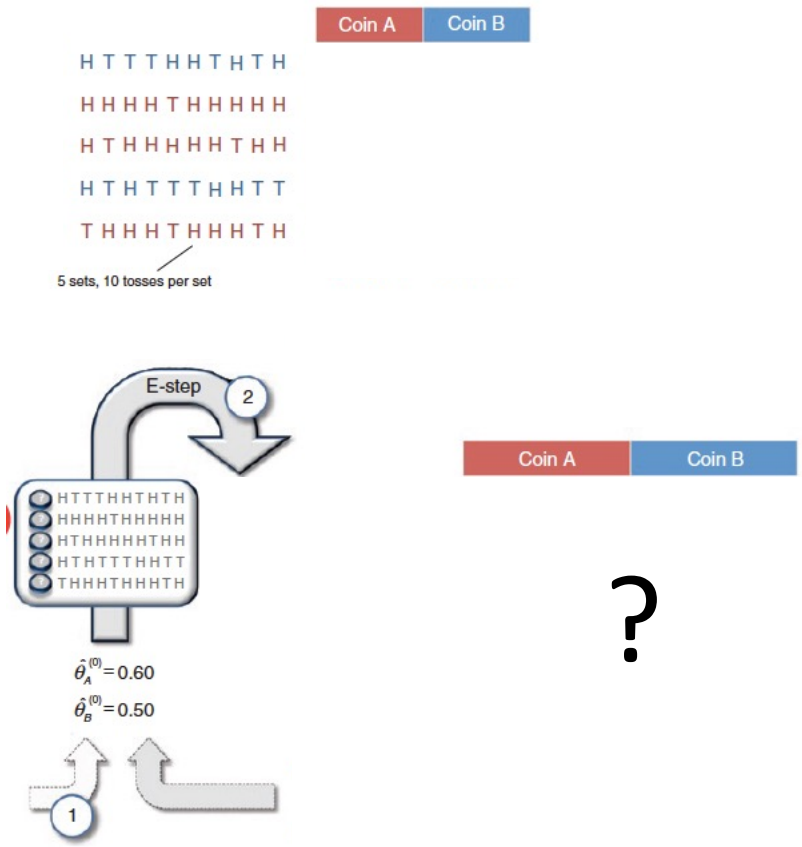
Chuong B. Do and Serafim Batzoglou are in the Computer Science Department, Stanford University, 318 Campus Drive, Stanford, California 94305-5428, USA.
e-mail: chuong@cs.stanford.edu

For a deep dive: Read the paper.
The following example is based on it.

Do, C., Batzoglou, S. What is the expectation maximization algorithm?. Nat Biotechnol 26, 897–899 (2008). https://doi.org/10.1038/nbt1406

Universität zu Köln

Expectation Maximization



Explain the application of the EM algorithm in light of the coin toss experiment.

Solution Part a): We have conducted an experiment with 5 rounds using two coins A and B, which are biased $\theta_A \neq \theta_B \neq 0.5$.

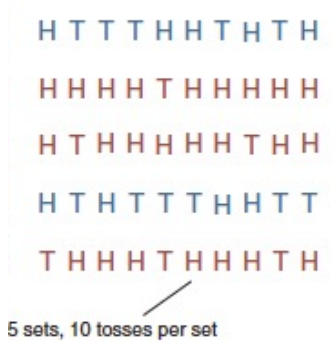
At each round, we tossed the selected coin 10 times.

However, we don't know which coin we used during each round (i.e., we cannot use MLE).

Our objective is to estimate the coefficients θ_A , θ_B.

Recall that coin tossing follows a binomial distribution.

Expectation Maximization



We solve the problem using EM.

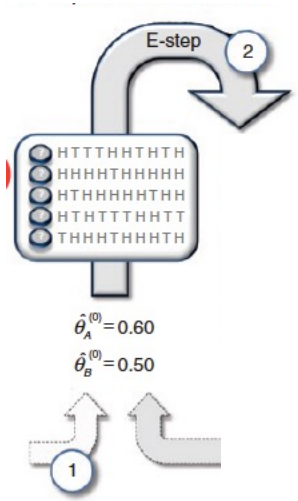
Solution Part b): Expectation Step.

1) Set initial (random) estimates

$$\theta_A^0 = 0.6 \neq \theta_B^0 = 0.5$$

2) Calculate the respective relative frequency for each round θ_i, which is also the MLE estimate within the round.

Expectation Maximization



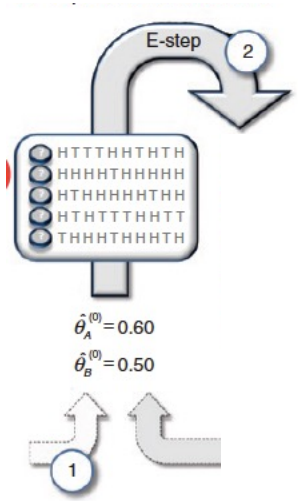
We solve the problem using EM.

Solution Part c): Calculate the weights, or in other words, the probability how likely a record is drawn from either coin A or coin B.

Therefore, we need to calculate the log likelihood values for each of the five records based on the current estimate θ_A^j, and θ_B^j according to:

$$\log_e(\mathcal{L}) = \log_e\left(\mathcal{L}(p \mid n, y)\right) = \log_e\binom{n}{y} + y \cdot log_e(p) + (n - y) \cdot \log_e(1 - p).$$

Expectation Maximization



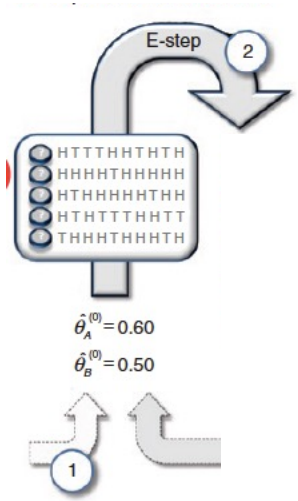
We solve the problem using EM.

Solution Part d): Based on these log liklihoods, we then calculate the weights for each round as follows:

Weight for Coin A: $\frac{\exp(LL\ A)}{\exp(LL\ A)+\exp(LL\ B)}$

Weight for Coin B: $\frac{\exp(LL\ B)}{\exp(LL\ A)+\exp(LL\ B)}$

Expectation Maximization



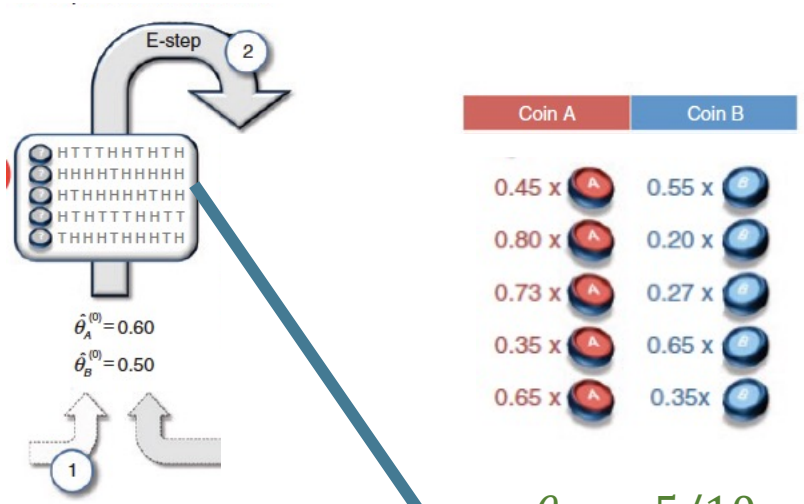
We solve the problem using EM.

Solution Part d): Based on these log liklihoods, we then calculate the weights for each round as follows:

Weight for Coin A: $\frac{\exp(LL\ A)}{\exp(LL\ A)+\exp(LL\ B)}$

Weight for Coin B: $\frac{\exp(LL\ B)}{\exp(LL\ A)+\exp(LL\ B)}$

Expectation Maximization



$$\theta_1 = 5/10$$
$$\theta_2 = 9/10$$
$$\theta_3 = 8/10$$
$$\theta_4 = 4/10$$
$$\theta_5 = 7/10$$

We solve the problem using EM.

Solution Part d): Maximization Step:

Maximization Step Coin A:

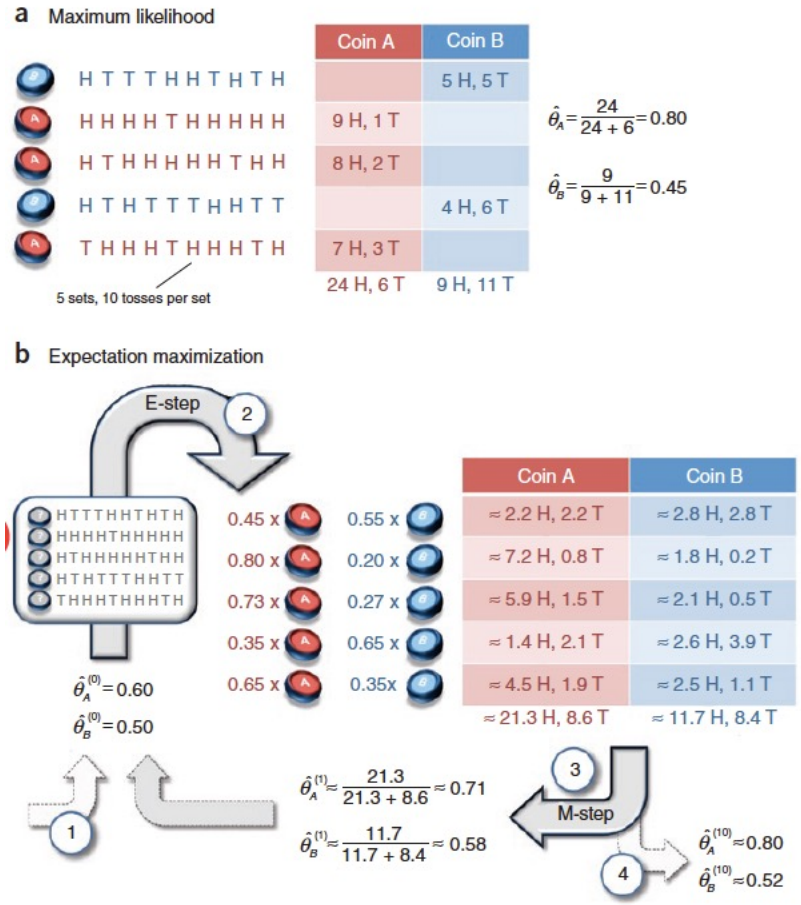
$$\theta_A^1 = \frac{0.45 * \left(\frac{5}{10}\right) + 0.8 * \left(\frac{9}{10}\right) + 0.73 * \left(\frac{8}{10}\right) + 0.35 * \left(\frac{4}{10}\right) + 0.65 * \left(\frac{7}{10}\right)}{0.45 + 0.8 + 0.73 + 0.35 + 0.65} = 0.71$$

Maximization Step Coin B:

$$\theta_B^1 = \frac{0.55 * \left(\frac{5}{10}\right) + 0.2 * \left(\frac{9}{10}\right) + 0.27 * \left(\frac{8}{10}\right) + 0.65 * \left(\frac{4}{10}\right) + 0.35 * \left(\frac{7}{10}\right)}{0.55 + 0.2 + 0.27 + 0.65 + 0.35} = 0.58$$

Then, repeat expectation and maximization step until convergence.

Expectation Maximization



Explain the difference between maximum likelihood estimation and EM approaches based on the coin toss example.

Solution: Maximum Likelihood Approach requires full information on the group assignment of each coin. EM doesn‘t require these information but tries to learn it from the data.

ML finds global optimum, EM splits the problem into sub problems and find for these sub problems global optima.

After that it combines the solutions of each sub problem, which in turn might end up in local optima of the overarching problem.

I Multiple Choice Questions

II Expectation Maximization Algorithm

III Programming

Contact

For general questions and enquiries on **research**, **teaching**, **job openings** and new **projects** refer to our website at www.is3.uni-koeln.de

For specific enquiries regarding this course contact us by sending an email to the **IS3 teaching** address at is3-teaching@wiso.uni-koeln.de