6.864 Advanced Natural Language Processing[1]

Lecture 4: EM Algorithm and Topic Model

24 September 2015

Earlier, we saw cases where the full data was observable and we could apply a straightforward maximum-likelihood estimator for the parameters. In Lecture 4, we introduce a Hidden Variable Model where data is partially observable. Then, we provide basic intuitions for the EM algorithm which can be used to optimize the likelihood on such Hidden Variable Models. We explore further by demonstrating the EM algorithm on a toy example with biased coins. Later, we connect our concept to more relevant topics in NLP such as topic modeling.

Hidden Variable Models

Motivation

Hidden Variable Models are very common in NLP applications. To gain intuition, see the following examples.

1. *Topic Model*: Consider the following traditional unigram model. Suppose document D is comprised of N words such that $D = \{w_1, w_2, ..., w_N\}$. Given a unigram distribution $p(w_i)$, the log-likelihood of any document will be given by the same form

$$\log p(D) = \sum_{i=1}^{N} \log p(w_i)$$

However in practice, each documents are different and therefore we may want to use different distributions over the words depending on the context. This idea can be easily implemented in Topic Models. A topic is denoted by z where $z \in \mathcal{Z}$, and $\mathcal{Z}$ is our pre-defined universe of topics of k dimension. Each word w_i is sampled from a corresponding topic z_i.

The problem in Topic Models arise from the fact that the topic information is not provided in the training corpus. If the entire data had been observed, we could simply estimate the conditional distribution as the empirical count,

$$\hat{p}(w|z) = \frac{count(w,z)}{count(z)} \tag{1}$$

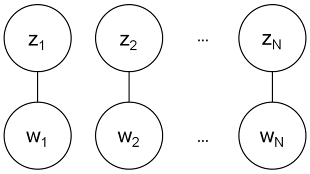


Figure 1: Topic model is where each word is associated with a conditional topic. For example, it is more likely to observe word $w_i = "Cat"$ if the topic is $z_i = "Pets"$.

However, such straightforward approach seen in (1) is no longer applicable if the variables z_i are hidden. Since it is unlikely to have each word in a document explicitly tagged with its related topics, topic models naturally arise as Hidden Variable Models.

[1] Instructors: Prof. Regina Barzilay, and Prof. Tommi Jaakkola.
TAs: Franck Dernoncourt, Karthik Rajagopal Narasimhan, Tianheng Wang.
Scribes: Dongyoung Kim.

2. *Machine Translation Alignment*: In Machine Translation, one is given two sentences in different languages with the same meaning. As seen in Figure 2, when the word *"I"* is paired together many times with the corresponding Korean word, then it is probably a good indicator that *"I"* should be translated to that word. Therefore, the first step in Machine Translation is to align corresponding translations. However, the alignments are not explicitly given in real life applications and should be treated as hidden variables one has to estimate.

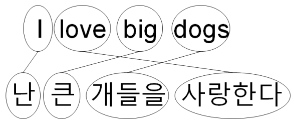


Figure 2: In machine translation, each word from one language has a hidden alignment to its counterpart.

3. *Part-of-Speech Tagging*: Part-of-speech tagging, which will be covered in the next lecture, matches each word in a sentence with its part of speech in English. For the example in Figure 3, *"I"* is tagged with *"pronoun"*, *"love"* with *"verb"*, *"big"* with *"adjective"*, and *"dogs"* with *"noun"*. When the full observed data observable, one can apply a supervised learning algorithm. However when the parts of speech are not observed, then they are treated as hidden variables.

pro.
v.
adj.
n.
I
love
big
dogs

Figure 3: Each word is associated with a hidden part-of-speech in English.

Observed Case vs. Unobserved Case

In previous lectures we looked at models where the full data was observed. We use

$$D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$$

to denote the given dataset in such cases, which is represented as a list of sampled tuples. Each data point (x, y) is independently sampled from some distribution $p_{X,Y}(x, y)$ parameterized by θ. The objective function we are trying to maximize is given by the following

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p_{X,Y}(x_i, y_i|\theta) \tag{2}$$

Now, let's look at the unobserved case where $\mathbf{y}$ is unseen. In other words, $\mathbf{x}$ is the observed variable and $\mathbf{y}$ is the hidden variable. In this case, the observed data is given by

$$D = \{x_1, x_2, ..., x_N\}$$

In this case, we replace the objective function with the log-likelihood of the *partially observed data*, which is

$$\begin{aligned} \theta^* &= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p_X(x_i|\theta) \\ &= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log \sum_{y \in \mathcal{Y}} p_{X,Y}(x_i, y|\theta) \end{aligned} \tag{3}$$

where the second equality is due to marginalization over y over $p_{X,Y}(x,y)$. The key point is to note that the criteria for the observed case and the unobserved case is different, and we should use a different objective function when we are trying to optimize for the parameter θ. In the following sections we will demonstrate how the EM algorithm finds the optimum parameter for (3).

Before we see how the EM algorithm can optimize (3), let's first how we can compute the solution for (2) through a simple example.

Example 1. Assume a simple unigram model where there are two words in the universe such that $V = \{\text{"dog"}, \text{"cat"}\}$. There is one free parameter given by

$$\begin{aligned} \theta &= p(\text{"dog"}) \\ 1-\theta &= p(\text{"cat"}) \end{aligned}$$

Given document $D = \{w_1, w_2, ..., w_N\}$, the log-likelihood of the document is given by

$$\begin{aligned} l(D|theta) &\triangleq \log p(D|\theta) \\ &= \log(\theta^d(1-\theta)^c) \\ &= d \cdot \log\theta + c \cdot \log(1-\theta) \end{aligned}$$

Where d is a constant equal to the number of occurrence of "dog" in the document, and c is equal to that of "cat". To compute θ^*, we take the partial derivative of the log-likelihood with respect to θ and set it to zero.

$$\begin{aligned} \frac{\partial l(D|theta)}{\partial\theta} &= \frac{d}{\theta^*} - \frac{c}{1-\theta^*} \\ &= 0 \end{aligned}$$

Multiplying $\theta^*(1-\theta^*)$ to both sides,

$$\begin{aligned} d(1-\theta^*) - c\theta^* &= 0 \\ d - d\theta^* - c\theta^* &= 0 \\ (d+c)\theta^* &= d \\ \rightarrow \qquad \theta^* &= \frac{d}{d+c} \end{aligned}$$

We can easily derive that the Maximum Likelihood Estimator is given by the fraction of the empirical counts.

EM Algorithm

Example 2. We will now set up a simple example to walk through how the EM algorithm works. Suppose we have a card and two different coins, where all probabilities are biased. We will take the following steps to sample a sequence of coin flips.

1. Flip the card, which either returns side A or side B.
2. If A was drawn from step 1, toss coin 1 three times. Otherwise if B was drawn, toss coin 2 three times.
3. repeat steps 1-2

Expressing this more formally, y represents the side of the card and x represents the outcome of three consecutive coin tosses. H and T denotes the head and tail of a coin, respectively.

$$\begin{aligned} y &\in \mathcal{Y} = \{A, B\} \\ x &\in \mathcal{X} = \{HHH, HTH, ..., TTT\} \end{aligned}$$

Furthermore, let us assume that we did not see the card values y, and we could only observe the sequence of the coin flips x. Since the data is partially observed, this example is an instance of a Hidden Variable Model.

There are three free parameters in this setup, given by $\theta = \{\alpha, p_A, p_B\}$. The parameters are described as follows

$$\begin{aligned} \alpha &= p(Card = A) \\ p_A &= p(Coin1 = H) \\ p_B &= p(Coin2 = H) \end{aligned}$$

It is easy to see that

$$\begin{aligned} p(x, y|\theta) &= p(y|\theta)p(x|y, \theta) \\ p(y|\theta) &= \begin{cases} \alpha & \text{if } y = A \\ 1 - \alpha & \text{if } y = B \end{cases} \\ p(x|y, \theta) &= \begin{cases} p_A^{h(x)}(1 - p_A)^{t(x)} & \text{if } y = A \\ p_B^{h(x)}(1 - p_B)^{t(x)} & \text{if } y = B \end{cases} \end{aligned}$$

Where we define $h(x)$ to be a function that is equal to the number of heads in sample x, and $t(x)$ to be the number of tails. For example, if $x_0 = HHT$, then $h(x_0) = 2$ and $t(x_0) = 1$.

Observed Case

Let's go back on our assumption that the card values $\mathbf{y}$ were hidden, and say the card flips and the coin tosses were entirely observable. Each card flip is associated with three coin tosses. Suppose after three iterations, the observed data was given as follows

$$D = \{(A, HHH), (A, HHH), (B, TTT)\}$$

where $(y, x) = (A, HHH)$ denotes that the observed card flip was A, followed by three heads. Earlier in Example 1, we have seen that applying partial derivative to the log-likelihood given in (2), the MLE parameters are simply given by the empirical counts. Using this fact for granted, the parameter values for these specific samples are given by

$$\left.\begin{aligned} \alpha^* &= \frac{\text{count(A)}}{3} = \frac{2}{3} \\ p_A^* &= \frac{\text{count}(A,H)}{\text{count}(A,H)+\text{count}(A,T)} = \frac{6}{6} \\ p_B^* &= \frac{\text{count}(B,H)}{\text{count}(B,H)+\text{count}(B,T)} = \frac{0}{3} \end{aligned}\right\} \text{ MLE parameters}$$

Shortly we will see that this analytic solution has a very similar form to the solution for each iteration in the EM algorithm, which is covered in the next section.

Unobserved Case

Continuing with the problem setup in Example 2, suppose the card flips are no longer observable. In other words, we will treat the card flip y as a hidden variable. Consider the following sample data

$$D = \{HHH, HHH, TTT\} = \mathbf{x}$$

Let us make an initial guess on the parameter values. Assign arbitrary prior values such that

$$\alpha^{(0)} = 0.1, \quad p_A^{(0)} = 0.8, \quad p_B^{(0)} = 0.5 \tag{4}$$

The initial guess in (4) is denoted with a superscript 0 to mark that this is the belief of the parameters at time 0. We will see how the belief is updated over each iteration of the EM algorithm.

For each x_i in the dataset, we compute $p_{Y|X,\theta}(y|x_i, \theta^{(0)})$ over all possible values of $y \in \mathcal{Y} = \{A, B\}$. For example, for our For $x_1 = HHH$ in the given example dataset,

$$\begin{aligned} &p(y_1 = A|x_1 = HHH, \theta^{(0)}) \\ &= \frac{p(x_1 = HHH, y_1 = A|\theta^{(0)})}{p(x_1 = HHH, y_1 = A|\theta^{(0)}) + p(x_1 = HHH, y_1 = B|\theta^{(0)})} \\ &= \frac{\alpha p_A^3}{\alpha p_A^3 + (1-\alpha)p_B^3} \\ &= \frac{0.1 \times 0.8^3}{0.1 \times 0.8^3 + (1-0.1) \times 0.5^3} \approx 0.3 \end{aligned}$$

$$p(y_1 = B|x_1 = HHH, \theta^{(0)})$$

$$
\begin{aligned}
&= \frac{p(x_1 = HHH, y_1 = B|\theta^{(0)})}{p(x_1 = HHH, y_1 = A|\theta^{(0)}) + p(x_1 = HHH, y_1 = B|\theta^{(0)})} \\
&= \frac{(1-\alpha)p_B^3}{\alpha p_A^3 + (1-\alpha)p_B^3} \\
&= \frac{(1-0.1) \times 0.5^3}{0.1 \times 0.8^3 + (1-0.1) \times 0.5^3} \approx 0.7
\end{aligned}
$$

Recall from the previous section that in the fully observed case, each x_i was explicitly associated with a *single* count of either $y_i = A$ or $y_i = B$. In the unobserved case, we assume that each x_i is associated with *fractional counts* over all possible values of y_i, to which we assign the *expected count* of y_i given x_i and our prior knowledge of θ.

x_i	y_i	empirical count
HHH	A	1
HHH	A	1
TTT	B	1

(a) y is observed

x_i	y_i	fractional count
HHH	A	0.3
	B	0.7
HHH	A	0.3
	B	0.7
TTT	A	0.5
	B	0.5

(b) y is unobserved

Table 1: In the fully observed case (a), a single value of y (which is the observed value itself) is associated with each i^{th} data sample. In the unobserved case (b), we assume that a entire list over the alphabet $\mathcal{Y}$, is associated with x_i. Each element in the spectrum is assigned a weight, or the *fractional count*, which corresponds to the expected value of y given x_i. The fractional count can also be understood as the confidence in the sample (x_i, y_i) given a prior belief over the parameters.

Hereafter the remaining steps are almost identical to the MLE maximization of the parameters. Again we will derive our distribution by counting, but since the actual counts are no longer available, we will count the fractional counts instead. The parameters for the next iteration, or time 1, will be computed as follows

$$
\begin{aligned}
\alpha^{(1)} &\leftarrow \frac{\text{fractional count of } A}{3} = \frac{0.3+0.3+0.5}{3} = \frac{1.1}{3} \\
p_A^{(1)} &\leftarrow \frac{\text{fractional count of } (A, H)}{\text{fractional count of } (A, H) + \text{fractional count of } (A, T)} \\
&= \frac{3 \times 0.3 + 3 \times 0.3}{3 \times 0.3 + 3 \times 0.3 + 3 \times 0.5} = \frac{1.8}{3.3} \\
p_B^{(1)} &\leftarrow \frac{\text{fractional count of } (B, H)}{\text{fractional count of } (B, H) + \text{fractional count of } (B, T)} \\
&= \frac{3 \times 0.7 + 3 \times 0.7}{3 \times 0.7 + 3 \times 0.7 + 3 \times 0.5} = \frac{4.2}{5.7}
\end{aligned}
$$

The EM algorithm guarantees that the likelihood given by Equation (3) is increased over every iteration as we update our parameters $\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, ..., \theta^{(T)}$. In summary, the EM algorithm is given by the following steps

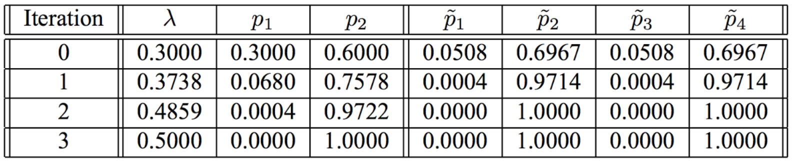
1. Randomly initialize $\theta^{(0)}$

2. Repeat until convergence

   - **E-step:** Based on $\theta^{(t)}$, compute $p_{(}y|x,\theta^{(t)})$ and count the fractionals
   - **M-step:** Re-estimate $\theta^{(t+1)}$

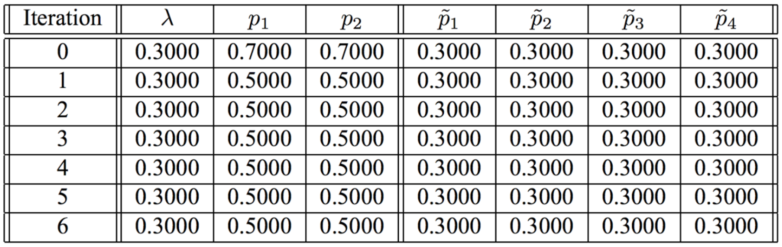
Properties of EM Algorithms

The EM algorithm is guaranteed to converge to a local maximum. As with many local-maximum search methods, initialization plays a crucial role in the EM algorithm. See the Figures 4 and 5 to understand when the EM algorithm works well and when it can get stuck, depending on the initialization values.



Iteration	λ	p_1	p_2	$\tilde{p}_1$	$\tilde{p}_2$	$\tilde{p}_3$	$\tilde{p}_4$
0	0.3000	0.3000	0.6000	0.0508	0.6967	0.0508	0.6967
1	0.3738	0.0680	0.7578	0.0004	0.9714	0.0004	0.9714
2	0.4859	0.0004	0.9722	0.0000	1.0000	0.0000	1.0000
3	0.5000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000

Figure 4: The coin example for $\mathbf{y} = \{HHH, TTT, HHH, TTT\}$. The solution that EM reaches is intuitively correct: the coin-tosser has two coins, one which always shows up heads, the other which always shows tails, and is picking between them with equal probability ($\lambda = 0.5$). The posterior probabilities $\tilde{p}_i$ show that we are certain on coin 1 (tail-biased) generated y_2 and y_4, whereas coin 2 generated y_1 and y_3.



Iteration	λ	p_1	p_2	$\tilde{p}_1$	$\tilde{p}_2$	$\tilde{p}_3$	$\tilde{p}_4$
0	0.3000	0.7000	0.7000	0.3000	0.3000	0.3000	0.3000
1	0.3000	0.5000	0.5000	0.3000	0.3000	0.3000	0.3000
2	0.3000	0.5000	0.5000	0.3000	0.3000	0.3000	0.3000
3	0.3000	0.5000	0.5000	0.3000	0.3000	0.3000	0.3000
4	0.3000	0.5000	0.5000	0.3000	0.3000	0.3000	0.3000
5	0.3000	0.5000	0.5000	0.3000	0.3000	0.3000	0.3000
6	0.3000	0.5000	0.5000	0.3000	0.3000	0.3000	0.3000

Figure 5: The coin example for $\mathbf{y} = \{HHH, TTT, HHH, TTT\}$, with p_1 and p_2 initialized to the same value. EM is stuck at a saddle point.

On final note, when y is hidden, it is also likely that we do not have previous knowledge of the dimension of the alphabet space, or $|\mathcal{Y}|$, such as the number of topics in the Topic Model. The dimension value can be configured by optimizing the model on a development corpus.

Topic Model

Introduction to Topic Model

In this section, we will briefly introduce Topic Model. Consider a simple unigram model where the probability of a word w_i is given by $p(w_i) = \theta_i$. We will also model the topic z such that $z \in \mathcal{Z}$ and $|\mathcal{Z} = k|$, where $\mathcal{Z}$ is our universe of topics and the dimension k is treated as a hyperparameter. There will be a distribution over topics z given by

“Arts”	“Budgets”	“Children”	“Education”
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

Figure 6: For each given topic, some words are more likely to appear than others.

θ_z, where $\sum_{z \in \mathcal{Z}} \theta_z = 1$. Furthermore, each word in the document will be generated from the multinomial distribution $p(w_i|z) = \theta_{w_i|z}$, where $\sum_{i=1}^{|V|} \theta_{w_i|z} = 1$. Consider the following steps to generate a document

1. Sample a topic z, such that $z \sim \theta_z$
2. **For** $i = 1$ to N:
   Sample words w_i given the topic z, such that $w_i \sim \theta_{w_i|z}$

Then, the likelihood of the document d of size N can be expressed as

$$p(d|\boldsymbol{\theta}) = \sum_{z \in \mathcal{Z}} \theta_z \prod_{i=1}^{N} \theta_{w_i|z}$$

However, this model is problematic because it assumes that the whole document comes from a single topic. To attain a more flexible model, we should allow multiple topics in the same document. This motivation leads us to the *mixture model* where each word can select its own topic.

In this alternative approach, we setup a new model where each word w_i is sampled from each corresponding topic z_i. In other words, every word is a mixture of topics. The sampling procedure will now become

1. **For** $i = 1$ to N:
   Sample a topic z_i, such that $z_i \sim \theta_{z|d}$
   Sample a word w_i given the topic z_i, such that $w_i \sim \theta_{w_i|z}$

The likelihood for the new model is

$$p_(d|\boldsymbol{\theta}) = \prod_{i=1}^{N} \sum_{z \in \mathcal{Z}} \theta_{z|d} \theta_{w_i|z}$$

The distribution over the topics is unique for each document, represented by $\theta_{z|d}$. Across all documents, the word conditional on a given topic is sampled from a single "shared" distribution, represented by $\theta_{w|z}$. It is easy to see that if both the words and topics are fully observed, we can derive the MLE parameters by looking at the empirical distributions

$$\hat{\theta}_{w_i|z} = \frac{\text{count}(w_i, z)}{\text{count}(z)}$$
$$\hat{\theta}_{z|d} = \frac{\text{count}(z)}{N}$$

In application, however, topic information are not given out in the training data and the problem should be treated as a Hidden Variable Model. We leave it to the readers to think about how the EM algorithm can be applied to solve this specific unigram topic model.