CSE 446: Machine Learning **Lecture**

(An example of) The Expectation-Maximization (EM) Algorithm

Instructor: Sham Kakade

1 An example: the problem of document clustering/topic modeling

Suppose we have N documents $x_1, \dots x_n$. Each document is is of length T, and we only keep track of the word count in each document. Let us say $\text{Count}^{(n)}(w)$ is the number of times word w appeared in the n-th document.

We are interested in a "soft" grouping of the documents along with estimating a model for document generation. Let us start with a simple model.

2 A generative model for documents

For a moment, put aside the document clustering problem. Let us instead posit a (probabilistic) procedure which underlies how our documents were generated.

2.1 "Bag of words" model: a (single) topic model

Random variables: a "hidden" (or *latent* topic) $i \in \{1 \dots k\}$ and T-word outcomes $w_1, w_2, \dots w_T$ which take on some discrete values (these T outcomes constitute a document).

Parameters: the *mixing weights* $\pi_i = \Pr(\text{topic} = i)$, the *topics* $b_{wi} = \Pr(\text{word} = w|\text{topic} = i)$

The generative model for a T-word document, where every document is only about one topic, is specified as follows:

1. sample a topic i, which has probability π_i
2. gererate T words $w_1, w_2, \dots w_T$, independently. in particular, we choose word w_t as the t-th word with probability $b_{w_t i}$.

Note this generative model ignores the word order, so it is not a particularly faithful generative model.

Due to the 'graph' (i.e. the conditional independencies implied by the generative model procedure), we can write the *joint* probability of the outcome topic i occurring with a document containing the words $w_1, w_2, \dots w_T$ as:

$$\begin{aligned}
\Pr(\text{topic} = i \text{ and } w_1, w_2, \dots w_T) &= \Pr(\text{topic} = i) \Pr(w_1, w_2, \dots w_T|\text{topic} = i) \\
&= \Pr(\text{topic} = i) \Pr(w_1|\text{topic} = i) \Pr(w_2|\text{topic} = i) \Pr(w_T|\text{topic} = i) \\
&= \pi_i b_{w_1 i} b_{w_2 i} \dots b_{w_T i}
\end{aligned}$$

where the second to last step follows due to the fact that the words are generated independently given the topic i.

Inference

Suppose we were given a document with $w_1, w_2, \ldots w_T$. One *inference* question would be: what is the probability the underlying topic is i? By Bayes rule, we have:

$$\begin{aligned}\Pr(\text{topic} = i | w_1, w_2, \ldots w_T) &= \frac{1}{\Pr(w_1, w_2, \ldots w_T)} \Pr(\text{topic} = i \text{ and } w_1, w_2, \ldots w_T) \\ &= \frac{1}{Z} \pi_i b_{w_1 i} b_{w_2 i} \ldots b_{w_T i}\end{aligned}$$

where Z is a number chosen so that the probabilities sum to 1. Critically, note that Z is not a function of i.

2.2 Maximum Likelihood estimation

Given the N documents, we could estimate the parameters as follows:

$$\widehat{b}, \widehat{\pi} = \arg\min_{b,\pi} - \log \Pr(x_1, \ldots x_n | b, \pi)$$

How can we do this efficiently?

3 The Expectation Maximization algorithm (EM): By example

The EM algorithm is a general procedure to estimate the parameters in a model with latent (unobserved) factors. We present an example of the algorithm. *EM improves the log likelihood function at every step and will converge.* However, it may not converge to the global optima. Think of it as a more general (and probabilistic) adaptation of the K-means algorithm.

3.1 The algorithm: An example for the topic modeling case

The EM algorithm is an *alternating minimization* algorithm. We start at some initialization and then alternate between the E and M steps as follows:

Initialization:

Start with some guess $\widehat{b}$ and $\widehat{\pi}$ (where the guess is not "symmetric").

The E step:

Estimate the *posterior* probabilities, i.e. the soft assignments, of each document:

$$\widehat{Pr}(\text{topic } i | x_n) = \frac{1}{Z} \widehat{\pi}_i \, \widehat{b_{w_1 i}} \, \widehat{b_{w_2 i}} \, \ldots \, \widehat{b_{w_T i}}$$

The M step:

Note that $\text{Count}^{(n)}(w)/T$ is the empirical frequency of word w in the n-th document.

Given the power probabilities (which we can view as "soft" assignments), we go back and re-estimate the topic probabilities and the mixing weights as follows

$$\widehat{b}_{wi} = \frac{\sum_{n=1}^{N} \widehat{Pr}(\text{topic } i | x_n) \ \text{Count}^{(n)}(w)/T}{\sum_{n=1}^{N} \widehat{Pr}(\text{topic } i | x_n)}$$

and

$$\widehat{\pi}_i = \frac{1}{N} \sum_{n=1}^{N} \widehat{Pr}(\text{topic } i | x_n)$$

Now got back to the E-step.

3.2 (local) Convergence

For a general class of latent variable models — models which have unobserved random variables — we can say the following about EM:

- If the algorithm has not converged, then, after every M step, the negative log likelihood function decreases in value.
- The algorithm will converge in the limit (to some point, under mild assumptions). Unfortunately, this point may *not* be the global minima. This is related to the that the log likelihood objective function (for these latent variable models) is typically not convex.