

Notes on Expectation Maximization

Tan Yee Fan

2009 May 17

1 Expectation Maximization

Let x be an observed random variable and z be a hidden random variable, and x and z are jointly parameterized by θ . In other words, we are given a complete data model $P(x, z|\theta)$. In this problem, we would like to find the θ that maximizes $P(x|\theta) = \sum_z P(x, z|\theta)$, known as the *maximum likelihood estimate* (MLE) for θ . Typically, people work with the *log likelihood* $\log P(x|\theta)$ and the *complete log likelihood* $\log P(x, z|\theta)$ instead. Thus, the problem is equivalent to finding $\arg \max_{\theta} \log P(x|\theta)$. However, maximizing $\log P(x|\theta)$ directly may be intractable. The *expectation maximization* (EM) algorithm aims to overcome this difficulty by producing an estimate for θ in an iterative manner.

1.1 Derivation

From the fact $P(x|\theta) = \frac{P(x, z|\theta)}{P(z|x, \theta)}$, we take logarithms:

$$\log P(x|\theta) = \log P(x, z|\theta) - \log P(z|x, \theta)$$

Let $\theta^{(t)}$ be an estimate of θ . Multiply over $P(z|x, \theta^{(t)})$ and sum over z :

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

Now define $Q(\theta, \theta^{(t)})$:

$$Q(\theta, \theta^{(t)}) = E_{z|x, \theta^{(t)}}[\log P(x, z|\theta)] = \sum_z P(z|x, \theta^{(t)}) \log P(x, z|\theta)$$

We also note that $\sum_z P(z|x, \theta^{(t)}) \log P(x|\theta) = \log P(x|\theta) \sum_z P(z|x, \theta^{(t)}) = \log P(x|\theta)$, and hence we have:

$$\log P(x|\theta) = Q(\theta, \theta^{(t)}) - \sum_z P(z|x, \theta^{(t)}) \log P(z|x, \theta)$$

We now compute $\log P(x|\theta) - \log P(x|\theta^{(t)})$:

$$\log P(x|\theta) - \log P(x|\theta^{(t)}) = Q(\theta, \theta^{(t)}) - Q(\theta^{(t)}, \theta^{(t)}) + D_{KL}(P(z|x, \theta^{(t)}) || P(z|x, \theta))$$

where $D_{KL}(P(z|x, \theta^{(t)})||P(z|x, \theta)) = \sum_z P(z|x, \theta^{(t)}) \log \frac{P(z|x, \theta^{(t)})}{P(z|x, \theta)}$ is the Kullback-Leibler divergence between $P(z|x, \theta^{(t)})$ and $P(z|x, \theta)$ which is always nonnegative. This means that:

$$\log P(x|\theta) - \log P(x|\theta^{(t)}) \geq Q(\theta, \theta^{(t)}) - Q(\theta^{(t)}, \theta^{(t)})$$

Recall that we want to maximize $\log P(x|\theta)$. Since $\log P(x|\theta^{(t)})$ and $Q(\theta^{(t)}, \theta^{(t)})$ are constants, we choose the next estimate of θ to be

$$\theta^{(t+1)} = \arg \max_{\theta} Q(\theta, \theta^{(t)})$$

The above equation describes one iteration of the EM algorithm, with the aim of maximizing the expected log likelihood of the complete data relative to the probability distribution over the hidden variable.

1.2 Algorithm

The EM algorithm starts by choosing an initial value for θ_0 . Then, for $t = 0, 1, 2, \dots$, execute the following steps:

1. *Expectation step* (E-step): Compute

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

which is a distribution over θ .

2. *Maximization step* (M-step): Compute

$$\theta^{(t+1)} = \arg \max_{\theta} Q(\theta, \theta^{(t)})$$

From the derivation, it is guaranteed that $\log P(x|\theta^{(t+1)}) \geq \log P(x|\theta^{(t)})$. We stop the EM algorithm when it has converged, i.e., when the difference between $\log P(x|\theta^{(t+1)})$ and $\log P(x|\theta^{(t)})$ is sufficiently small, for which we take $\theta^{(t+1)}$ to be an estimate of $\arg \max_{\theta} P(x|\theta)$.

It is known that the EM algorithm will always converge to a stationary point of $\log P(x|\theta)$. This stationary point is usually a local maximum, but in some unusual cases, the EM algorithm can converge on a saddle point or even a local minimum. Therefore, the EM algorithm is typically executed multiple times, each with a random initialization for θ_0 . This increases the chance of finding the global maximum of $P(x|\theta)$.

In problems where it is difficult to compute the maximization in the M-step directly, we can modify the M-step to select a $\theta^{(t+1)}$ that satisfies $\log P(x|\theta^{(t+1)}) \geq \log P(x|\theta^{(t)})$. This modified form is known as the *generalized expectation maximization* (GEM) and is guaranteed to converge as well.

1.3 Alternate View

Note that the E-step involves computing the distribution R that satisfies $R(z|x) = P(z|x, \theta^{(t)})$. We note that when $R(z|x) = P(z|x, \theta^{(t)})$, we have $E_R[\log P(x, z|\theta)] = Q(\theta, \theta^{(t)})$. We now define the function

$$F(R, \theta) = E_R[\log P(x, z|\theta)] + H(R)$$

It can be shown that $F(R, \theta)$ can be rewritten as follows:

$$F(R, \theta) = -D_{KL}(R||P_\theta) + \log P(x|\theta)$$

where $P_\theta(z|x) = P(z|x, \theta)$. Note that if we hold θ constant, then $F(R, \theta)$ is maximized when $R = P_\theta$, and at this maximum, $F(R, \theta) = \log P(x|\theta)$. Thus, the EM algorithm is equivalent to the following:

1. E-step: Compute

$$R^{(t+1)} = \arg \max_R F(R, \theta^{(t)})$$

2. M-step: Compute

$$\theta^{(t+1)} = \arg \max_\theta F(R^{(t+1)}, \theta)$$

In this formulation, whenever $F(R, \theta)$ is a local maximum, $\log P(x|\theta)$ is also a local maximum. Also, whenever $F(R, \theta)$ is a global maximum, $\log P(x|\theta)$ is also a global maximum. Therefore, for GEM, it is sufficient to have the EM steps increase the function F .

1.4 Multiple Examples

The EM algorithm is often run when the random variable x consists of multiple examples, i.e., $x = (x_1, \dots, x_M)$, which are assumed to be independently and identically distributed. This can happen when we train a model using the EM algorithm. When multiple examples, we have:

$$P(x_1, \dots, x_M|\theta) = \prod_{i=1}^M P(x_i|\theta)$$

Hence, the log likelihood $\log P(x|\theta)$ becomes:

$$\log P(x_1, \dots, x_M|\theta) = \sum_{i=1}^M \log P(x_i|\theta)$$

and thus $Q(\theta, \theta^{(t)})$ becomes:

$$Q(\theta, \theta^{(t)}) = \sum_{i=1}^M \sum_z P(z|x_i, \theta^{(t)}) \log P(x_i, z|\theta)$$

2 Hidden Markov Model

A *Hidden Markov Model* (HMM) consists of N hidden states. The HMM starts in an initial state s . At each time step, it emits one observed symbol α from an output alphabet Σ and transitions into another state s' . It should be emphasized that the states of the HMM are hidden and only the output sequence is observed. The sequence of states a HMM is in over time is governed by initial state probabilities $P_\theta(s)$ and state transition probabilities $P_\theta(s'|s)$, and the output is governed by the symbol emission probabilities $P_\theta(\alpha|s)$. Therefore, the probability of observing an output sequence $\alpha_1, \dots, \alpha_T$ together with a state sequence s_1, \dots, s_T is

$$P_\theta(\alpha_1, \dots, \alpha_T, s_1, \dots, s_T) = P_\theta(s_1) \prod_{t=1}^T P_\theta(s_{t+1}|s_t) \prod_{t=1}^T P_\theta(\alpha_t|s_t)$$

The initial state probabilities $P_\theta(s)$, state transition probabilities $P_\theta(s'|s)$, and the symbol emission probabilities $P_\theta(\alpha|s)$ form the parameters θ of a HMM.

One use of the HMM is to recover the hidden state sequence when given an output sequence. Part-of-speech (POS) tagging is one such example, where each state is a POS tag, and each symbol in the output sequence is a token, which can be either a word or a punctuation.

Consider the task of training a HMM from a set of sequences x , whose corresponding states z are known. We count the following:

- $C(s)$, the number of times state s is the initial state.
- $C(s'|s)$, the number of times state s is followed by state s' .
- $C(\alpha|s)$, the number of times symbol α is emitted in state s .

Thus, the complete data is described by:

$$P(x, z|\theta) = \prod_s P_\theta(s)^{C(s)} \prod_{s, s'} P_\theta(s'|s)^{C(s'|s)} \prod_{s, \alpha} P_\theta(\alpha|s)^{C(\alpha|s)}$$

and the complete log likelihood is:

$$\log P(x, z|\theta) = \sum_s C(s) \log P_\theta(s) + \sum_{s, s'} C(s'|s) \log P_\theta(s'|s) + \sum_{s, \alpha} C(\alpha|s) \log P_\theta(\alpha|s)$$

The $Q(\theta, \theta^{(t)})$ function can be expressed by:

$$Q(\theta, \theta^{(t)}) = \sum_s \bar{C}_{\theta^{(t)}}(s) \log P_\theta(s) + \sum_{s, s'} \bar{C}_{\theta^{(t)}}(s'|s) \log P_\theta(s'|s) + \sum_{s, \alpha} \bar{C}_{\theta^{(t)}}(\alpha|s) \log P_\theta(\alpha|s)$$

where

$$\begin{aligned}\bar{C}_{\theta^{(t)}}(s) &= \sum_z P(z|x, \theta^{(t)})C(s) \\ \bar{C}_{\theta^{(t)}}(s'|s) &= \sum_z P(z|x, \theta^{(t)})C(s'|s) \\ \bar{C}_{\theta^{(t)}}(\alpha|s) &= \sum_z P(z|x, \theta^{(t)})C(\alpha|s)\end{aligned}$$

are the expected counts which can be efficiently computed using the forward and backward procedure. We maximize $Q(\theta, \theta^{(t)})$ with respect to θ to obtain $\theta^{(t+1)}$. This is a constrained optimization problem, and its solution for $\theta^{(t+1)}$ results in the following update equations:

$$\begin{aligned}P_{\theta^{(t+1)}}(s) &= \frac{\bar{C}_{\theta^{(t)}}(s)}{\sum_s \bar{C}_{\theta^{(t)}}(s)} \\ P_{\theta^{(t+1)}}(s'|s) &= \frac{\bar{C}_{\theta^{(t)}}(s'|s)}{\sum_{s'} \bar{C}_{\theta^{(t)}}(s'|s)} \\ P_{\theta^{(t+1)}}(\alpha|s) &= \frac{\bar{C}_{\theta^{(t)}}(\alpha|s)}{\sum_{\alpha} \bar{C}_{\theta^{(t)}}(\alpha|s)}\end{aligned}$$

In summary, the EM algorithm for training a HMM is as follows:

1. E-step: Compute the expected counts $\bar{C}_{\theta^{(t)}}(s)$, $\bar{C}_{\theta^{(t)}}(s'|s)$, and $\bar{C}_{\theta^{(t)}}(\alpha|s)$.
2. M-step: Compute $P_{\theta^{(t+1)}}(s)$, $P_{\theta^{(t+1)}}(s'|s)$, and $P_{\theta^{(t+1)}}(\alpha|s)$ using the update equations.

For POS tagging, empirical results have indicated that running the EM algorithm to convergence can lead to overfitting. As such, a separate validation set is used to stop the EM algorithm when the tagging accuracy starts to decrease. Typically, only a few iterations of the EM algorithm is needed to train the POS tagger.

References

- [Borman, 2004] Borman, S. (2004). The expectation maximization algorithm – a short tutorial. Available at http://www.seanborman.com/publications/EM_algorithm.pdf.
- [Neal and Hinton, 1999] Neal, R. and Hinton, G. E. (1999). A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, pages 355–368.
- [Rabiner, 1990] Rabiner, L. R. (1990). A tutorial on hidden markov models and selected applications in speech recognition. *Readings in speech recognition*, pages 267–296.