The Baum–Welch algorithm for estimating a Hidden Markov Model[i]

In this page we present a detailed description of the Baum-Welch algorithm as used in our recent work[ii].
A Hidden Markov Model is a probabilistic model of the joint probability of a collection of random variables $\{O_1,....O_T,Q_1,....Q_T\}$. The O_t variables are discrete observations and the Q_t variables are "hidden" and discrete. Under an HMM, there are two conditional independence assumptions made about these random variables that make associated algorithms tractable. These independence assumptions are:

1. the t^{th} hidden variable, given the $(t-1)^{st}$ hidden variable, is independent of previous variables, or: $P(Q_t \mid Q_{t-1}, O_{t-1}, ..., Q_1, O_1) = P(Q_t \mid Q_{t-1})$.
2. the t^{th} observation depends only on the t^{th} state. $P(O_t \mid Q_t, O_t, ..., Q_1, O_1) = P(O_t \mid Q_t)$.

In the following, we present the EM algorithm for finding the maximum-likelihood estimate of the
parameters of a hidden Markov model given a set of observed feature vectors. This algorithm is also known as the Baum-Welch algorithm.

Q_t is a discrete random variable with N possible values $\{1....N\}$. We further assume that the underlying "hidden" Markov chain defined by $P(Q_t \mid Q_{t-1})$ is time-homogeneous (i.e., is independent of the time t). Therefore, we can represent $P(Q_t \mid Q_{t-1})$ as a time-independent stochastic transition matrix $A=\{a_{ij}\}=p(Q_t=j|Q_{t-1}=i\}$. The special case of time t=1 is described by the initial state distribution $\pi_i=P(Q_1=i)$. We say that we are in state j at time t if $Q_t = j$. A particular sequence of states is described by $q = (q_1 \ldots q_T)$ where $q_t \in \{1 \ldots N\}$ is the state at time t.

The observation is one of L possible observation symbols, $O_t \in \{o_1 \ldots o_L\}$. The probability of a particular observation vector at a particular time t for state j is described by: $b_j(o_t) = p(O_t = o_t|Q_t = j)$. ($B=\{b_{ij}\}$ is an L by N matrix). A particular observation sequence O is described as $O = (O_1 = o_1, \ , \ , \ O_T = o_T)$.

Therefore, we can describe a HMM by: $\lambda = (A, B, \pi)$. Given an observation O, the Baum-Welch algorithm finds: $\lambda^* = \max_\lambda P(O \mid \lambda)$ - that is, the HMM λ, that maximizes the probability of the observation O.

The Baum-Welch algorithm

Initialization: set $\lambda = (A, B, \pi)$ with random initial conditions. The algorithm updates the parameters of λ iteratively until convergence, following the procedure below:

The forward procedure: We define: $\alpha_i(t) = p(O_1 = o_1, \ , \ , O_t = o_t, Q_t = i| \lambda)$, which is the probability of seeing the partial sequence $o_1, , , o_t$ and ending up in state i at time t.

We can efficiently calculate $\alpha_i(t)$ recursively as:

1. $\alpha_i(t) = \pi_i b_i(o_1)$

2. $\alpha_j(t+1) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_i(t) \cdot a_{ij}$

The backward procedure: This is the probability of the ending partial sequence $o_{t+1}, , , o_T$ given that we started at state *i*, at time *t*. We can efficiently calculate $\beta_i(t)$ as:

1. $\beta_i(T) = 1$
2. $\beta_i(t) = \sum_{j=1}^{N} \beta_j(t+1) a_{ij} b_j(o_{t+1})$

using α and β, we can calculate the following variables:

$$\gamma_i(t) \equiv p(Q_t = i \mid O, \lambda) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t)\beta_j(t)}$$

$$\xi_{ij}(t) \equiv p(Q_t = i, Q_{t+1} = j \mid O, \lambda) = \frac{\alpha_i(t) a_{ij} \beta_j(t+1) b_j(o_{t+1})}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i(t) a_{ij} \beta_j(t+1) b_j(o_{t+1})}$$

having γ and ξ , one can define update rules as follows:

$$\overline{\pi}_i = \gamma_i(1)$$

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

$$\overline{b}_i(k) = \frac{\sum_{t=1}^{T} \delta_{O_t, o_k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

(note that the summation in the nominator of $\overline{b}_i(k)$ is only over observed symbols equal to o_k). Using the updated values of *A, B and* π, a new iteration is preformed until convergence.

---

[i]) The above brief description is based on "A gentle tutorial of the EM algorithm and its application to Parameter Estimation for gaussian mixture and Hidden Markov Models", J. A. Bilmes (http://www.vision.ethz.ch/ml/slides/em_tutorial.pdf)

[ii]) cond-mat/0402546, cond-mat/0308308