HMM algorithm

1 Backward

$P(Y_{k+1},...,Y_n|X_k = C) = \sum_q P(P(Y_{k+2},...,Y_n|t_{k+1} = q)P(q|C)P(x_{k+1}|q)$

Define:

$\beta_k(C) = P(Y_{k+1},...,Y_n|X_k = C)$

Inductive step:

$\beta_k(C) = \sum_q \beta_{k+1}(q)P(q|C)P(Y_{k+1}|q)$

2 Forward

$P(Y_1,...,Yn) = \sum_t(\prod_i P(Y_i|X_i)P(X_i|X_{i-1})$

$P(Y_1,...,Y_k,X_k = q) = \sum_{q1} P(Y_1,...,Y_k,X_{k-1} = q1,X_k = q) = \sum_{q1} P(Y1,...,Y_{k-1},X_{k-1} = q1)P(X_k = q|X_{k-1} = q1)P(Y_k|X_k = q)$

Define:

$\alpha_k(q) = P(Y_1,...,Y_k,X_k = q)$

$\alpha_1(q) = p(Y_1,X_1 = q) = P(X_1 = q|X_0)p(Y_1|X_1 = q)$

Inductive step:

$\alpha_k(q) = \sum_{q1} \alpha_{k-1}(q1)P(X_k = q|X_{k-1} = q1)P(Y_k|X_k = q)$

3 Viterbi

$max_{X_1,...X_k}P(X_1,...X_k,Y1,...,Y_k) = max_q max_{X_1,...X_{k-1}}P(X_1,...X_{k-1},X_k = q,Y1,...,Y_k)$

Define:

$\theta_k(q) = max_q max_{X_1,...X_{k-1}}P(X_1,...X_{k-1},X_k = q,Y1,...,Y_k)$

Inductive step:

$\theta_k(q) = max_{q1}\theta_{k-1}(q1)P(X_k = q|X_{k-1} = q1)P(Y_k|X_k = q)$

4 Baum Welch

E step:

$\gamma_t(j) = \alpha_t(j)\beta_t(j)/\alpha_T(q_F)$

$\xi_t(i,j) = \alpha_t(i)T_{ij}O_{j,O_{t+1}}\beta_{t+1}(j)/\alpha_T(q_F)$

M step:

$ahat_{ij} = \sum_{t=1}^{T-1} \xi_t(i,j)/\sum_{t=1}^{T-1}\sum_{k=1}^{N} \xi_t(i,k)$

$bhat_j(v_k) = \sum_{t=1 s.t. O_t = v_k}^{T} \gamma_t(j) / \sum_{t=1}^{T} \xi_t(j)$

5 Loss function for Baum Welch

EM optimize the log likelihood of the input:

$$\text{logP}(\text{Y}_1, ..., Y_k|T, O, pi)$$
$$= log \sum_{X_1,...X_k} P(X_1, ...X_k, Y_1, ..., Y_k|T, O, pi)$$
$$= log \sum_{X_1,...X_k} \Pi_{i=1}^{n} P(X_i|X_{i-1})P(Y_i|X_i)$$

If I use ω to represent $Y_1, ..., Y_k$, and t to represent $X_1, ..., X_k$, and λ to represent the model parameter.

Then according to Jensen's inequality,

$$\log \sum_t (\omega, t|\lambda)$$
$$= log \sum_t (P(\omega, t|\lambda)/P(t|\omega, \lambda^{(s)}))P(t|\omega, \lambda^{(s)})$$
$$\geq \sum_t P(t|\omega, \lambda^{(s)}) log(P(\omega, t|\lambda)/P(t|\omega, \lambda^{(s)}))$$

Define

$f(\lambda) = log \sum_t P(\omega, t|\lambda)$

$g_s(\lambda) = \sum_t P(t|\omega, \lambda^{(s)}) logP((\omega, t|\lambda)/P(t|\omega, \lambda^{(s)}))$

So we have $f(\lambda) \geq g_s(\lambda)$

Optimizing $g_s(\lambda)$

$$\text{g}_s(\lambda)$$
$$= \sum_t P(t|\omega, \lambda^{(s)}) logP(\omega, t|\lambda)/P(t|\omega, \lambda^{(s)})$$
$$= \sum_t P(t|\omega, \lambda^{(s)})(logP(\omega, t|\lambda) - P(t|\omega, \lambda^{(s)}))$$

$$\text{max}_\lambda g_s(\lambda)$$
$$= max_\lambda \sum_t P(t|\omega, \lambda^{(s)}) logP(\omega, t|\lambda)$$
$$= max_\lambda \sum_t P(t|\omega, \lambda^{(s)}) \sum_i (logP(t_i|t_{i-1}) + logP(\omega_t|t_i))$$

In a word, we can minimize the negative log loss (or maximize the log likelihood $max_\lambda P(w|\lambda)$) by maximizing $g_s(\lambda) = \sum_t P(t|w, \lambda^{(s)}) logP(w, t|\lambda)$ at iteration s.

$max_\lambda g_s(\lambda)$ has a closed-form solution such that the parameters can be directly calcuated by the expected counts.