Bioinformatics – Lecture Notes

Announcements

Remember: Project Proposals are due this Thursday April 11. I have only heard from one group so far.

Only 8 classes left including this lecture and project presentations.

Class 23 – April 9, 2002

B. Parameter Estimation for Hidden Markov Models

1) Baum-Welch Method

This algorithm calculates *A_{kl}* and *$E_k(b)$* as the expected number of times each transition or emission is used in the training sequences. This is done using the same forward and backward values as the posterior probability decoding method. The probability that a_{kl} is used at position i in the sequence x is as follows:

$$\Pr[p_i = k, p_{i+1} = l \mid x, q] = \frac{f_k(i) a_{kl} e_l(x_{i+1}) b_l(i+1))}{\Pr[x]}$$

where q is the entire current set of values of the parameters in the model. This can be used to find the expected number of time that a_{kl} is used by summing over all positions and over all training sequences:

$$A_{kl} = \sum_j \frac{f_k^j(i) a_{kl} e_l(x_{i+1}^j) b_l^j(i+1)}{\Pr[x^j]}$$

The expected number of times b appears in state k is given by

$$E_k(b) = \sum_{i \mid x_i^j = b} \frac{f_k^j(i) b_k^j(i)}{\Pr[x^j]}$$

where this sum is over only the positions i where b is emitted.

We can compute the maximum likelihood estimators for the model parameters a_{kl} and $e_k(b)$ by

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \quad \text{and} \quad e_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')}$$

We iterate back and forth between the values of the A's and E's and the a's and e's until they converge on a value. To do this, a convergence criterion is chosen. The criterion typically used is when the change in the log likelihood is sufficiently small. Other criteria are possible.

An algorithm for this is given on page 64 of Durbin et al. As we mentioned earlier, Viterbi training, which is a modification of Baum-Welch, can also be used to estimate parameters. The *Viterbi Score* is the same as the Baum Welch Score except with the maximum in place of sum over all paths of states.

Example – The occasionally dishonest casino, part 5 (see page 65 Durbin et al.)

2) Expectation Maximization (EM)

Consider that we have a set of observations that forms a set *Y* of *incomplete data.* Then there is a set *X* of *complete data* that consists of the observed and hidden data. Let $y \in Y$ be an sequence of observations *O* of length *T*.

Assume that there is $x \in X$ that are mapped to y in a *many-to-one* mapping. x consists of *p*, *O*, where $p \in Q^T$ is a sequence of state of length *T*. Let M denote a model and suppose that *g(y|M)* is a conditional probability density for the space *Y*, that *f(x|M)* is a conditional probability density of *x* and that

$$g(y \mid M) = \int_{x \in X(y)} f(x \mid M)$$

where *X(y)* is the set of $x \in X$ that are mapped to *y*.

Question: Is *y* complete or incomplete data? What about *x*?

In the discrete case, we have analogous conditional probabilities with

$$g(y \mid M) = \sum_{x \in X(y)} f(x \mid M)$$

Define

$$\begin{aligned} Q(\tilde{M} \mid M) &= E[\log\ f(x \mid \tilde{M}) \mid y, M] \\ &= \sum_{x \in X(y)} \Pr[x \mid y, M] \log\ f(x \mid \tilde{M}) \end{aligned}$$

The EM algorithm consists of two steps

i) Compute $Q(\tilde{M} \mid M)$.

ii) Determine $\arg\max_{\tilde{M}} Q(\tilde{M} \mid M)$

3) Baldi-Chauvin Gradient Descents

The gradient descent equations on the negative log likelihood can be derived by first reparameterizing the hidden Markov model using normalized exponentials:

$$a_{i,j} = \frac{e^{lw_{i,j}}}{\sum_k e^{lw_{i,k}}} \qquad \text{and} \qquad b_{i,c} = \frac{e^{ln_{i,c}}}{\sum_k e^{ln_{i,k}}}$$

This reparametrization has two advantages: i) modification of the ω's automatically preserves normalization constraints on emission and transition distributions and ii) transition and emission probabilities can never reach the value of 0. For the emission parameters

$$\frac{\partial b_{i,c}}{\partial w_{i,c}} = b_{i,c}(1 - b_{i,c}) \qquad \text{and} \qquad \frac{\partial b_{i,c}}{\partial w_{i,k}} = -b_{i,c} b_{i,k}$$

and similarly for the transitions parameters. By the chain rule

$$\frac{\partial \log \Pr[O \mid w]}{w_{i,c}} = \sum_k \frac{\partial \log \Pr[O \mid w]}{b_{i,k}} \frac{\partial b_{i,k}}{w_{i,c}}$$

Applying Lagrange multipliers we can optimize for the parameters to give the maximum likelihood, which yields the negative log likelihood gradient descent equations

$$\Delta w_{i,c} = h(n_{i,c} - n_i b_{i,c}) \quad \text{and} \qquad \Delta w_{j,i} = h(n_{j,i} - n_i a_{j,i})$$

where η is the learning rate, $n_{i,c}$ and $n_{j,i}$ are the expected counts derived from the forward-backward procedure for each single sequence ie.

$$n_{i,c} = \sum_p n(i, c, p, O) Q(p)$$

with $n(i,c,p,O)$ being the number of times letter c is emitted from i given π and Q and

$$n_i = \sum_p n(i,\boldsymbol{p},O)\,\mathrm{Pr}[\boldsymbol{p}\,|\,O,\boldsymbol{w}]$$

with $n(i,\boldsymbol{p},O)$ being the number of times i is visited given π and Q and

4) Mamitsuka's MA Algorithm

The likelihood of a the sth sequence with respect to a hidden Markov model is given by

$$p_s = \mathrm{Pr}[O^s\,|\,M]$$

and is called the *target value likelihood* of the sth sequence. Define

$$d_s = \log\left(\frac{p_s^*}{p_s}\right)$$

$$d_{\max} = \log\left(\frac{p_{\max}^*}{p_{\min}^*}\right)$$

where $p_{\max}^*$ and $p_{\min}^*$ are the maximum and minimum of the p_s^*, respectively.

The goal of this method is to minimize the *error distance* given by

$$\sum -\log\left(\frac{d_{\max}^2 - d_s^2}{d_{\max}^2}\right)$$

during parameter fitting because d_s should be close to 0 after training.

C. Applications

a) Multiple sequence alignment

Hidden Markov models have been used for multiple sequence alignment. To do this, build a linear hidden Markov model with 3n states where n is the average sequence length in the training set. The training set of sequences to be aligned is treated as a collection of observation sequences. There are n states for each letter in the sequence along with two additional states for insertions and deletions. Once the model is trained, each sequence from the training set can be scored using the Viterbi algorithm which aligns the sequence against the stochastic model.

Figure 5.3 (Clote and Backofen) shows such a model. There are two backbone end states that have no emissions. Insertion states and the other backbone states have emissions. The deletion states have no emissions.

Example – If we want to align GGCT, ACCGAT, and CT we get the followingViterbi path after using the Baum-Welch algorithm to determine the parameters:

GGCT	m0, m1, m2, m3, m4, m5
ACCGAT	m0, i0, m1, d2, m3, i3, i3, m4, m5
CT	mo, m1, d2, d3, m4, m5

This yields the following alignment

```
A C – C g a T
.  G G C .  . T
.  C –  –  .  . T
```

b) Protein motifs

Mamitsuka studied the problem of aligning sugar transport proteins that had a certain consensus sequence. There were 49 sugar transport proteins with this motif and 19 non-sugar transport proteins with this motif obtained from Swiss-Prot database. The hidden Markov model is shown in Fig. 5.4 (Clote and Backofen). He applied several methods including his own.

c) Eukaryotic DNA promoter regions

Hidden Markov models have been used to recognize promotor sites in eucaryotic DNA. (See Clote and Backofen)