Marginal model likelihood for standard HMM
(independent base transitions)

Darren Wilkinson

Definitions

Data $y^t = (y_1, \ldots, y_t)$, $y^{[t]} = (y_{t+1}, \ldots, y_n)$, $y = (y^t, y^{[t]})$, Λ is $r \times r$ transition matrix for S, with ith column $\Lambda^{(k)}$. In state k, emission probabilities are 4×1 vectors $P^{(k)}$ with elements $p_i^{(k)}$. P is the $4 \times r$ matrix $P = (P^{(1)}, \ldots, P^{(n)})$. The ith row of P is the $1 \times r$ vector P_i.

Basic idea

Define $l_k(t) = \mathrm{P}(S_t = k, y^t)$, and $l(t) = (l_1(t), \ldots, l_r(t))$. Initialise using $l(0) = \pi$, where π is the equilibrium distribution of the hidden states (a solution to $\pi\Lambda = \pi$).

Now suppose we know $l(0), \ldots, l(t-1)$ and want to know $l(t)$.

$$\begin{aligned} l_k(t) &= \mathrm{P}\big(S_t = k, y^t\big) \\ &= \mathrm{P}\big(S_t = k, y^{t-1}\big)\ \mathrm{P}\big(y_t | S_t = k, y^{t-1}\big) \\ &= \mathrm{P}\big(S_t = k, y^{t-1}\big)\, p_{y_t}^{(k)} \\ &= p_{y_t}^{(k)} \sum_{j=1}^{r} \mathrm{P}\big(S_t = k, y^{t-1} | S_{t-1} = j, y^{t-1}\big)\ \mathrm{P}\big(S_{t-1} = j, y^{t-1}\big) \\ &= p_{y_t}^{(k)} \sum_{j=1}^{r} \lambda_{jk} l_j(t-1) \\ &= p_{y_t}^{(k)} [l(t-1)\Lambda^{(k)}] \end{aligned}$$

So,

$$l(t) = P_{y_t} \star [l(t-1)\Lambda],$$

where $\star$ denotes element-wise product.

So the marginal model likelihood is just

$$\mathrm{P}(y) = \sum_{k=1}^{r} \mathrm{P}(S_n = k, y) = \sum_{k=1}^{r} l_k(n),$$

ie. the sum of the elements of the vector $l(n)$.

Note that there are numerical issues in computing this for large sequences, due to numerical underflow.

Numerically stable computation

In fact it is easy to adapt the above idea in order to compute the log of the marginal likelihood by utilising the similarity between the likelihood terms $l(t)$ and the filtered probabilities $f(t)$ (which are numerically stable to compute, due to the normalisation which occurs at each step).
Recall that the filtered probabilities are defined as

$$f(t) = \mathrm{P}\big(S_t = k|y^t\big)$$

and are initialised as $f(0) = \pi = l(0)$. Now using the argument given in the notes (or a simple adaptation of the argument for $l(t)$), these can be computed sequentially using the recursion

$$f(t) \propto P_{y_t} \star [f(t-1)\Lambda].$$

We can explicitly include the normalisation constant to get

$$f(t) = \frac{P_{y_t} \star [f(t-1)\Lambda]}{\mathrm{sum}(P_{y_t} \star [f(t-1)\Lambda])},$$

where $\mathrm{sum}(\cdot)$ denotes the sum of the elements of the vector (equivalent to the 1-norm of the vector in the case of non-negative vectors such as these). It is helpful to define

$$\xi_t = \mathrm{sum}(P_{y_t} \star [f(t-1)\Lambda]),$$

to get

$$f(t) = \frac{P_{y_t} \star [f(t-1)\Lambda]}{\xi_t}.$$

Now it is clear that computing $f(t)$ and $l(t)$ is the same apart from an additional factor of ξ_t at each step. Therefore $f(n)$ and $l(n)$ are related by

$$l(n) = f(n) \prod_{i=1}^{n} \xi_i.$$

So the marginal likelihood is given by

$$\begin{aligned}\mathrm{P}(y) &= \sum_{k=1}^{r} l_k(n) = \sum_{k=1}^{r} f_k(n) \prod_{i=1}^{n} \xi_i \\ &= \left[\prod_{i=1}^{n} \xi_i\right] \sum_{k=1}^{r} f_k(n) = \prod_{i=1}^{n} \xi_i\end{aligned}$$

Now like all likelihood computations, this will lead to numerical underflow if computed directly, but the log-likelihood can be computed stably as

$$\log\ \mathrm{P}(y) = \sum_{i=1}^{n} \log \xi_i.$$

So the simplest way to compute the log marginal likelihood in a numerically stable way is to compute the filtered probabilities in the usual way but at each stage accumulate the log of the normalisation constant.
It is also worth noting that this can be computed in a “read once” fashion, without using any data storage of dimension n. Therefore, likelihood computations can be carried out on huge sequences such as whole-genomes using this technique.