Expectation Maximization

Fast probabilistic optimization

I. Holmes

Department of Bioengineering
University of California, Berkeley

Spring semester

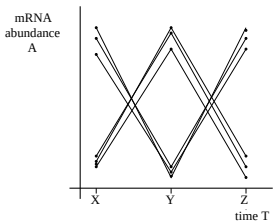
Outline

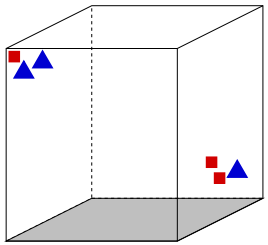
The K-means algorithm

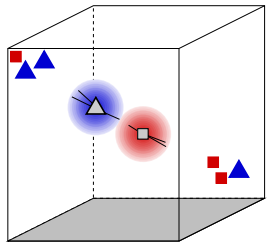
Example of iterative re-estimation: K-means algorithm.

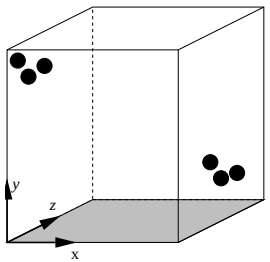
- N points $\{\mathbf{y}^{(i)}\}$ in D dimensions; K clusters; point i has cluster label $x^{(i)}$
- Start by randomising all $x^{(i)}$
- Re-estimate cluster centroids; set each $x^{(i)}$ to closest cluster; iterate to convergence
- K-means is commonly used for clustering, e.g. (in bioinformatics) microarray data
- We will see that it's very similar to EM on a mixture-of-Gaussians model

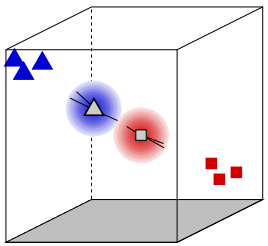
The *K*-means algorithm (visual)

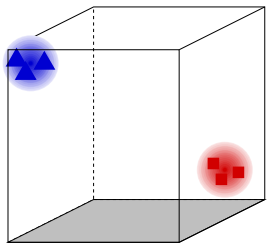












Essence of the K-means algorithm

Alternate between two steps

- Estimate *missing data* (cluster assignments)
- Estimate *model parameters* (cluster centroids)

As we will see, this is very close in essence to EM.

Variations on the K-means algorithm

- "K-medians": use cluster medians instead of centroids (a bit more stable)
- "Soft K-means": allow dataset$\rightarrow$cluster assignments to be probabilistic ("soft"), rather than deterministic ("hard"); centroids are then probability-weighted averages

We will see that soft K-means is, in fact, EM on a particular model.

What is the EM algorithm for?

EM (Expectation-Maximization) is a very broad family of algorithms for finding a maximum-likelihood point estimate of some model parameters

$$\begin{aligned} \hat{\theta} &= \text{argmax}_\theta P(Y|\theta) \\ &= \text{argmax}_\theta \sum_X P(X, Y|\theta) \end{aligned}$$

where

- X represents missing data (unknown, to be summed out)
- Y represents observed data

Characteristics of EM

- Algorithm is iterative
  - Guaranteed to converge
  - Convergence often rapid at first, then slow
- Works for a lot of models (but not all)
- Missing data (X) can often be summarized by counts
- Many generalizations (approximate, stochastic, etc.)

Formal statement of the EM algorithm

$$\begin{aligned}
\theta^{(n+1)} &= \text{argmax}_\theta \, \mathcal{E}(\theta|\theta^{(n)}) \\
\mathcal{E}(\theta|\theta^{(n)}) &= \sum_x P(X|Y,\theta^{(n)}) \log P(X,Y|\theta) \\
&= \langle \log P(X,Y|\theta) \rangle_{P(X|Y,\theta^{(n)})}
\end{aligned}$$

- Find posterior of missing data, $P(X|Y,\theta^{(n)})$
- Maximize expected log-likelihood under this posterior

Example: two-coin mixture

- I have two coins, each of which has probability p_k of returning heads and $q_k = 1 - p_k$ of tails ($k \in \{1, 2\}$)
- We perform E experiments.
- In the e'th experiment, I pick one of the coins and flip it F times, yielding y_e heads and $z_e = F - y_e$ tails
- Let x_e be the coin I picked for the e'th experiment.
- Missing data: $X = (x_1 \ldots x_E)$
- Observed data: $Y = (y_1 \ldots y_E)$
- Parameters: $\theta = (p_1, p_2)$

$$P(X, Y|\theta) = \prod_{e=1}^{E} \frac{1}{2} \begin{pmatrix} F \\ y_e \end{pmatrix} p_{x_e}^{y_e} q_{x_e}^{z_e}$$

EM algorithm for two-coin mixture

First calculate the posterior distribution over $X|Y$ for given θ

$$\begin{aligned} P(x_e = x, y_e = y|\theta) &= \frac{1}{2}\binom{F}{y} p_x^y q_x^z \\ P(y_e = y|\theta) &= \sum_{x\in\{1,2\}} \frac{1}{2}\binom{F}{y} p_x^y q_x^z \\ P(x_e = x|y_e = y, \theta) &= P(x_e = x, y_e = y|\theta) \;/\; P(y_e = y|\theta) \\ &= W_{e,x} \end{aligned}$$

where $z = F - y$

EM algorithm for two-coin mixture

Next write down the expected log-likelihood

$$\begin{aligned}\mathcal{E}(\theta|\theta^{(n)}) &= \sum_X P(X|Y,\theta^{(n)}) \log P(X,Y|\theta) \\ &= \sum_e \sum_{x\in\{1,2\}} W_{e,x}\,(y_e \log p_x + z_e \log q_x + K)\end{aligned}$$

where K is independent of θ (it includes the binomial coefficient and the factor of $1/2$ corresponding to $P(x_e)$) — we can drop this term

EM algorithm for two-coin mixture

Here's a longer derivation of that

$$
\begin{aligned}
&\mathcal{E}(\theta|\theta^{(n)}) \\
&\quad = \sum_X P(X|Y,\theta^{(n)}) \log P(X,Y|\theta) \\
&\quad = \sum_X P(X|Y,\theta^{(n)}) \sum_e \log P(x_e,y_e|\theta) \\
&\quad = \sum_e \sum_{x\in\{1,2\}} P(x_e = x|y_e,\theta^{(n)}) \log P(x_e = x,y_e|\theta) \\
&\quad = \sum_e \sum_{x\in\{1,2\}} W_{e,x}\,(y_e \log p_x + z_e \log q_x + K)
\end{aligned}
$$

EM algorithm for two-coin mixture

$$\mathcal{E}(\theta|\theta^{(n)}) = \sum_{x\in\{1,2\}} \left[\left(\sum_e W_{e,x} y_e \right) \log p_x + \left(\sum_e W_{e,x} z_e \right) \log q_x \right]$$

- Experiment e yielded y_e heads and z_e tails
- Posterior probability that experiment e used coin x is $W_{e,x}$
- For coin x, expected counts are $h_x = \sum_e W_{e,x} y_e$ heads, $t_x = \sum_e W_{e,x} z_e$ tails, $f_x = h_x + t_x = \sum_e W_{e,x} F$ total flips

Subject to probabilistic constraints on the p_x the EM update is

$$p_x \leftarrow \frac{h_x}{f_x}$$

Proof/derivation of EM

Following Anders Krogh (Durbin, Krogh *et al*, chapter 11.6)

- Suppose we have some estimate $\theta^{(n)}$ and we want to choose $\theta^{(n+1)}$ such that $P(y|\theta^{(n+1)}) \geq P(y|\theta^{(n)})$
- Since $P(x, y|\theta) = P(y|\theta)P(x|y, \theta)$ we can write $\log P(y|\theta) = \log P(x, y|\theta) - \log P(x|y, \theta)$ for some θ
- Multiplying by $P(x|y, \theta^{(n)})$ and summing over x gives

$$\log P(y|\theta) = \sum_x P(x|y, \theta^{(n)}) \log P(x, y|\theta) - \sum_x P(x|y, \theta^{(n)}) \log P(x|y, \theta)$$

Derivation of EM

- Let first term on RHS be $\mathcal{E}(\theta|\theta^{(n)}) = \sum_x P(x|y,\theta^{(n)}) \log P(x,y|\theta)$. Then

$$\log P(y|\theta) = \mathcal{E}(\theta|\theta^{(n)}) - \sum_x P(x|y,\theta^{(n)}) \log P(x|y,\theta)$$

$$\log P(y|\theta^{(n)}) = \mathcal{E}(\theta^{(n)}|\theta^{(n)}) - \sum_x P(x|y,\theta^{(n)}) \log P(x|y,\theta^{(n)})$$

- Subtracting gives

$$\log P(y|\theta) - \log P(y|\theta^{(n)})$$

$$= \mathcal{E}(\theta|\theta^{(n)}) - \mathcal{E}(\theta^{(n)}|\theta^{(n)}) + \sum_x P(x|y,\theta^{(n)}) \log \frac{P(x|y,\theta^{(n)})}{P(x|y,\theta)}$$

$$= \mathcal{E}(\theta|\theta^{(n)}) - \mathcal{E}(\theta^{(n)}|\theta^{(n)}) + D\left(P(x|y,\theta^{(n)})||P(x|y,\theta)\right)$$

Proof of convergence of EM

$$\begin{aligned}\log P(y|\theta) - \log P(y|\theta^{(n)}) \\ = \quad \mathcal{E}(\theta|\theta^{(n)}) - \mathcal{E}(\theta^{(n)}|\theta^{(n)}) + D\left(P(x|y,\theta^{(n)})||P(x|y,\theta)\right)\end{aligned}$$

- Since last term on RHS is always ≥ 0, we have $P(y|\theta^{(n+1)}) \geq P(y|\theta^{(n)})$ as long as

$$\theta^{(n+1)} = \text{argmax}_\theta \, \mathcal{E}(\theta|\theta^{(n)})$$

- If $\theta^{(n+1)} = \theta^{(n)}$, then a maximum has been reached and so $P(y|\theta^{(n+1)}) = P(y|\theta^{(n)})$

Interpretation

- Again, $\mathcal{E}(\theta|\theta^{(n)})$ is the **expected joint log-likelihood of the missing data** x **and observed data** y **as a function of** θ, with the expectation taken over the posterior distribution of x as estimated using $\theta^{(n)}$
- Computing $P(x|y,\theta^{(n)})$, or statistics that are sufficient to summarize this distribution, is called the **E-step** of EM.
- Computing $\theta^{(n+1)} = \text{argmax}_\theta \mathcal{E}(\theta|\theta^{(n)})$ is the **M-step** of EM.

Expected counts

Note if $P(x, y|\theta)$ has the form $\prod_i \theta_i^{x_i}$, where x_i is the number of times an event occurs and θ_i is the probability of that event, then $\mathcal{E}(\theta|\theta^{(n)})$ has the form $\sum_i \langle x_i \rangle_{P(x|y,\theta^{(n)})} \log \theta_i$

- $\mathcal{E}$ typically involves **expected counts** $\langle x_i \rangle$ for missing data
  - In this case, EM is making use of the first derivatives:

$$\begin{aligned}
\frac{\partial(\log P(y|\theta))}{\partial(\log \theta_i)} &= \frac{\theta_i}{P(y|\theta)} \frac{\partial P(y|\theta)}{\partial \theta_i} \\
&= \frac{\theta_i}{P(y|\theta)} \frac{\partial}{\partial \theta_i} \sum_x P(x, y|\theta) \\
&= \frac{\theta_i}{P(y|\theta)} \sum_x \left(\frac{x_i}{\theta_i} \right) P(x, y|\theta) \\
&= \frac{\theta_i}{P(y|\theta)} \sum_x \left(\frac{x_i}{\theta_i} \right) P(y|\theta) P(x|y, \theta) = \langle x_i \rangle_{P(x|y,\theta)}
\end{aligned}$$

Expected wait times

- Likewise if $P(x, y|\theta)$ contains terms of the form $\exp(-\theta_i x_i)$, where θ_i is an event rate and x_i is the time that elapses before the event occurs, the corresponding terms in $\mathcal{E}$ will be $-\theta_i \langle x_i \rangle$ involving the **expected wait times** $\langle x_i \rangle$

Neal and Hinton's variational view of EM

- Neal & Hinton (Learning in Graphical Models, Jordan, 1998)
  - Let $\tilde{P}(x)$ be a probability distribution over the missing data and let $H(\tilde{P})$ be the entropy of $\tilde{P}$. Define

$$\begin{aligned} F(\theta, \tilde{P}) &= \langle \log P(x, y|\theta) \rangle_{\tilde{P}} + H(\tilde{P}) \\ &= \sum_x \tilde{P}(x) \left(\log P(x, y|\theta) - \log \tilde{P}(x) \right) \\ &= \sum_x \tilde{P}(x) \left(\log P(y|\theta) + \log P(x|y, \theta) - \log \tilde{P}(x) \right) \\ &= \log P(y|\theta) - D\left(\tilde{P}(x) || P(x|y, \theta) \right) \end{aligned}$$

  where $D(\tilde{P}(x)||P(x, y, \theta))$ is the relative entropy.

Neal and Hinton's variational view of EM

$$F(\theta, \tilde{P}) = \langle \log P(x, y|\theta) \rangle_{\tilde{P}} + H(\tilde{P}) = \log P(y|\theta) - D\left(\tilde{P}(x) || P(x|y, \theta)\right)$$

- Suppose we fix $\theta = \theta^{(n)}$ and maximise $F(\theta^{(n)}, \tilde{P})$ w.r.t. $\tilde{P}$. Then the latter expression for F shows that the maximum is at $\tilde{P}(x) = P(x|y, \theta^{(n)})$ (due to Gibbs' inequality). This is the E-step of EM.
- If we then fix $\tilde{P}$ at this value, we have $F(\theta, \tilde{P}) = F(\theta, P(x|y, \theta^{(n)})) = \mathcal{E}(\theta|\theta^{(n)}) + H(\tilde{P})$. Maximising this w.r.t. θ is the M-step of EM.
- Thus EM can be viewed as a two-step maximization of F. If $[-\log P(x, y|\theta)]$ is analogous to the "energy" of state x, then $[-F]$ is like a "free energy" (energy minus entropy).

Specific examples of EM

We will look at two specific examples of how EM can be applied:

- EM on a mixture of Gaussians ("soft K-means")
  - for clustering general multi-dimensional data
- EM on a continuous-time finite-state Markov chain ("phylo-EM")
  - for estimating a substitution model from aligned sequence data

K-means again

Mixture-of-Gaussians example

- Again, suppose we have N datapoints $\{y_i\}$ (restricted to one dimension for simplicity)
- Probabilistic model: mixture of K Gaussian components; component k has mean m_k and variance v_k. Parameters $\theta = \{m_k, v_k\}$. Each component has equal probability $1/K$.

Mixture of Gaussians: likelihoods

- If component label of point i is x_i, then joint likelihood for point i is

$$P(x_i, y_i|\theta) = \frac{1}{K}(2\pi v_{x_i})^{-\frac{1}{2}} \exp(-\frac{1}{2}(y_i - m_{x_i})^2/v_{x_i})$$

- Marginal likelihood for observed data is

$$P(y_i|\theta) = \sum_{x_i} P(x_i, y_i|\theta)$$

- Joint likelihood for all observed data and missing component labels is

$$P(x, y|\theta) = \prod_{i=1}^{N} P(x_i, y_i|\theta)$$

Posterior expectations

- Posterior probability of i'th component label (the E-step) is

$$P(x_i|y_i,\theta) = \frac{P(x_i,y_i|\theta)}{P(y_i|\theta)}$$

- Expected log-likelihood: let $W_i(x_i) = P(x_i|y_i,\theta^{(n)})$. Then

$$\begin{aligned}
\mathcal{E}(\theta|\theta^{(n)}) &= \langle \log P(x,y|\theta)\rangle_{P(x|y,\theta^{(n)})} \\
&= \sum_{i=1}^{N} \langle \log P(x_i,y_i|\theta)\rangle_{P(x_i|y_i,\theta^{(n)})} \\
&= -N\log K - \frac{N}{2}\log(2\pi) \\
&\quad -\frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K} W_i(k)\left(\log(v_k) + \frac{(y_i-m_k)^2}{v_k}\right)
\end{aligned}$$

Deriving K-means

$$\begin{aligned}\mathcal{E}(\theta|\theta^{(n)}) &= -N\log K - \frac{N}{2}\log(2\pi) \\ &\quad -\frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K} W_i(k)\left(\log(v_k) + \frac{(y_i - m_k)^2}{v_k}\right)\end{aligned}$$

Partial derivatives of $\mathcal{E}(\theta|\theta^{(n)})$

$$\frac{\partial \mathcal{E}}{\partial m_k} = \sum_{i=1}^{N} W_i(k)\frac{y_i - m_k}{v_k}$$

$$\frac{\partial \mathcal{E}}{\partial v_k} = -\frac{1}{2}\sum_{i=1}^{N} W_i(k)\left(\frac{1}{v_k} - \frac{(y_i - m_k)^2}{v_k^2}\right)$$

(Soft) K-means recovered

- Setting the partial derivatives to zero gives

$$\begin{aligned} m_k &= \frac{\sum_i W_i(k) y_i}{\sum_i W_i(k)} \\ v_k &= \frac{\sum_i W_i(k)(y_i - m_k)^2}{\sum_i W_i(k)} \\ &= \frac{\sum_i W_i(k) y_i^2}{\sum_i W_i(k)} - \left(\frac{\sum_i W_i(k) y_i}{\sum_i W_i(k)} \right)^2 \end{aligned}$$

  which are immediately recognisable as the mean and variance of the y_i, weighted by $W_i(k)$.
- As variance $\to 0$, "soft" K-means becomes "hard" K-means

Notes on soft K-means

- The original K-means algorithm is equivalent to (a) only estimating the m_k and (b) taking the limit $v_k \to 0 \; \forall \; k$ so that $W_i(k) \to 1$ for the most probable cluster and 0 for all other clusters. Neal and Hinton refer to this as a "winner-take-all" variant of EM. It's also called "hard" (vs "soft") K-means.
- Note that soft K-means can get stuck in an infinite-likelihood "trap" if a single point gets assigned to a cluster and $v_k \to 0$. This can be fixed by putting a prior distribution on the parameters (m_k, v_k).
- Works on pretty much any mixture (not just Gaussians)

Substitution models

Short-time approximation (Dayhoff *et al*)

- take a pairwise alignment of two closely related sequences
- count the number of instances C_{ij} of each aligned residue-pair (i, j)
- estimate the evolutionary distance Δt separating the two sequences
- set $R_{ij} \leftarrow C_{ij}/\Delta t$.

Beyond the short-time approximation

- Drawback: ignores multiple substitutions. We seek a maximum likelihood version, with the likelihood implicitly taking multiple substitutions into account.
- We will see that this amounts (at least for the discrete-time approximation) to getting an "unbiased" estimate of **C**. These correspond to the *expected* number of times that each $i \to j$ transition occurred.
- Our unbiased estimate of **C** depends on our current estimate of the rate matrix: if we think that the R_{ij} are small, there will be few multiple substitutions, but if the R_{ij} are large, there will be many. Thus the two things that we are trying to estimate are inter-related, but that's how EM works: we fix one and estimate the other, then do it the other way round, then iterate to convergence.

Beyond the short-time approximation

- We start with a discrete-time approximation (breaking the time interval into small, finite steps). We then consider the limit where the time-steps get infinitely small. In this continuous-time limit, there are an infinite number of $i \rightarrow i$ transitions. It then makes more sense to consider the amount of *time* spent in state i.
- Take the pairwise case first. The derivation is laborious but the result we're aiming for is that we can get our expected counts $\mathbf{C}$ by conditioning on, then summing over, all possible times at which a substitution can occur.

Discrete-timestep version

- Use discrete-time approximation: break T into discrete steps of size Δt with discrete-time transition matrix $\mathbf{Q} = \mathbf{I} + \mathbf{R}\Delta t$ (later we'll take limit $\Delta t \to 0$). Let x_n be the state at time $t = n\Delta t$. Let the p.d.f. over x_0 be π_{x_0}. Suppose that $x_0 = a$ and $x_N = b$ are observed (where $N = T/\Delta t$), while states $x_1 \ldots x_{N-1}$ are missing data. Thus

$$
\begin{aligned}
P(x_0 \ldots x_N|\theta) &= \pi_{x_0} \prod_{n=0}^{N-1} Q_{x_n x_{n+1}} \\
P(x_0, x_N|\theta) &= \pi_{x_0} \left[\mathbf{Q}^N\right]_{x_0 x_N} \\
P(x_1 \ldots x_{N-1}|x_0, x_N, \theta) &= \frac{P(x_0 \ldots x_N|\theta)}{P(x_0, x_N|\theta)}
\end{aligned}
$$

Expected transition counts

- Let $\theta' = (\pi', \mathbf{Q}')$ be the old parameters and $\theta = (\pi, \mathbf{Q})$ be the new parameters. The EM function $\mathcal{E}(\theta|\theta')$ is

$$\mathcal{E}(\theta|\theta') = \log \pi_{x_0} + \sum_i \sum_j C_{ij} \log Q_{ij}$$

where C_{ij} is the **expected number of times that the transition $i \to j$ occurred.** This can be seen immediately from the fact that the joint likelihood for observed & missing data can be written in the form

$$P(x_0 \ldots x_N|\theta) = \pi_{x_0} \prod_i \prod_j Q_{ij}^{\xi_{ij}}$$

where ξ_{ij} counts the usage of transition $i \to j$.
Then we can deduce that $C_{ij} = \langle \xi_{ij} \rangle$.

A longer derivation

$$
\begin{aligned}
\mathcal{E}(\theta|\theta') &= \sum_{x_1}\sum_{x_2}\cdots\sum_{x_{N-1}} P(x_1 \ldots x_{N-1}|x_0, x_N, \theta') \log P(x_0 \ldots x_N|\theta) \\
&= \sum_{x_1}\sum_{x_2}\cdots\sum_{x_{N-1}} P(x_1 \ldots x_{N-1}|x_0, x_N, \theta') \left(\log \pi_{x_0} + \sum_{n=0}^{N-1} \log Q_{x_n x_{n+1}} \right) \\
&= \log \pi_{x_0} + \sum_{n=0}^{N-1} \sum_{x_n} \sum_{x_{n+1}} \left[\sum_{\{x_k:\, 1\le k<n,\ n+1<k\le N\}} P(x_1 \ldots x_{N-1}|x_0, x_N, \theta') \right] \log Q_{x_n x_{n+1}} \\
&= \log \pi_{x_0} + \sum_{n=0}^{N-1} \sum_{x_n} \sum_{x_{n+1}} \left[\sum_{\sim\{x_n, x_{n+1}\}} P(x_1 \ldots x_{N-1}|x_0, x_N, \theta') \right] \log Q_{x_n x_{n+1}} \\
&= \log \pi_{x_0} + \sum_{n=0}^{N-1} \sum_{x_n} \sum_{x_{n+1}} P(x_n, x_{n+1}|x_0, x_N, \theta') \log Q_{x_n x_{n+1}} \\
&= \log \pi_{x_0} + \sum_i \sum_j \left[\sum_{n=0}^{N-1} P(x_n = i, x_{n+1} = j|x_0, x_N, \theta') \right] \log Q_{ij} \\
&= \log \pi_{x_0} + \sum_i \sum_j C_{ij} \log Q_{ij}
\end{aligned}
$$

Calculating the expected transition counts

C_{ij} is the expected number of transitions $i \to j$ that occurred among the missing data. Recalling that the start and end states are $x_0 = a$ and $x_N = b$, and observing that the expectation C_{ij} is additive over all timepoints n at which the transition could occur, we have

$$
\begin{aligned}
C_{ij} &= \sum_{n=0}^{N-1} P(x_n = i, x_{n+1} = j | x_0 = a, x_N = b, \theta') \\
&= \sum_{n=0}^{N-1} \frac{P(x_n = i | x_0 = a, \theta') P(x_{n+1} = j | x_n = i, \theta') P(x_N = b | x_{n+1} = j, \theta')}{P(x_N = b | x_0 = a, \theta')} \\
&= \sum_{n=0}^{N-1} \frac{[\mathbf{Q}^n]_{ai} Q_{ij} [\mathbf{Q}^{N-n-1}]_{jb}}{[\mathbf{Q}^N]_{ab}}
\end{aligned}
$$

Summary so far

So far we have the counts in terms of something like a convolution of two matrix exponentials

$$C_{ij} = \frac{Q_{ij}}{[\mathbf{Q}^N]_{ab}} \sum_{n=0}^{N-1} [\mathbf{Q}^n]_{ai} [\mathbf{Q}^{N-n-1}]_{jb}$$

We can get explicit forms for terms like $[\mathbf{Q}^n]_{ai}$ using the diagonalized matrix form.

Eigenform of transition counts

Eigenvector decomposition $\mathbf{Q} = \mathbf{U}\mathbf{D}\mathbf{U}^{-1}$, where $\mathbf{D}$ is diagonal (with $D_{kk} = \lambda_k$), gives us $\mathbf{Q}^n = \mathbf{U}\mathbf{D}^n\mathbf{U}^{-1}$ and so

$$
\begin{aligned}
C_{ij} &= \frac{Q_{ij}}{[\mathbf{U}\mathbf{D}^N\mathbf{U}^{-1}]_{ab}} \sum_{n=0}^{N-1} [\mathbf{U}\mathbf{D}^n\mathbf{U}^{-1}]_{ai} [\mathbf{U}\mathbf{D}^{N-n-1}\mathbf{U}^{-1}]_{jb} \\
&= \frac{Q_{ij}}{\sum_m U_{am}\lambda_m^N U_{mb}^{-1}} \sum_{n=0}^{N-1} \left(\sum_k U_{ak}\lambda_k^n U_{ki}^{-1} \right) \left(\sum_l U_{jl}\lambda_l^{N-n-1} U_{lb}^{-1} \right) \\
&= \frac{Q_{ij}}{\sum_m U_{am}\lambda_m^N U_{mb}^{-1}} \sum_k U_{ak} U_{ki}^{-1} \sum_l U_{jl} U_{lb}^{-1} \sum_{n=0}^{N-1} \lambda_k^n \lambda_l^{N-n-1}
\end{aligned}
$$

Eigenvector interaction matrix

$$
\begin{aligned}
C_{ij} &= \frac{Q_{ij}}{\sum_m U_{am}\lambda_m^N U_{mb}^{-1}} \sum_k U_{ak} U_{ki}^{-1} \sum_l U_{jl} U_{lb}^{-1} \sum_{n=0}^{N-1} \lambda_k^n \lambda_l^{N-n-1} \\
&= \frac{Q_{ij}}{\sum_m U_{am}\lambda_m^N U_{mb}^{-1}} \sum_k U_{ak} U_{ki}^{-1} \sum_l U_{jl} U_{lb}^{-1} \Lambda_{kl}
\end{aligned}
$$

where

$$
\Lambda_{kl} = \begin{cases} \dfrac{\lambda_l^N - \lambda_k^N}{\lambda_l - \lambda_k} & \text{if } \lambda_k \neq \lambda_l \\ N\lambda_l^{N-1} & \text{if } \lambda_k = \lambda_l \end{cases}
$$

To obtain Λ_{kl} we have used the identity $\sum_{k=0}^{N-1} a^k = (a^N - 1)/(a - 1)$.

From one column to an alignment

So far we've estimated a summary of the missing data for a single column of an alignment.
Hopefully it should also be clear that if we have two sequences X, Y of length L then the expected counts matrix $\mathbf{C}$ can be obtained by summing over all sites X_l, Y_l, so

$$C_{ij} = \sum_{l=1}^{L} C_{ij}^{[X_l \overset{N}{\to} Y_l]}$$

where $C_{ij}^{[a \overset{N}{\to} b]}$ is the C_{ij} derived above for a process beginning in state $x_0 = a$ and ending in $x_N = b$.

Discrete-timestep EM algorithm

- To summarise: the EM algorithm for parameterising a discrete-time Markov chain from a pairwise alignment is
  1. Start with some initial estimate of the probability matrix $\mathbf{Q}$.
  2. Estimate the transition counts C_{ij} by summing over all sites X_l, Y_l as above.
  3. Update $Q_{ij} \leftarrow C_{ij} / \sum_k C_{ik}$ (that this maximizes $\mathcal{E}$ can be seen using Lagrange multipliers)
  4. Repeat until the algorithm converges on a fixed matrix $\mathbf{Q}$ (i.e. until the likelihood $P(Y|X, \mathbf{Q})$ stabilises).

From discrete to continuous

- Continuous limit: as $\Delta t \to 0$, so $N \to \infty$. The upshot of this is that C_{ij} converges and is meaningful for $i \neq j$, but $C_{ii} \sim 1/N \to \infty$ because during most of the short time intervals, the chain stays in the same state.
  - More detailed argument: the N-dependent terms in the expression for C_{ij} are $Q_{ij}\Lambda_{kl}/\mathbf{Q}^N_{ab}$ with $Q_{ij} = \delta_{ij} + R_{ij}/N$. The factors of λ^N approximately cancel in Λ_{kl} and $\mathbf{Q}^N_{ab}$, and the factor of N in Λ_{kk} cancels the $1/N$ in Q_{ij} as long as $i \neq j$. However, when $i = j$, there is an unaccounted factor of N from Λ_{kk} (indeed from all Λ_{kl} where $\lambda_k = \lambda_l$).
  - To get round this, we can define $W_i = C_{ii}T/N = C_{ii}\Delta t$ to be the expected *time* spent waiting in state i. This then converges to a meaningful, finite value as $\Delta t \to 0$.

Continuous-time transition counts and wait times

Since $\mathbf{M}(t) = \exp(\mathbf{R}t)$, by analogy to the discrete case

$$\begin{aligned} C_{ij} &= \frac{1}{M(T)_{ab}} \int_0^T M(t)_{ai} (R_{ij} dt) M(T-t)_{jb} \\ &= \frac{R_{ij}}{\sum_m U_{am} \exp(\lambda_m T) U^{-1}_{mb}} \sum_k U_{ak} U^{-1}_{ki} \sum_l U_{jl} U^{-1}_{lb} \mathcal{J}_{kl}(T) \\ W_i &= \frac{1}{M(T)_{ab}} \int_0^T M(t)_{ai} (dt) M(T-t)_{jb} \\ &= \frac{1}{\sum_m U_{am} \exp(\lambda_m T) U^{-1}_{mb}} \sum_k U_{ak} U^{-1}_{ki} \sum_l U_{jl} U^{-1}_{lb} \mathcal{J}_{kl}(T) \end{aligned}$$

where

$$\mathcal{J}_{kl}(T) = \int_0^T \exp(\lambda_k t) \exp(\lambda_l (T-t)) dt = \begin{cases} \dfrac{\exp(\lambda_l T) - \exp(\lambda_k T)}{\lambda_l - \lambda_k} & \text{if } \lambda_k \neq \lambda_l \\ T \exp(\lambda_k T) & \text{if } \lambda_k = \lambda_l \end{cases}$$

- The estimates for C_{ij} and W_i can be tested by simulation.
- The λ_k in the above expression are the eigenvalues for $\mathbf{R}$, not $\mathbf{Q}$. Since $\mathbf{Q} = \mathbf{I} + \mathbf{R}$, they are related by $\lambda_k^{(\mathbf{Q})} = 1 + \lambda_k^{(\mathbf{R})}$. The eigenvectors are the same.
- To estimate R_{ij}, note that $R_{ij} = \lim_{\Delta t \to 0} \frac{Q_{ij}}{\Delta t} = \lim_{\Delta t \to 0} \frac{C_{ij}}{\sum_k C_{ik} \Delta t} = \frac{C_{ij}}{W_i}$ since $\lim_{\Delta t \to 0} C_{ik} \Delta t = \delta_{ik} W_i$.

The EM algorithm for substitution models

- To summarise, the EM algorithm for *continuous-time* Markov chains and pairwise alignments is as follows:
  1. Start with some initial estimate of the probability matrix **R**.
  2. Estimate the transition counts C_{ij} by summing over all sites X_l, Y_l.
  3. Update $R_{ij} \leftarrow C_{ij}/W_i$.
  4. Repeat until the algorithm converges on a fixed matrix **R** (i.e. until the likelihood stabilises).
- Strictly speaking, the expected likelihood $\mathcal{E}$ probably needs to be recast as an expected likelihood *density* w.r.t. the W_i (which are continuous variables) if this EM algorithm is to be made rigorous, but life is too short.

The phylo-EM algorithm

- EM for a multiple alignment (with a known phylogenetic tree): sum over all alignment columns, all tree branches, and all possible states $a \to b$ of each branch. Use peeling algorithm to find posterior probabilities of each $a \to b$ state.
  - Can accumulate counts in eigenvector space to save time.

Connection to message-passing

Detailed derivation: recall, for a parent-node-sibling triplet (p, n, s):

$$P(x_p = a, x_n = b|Y) = \frac{G_p(a)M(t_{pn})_{ab}F_n(b)E_s(a)}{P(Y)}$$

where Y represents the observed states at the tree leaves, and $\{F_n, G_p, E_s\}$ are the pruning and peeling likelihoods corresponding to messages on the factor graph.
Summing over alignments A, columns C and branches (p, n, s):

$$\begin{aligned}
C_{ij} &= \sum_A \sum_C \sum_{(p,n,s)} \sum_{a,b} P(x_p = a, x_n = b|Y_{AC})C_{ij}(a, b, t_{pn}) \\
&= \sum_A \sum_C \sum_{(p,n,s)} \sum_{a,b} \left(\frac{G_p(a)M(t_{pn})_{ab}F_n(b)E_s(a)}{P(Y_{AC})} \right) \left(\frac{1}{M_{ab}(t_{pn})} \sum_k U_{ak}U_{ki}^{-1} \sum_l U_{jl}U_{lb}^{-1}\mathcal{J}_{kl}(t_{pn}) \right) \\
&= \sum_k U_{ki}^{-1} \sum_l U_{jl} \sum_A \sum_C \frac{1}{P(Y_{AC})} \sum_{(p,n,s)} \left(\sum_a U_{ak}G_p(a)E_s(a) \right) \left(\sum_b U_{lb}^{-1}F_n(b) \right) \mathcal{J}_{kl}(t_{pn})
\end{aligned}$$

The terms $\sum_a U_{ak}G_p(a)E_s(a)$ and $\sum_b U_{lb}^{-1}F_n(b)$ are projections of the peeling and pruning messages onto the eigenvector basis.

Summary

- The EM algorithm
  - Maximizes posterior expectation of log-likelihood, $\mathcal{E}(\theta|\theta^{(n)})$
  - Alternates between two steps:
    - Estimating posterior $\tilde{P}(x) \equiv P(x|y,\theta^{(n)})$ (the *missing data*)
    - Maximizing $\langle \log P(x,y|\theta) \rangle_{\tilde{P}}$ w.r.t. θ (the *model parameters*)
- Specific applications in bioinformatics
  - EM + Mixture of Gaussians $\rightarrow$ Soft K-Means
  - EM + CTMC + Tree $\rightarrow$ Phylo-EM
  - There are many others