Computational Biology

Lecture 10

Saad Mneimneh

Hidden Markov Model

A Hidden Markov Model HMM is defines as:

decouple states from symbols

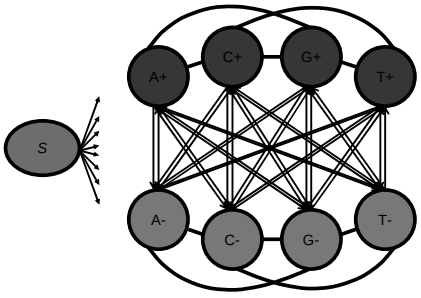
- A set of *hidden* states
- [transitional probabilities]
  For each pair of states i and j,
  a transition probability a_{ij}.
- $\Sigma_j a_{ij} = 1$

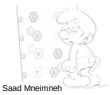
- An alphabet of symbols Σ
- [emission probabilities]
  For each state k, and symbol b
  $e_k(b) = p(x_i = b \mid \pi_i = k)$
  [now we use variable π for states and variable x for symbols]
- $\Sigma_{b \in \Sigma} e_k(b) = 1$ for each state k

Markov property: $p(\pi_n = j \mid x_0 \ldots x_m, \pi_0 \ldots \pi_{m-1}, \pi_m = i) = p(\pi_n = j \mid \pi_m = i) \quad m < n$
if $m = n - 1$, then this is a_{ij}

Saad Mneimneh

HMM for CpG islands





Saad Mneimneh

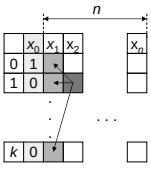
Questions with HMMs

- ***Evaluation***: given x, what is the probability $p(x)$ that it was produced by the model?
- ***Decoding***: given x, what is the most probable path that produces x in the model?
- ***Learning***: given x, what are the parameters (transitional probabilities and emission probabilities) of the model that maximize $p(x)$.

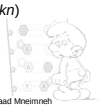
Saad Mneimneh

Viterbi *decoding* algorithm

- Initialization
  $v_0(0) = 1$, $v_k(0) = 0$ for $k > 0$
- Main iteration
  for $i = 1 \ldots n$
  $v_l(i) = e_l(x_i).\max_k (v_k(i-1).a_{kl})$
  $\text{ptr}_l(i) = \text{argmax}_k(v_k(i-1)a_{kl})$
- Termination
  $p(x,\pi^*) = \max_k(v_k(n)a_{k0})$



Time = $O(k^2n)$
Space = $O(kn)$



Saad Mneimneh

Computing $p(x)$

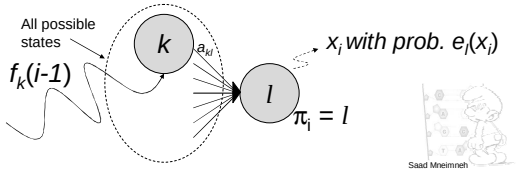
- Before, $p(x) = a_{sx1}\Pi_{i=2\ldots n}\, \boldsymbol{a}_{x^{i-1}\, x^i}$
- Now, $p(x)=\Sigma_\pi p(x,\pi)$
- Enumerating all π is exponential!
- Use Viterbi, same as before, but change **max** to $\boldsymbol{\Sigma}$

Saad Mneimneh

Forward *evaluation* algorithm

- Let $f_l(i) = p(x_1 \ldots x_i, \pi_i = l)$
- Then,

$f_l(i) = e_l(x_i)\Sigma_k f_k(i-1)a_{kl}$



Saad Mneimneh

Derivation

$f_l(i)$

$= \Sigma_{\pi 1 \ldots \pi i-1} p(x_1 \ldots x_i, \pi_1 \ldots \pi_{i-1}, \pi_i = l)$

$= \Sigma_{\pi 1 \ldots \pi i-1} p(x_i, \pi_i = l, x_1 \ldots x_{i-1}, \pi_1 \ldots \pi_{i-1})$

$= \Sigma_{\pi 1 \ldots \pi i-1} p(x_i, \pi_i = l \mid x_1 \ldots x_{i-1}, \pi_1 \ldots \pi_{i-1}).p(x_1 \ldots x_{i-1}, \pi_1 \ldots \pi_{i-1})$

$= \Sigma_{\pi 1 \ldots \pi i-1} p(x_i, \pi_i = l \mid \pi_{i-1}).p(x_1 \ldots x_{i-1}, \pi_1 \ldots \pi_{i-1})$

$= \Sigma_{\pi 1 \ldots \pi i-2, k} p(x_i, \pi_i = l \mid \pi_{i-1} = k).p(x_1 \ldots x_{i-1}, \pi_1 \ldots \pi_{i-2}, \pi_{i-1} = k)$

$= \Sigma_{\pi 1 \ldots \pi i-2, k} e_l(x_i)a_{kl}.p(x_1 \ldots x_{i-1}, \pi_1 \ldots \pi_{i-2}, \pi_{i-1} = k)$

$= \Sigma_k \Sigma_{\pi 1 \ldots \pi i-2} e_l(x_i)a_{kl}.p(x_1 \ldots x_{i-1}, \pi_1 \ldots \pi_{i-2}, \pi_{i-1} = k)$

$= \Sigma_k \Sigma_{\pi 1 \ldots \pi i-2} p(x_1 \ldots x_{i-1}, \pi_1 \ldots \pi_{i-2}, \pi_{i-1} = k)\ e_l(x_i)a_{kl}$

$= \Sigma_k f_k(i\text{-}1)e_l(x_i)a_{kl} = e_l(x_i)\ \Sigma_k f_k(i\text{-}1)a_{kl}$

Saad Mneimneh

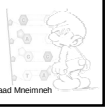
Forward *evaluation* algorithm

- Initialization
  $f_0(0) = 1$, $f_k(0) = 0$ for $k > 0$
- Main iteration
  for $i = 1 \ldots n$
  $f_l(i) = e_l(x_i).\Sigma_k\ (f_k(i-1).a_{kl})$
- Termination
  $p(x) = \Sigma_k f_k(n)a_{k0}$

Saad Mneimneh

Problem of small numbers

- In Viterbi and Forward algorithm we multiply probabilities → numbers will soon be very small and we loose precision.
- Use log space → addition instead of multiplication.



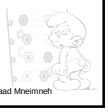
Saad Mneimneh

Log space Viterbi

$v_l(i) = e_l(x_i).\max_k (v_k(i-1).a_{kl})$

Let $V_l(i) = \log v_l(i)$

$$\begin{aligned} V_l(i) &= \log [e_l(x_i).\max_k (v_k(i-1).a_{kl})] \\ &= \log e_l(x_i) + \log [\max_k (v_k(i-1).a_{kl})] \\ &= \log e_l(x_i) + \max_k \log [(v_k(i-1).a_{kl})] \\ &= \log e_l(x_i) + \max_k (V_k(i-1) + \log a_{kl}) \end{aligned}$$



Saad Mneimneh

Log space Forward

$f_l(i) = e_l(x_i).\Sigma_k (f_k(i-1).a_{kl})$

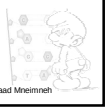
Let $F_l(i) = \log f_l(i)$

$$\begin{aligned} F_l(i) &= \log [e_l(x_i).\Sigma_k (f_k(i-1).a_{kl})] \\ &= \log e_l(x_i) + \log \Sigma_k (f_k(i-1).a_{kl}) \\ &\neq \log e_l(x_i) + \Sigma_k \log [(f_k(i-1).a_{kl})] \\ &= \log e_l(x_i) + \log \Sigma_k e^{(Fk(i-1) + \log akl)} \end{aligned}$$

Saad Mneimneh

Back to the most probable path

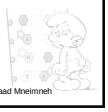
- The Viterbi algorithm finds it!
- The most probable path might not be the most appropriate basis for judgment.
- We might want, for instance, the most probable state for an observation x_i.
- More generally, we are interested in $p(\pi_i = k \mid x)$



Saad Mneimneh

Computing $p(\pi_i = k \mid x)$

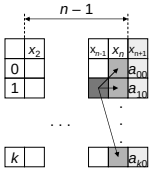
- $p(\pi_i = k \mid x) = p(x, \pi_i = k)/p(x)$
- I know how to compute $p(x)$: forward alg.
- $p(x, \pi_i = k)$
  $= p(x_1 \ldots x_i, \pi_i = k).p(x_{i+1} \ldots x_n \mid x_1 \ldots x_i, \pi_i = k)$
  $= p(x_1 \ldots x_i, \pi_i = k).p(x_{i+1} \ldots x_n \mid \pi_i = k)$
  $= f_k(i).b_k(i)$



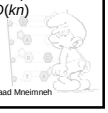
Saad Mneimneh

Backward *evaluation* algorithm

- Initialization
  $b_k(n) = a_{k0}$ for all k
- Main iteration
  for $i = n - 1 \ldots 1$
  $b_k(i) = \Sigma_l \, a_{kl} e_l(x_{i+1}) b_l(i+1)$
- Termination
  $p(x) = \Sigma_l \, a_{0l} e_l(x_1) b_l(1)$



Time = $O(k^2 n)$
Space = $O(kn)$



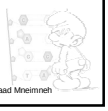
Saad Mneimneh

Learning (training the HMM)

- Let θ be the parameters of the HMM (transition probabilities and emission probabilities, the a's and e's)
- Given independent sequences $x^1, \ldots, x^n$, we would like to find θ that will maximize:

$$\log p(x^1 \ldots x^n \mid \theta) = \Sigma_{j=1..n} \log p(x^j \mid \theta)$$

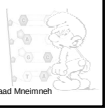
This is called the maximum likelihood parameters.



Saad Mneimneh

State sequence is known

- Assume the path for each x^j is known
  - For instance, we have sequences in which CpG islands are already labeled
- Paths are known, let
  - A_{kl} = number of transitions from k to l
  - $E_k(b)$ = number of times b emitted in state k
- The maximum likelihood parameters are given by:
  - $a_{kl} = A_{kl} / \Sigma_{l'} A_{kl'}$
  - $e_k(b) = E_k(b) / \Sigma_{b'} E_k(b')$

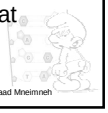


Saad Mneimneh

Maximum likelihood from counts

- Assume we have a sequence of independent observations $x_1 \ldots x_n$ and that we count n_i occurrences of outcome i, i=1…k.
- Let θ_i = probability of i.
- Then $\theta^{ML} = \{\theta_i = n_i/n, i=1 \ldots k\}$ is the maximum likelihood solution for θ.
- Consider any other θ. We want to show that

$$p(x \mid \theta^{ML}) > p(x \mid \theta)$$



Saad Mneimneh

Proof

$$\log \frac{p(x \mid \theta^{ML})}{p(x \mid \theta)} = \log \frac{\prod_i (\theta_i^{ML})^{n_i}}{\prod_i \theta_i^{n_i}}$$
$$= \sum_i n_i \log \frac{\theta_i^{ML}}{\theta_i}$$
$$= n \sum_i \theta_i^{ML} \log \frac{\theta_i^{ML}}{\theta_i} > 0$$

The last summation is the relative entropy of θ^{ML} and θ which is always positive and 0 iff $\theta^{ML} = \theta$ (from information theory)

Saad Mneimneh

Some problems

- Maximum likelihood are vulnerable to overfitting if insufficient data.
- For instance, if a state *k* was never used in the set of training sequences, then
  - $a_{kl} = 0$ for all *l*
  - $e_k(b) = 0$ for all *b*
- To avoid such problem, start with pseudocounts of r_{kl} for A_{kl} and $r_k(b)$ for $E_k(b)$.
- Large pseudocount indicates strong prior belief about the probabilities (will require more data to modify)
- Small pseudocount just to avoid zero probability

Saad Mneimneh

Example

Dishonest Casino HMM

$r_{0F} = r_{0L} = r_{F0} = r_{L0} = 1$; [avoid zero probability]
$r_{FL} = r_{LF} = r_{FF} = r_{LL} = 1$; [avoid zero probability]

$r_F(1) = r_F(2) = \ldots = r_F(6) = 20$ [strong belief that fair is fair]

$r_L(1) = r_L(2) = \ldots = r_L(6) = 5$ [wait and see for loaded]

Saad Mneimneh

New species comes in…

- New species with different distribution of CpG islands.
- We do not have labeled genomic sequences for the new species.
- Need to find maximum likelihood θ of HMM without knowing the paths!

Saad Mneimneh

Baum–Welsh algorithm

start at iteration 0 with some θ, call it θ^0
$L^0 \leftarrow \log p(x^1 \ldots x^n \mid \theta^0)$
$i \leftarrow 0$

repeat
$i \leftarrow i + 1$

$A_{kl}^i \leftarrow \mathrm{E}\,[A_{kl} \mid x^1 \ldots x^n, \theta^{i-1}]$ (expected value)
$E_k(b)^i \leftarrow \mathrm{E}\,[E_k(b) \mid x^1 \ldots x^n, \theta^{i-1}]$ (expected value)

calculate θ^i using maximum likelihood estimators from counts A_{kl}^i and $E_k(b)^i$ *as before.*

$L^i \leftarrow \log p(x^1 \ldots x^n \mid \theta^i)$ (new likelihood)

until $L^i - L^{i-1} <$ threshold

Saad Mneimneh

What is the guarantee?

- Baum–Welsh algorithm is a special case of a general algorithm known as Expectation Maximization (EM)
- EM guarantees that $p(\mathrm{X} \mid \theta^{i+1}) \geq p(\mathrm{X} \mid \theta^i)$
- It will therefore converge to a local maximum (not necessarily the maximum)

Saad Mneimneh

We need to…

Compute:

– E $[A_{kl} \mid x^1 \ldots x^n, \theta]$

– E $[E_k(b) \mid x^1 \ldots x^n, \theta]$

Saad Mneimneh

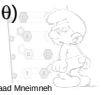
E $[A_{kl} \mid x^1 \ldots x^n, \theta]$

By linearity of expectation:
E $[A_{kl} \mid x^1 \ldots x^n, \theta] = \Sigma_j$ E $[A_{kl}^j \mid x^j, \theta]$

By linearity of expectation, again:
E $[A_{kl}^j \mid x^j, \theta] = \Sigma_i$ E[# of $k \rightarrow l$ at $x_i^j \mid x^j, \theta$]
$= \Sigma_i p(k \rightarrow l$ at $x_i^j \mid x^j, \theta)$

$p(k \rightarrow l$ at $x_i^j \mid x^j, \theta) = p(\pi_i = k, \pi_{i+1} = l \mid x^j, \theta)$

$p(\pi_i = k, \pi_{i+1} = l \mid x^j, \theta) = p(\pi_i = k, \pi_{i+1} = l, x^j \mid \theta) \,/\, p(x^j \mid \theta)$
$= f_k^j(i) a_{kl} e_l(x_{i+1}) b_l^j(i+1) \,/\, p(x^j \mid \theta)$



Saad Mneimneh

We get:

$$E[A_{kl} \mid x^1 \ldots x^j, \theta] = \sum_j \frac{1}{p(x^j \mid \theta)} \sum_i f_k^j(i) a_{kl} e_l(x_{i+1}^j) b_l^j(i+1)$$

$$E[E_k(b) \mid x^1 \ldots x^j, \theta] = \sum_j \frac{1}{p(x^j \mid \theta)} \sum_{i \mid x_i^j = b} f_k^j(i) b_k^j(i)$$

Saad Mneimneh

Viterbi training

start at iteration 0 with some θ, call it θ^0
$i \leftarrow 0$

repeat
$i \leftarrow i + 1$

$A_{kl}^i \leftarrow$ number of transitions $k \rightarrow l$ on the most probable paths $\pi^{1*}, \ldots, \pi^{i*}$

$E_k(b)^i \leftarrow$ number of times k emits b on the most probable paths $\pi^{1*}, \ldots, \pi^{i*}$

calculate θ^i using maximum likelihood estimators
from counts A_{kl}^i and $E_k(b)^i$ *as before.*

until none of the optimal paths change

Saad Mneimneh

What is the guarantee

- It will converge
- It will not necessarily maximize the true likelihood $p(x_1 \ldots x^n \mid \theta)$, but $p(x_1 \ldots x^n \mid \theta, \pi^{1*}, \ldots, \pi^{i*})$
- Usually performs less well than Baum–Welsh
- Practical, don't have to perform Forward and Backward algorithms, only Viterbi!
- Makes sense if we are using only Viterbi decoding

Saad Mneimneh