Knowledge-based systems in Bioinformatics, 1MB602

Lecture 8: Probabilistic Approaches

Lecture overview

- Elementary probability theory
- Frequentist vs. Bayesian philosophy
- Machine learning
- Bayesian networks
- Markov processes
- Hidden Markov Models

Role of probability theory in AI

- A problem with an agent based on first-order logic is that the agent almost never have access to the whole truth about its environment
- Many aspects are either unknown or not precisely known
- The agent must therefore act under uncertainty

Probability and decision

- Probability
  - A way of summarizing uncertainty (0 to 1)
- Probability theory
  - Our main tool for dealing with degrees of belief
- Decision theory
  - Probability theory + utility theory
  - Utility theory: all states have a degree of usefulness (utility) to the agent, and the agent will prefer states with high utility
  - Rational agent: chooses the action that yields the highest expected utility, averaged over all the possible outcomes of the action (i.e. we weight the utility of a particular outcome by the probability that it occurs)

Probability as frequency

- Drawing cards from a standard deck
  - P(card is jack of hearts | standard deck) = 1/52
  - P(card is of color hearts | standard deck) = 13/52
- Probability of drawing a pair in 5-card poker
  - P(hand contains a pair | standard deck) = $\frac{\text{\# of hands with pairs}}{\text{total \# of hands}}$
  - Use combinatorics to calculate the answer
- General probability of event given some conditions (conditional probability)
  - P(event | conditions)

Joint probability vs conditional probability

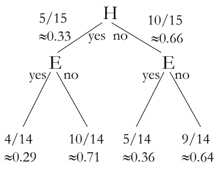
- Flipping a 'non-biased' coin
- Probability of four consecutive flips resulting in four heads
  - $P(F_1=\text{head},F_2=\text{head},F_3=\text{head},F_4=\text{head}) = P(F_1=\text{head}) * P(F_2=\text{head}) * P(F_3=\text{head}) * P(F_4=\text{head}) = 1/2 * 1/2 * 1/2 * 1/2 = 0.0625$
  - Independent events: joint probability
- Probability of taking four American coins from a bag of 10 American and 10 British coins
  - $P(C_1=\text{amer},C_2=\text{amer},C_3=\text{amer},C_4=\text{amer}) = P(C_1=\text{amer}) * P(C_2=\text{amer} \mid C_1=\text{amer}) * P(C_3=\text{amer} \mid C_1=\text{amer}, C_2=\text{amer}) * P(C_4=\text{amer} \mid C_1=\text{amer}, C_2=\text{amer}, C_1=\text{amer}) = 10/20 * 9/19 * 8/18 * 7/17 = 0.0625$
  - Each event dependent on previous events: conditional probability

Bayes’ Theorem

- $$\overset{\substack{\text{Posterior}\\\downarrow}}{P(H|E)} = \frac{\overset{\substack{\text{Likelihood}\\\downarrow}}{P(E|H)} * \overset{\substack{\text{Prior}\\\swarrow}}{P(H)}}{\underset{\substack{\uparrow\\\text{Normalization constant}}}{P(E)}}$$
- **The** posterior **(a posteriori) probability of an hypothesis (H) after considering the evidence (E) is the** likelihood **of the evidence given the hypothesis times the** prior **(a priori) probability of the hypothesis scaled by a normalization constant**

Bayes’ theorem and reasoning under uncertainty

- **Allows us to reason about a prior event H if a subsequent event has occurred**
  - **We need not know whether event H has occurred**
- **Example (10 British, 5 American coins)**
  - **H=first coin American, E=second coin American**
  - **P(H|E) = P(E|H) * P(H) / P(E)**

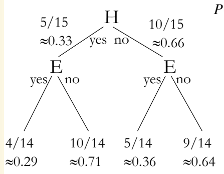


$$P(H|E) \approx \frac{0.29 * 0.33}{P(E)}$$

How to calculate P(E)?

Bayes’ theorem and reasoning under uncertainty

- **H=first coin American, E=second coin American**



$$\begin{aligned} P(H|E) &= \frac{P(E|H)P(H)}{P(E)} \\ &\approx \frac{0.29\cdot 0.33}{P(E)} \\ &= \frac{0.29\cdot 0.33}{P(E|H)P(H)+P(E|\neg H)P(\neg H)} \\ &\approx \frac{0.29\cdot 0.33}{(0.29\cdot 0.33)+(0.36\cdot 0.66)} \\ &\approx 0.31 \end{aligned}$$

$$\boxed{\begin{array}{l} P(H|E) = \frac{P(E|H)P(H)}{P(E)} \\ P(\neg H|E) = \frac{P(E|\neg H)P(\neg H)}{P(E)} \\ P(H|E)+P(\neg H|E) = 1 \\ P(E) = P(E|H)P(H)+P(E|\neg H)P(\neg H) \end{array}}$$

Machine learning

- **A learning agent can be divided into two main conceptual components**
  - **The learning element: responsible for making improvements**
  - **The performance element: responsible for selecting external actions**
- **The learning element takes some knowledge from the performance element and some feedback on how the agent is doing**
  - **Based on this it determines how the performance element should be modified to (hopefully) do better in the future**
- **The feedback generally tells the agent what the correct outcome is**

Supervised learning

- **Any situation in which both the inputs and the outputs can be perceived by the agent is called** supervised learning
- **In supervised learning the learning element is given the (approximately) correct value of the function for particular inputs**
  - **Based on this value it changes its representation of the function to try to match the information provided by the feedback**
- **Formally: an example is a pair (*x*,*f*(*x*)), where *x* is the input and *f(x)* the output**
- **Pure inductive learning:**
  - **Given a collection of examples of *f*, return a function *h* that approximates *f***
  - ***h* is called a hypothesis**

Choosing hypotheses

- **Generally we want the most probable hypothesis given the training data**

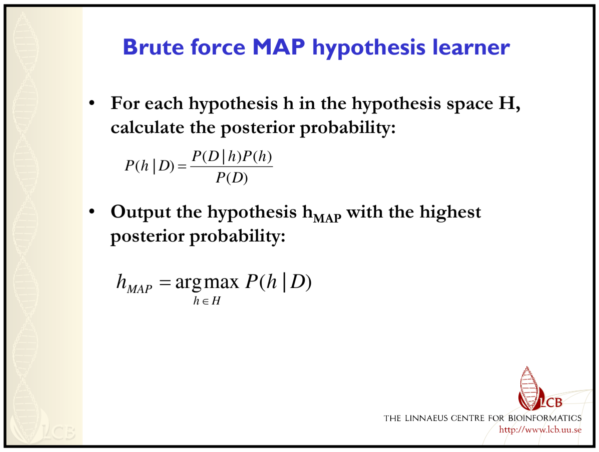
$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- **Maximum a posteriori hypothesis h_{MAP}**

$$\begin{aligned} h_{MAP} &= \underset{h\in H}{\arg\max}\, P(h|D) \\ &= \underset{h\in H}{\arg\max}\, \frac{P(D|h)P(h)}{P(D)} \\ &= \underset{h\in H}{\arg\max}\, P(D|h)P(h) \end{aligned}$$

- **If $P(h_i) = P(h_j)$, then we can choose the Maximum likelihood hypothesis h_{ML}**

$$h_{ML} = \underset{h\in H}{\arg\max}\, P(D|h)$$



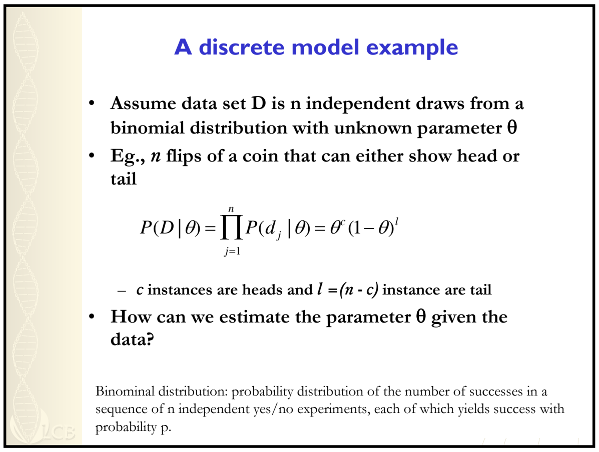
Brute force MAP hypothesis learner

- For each hypothesis h in the hypothesis space H, calculate the posterior probability:

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

- Output the hypothesis h_{MAP} with the highest posterior probability:

$$h_{MAP} = \arg\max_{h \in H} P(h \mid D)$$



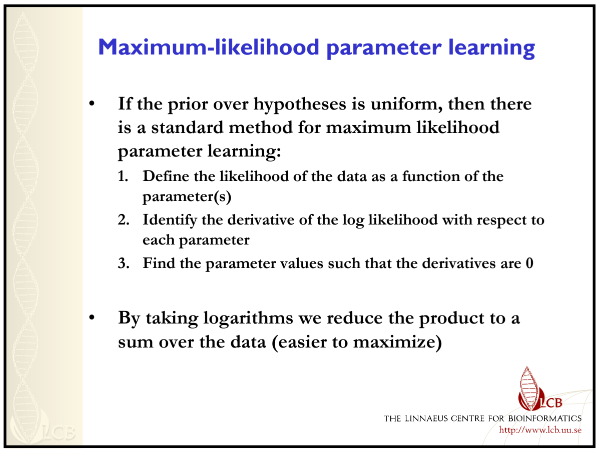
A discrete model example

- Assume data set D is n independent draws from a binomial distribution with unknown parameter θ
- Eg., *n* flips of a coin that can either show head or tail

$$P(D \mid \theta) = \prod_{j=1}^{n} P(d_j \mid \theta) = \theta^c (1-\theta)^l$$

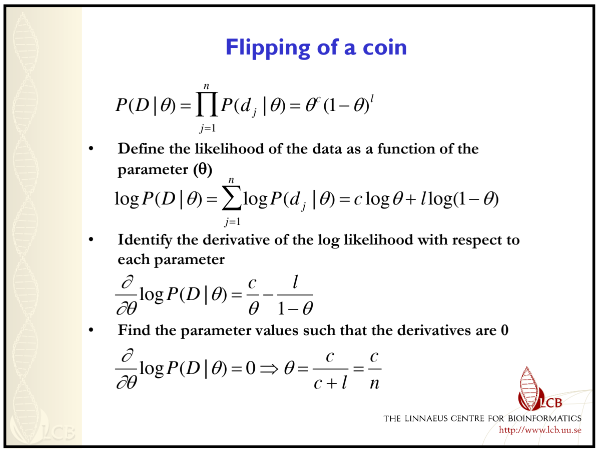
  - *c* instances are heads and *l* =*(n - c)* instance are tail
- How can we estimate the parameter θ given the data?

Binominal distribution: probability distribution of the number of successes in a sequence of n independent yes/no experiments, each of which yields success with probability p.



Maximum-likelihood parameter learning

- If the prior over hypotheses is uniform, then there is a standard method for maximum likelihood parameter learning:
  1. Define the likelihood of the data as a function of the parameter(s)
  2. Identify the derivative of the log likelihood with respect to each parameter
  3. Find the parameter values such that the derivatives are 0
- By taking logarithms we reduce the product to a sum over the data (easier to maximize)



Flipping of a coin

$$P(D \mid \theta) = \prod_{j=1}^{n} P(d_j \mid \theta) = \theta^c (1-\theta)^l$$

- Define the likelihood of the data as a function of the parameter (θ)

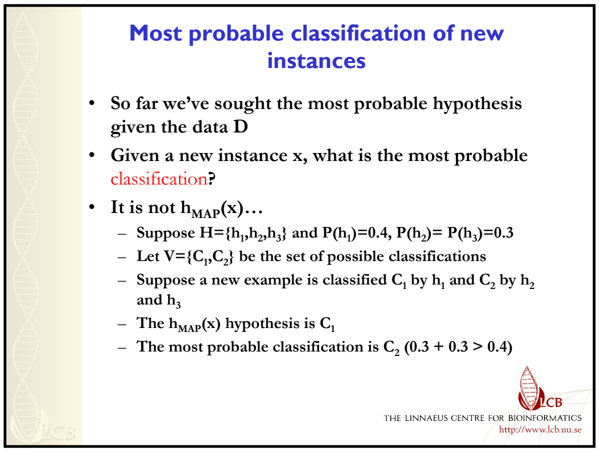
$$\log P(D \mid \theta) = \sum_{j=1}^{n} \log P(d_j \mid \theta) = c \log \theta + l \log(1-\theta)$$

- Identify the derivative of the log likelihood with respect to each parameter

$$\frac{\partial}{\partial \theta} \log P(D \mid \theta) = \frac{c}{\theta} - \frac{l}{1-\theta}$$

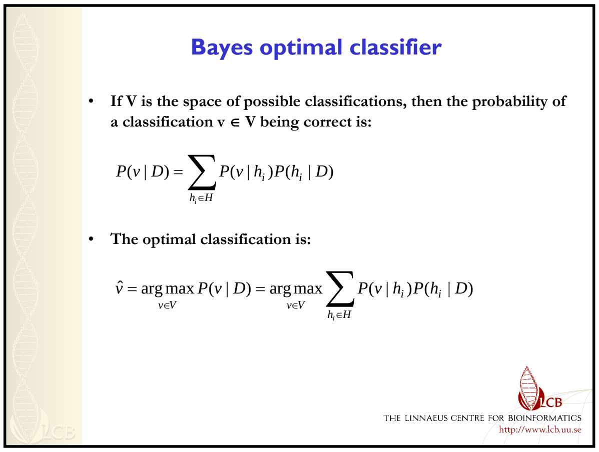
- Find the parameter values such that the derivatives are 0

$$\frac{\partial}{\partial \theta} \log P(D \mid \theta) = 0 \Rightarrow \theta = \frac{c}{c+l} = \frac{c}{n}$$



Most probable classification of new instances

- So far we've sought the most probable hypothesis given the data D
- Given a new instance x, what is the most probable classification?
- It is not h_{MAP}(x)…
  - Suppose H={h_1,h_2,h_3} and P(h_1)=0.4, P(h_2)= P(h_3)=0.3
  - Let V={C_1,C_2} be the set of possible classifications
  - Suppose a new example is classified C_1 by h_1 and C_2 by h_2 and h_3
  - The h_{MAP}(x) hypothesis is C_1
  - The most probable classification is C_2 (0.3 + 0.3 > 0.4)



Bayes optimal classifier

- If V is the space of possible classifications, then the probability of a classification v ∊ V being correct is:

$$P(v \mid D) = \sum_{h_i \in H} P(v \mid h_i) P(h_i \mid D)$$

- The optimal classification is:

$$\hat{v} = \arg\max_{v \in V} P(v \mid D) = \arg\max_{v \in V} \sum_{h_i \in H} P(v \mid h_i) P(h_i \mid D)$$

Gibbs classifier

- Why can't we just use the Bayes optimal classifier every time?
  - Can be expensive if many hypotheses
- An alternative to the Bayes optimal classifier is a slightly less optimal procedure known as the Gibbs classifier
  1. Choose a hypothesis h from H at random according to the posterior distribution (i.e. P(h|D))
  2. Use h to predict the classification of the next instance x
- The misclassification error for the Gibbs algorithm is at most twice the expected error of the Bayes optimal classifier!

Bayesian (belief) networks

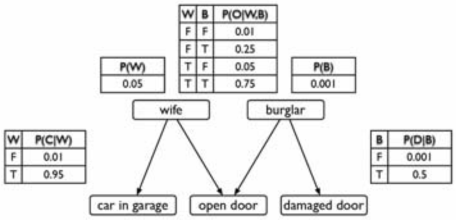
- X is conditionally independent of Y given Z if the probability distribution governing X is independent of the value of Y given the value of Z;
  - $(\forall x_i, y_j, z_k)\ P(X=x_i|Y=y_j, Z=z_k) = P(X=x_i|Z=z_k)$
  - $P(X,Y|Z) = P(X|Z)$
- A Bayesian network represents a set of conditional independence assertions:
  - Each node is asserted to be conditionally independent of its nondescendants, given its immediate predecessors

Bayesian networks

- A Bayesian network is a directed, acyclic graph
  - Nodes represent features or attributes
  - Arcs denote dependencies
  - Root node is the start node with no dependencies
- A node X is linked to another node Y provided that there is direct influence of X on Y

A burglar network

- How to compute the probability of a burglar given that we see that the door is open?



W	B	P(O|W,B)
F	F	0.01
F	T	0.25
T	F	0.05
T	T	0.75

P(W)
0.05

P(B)
0.001

W	P(C|W)
F	0.01
T	0.95

B	P(D|B)
F	0.001
T	0.5

Inference in Bayesian networks

- Inferences in Bayesian networks consist of computing P(X|E), the posterior probability of the query (e.g. burglar) given the evidence (e.g. open door):

$$p(x|e) = \frac{p(x,e)}{p(e)} = \alpha\, p(x,e) = \alpha \sum_y p(x,e,y)$$

  - y are non-evidence variables (wife, car in garage, door broken)
  - Summation is done over all non-evidence variables
- A joint distribution is defined by the product of the conditional probabilities:

$$p(z_1, z_2, \ldots, z_n) = \prod_{i=1}^{n} p(z_i \mid parents(z_i))$$

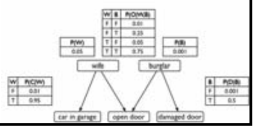
  - The product is taken over all variables in the network

Burglar network

- Probability of a burglar given an open door
- Straight summation:

$$p(b|o) = \alpha \sum_{w,c,d} p(o|w,b)p(c|w)p(d|b)p(w)p(b)$$

- The number of terms in the sum is exponential in the non-evidence variables!
- Variable elimination could be used



Variable elimination

$$p(b|o) = \alpha \sum_{w,c,d} p(o|w,b)p(c|w)p(d|b)p(w)p(b)$$

- When we've pulled out all the redundant terms we get:

$$p(b|o) = \alpha\, p(b) \sum_{d} p(d|b) \sum_{w} p(w)p(o|w,b) \sum_{c} p(c|w)$$

- We can also note the last term sums to one.
- In fact, every variable that is not an ancestor of a query variable or evidence variable is irrelevant to the query, so we get

$$p(b|o) = \alpha\, p(b) \sum_{d} p(d|b) \sum_{w} p(w)p(o|w,b)$$

Example: Pathfinder

- Pathfinder system (Heckerman et al., 1992).
  - Diagnostic system for lymph-node diseases
  - 60 diseases and 100 symptoms and test-results
  - 14,000 probabilities
  - Experts consulted to make net
    - 8 hours to determine variables
    - 35 hours for net topology
    - 40 hours for probability table values
- Pathfinder is said to outperform the world experts in diagnosis
- Being extended to several dozen other medical domains

Markov chains

- Desire: being able to deal with probabilistic sequences
- A Markov chain is described by the following:
  - a set of states S = $\{s_1,\ldots,s_n\}$
  - a set of transition probabilities $T(s_i,s_j) = P(s_j | s_i)$
  - an initial state $s_0 \in S$
- The Markov assumption
  - The state at time t, s_t, depends only on the previous state s_{t-1} and not the previous history, i.e.:

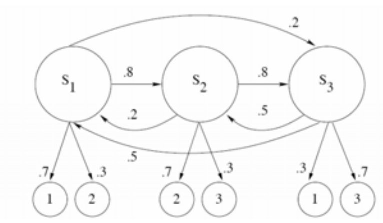
$$p(s_t|s_{t-1}, s_{t-2}, s_{t-3}, s_0) = p(s_t|s_{t-1})$$

Hidden Markov Models

- Extension of Markov chains to partially observable worlds
- A HMM is described by the following:
  - a set of states S = $\{s_1,\ldots,s_n\}$
  - a set of observations Z = $\{z_1,\ldots,z_t\}$
  - a set of transition probabilities $T(s_i,s_j) = P(s_j | s_i)$
  - a set of emission probabilities $O(z_i,s_j) = P(z_i | s_j)$
  - an initial state distribution $P_0(s)$
- We never know the true state of the system
- At each point in time, we get some observation z

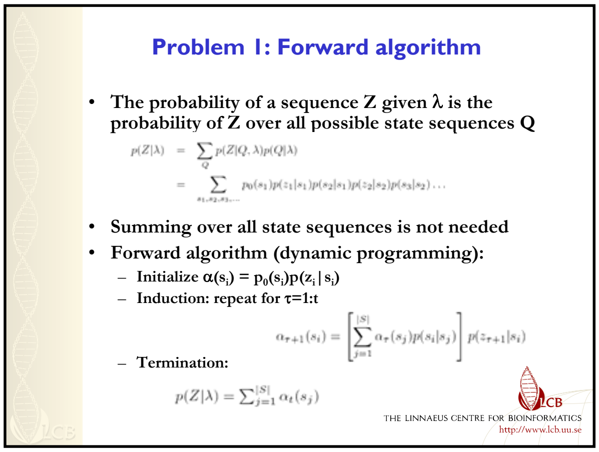
Discrete HMM example

- Three states $\{s_1,s_2,s_3\}$
- Three possible observations {1,2,3}



Rabiner's 3 famous questions

1. Given the observation sequence Z and a model $\lambda=(T,O,p_0)$, how do we efficiently compute $P(Z|\lambda)$?
2. Given the observation sequence Z and a model $\lambda=(T,O,p_0)$, how do we find the most probable state sequence (path in the HMM) Q = $s_1,\ldots,s_t$ (the sequence that best "explains" the observations)?
3. How do we adjust the model parameters $\lambda=(T,O,p_0)$ to maximize $P(Z|\lambda)$?



Problem 1: Forward algorithm

- The probability of a sequence Z given λ is the probability of Z over all possible state sequences Q

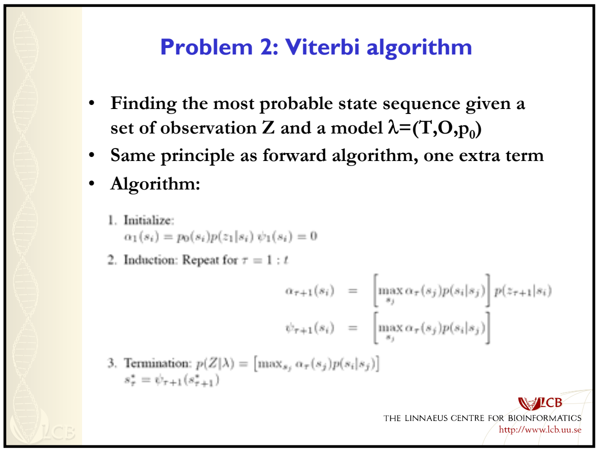
$$\begin{aligned} p(Z|\lambda) &= \sum_{Q} p(Z|Q,\lambda)p(Q|\lambda) \\ &= \sum_{s_1,s_2,s_3,\ldots} p_0(s_1)p(z_1|s_1)p(s_2|s_1)p(z_2|s_2)p(s_3|s_2)\ldots \end{aligned}$$

- Summing over all state sequences is not needed
- Forward algorithm (dynamic programming):
  - Initialize $\alpha(s_i) = p_0(s_i)p(z_i | s_i)$
  - Induction: repeat for τ=1:t

$$\alpha_{\tau+1}(s_i) = \left[\sum_{j=1}^{|S|} \alpha_\tau(s_j)p(s_i|s_j)\right] p(z_{\tau+1}|s_i)$$

  - Termination:

$$p(Z|\lambda) = \sum_{j=1}^{|S|} \alpha_t(s_j)$$



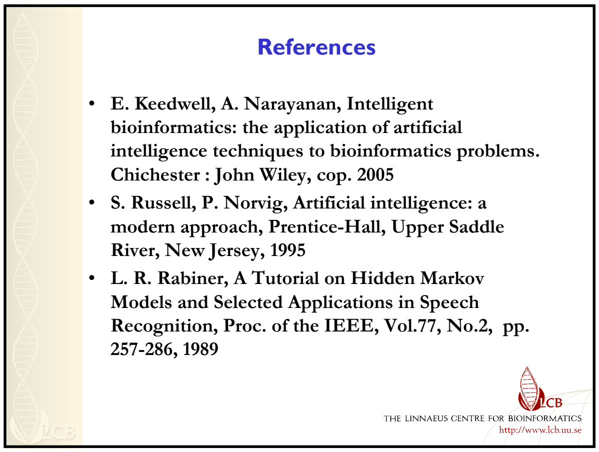
Problem 2: Viterbi algorithm

- Finding the most probable state sequence given a set of observation Z and a model $\lambda=(T,O,p_0)$
- Same principle as forward algorithm, one extra term
- Algorithm:

1. Initialize:
   $\alpha_1(s_i) = p_0(s_i)p(z_1|s_i)\ \psi_1(s_i) = 0$
2. Induction: Repeat for $\tau = 1 : t$

$$\begin{aligned} \alpha_{\tau+1}(s_i) &= \left[\max_{s_j} \alpha_\tau(s_j)p(s_i|s_j)\right] p(z_{\tau+1}|s_i) \\ \psi_{\tau+1}(s_i) &= \left[\max_{s_j} \alpha_\tau(s_j)p(s_i|s_j)\right] \end{aligned}$$

3. Termination: $p(Z|\lambda) = \left[\max_{s_j} \alpha_\tau(s_j)p(s_i|s_j)\right]$
   $s^*_\tau = \psi_{\tau+1}(s^*_{\tau+1})$



References

- E. Keedwell, A. Narayanan, Intelligent bioinformatics: the application of artificial intelligence techniques to bioinformatics problems. Chichester : John Wiley, cop. 2005
- S. Russell, P. Norvig, Artificial intelligence: a modern approach, Prentice-Hall, Upper Saddle River, New Jersey, 1995
- L. R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proc. of the IEEE, Vol.77, No.2, pp. 257-286, 1989