CSC 448 Bioinformatics Algorithms Alexander Dekhtyar

(Hidden) Markov Models for Bioinformatics

Markov Models

Stochastic Processes. A stochastic process is a collection of *random variables* that represents evolution of a system over time.

Consider a collection of random variables $V_1, \ldots, V_m$ over some domains $dom(V_1), \ldots, dom(V_m)$ of values. We consider the discrete time case, where time t comes from a discrete set $t_1, \ldots, t_n, \ldots$ of values. Let $v_{i,t} = V_i(t)$ be the value of random variable V_i at time t.

A state of the stochastic system $D = \langle V_1, \ldots, V_m \rangle$ at time t, denoted S_t is $\langle v_{1,t}, \ldots v_{m,t} \rangle$.

For each random variable $V_i \in D$, its transition probability function $Pr_i()$ is defined as

$$Pr_i : 2^{dom(V_1) \times \ldots \times dom(V_n) \times T} \times dom(V_i) \longrightarrow [0,1],$$

that is, a probability of random variable V_i taking value $a \in dom(V_i)$ at time t potentially depends on all prior states of the system.

In practice, generic stochastic processes are hard to reason with, and restrictions on their power are considered. The key restriction is defined by a Markov Property.

Markov Property. A stochastic process D has a Markov Property if the transition probability function depends only on the current state of the system, i.e., if for each $V_i \in D$, the transition probability function has the shape:

$$Pr_i : dom(V_1) \times \ldots \times dom(V_n) \times T \times dom(V_i) \longrightarrow [0,1].$$

In other words, the probability of V_i taking value $a \in dom(V_i)$ at time t_j can be expressed as a conditional probability distribution function:

$$Pr_i(V_i = a, t_j | S_1, \ldots, S_{j-1}) = Pr_i(V_i = a, t_j | S_{j-1}) = Pr_i(V_i = a, t_j | = v_{1,j-1} = x_1, v_{2,j-1} = x_2, \ldots v_{m,j-1} = x_m).$$

Markov Process. A Markov process is a stochastic process that possesses a Markov property.

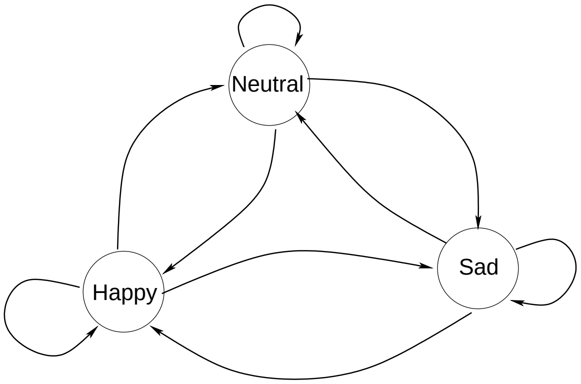

Figure 1: Graphical representation of the human mood changes.

Markov Processes are classified based on whether the domains of random variables are continuous or discrete, and based on whether the time is continuous or discrete.

In Bioinformatics, of special interest are stochastic systems with discrete random variables and discrete time. Stochastic processes of these type are called Markov Chains.

Markov Chains

A Markov Chain is a stochastic model emulating a process where a state of a random variable at each moment of time stochastically (i.e., through a conditional probability distribution) depends on its state at a previous time.

Formal definition. A Markov chain is a sequence of random variables $X_1, \ldots, X_n, \ldots$ over the same domain $\mathcal{X} = dom(X_1) = dom(X_2) = \ldots dom(X_n) = \ldots$ such that

$$Pr(X_{i+1} = x | X_1 = x_1, X_2 = x_2, \ldots X_i = x_i) = Pr(X_{i+1} = x | X_i = x_i)$$

(assuming $Pr(X_1 = x_1, \ldots X_i = x_i) > 0$).

Here, X_1 represents the state of the Markov chain at time t_1, X_2 – its state at time t_2, and so on.

Example. Consider a simple model explaining how people's moods change over time. For simplicity we consider three possible moods : happy, neutral and sad. We assume that one's mood can either stay the same, or change to one of the other two over time. Generally speaking, a human can transition from any mood to any other mood.

We can use a graph representation of this scenario. In Figure 1, the moods are represented as nodes, and all possible mood transitions are represented as edges.

Humans tend to show a certain level of resilience in their moods - so if a person is happy at a specific moment of time, it would be more difficult for them to become sad the next moment of time, then to just stay happy, or switch to a neutral mood. We can attempt to quantify mood transitions, by coming up with a conditional probability disitribution

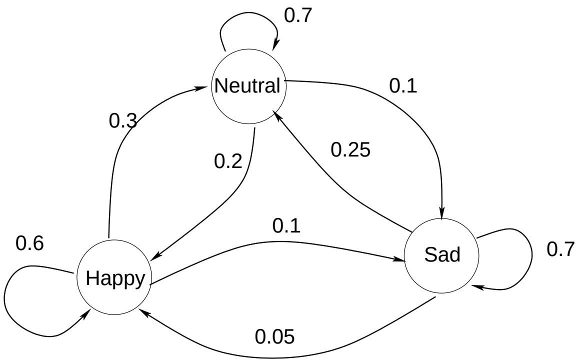

Figure 2: Markov Chain for Human Moods.

$$Pr(Mood_{i+1} = m).$$

In a stochastic model with a Markov property, this probability only depends on one's mood at a previous moment of time:

$$Pr(Mood_{i+1} = m | Mood_i = m').$$

The transition probabilities may be collected either following an empirical/frequentist approach (observe a lot of humans and their moods and find out the frequency of mood changes from one state to another for any given pair of states), or following a Bayesian approach (ask experts to estimate the probabilities directly).

A sample transition probability function is documented in the table below:

M_{i+1}	M_i		
	Happy	Neutral	Sad
Happy	0.6	0.2	0.05
Neutral	0.3	0.7	0.25
Sad	0.1	0.25	0.7

Essentially, this model assumes that humans are relatively stable in their states. From the extreme states (happiness, sadness) they are more likely to recover into the neutral state, than directly to the opposite extreme. Becoming sad or happy is somewhat difficult - neutral state is very stable, but it is more in the human nature to become happy. On the other hand sadness is harder to quit than happiness.

A graphical model representing this Markov chain is presented in Frigure 2

Computing probabilities of state sequences. The key operation one can do with Markov chains is checking the probability of a certain sequence of states.

Assuming an initial probability distribution function $Pr(X_1 = x)$ that is unconditional (the probability distribution for the initial state), the probability for a sequence of states $x_1, x_2, \ldots, x_n$ of the Markov chain can be computed as follows:

$$Pr(X_1 = x_1, \ldots X_n = x_n) = Pr(X_1 = x_1)Pr(X_2 = x_2 | X_1 = x_1) \ldots Pr(X_n = x_n | X_{n-1} = x_{n-1}.$$

Example. Consider the human moods Markov chain described above. Let the initial probability distribution be $Pr(\mathsf{Neutral}) = 0.5, Pr(\mathsf{Happy}) = 0.3, Pr(\mathsf{Sad}) = 0.2$. We ask ourselves, which sequence of human moods is more probable:

Neutral, Neutral, Happy, Happy, or
Happy, Neutral, Sad, Sad

We compute the first probability as follows:

$$Pr(\mathsf{Neutral}, \mathsf{Neutral}, \mathsf{Happy}, \mathsf{Happy}) = Pr(\mathsf{Neutral}) \cdot Pr(\mathsf{Neutral}|\mathsf{Neutral}) \cdot Pr(\mathsf{Happy}|\mathsf{Neutral}) \cdot Pr(\mathsf{Happy}|\mathsf{Happy})$$

$$= 0.5 \cdot 0.7 \cdot 0.2 \cdot 0.6 = 0.042.$$

The probability of the second sequence can be computed similarly:

$$Pr(\mathsf{Happy}, \mathsf{Neutral}, \mathsf{Sad}, \mathsf{Sad}) = Pr(\mathsf{Neutral}) \cdot Pr(\mathsf{Happy}|\mathsf{Neutral}) \cdot Pr(\mathsf{Sad}|\mathsf{Happy}) \cdot Pr(\mathsf{Sad}|\mathsf{Sad})$$

$$= 0.3 \cdot 0.3 \cdot 0.1 \cdot 0.7 = 0.0063.$$

As can be seen from these two computations, the Neutral, Neutral, Happy Happy sequence of human moods is much more probable according to our model.

Hidden Markov Models

In the example above, the actual states of the Markov model (happy, sad, neutral) assumed to be observable.

However, often, we are in a situation where we cannot directly observe the state (i.e., current value) of a random variable, but rather, can observe some *stochastic effect* of a system being in a specific state.

Example. Consider the situation where a Conclave of Cardinals must meet in Vatican City to elect new Pope. We know that the conclave meets and casts ballots for the Pope. At each moment of time the Conclave either has reach a decision and elected a new Pope, or has not reached a decision and continues the deliberations.

The Conclave itself is a closed-door affair and no one outside of the Conclave itself knows whether the Pope has been elected at each moment of time. However, the Conclave announces the election of the new Pope via the white smoke coming out of the chimney in the Sistine chapel.

We have two possible states: New Pope Elected and No New Pope. We have three possible observations we can make: White Smoke, Black Smoke and No Smoke. We can make certain observations:

- Black Smoke means No New Pope has been elected.
- White Smoke means New Pope Elected.
- No Smoke most likely means No New Pope yet, but sometimes it is possible that the Conclave has reached its decision, but has not announced it yet.
- Once the decision to elect a new Pope is made, the Conclave will try to be fast in announcing it.

Based on these observations we can create the followin *emission probabilities* for the observations:

State	Observation		
	Black Smoke	No Smoke	White Smoke
No New Pope	0.1	0.9	0.0
New Pope Elected	0.0	0.67	0.33

In addition to this, we have the following transition probability table:

Current State	Previous State	
	No New Pope	New Pope Elected
No New Pope	0.95	0.0
New Pope Elected	0.05	1.0

Markov Models with hidden states are called Hidden Markov Models (HMMs). Basically, an HMM specifies two sets of probabilities: the transition probabilities between (now directly unobservable) states, and the *emission probabilities* - the unconditional probability distributions of emitting a specific **observable** output when entering a given (hidden) state.

Viterbi Algorithm

The key question of interest given a Hidden Markov Model M is specified as follows:

> Given a sequence $A_1, A_2, \ldots, A_k$ of observed outputs of M what is *the most likely* sequence of states $X_1, \ldots, X_k$ of to have produced this output?

Viterbi Algorithm. The problem of finding the most likely sequence of states given the set of outputs can be solved using dynamic programming techniques. The first person to propose the dynamic programming solution was Andrew Viterbi. The algorithm bears his name.

Algorithm Inpupts. Viterbi Algorithm takes as input the following information:

- State space $S = \{s_1 \ldots s_n\}$ (these are hidden states)
- Output space $Y = \{y_1, \ldots y_m\}$ (these are observable outputs)
- Initial distribution probability π. $\pi(k) = \pi_k = Pr(S_1 = s_k)$
- Transition probability matrix $a[1..n, 1..n]$. $a[i, j] = Pr(s_j|s_i)$
- Emission probability matrix $P[1..m, 1..n]$. $P[l, j] = Pr(y_l|s_j)$
- Output sequence $Y_1, \ldots, Y_n$ of outputs $Y_k \in Y$.

Algorithm Outputs. The algorithm outputs a sequence $S_1, \ldots, S_T$ of T states that is the most likely cause of the observed outputs $Y_1, \ldots, Y_T$.

Dynamic Programming Recurrence. The Markov Property suggests that if our HHM is in the state s at some time t_i, then we can simply assume that it has arrived to this state **via the most probable path**: i.e., the actual path by which we arrived as state s at time t_i will not affect the probability of transitioning to state s' at time t_{i+1}. This principle can be exploited in a dynamic programming scheme.

Let $V[1..T, 1..n]$ be an array of values from $[0,1]$. We interpret $V[t,k]$ as *the probability of the most probable state sequence responsible for the observations* $Y_1, \ldots Y_t$ ***that ends in state*** s_k.

We have the following recurrence relations for computing $V[t,k]$:

$$V[1,k] = P[Y_1, k] \cdot \pi_k.$$

$$V[t,k] = P[Y_t, k] \cdot \max_{i=1..n}(a[i,k] \cdot V[t-1, i])$$

Given the completed matrix $V[\_,\_]$, we recover the most probable path as follows.

Let $Ptr(k,t)$ be set to $r \in [1..n]$, where $a[r,k] \cdot V[t-1,r] = \max_{i=1..n}(a[i,k] \cdot V[t-1,i])$ (i.e, this is the prior state that gives us the highest probability of transitioning to state k of a given output at the specific point). We recover the entire sequence of states $S_1, \ldots, S_T$ as follows:

$$S_T = arg \max_{i=1..n}(V[T,i])$$
$$S_{t-1} = Ptr(S_T, t)$$