CS 252 - Markov Chains
Additional Reading 1
and
Homework problems

1 Markov Chains

The first model we discussed, the DFA, is deterministic—it's transition function must be total and it allows for transition from one state to exactly one other state at each processing step. Our second model, the NFA, relaxed this deterministic constraint in a "magical" sort of way, and provided us with some interesting new insights into regular languages, and the potential for additional efficiency of computation (though this is something we have ignored for now). Even though NFAs are "magical" in the sense that we can't actually implement them, we know we can simulate them with a DFA that has (in the worst case) an exponential number of states. What if we relaxed our determinism in another way? What if, instead of having to track an exponential number of states, we allowed the transition function to be *probabilistic*—at each computational step, we allow a transition to only a single next state, but we allow it to happen with some probability over many different states? In essence, we avoid the exponential state explosion at the cost of introducing uncertainty into our model—we no longer can be certain about what path the computation follows and in what state the machine is in at any point in the computation. Such a model exists and it is called a Markov chain.

Figure 1 shows a Markov chain that models the flipping of a fair coin. Probablilities on the edges show that from the *start* state, there is a 50% chance of transitioning to the *heads* state and a 50% chance of transitioning to the *tails* state. Transitioning from the *heads* state

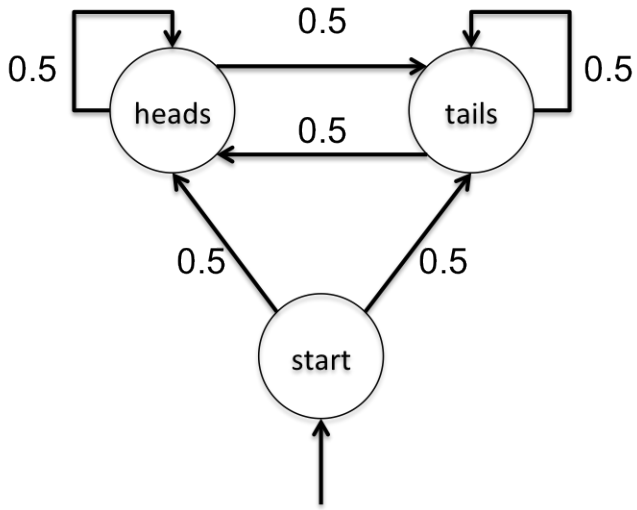

Figure 1: A Markov chain C_1 that models the flipping of a fair coin

to the *tails* state is as likely as remaining in the *heads* state, and transitioning from the *tails* state to the *heads* state is as likely as remaining in the *tails*.

Using this model, we can simulate a series of (fair) coin flips and calculate their probabilities. We will use the notation $P(x|y)$ to represent the *conditional* probability of event x happening, given that event y has happened. So, given the model of Figure 1, we can write

- $P(heads|start) = 0.5$
- $P(tails|start) = 0.5$
- $P(heads|heads) = 0.5$
- $P(tails|heads) = 0.5$
- $P(heads|tails) = 0.5$
- $P(tails|tails) = 0.5$

What is the probability $P(H)$ of flipping the coin a single time and having it land heads up? It is $P(heads|start) = 0.5$, which is represented by beginning in the *start* state and transitioning to the *heads* state in our diagram[1]. What about the probability of flipping three heads in a row $P(HHH)$? The probability of both event x *and* event y happening is computed by multiplying their individual probabilities together, $P(xy) = P(x)P(y)$[2]. Thus the probability of flipping three heads in a row is

$$P(HHH) = P(heads|start)P(heads|heads)P(heads|heads) = 0.5 \times 0.5 \times 0.5 = 0.125$$

Figure 1 shows a computation tree for all possible sequences of three or fewer coin flips. Note that each unique path in the tree corresponds to a unique sequence of events (coin flip results), and the probability of a particular sequence is computed by multiplying the probabilities on the edges of the path.

What about the probability of flipping the coin three times and having exactly two of them be heads? This is a bit trickier, as there are several possible ways this can happen (HHT, HTH, THH). The probability of either event x *or* event y happening is computed by adding their individual probabilities together, $P(x \text{ or } y) = P(x) + P(y)$. So, to compute the probability of seeing exactly two heads in three coin flips, we want to compute the probability of each path in the tree that has exactly two heads and one tail and then add

[1] Because the first event in a sequence cannot be conditioned on any prior event, this initial probability $P(heads|start)$ is sometimes written as the *unconditional* probability $P(heads)$; here, we write all probabilities as conditional and use a special start state that does not have an associated event and whose unconditional probability $P(start) = 1$.

[2] Note that this is actually only true when x and y are *independent* events. In general, $P(xy) \leq P(x)P(y)$, but for our discussion, we will only be concerned with the independent case, as, in particular, it is *the* defining characteristic of Markov chains.

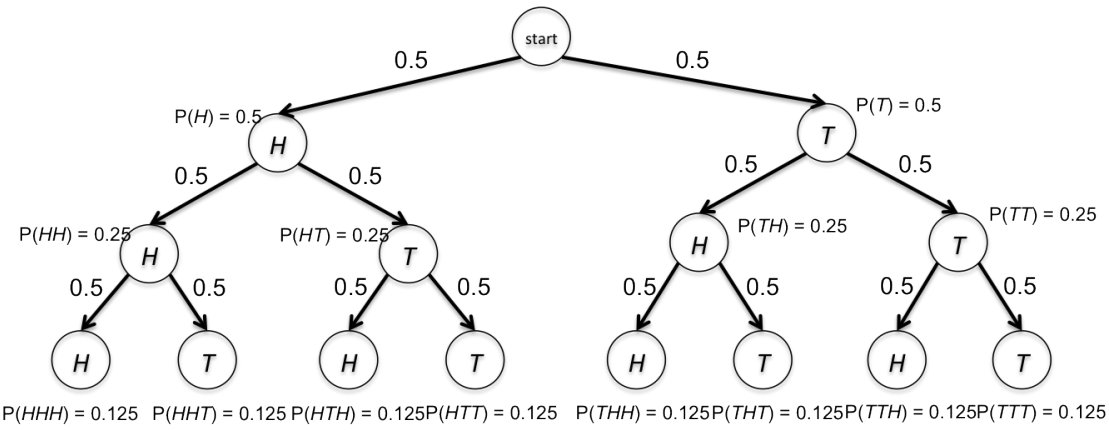

Figure 2: A partial tree showing computation paths for the Markov chain C of Figure 1.

them together. In this case, because the coin is fair, each path will have the same probability and the total probability of seeing exactly two heads is

$$P(\text{exactly 2 heads}) = P(HHT) + P(HTH) + P(THH) = 0.125 + 0.125 + 0.125 = 0.375$$

1.1 Formal Definition of Markov Chains

Like in previous readings, we have started with some informal discussion. We will now give a formal definition of Markov chains, similar to those we've seen in the past.

Definition 0.0.1. A ***Markov Chain*** is a 5-tuple $(Q, \Sigma, \delta, \gamma, q_0)$, where

1. Q is a finite set called the ***states***
2. Σ is a finite set called the ***alphabet***
3. $\delta : Q \cup \{q_0\} \times Q \to [0 \dots 1]$ is the ***state transition function***
4. $\gamma : Q \to \Sigma$ is the ***state emission function***
5. $q_0 \notin Q$ is the ***start state***

Note the similarities to our definitions for DFAs and NFAs, but also note some important differences. Like previous models, Markov chains have a set of states, a set of alphabet symbols, a transition function and a start state. However, unlike earlier models, the start state is a special kind of state that is *not* included in the set of states Q. Also, the transition function is (as you might expect) again different from what we've seen in the past. This time, it maps a pair of states to a real number between 0 and 1, which we will interpret as a probability. That is, the transition function δ is now interpreted as the probability that

the Markov chain will transition from one state to another. Note that this function takes as input only two states: the state from which the transition occurs and the state to which it goes. Another way to think about this is to realize that the probability of transition does not depend on any states occupied prior to the current state—history beyond the current state does not matter in determining the transition probability. This property is called *Markovian*, after Andrey Markov, a Russian mathematician interested in stochastic processes, and it is this property that gives this model its name. In addition to the now familiar transition function δ, we now have an entirely new function γ, called the *emission function*. Because we now associate a probability with transitions, we will associate alphabet symbols with states instead. The emission function tells us how symbols are associated with states—it maps a symbol from Σ to each state in Q. Each time the Markov chain visits a state, it consumes (or emits) its associated alphabet symbol. Finally, note that Markov chains do not have a set of accept states; every state in the model is a valid ending state.

Revisiting the Markov chain C of Figure 1, the formal definition of C is $(Q, \Sigma, \delta, \gamma, q_0)$, where

1. $Q = \{heads, tails\}$
2. $\Sigma = \{H, T\}$
3. δ is given as

	heads	*tails*
start	0.5	0.5
heads	0.5	0.5
tails	0.5	0.5

4. γ is given as

q	$\gamma(q)$
heads	H
tails	T

5. $q_0 = start$

1.2 Computing with Markov Chains

Similar to DFAs, Markov chains can be thought of as language recognizers or as machines that accept some strings and reject others. However, similar to regular expressions, Markov chains can also be thought of as generators of strings, and, in fact, it is the latter that is perhaps the more common case. When used as acceptors, we say that the Markov chain is a *discriminative* model, and when used as a generator, we say that the Markov chain is a *generative* model.

1.2.1 Discriminative models

When used discriminatively, Markov chains have a formal definition of computation that is similar to those we've seen for DFAs and NFAs. Let $C = (Q, \Sigma, \delta, \gamma, q_0)$ be a Markov chain and w be a string over the alphabet Σ. Then we say that C accepts w if we can write $w = y_1 y_2 \cdots y_m$, where each $y_i \in \Sigma$, and one or more sequences of states $r_0^1, r_1^1, \ldots, r_m^1$, $r_0^2, r_1^2, \ldots, r_m^2, \ldots, r_0^n, r_1^n, \ldots, r_m^n$ exist in $Q \cup \{q_0\}$ with three conditions:

1. $r_0 = q_o$
2. $\sum_{i=1}^{n} \prod_{j=0}^{m-1} \delta(r_j^i, r_{j+1}^i) \geq \theta \geq 0$
3. $\gamma(r_j^i) = y_j$, for $i = 1, \ldots, n$ and $j = 1, \ldots, m$

Condition 1 says that the machine starts out in the start state. Condition 2 says that the total probability over all sequences of states is greater than some positive threshold θ (which can be chosen in many ways, and might, for example, be a function of the length of the string). Condition 3 says that for each sequence, each state in the sequence is associated with the appropriate alphabet letter in the string w—in other words, it says that each sequence of states can generate the string w.

1.2.2 Generative models

When used generatively, Markov chains give us the ability to compute the probability of various sequences (or subsequences) occurring. Let $C = (Q, \Sigma, \delta, \gamma, q_0)$ be a Markov chain, $w = y_1 y_2 \cdots y_m$, where each $y_i \in \Sigma$, be a string over the alphabet Σ and $R = \{(r_1, \ldots, r_m) | r_i \in Q, \gamma(r_i) = y_i, 1 \leq i \leq m\}$ be the set of state sequences that can generate w. Then, we can compute the probability of C generating w as

$$P(w) = \sum_{r \in R} \delta(q_0, r_1) \prod_{j=1}^{m} \delta(r_j, r_{j+1})$$

The total probability is computed by summing over all generating sequences, and, for each sequence, computing the probability of the entire sequence as a product of conditional probabilities, each transition to a new state conditioned on the previous state.

1.3 Exercises

Exercise 1.1. Consider the transition diagram for C_1 in Figure 1.

a. Draw a modified transition diagram for C_1 to represent a coin that is biased to land on heads 75% of the time.

b. Compute the probability of flipping the coin three times and seeing the sequence tails, heads, tails.

c. Compute the probability of flipping the coin three times and seeing an odd number of tails.

Exercise 1.2. Given the following formal description of a Markov model, $C = (Q, \Sigma, \delta, \gamma, q_0)$

1. $Q = \{0, 1\}$
2. $\Sigma = \{0, 1\}$
3. δ is given as

	0	*1*
start	0.25	0.75
0	0.25	0.75
1	1.0	0.0

4. γ is given as

q	$\gamma(q)$
0	0
1	1

5. $q_0 = start$

a. Draw a transition diagram for C.

b. Considering C as a generative model, what string of length 4 is most likely to be generated?

c. What string is least likely to be generated?

d. Considering C as a discriminative model, let $\theta = (0.5)^m$, where m is the length of the string in question. Will the the string 1001 be accepted by C? Why or why not?

e. Will the string 0010 be accepted by C? Why or why not?

f. List all strings of length 4 or less that are in $L(C)$?

g. What happens to $L(C)$ if we make θ a constant?