CS265/CME309: Randomized Algorithms and Probabilistic Analysis
Lecture #13: Introduction to Markov Chains, and a Randomized Algorithms for 2-SAT

Gregory Valiant*

November 6, 2019

1 Markov Chains

The formal development of the theory of Markov chains was initially motivated by Markov's observation that sequences of *dependent* events often exhibited similar sorts of concentration in their long-term behavior, as sequences of independent events. Specifically, Markov examined the statistics of word frequencies (and vowel/consonant frequencies) in Pushkin's novel *Eugene Onegin*, and found that, for example, the statistics of these frequencies in different parts of the novel were all very similar. This would be expected if words were chosen independently; however, language has extremely strong dependencies—the next word is very dependent on the previous words. How can we model such dependencies, and how can we understand the long-term behavior of these processes?

Definition 1. *A sequence of random variables indexed by the integers, $X_0, X_1, X_2, \ldots$ is a* Markov Chain *or* Markov process *if for all t, the distribution of X_t conditioned on all of $X_0, \ldots, X_{t-1}$ is equal to the distribution of X_t conditioned on only X_{t-1}. Namely, for any sequence of values c_i,*

$$\Pr[X_t = c_t | X_0 = c_0, X_1 = x_1, \ldots, X_{t-1} = c_{t-1}] = \Pr[X_t = c_t | X_{t-1} = c_{t-1}].$$

This property is often referred to either as the "Markov property" or the "memoryless property", since it implies that, to know the distribution of future random variables, all you need to know is the current state of the current random variable—there is no need to remember the history of previous random variables.

While it is certainly possible to have Markov chains for which $\Pr[X_t = c_t | X_{t-1} = c_{t-1}]$ is a function of t and the values c_t and c_{t-1}, in many cases this probability is independent of t, in which case we refer to the chain as being *time homogeneous*.

Definition 2. *A Markov chain $X_0, X_1, \ldots$ is* time homogeneous *if, for all values a, b and all times $t \geq 0$,*

$$\Pr[X_t = a | X_{t-1} = b] = P_{b \to a},$$

for some probability $P_{b \to a}$ that does not depend on t.

For time homogeneous Markov chains, it is convenient to think of these conditional probabilities as forming a *transition matrix*, P, whose rows and columns are indexed by the possible values that the variables $\{X_t\}$ can take, and entry $P_{b,a} = \Pr[X_t = a | X_{t-1} = b]$. This matrix representation is especially convenient, because evolving the chain by one timestep corresponds to multiplication by P. Supposing the value of X_t is drawn according to a distribution, we can represent this distribution as a (row) vector v, whose coordinates are indexed by the same values as matrix P. The vector-matrix product vP corresponds to the distribution of X_{t+1}, given that X_t was selected according to the distribution represented by v. Hence, given that $X_0 = c_0$ the distribution of $X_t | X_0 = c_0$ is given by vP^t, where v is the vector with 1 in the coordinate corresponding to c_0.

Example 3. *Consider the following time homogeneous Markov chain representing my transitions between three states: $\{Awake, Tired, Sleeping\}$. We can represent this chain either in graphical form, where the directed weighted edges denote the transition probabilities (and hence the sum of weights of all edges leaving a state—including the self-loops—must be 1).*

TODO[add figure, finish example]

2 Randomized 2-SAT

As many of you have seen in previous classes, there is an efficient deterministic algorithm for 2-SAT. Here, we will describe a very intuitive efficient randomized algorithm, which resembles the randomized algorithms we discussed in the constructive Lovasz Local Lemma, but is slightly different. The analysis of the algorithm will leverage the analysis of a Markov chain.

Algorithm 4. RANDOMIZED 2-SAT ALGORITHM

Input: 2-SAT formula over n variables, $x_1, \ldots, x_n$.

- *Let A_0 denote some assignment to the variables (e.g. all set to false).*
- *For $t = 1$ to cn^2 (for some parameter $c > 1$):*
  - *If there exists an unsatisfied clause, arbitrarily (e.g. the lowest index such clause) select one. Let x_i, x_j denote the two variables in this clause.*
  - *With probability $1/2$, let A_t be identical to A_{t-1} except with the assignment to variable i flipped, and with the remaining probability $1/2$ let A_t be identical to A_{t-1} except with the assignment to variable j flipped.*
- *If no satisfying assignment has been found within cn^2 flips, then return ''The formula is not satisfiable''.*

Theorem 1. *If the formula is satisfiable, the above algorithm will return a satisfying assignment with probability at least $1 - \frac{1}{2^{c/2}}$. If the formula is not satisfiable, the algorithm will (with probability 1) return "The formula is not satisfiable".*

If the formula is not satisfiable, the algorithm will never return a satisfying assignment. Hence, for the remainder of our analysis, we will assume the formula is satisfiable, and analyze the probability that the algorithm fails to find a satisfying assignment. To this end, let S denote a satisfying assignment (there might be multiple satisfying assignments, in which case let S be any one of them). Consider the random variables, $X_0, X_1, \ldots,$ where X_i denotes the number of variables whose assignment in A_i agrees with the assignment in S. Hence $X_i \in \{0, 1, \ldots, n\}$, and if $X_i = n$, then $A_i = S$ and we must have found a satisfying assignment.

Lets analyze the behavior of the X_i's. First, note that if A_{t-1} is not a satisfying assignment, then the clause considered in the tth step of the algorithm must have at least one of the two variables taking the opposite value from their assignment in S. Hence, with probability at least $1/2$, the algorithm will flip the "right" one, and $X_t = X_{t-1} + 1$, and in the remaining case, $X_t = X_{t-1} - 1$. Additionally, If A_{t-1} is not a satisfying assignment and $X_{t-1} = 0$, then with probability 1, $X_t = 1$.

The sequence $X_0, X_1, \ldots$ is *not* a Markov process, because the sequence of values $X_0, X_1, \ldots, X_{t-1}$ contains information about which variables were flipped, and hence the distribution of X_t conditioned on the entire history might be different than the distribution conditioned on just X_{t-1}. Our analysis will proceed by considering a related process, $Y_0, Y_1, \ldots$, defined to satisfy the following properties:

- $Y_0 = X_0$, and for all t, $Y_t \leq X_t$.
- If $Y_{t-1} \in \{1, \ldots, n-1\}$, then $\Pr[Y_t = Y_{t-1} + 1 | Y_{t-1}] = 1/2$, and $\Pr[Y_t = Y_{t-1} - 1 | Y_{t-1}] = 1/2$, and $\Pr[Y_t = 1 | Y_{t-1} = 0] = 1$.

Given that we can construct such a sequence of Ys, the expected number of steps until either $X_t = n$ or we have found a satisfying assignment (other than S) is bounded by the expected time t before $Y_t = n$. Since Y_t is a relatively simple Markov chain, analyzing this expected time will not be too hard.

To see why the sequence $Y_0, Y_1, \ldots$ can be constructed so as to satisfy the above conditions, note that when $X_{t-1} > 0$ $X_t = X_{t-1} \pm 1$, and the probability of being $X_{t-1} + 1$ is at least a half, hence there is "probability to spare" to ensure that $Y_t = Y_{t-1} + 1$ with probability exactly $1/2$. Formally, we can construct such a sequence by defining a joint distribution over the pairs (X_i, Y_i). We will talk more about this sort of thing next week when we discuss *couplings*, though for the sake of completeness, consider defining Y_t as a function of A_{t-1}, Y_{t-1}, and X_t as follows: If $Y_{t-1} = 0$, then $Y_t = 1$. Otherwise, let p_t denote $\Pr[X_t = X_{t-1} + 1 | A_{t-1}]$, and note that $p_t \geq 1/2$. We define Y_t such that in the $1 - p_t$ probability event that if $X_t = X_{t-1} - 1$, then $Y_t = Y_{t-1} - 1$. In the remaining $p_t \geq 1/2$ probability case, we flip a coin with probability $(1/2)/p_t$ we set $Y_t = Y_{t-1} + 1$, and with the remaining probability $(p_t - 1/2)/p_t$ we set $Y_t = Y_{t-1} - 1$. This definition of Y_t, while slightly pedantic, clearly satisfies the bulleted properties specified above.

Lemma 5. *Given the Markov chain $Y_0, Y_1, \ldots$ defined by the bulleted properties above, then for any initial value of $Y_0 \in \{0, \ldots, n\}$,*

$$\mathbf{E}[\min(t : Y_t = n)] = \mathbf{E}[\min(t : Y_t = n) | Y_0 = 0] = n^2.$$

Proof. Letting $r_i = \mathbf{E}[\min(t : Y_t = n) | Y_0 = i]$, we have that $r_0 = 1 + r_1$, since when the chain starts at value 0, after 1 timestep, the value is 1, and then the expected additional time until it hits n is simply r_1, by the Markov property. Similarly, for $i > 0$, $r_i = 1 + \frac{1}{2} r_{i-1} + \frac{1}{2} r_{i+1}$. Additionally,

$r_n = 0$, by definition. Hence we have $n+1$ linear equations in $n+1$ variables, $r_0, \ldots, r_n$, and hence there is a unique solution. I now claim that the unique solution has $r_i = r_{i+1} + 2i + 1$. To see that this satisfies the equations, note that this certainly satisfies the equation $r_0 = 1 + r_1$. For the other equations, consider plugging this in for r_{i-1} in the right side of the equation $r_i = 1 + \frac{1}{2}r_{i-1} + \frac{1}{2}r_{i+1}$: This yields $1 + \frac{1}{2}(r_i + 2(i-1)+1) + \frac{1}{2}r_{i+1} = \frac{1}{2}(r_i + 2i + 1 + r_{i+1})$, and if we replace the $2i + r_{i+1} + 1$ by r_i, then this expression simplifies to r_i, showing that $r_i = r_{i+1} + 2i + 1$ is a solution to the system of equations, with $r_n = 0$. Hence $r_i = \sum_{j=i}^{n-1}(2i+1)$, and hence $r_i \leq r_0 = 1+3+5+\ldots = n^2$. $\square$

To finish our proof of Theorem 1, note that by Markov's inequality, no matter the value of Y_t, with probability at least $1/2$, there will be a $t' \leq t + 2n^2$ for which $Y_{t'} = n$. Hence, for a satisfiable formula, the probability that the algorithm fails to find a satisfying formula during the first cn^2 steps is the probability that ALL $c/2$ blocks of $2n^2$ steps fail to find a formula, which is at most $1/2^{c/2}$.

2.1 Extensions to 3-SAT

We can try to apply the natural analog of the randomized 2-SAT algorithm to an instance of 3-SAT. Suppose we try to analyze it in the same way, and let X_t denote the number of variables whose assignment differs between the assignment at step t, and the fixed satisfying assignment, S. Since we are dealing with a 3-SAT formula, we will have that $\Pr[X_t = X_{t-1} + 1] \geq 1/3$, and $\Pr[X_t = X_{t-1} - 1] \leq 2/3$, instead of the case of 2-SAT where these bounds were $1/2$. Now, since the probability of making progress might only be $1/3$, if we start at $X_t = 0$, (or even $X_t = n/2$, corresponding to initializing via a random guess), then the expected time until we reach $X_t = n$ will be exponential in n, as we would expect given that 3-SAT is NP-hard. We will see techniques for formalizing this, though the intuition is that we do really expect to be losing ground fast than we gain ground, and hence if we end up at n, we must have "beaten the odds" a significant number of times, which will happen only with an inverse exponential probability.

It turns out, however, that a variant of this algorithm due to Schoning [1], *does* yield the best know runtime for 3-SAT. That runtime is still exponential, but is at most $O(1.334^n)$ improving upon the previous best of (1.36^n). The algorithm is slightly different than the 2SAT algorithm, in that it frequently re-initializes the assignment.

Algorithm 6. SCHONING'S RANDOMIZED 3-SAT ALGORITHM
Input: 3-SAT formula over n variables, $x_1, \ldots, x_n$.

- *Repeat the following:*
  - *Select a uniformly random assignment to the n variables.*
  - *For $t = 1$ to $3n$:*
    - *If there exists an unsatisfied clause, arbitrarily (e.g. the lowest index such clause) select one, and flip the assignment to one of the three variables (chosen uniformly at random) in the offending clause.*

The intuition for the frequent randomized restarting is the following: In the analysis, if we make positive progress $1/3$ of the time, and negative progress $2/3$ of the time, then if we are close to a satisfying solution, S, at some time t, for example if $X_t = n-1$, then conditioned on not having reached n within the next, say, w timesteps, we would expect $X_{t+w} \approx n + (\frac{1}{3} - \frac{2}{3})w$, and hence we expect to be even worse than we started. Hence, the algorithm starts by randomly guessing, hopes that the initial guess is close to a satisfying assignment S, in which case there is a not-too-small probability that the random procedure might find the satisfying assignment. If, however, after $3n$ steps, we haven't found the satisfying assignment, then the analysis says that we are probably not so close, and hence it would be better to randomly restart, rather than continuing down the hole that we might be in. The analysis of this algorithm closely mirrors the 2SAT analysis, and the entire paper [1] is 4 or 5 pages!

NOTE: At the end of lecture, we defined "irreducible" Markov Chains; this will be at the beginning of the notes for next lecture, along with the other definitions necessary to state the Fundamental Theorem of Markov chains.

References

[1] T Schoning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *40th Annual Symposium on Foundations of Computer Science*, pages 410–414, 1999.