Markov Chains

Eugene Kolinko

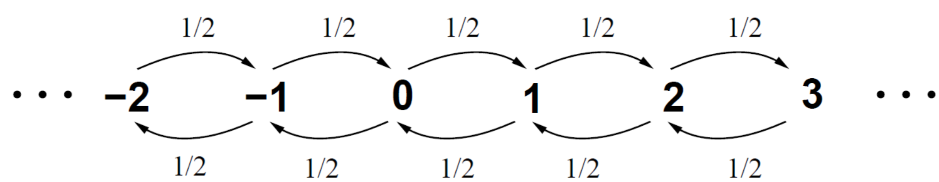
March 6, 2015

Introduction

A Markov chain is a mathematical model of a random variable that describes a sequence of events in which the probability of each events depends only on the state of the previous event. A Markov chain can be used to represent a stochastic process, which is a model representing how a distribution of a set of random variables changes over over time.

Drunkard's Walk

Consider a drunkard roaming the city trying to get home from the bar. Every time he takes a step, it is either towards his home with probability $\frac{1}{2}$ or away from his home with probability $\frac{1}{2}$. If the drunkard is n steps away from his home, what is the expected number of steps he is needs to take to get back home?



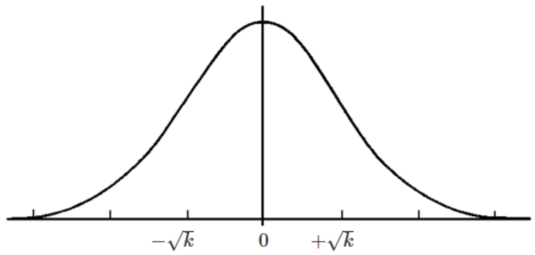
The distribution of the drunkard's location can be modeled using the following variables:

$$X = \sum_{i=1}^{n} X_i$$
$$X_i = \begin{cases} 1 & \text{with probability } \frac{1}{2} \\ -1 & \text{otherwise} \end{cases}$$

Assuming that k is the number of steps taken and k is even, the probability that the drunkard stays in the same location is:

$$\begin{aligned} P[X = 0] &= \binom{k}{k/2} 2^{-k} \\ &\approx \frac{1}{\sqrt{2\pi k}} \qquad \text{(Sterling's Approximation)} \end{aligned}$$

The distribution of X is the following:



The probability that X takes a value within one standard deviation in this case is:

$$P[-\sqrt{k} \leq X \leq \sqrt{k}] = \frac{2\sqrt{k}}{\sqrt{2\pi k}} = \sqrt{\frac{2}{\pi}}$$
$$P[X > \sqrt{k}] \geq C$$

Hence, in order to make sure that $P[X > n]$ with high probability, k needs to be at least n^2.

Stochastic Process

Markov chains are a stochastic process. A stochastic process is a sequence of random variables representing the change in the distribution of those variables over time. For example, given a sequence of random variables X_t with $t \in R^t$, the variables X_0, X_1, and X_2 represent sequence of random variables the distribution of which changes with respect to t.

Discrete Stochastic Process

The variables X_1, X_2, X_3, X_4, X_5, X_6, X_7 can also be viewed as state variables that can be discrete or continuous. In the example of the drunkard's walk problem, one view of the states can be that each variable represent either the state -1 or $+1$ with the state indicating the direction of the walk. In this case, for an independent distribution there are 2^n when considering the joint distribution of X. However, the distribution of X_{i+1} is completely determined if given X_i and therefore it is possible to describe the distribution by using fewer than 2_n numbers.

$$\{X_t\}_{t \in IN} \text{ is Markov}$$
$$\forall i \geq 1, \forall k$$
$$\text{if } P[X_i | x_0, ..., x_{i-1}] = P[x_i | x_{i-1}]$$

This is an important characteristic of a stochastic process since it is only important to consider the previous time step rather than the whole history.

Probability Distribution

Another important characteristic regarding the stochastic process is that there's a limited amount of independence between the states, which helps reduce the number of values used to represent the distribution. Consider a distribution of t random variables where each variable can represent any one of t states. Assuming nothing is known regarding independence, the number of parameters needed to represent this distribution is $d^t - 1$ and requires an exponential table to represent. On the other hand, if all the variables are independent, only $(d-1)t$ parameters are needed to specify the distribution.

A stochastic process is an intermediate representation between the two extremes (no independence and complete independence). Only the previous step is important:

$$\forall t \geq 2$$
$$P[X_t = V_t | X_{t-1} = V_{t-1}, ..., X_1 = V_1] = P[X_t = V_t | X_{t-1} = V_{t-1}]$$
$$P[X_t | X_{t-1}, ..., X_1] = P[X_t | X_{t-1}]$$

In the case of a stochastic process, $(d-1) + (t-1)d(d-1)$ parameters are needed to represent the distribution.

Stochastic Matrix

The stochastic matrix is a matrix that describes the transitions of a Markov chain. Running a Markov chain involves multiplying an initial distribution π by the stochastic matrix T any $i \geq 0$ number of times in order to get to get the distribution of states at step i.

Here's an example of a stochastic matrix where all the probabilities are the same regardless of t:

$$T^t(i,j) = P[X_t = j | X_{t-1} = i]$$
$$T(i,j) = P[X_t = j | X_{t-1} = i]$$

The matrix T can be used to run Markov chains on a initial distribution π using the following operation:

$$i \geq 0$$
$$\pi T^i = \text{ distribution of states at step } i$$

The distribution at step i is given by πT^i.

Satisfiability Assignment (2-SAT)

Markov chains can be used in an algorithm to determine the satisfiability of a set of clauses. Consider the formula $F = \bigwedge_m^{i=1} C_i$ that contains n variables $x_1, x_2, ..., x_n$. The problem asks to find an assignment of the variables that satisfy F.

Deterministic Check for Satisfiability

The solution for the 2-SAT problem can be found deterministically by decomposing the original formula F into F_1 and F_2:

$$F_1 = F_{-x_1} \bigwedge D^-$$
$$F_2 = F_{-x_1} \bigwedge D^+$$

where D^- is the set of clauses that contain $\overline{x_1}$ and D^+ is the set of clauses that contain x_1. F is satisfiable if $F_1 \bigwedge F_2$ is satisfiable, so the problem can be decomposed in a tree-like fashion. However, this approach takes 2^n time so it's exponential in terms of the number of variables.

Randomized Check for Satisfiability using Markov Chains

The randomized algorithm for solving the satisfiability problem is:

Start with a random assignment of variables
while There exists and unsatisfied clause **do**
  Pick one of the variables in the clause at random and flip its value
end while

The analysis of the algorithm is as follow:

- Fix the initial state A
- Fix a satisfying assignment S
- Find the probability of reaching state S in terms of k, where k is the number of steps
- Let $H(A, S)$ be the Hamming Distance between A and S, in other words the shortest distance between A and S
- Let X_t represent the distance between the state at time t and S
- The distribution of states can be described as follows:

$$0 < j < n$$
$$P[X_t = j + 1 | X_{t-1} = j] \leq \frac{1}{2}$$
$$P[X_t = j - 1 | X_{t-1} = j] \geq \frac{1}{2}$$
$$P[X_t = 0 | X_{t-1} = 0] = 1$$
$$P[X_t = n - 1 | X_{t-1} = n] = 1$$

- The range of state that X_t can take is $0, 1, ..., n$. The state n is an absorbing state.
- The recurrence relation can be written as follows:

$h_i =$ expected number of steps to reach given $H(A, S) = i$ where A is the current assignment
$$h_0 = 0$$
$$h_n = 1 + h_{n-1}$$
Rewrite: $h_n - h_{n-1} = 1$
$$h_i = 1 + \frac{1}{2} h_{i-1} + \frac{1}{2} h_{i+1}$$
Rewrite: $h_i - hi - 1 = 2 + h_{i+1} - h_i$
$$h_i - h_{i-1} = 2i + 1$$
$$\begin{aligned} h_n &= \sum_{i=0}^{n-1} h_i - h_{i-1} \\ &= \sum_{i=0}^{n-1} 2i + 1 \\ &= 1 + 3 + ... + (2n - 1) \\ &= n^2 \end{aligned}$$

Hence, the probability that state S won't be reached in n^2 is $\leq \frac{1}{2}$. At this point, k Bernoulli trials can be applied to reduce the probability that state S won't be reached in k trials to $\leq (\frac{1}{2})^k$.

Satisfiability Assignment (3-SAT)

Markov chains provide a polynomial solution to the 2-SAT problem as proven above. But what happens when the same algorithm is applied to the 3-SAT problem?

In the 3-SAT problem, the following probabilities change:

$$P[X_t = j+1 | X_{t-1} = j] \leq \frac{2}{3}$$
$$P[X_t = j-1 | X_{t-1} = j] \geq \frac{1}{3}$$

The issue with applying this algorithm to the 3-SAT problem is that the probability of getting closer to the solution in every time step has been reduced to $\geq \frac{1}{3}$. In the worst-case scenario, this means that the expected time to find a satisfying assignment in this case is 2^n which is just as bad as iterating through all the possible states in a deterministic manner. It's unclear how to design an algorithm that can perform in polynomial time in this scenario. However, there exists an algorithm that is able to find a satisfying assignment in $2^{0.5n}$ time steps with good probability.

The algorithm to do so is the following:

- Start with a random assignment of variables. There is a good chance that this assignment is $\frac{n}{2}$ steps away S
- Run the algorithm for $3n$ steps to avoid the strong pull away from S
- If S isn't found within $3n$ steps, reselect a random assignment of variables and try again

The above algorithm finds a satisfying assignment in $2^{0.5n}$ time with good probability. The way it does so is by selecting a random starting assignment so that there is a good chance that it is $\sim \frac{n}{2}$ away from S. Then it runs the algorithm for a short amount of time. If the answer is not found within this time, it is very likely that the starting assignment was strongly pulled away from S and there's a better chance of finding the solution by reselecting a starting assignment rather than continuing to run the algorithm.