Design and Analysis of Algorithms March 8, 2013
Massachusetts Institute of Technology 6.046/18.410
Profs. Constantinos Daskalakis and Erik Demaine Recitation 5

Random Walks and Satisfiability

1 Test Reminder

Quiz 1 is *next* Thursday, March 14.

Covers material though today.

Allowed 1 page crib sheet.

Blank questions will receive 30% credit.

2 Random Walks on a Line

A random walk on a line is a simple random walk on the one dimensional integer lattice. This means that one travels either right one step with probability p or left one step with probability $1-p$. Random walks can also be defined on the real number line, but being Computer Scientists, we will only consider the discrete problem. Further, we will just consider the symmetric case where the probability of transition in either direction is $\frac{1}{2}$. The mean location at any point in the random walk is 0. This can be seen by symmetry or linearity of expectation (the mean at each time-step is zero).

We may also ask what the expected distance from the origin is after n time steps. We can define the random variable Z_i which is ± 1 each with probability $\frac{1}{2}$. Now for intuition, we may ask about the variance of this distribution.

$$\mathbb{E}\left[\sum_{i=0}^{n} Z_i\right] = 0$$

$$Z_i^2 = \frac{1}{2}((-1)^2 + 1^2) = 1$$

$$|\mathbb{E}\left[\sum_{i=0}^{n} Z_i\right]^2 - \mathbb{E}\left[\sum_{i=0}^{n} Z_i^2\right]| = n$$

Thus, the variance is n, so intuition suggests $\sqrt{n}$ might be the rate of growth of the expected distance.

Now let's prove that relation. Without loss of generality, we'll look at the probability of taking k steps to the right out of N total steps. This is given by $\mathbb{P}(k, N) = \binom{N}{k}(\frac{1}{2})^N$. We want to know what the expected deviation of k from its average value is. This is known as the dispersion

$\mathbb{E}[\Delta k] = \sqrt{\mathbb{E}[k^2] - (\mathbb{E}[k])^2}$.

Now we may calculate:

$$\mathbb{E}[k] = \sum_{k=0}^{N} k\mathbb{P}(k,N) = \sum_{k=0}^{N} k\binom{N}{k}(\frac{1}{2})^N = (\frac{1}{2})^N \sum_{k=0}^{N} \frac{N!}{(k-1)!(N-k)!} = (\frac{1}{2})^N N 2^{N-1} = \frac{N}{2}$$

And next:

$$\mathbb{E}[k^2] = \sum_{k=0}^{N} k^2\mathbb{P}(k,N) = (\frac{1}{2})^N \sum_{k=1}^{N} kN\binom{N-1}{k-1}$$

$$= (\frac{1}{2})^N N \left[\sum_{k=2}^{N} k(N-1)\binom{N-2}{k-2} + \sum_{k=1}^{N} k\binom{N-1}{k-1}\right] = (\frac{1}{2})^N N\left[(N-1)2^{N-2} + 2^{N-1}\right] = \frac{N^2}{4} + \frac{N}{4}$$

Thus: $\mathbb{E}[\Delta k] = \sqrt{\frac{N^2}{4} + \frac{N}{4} - (\frac{N}{2})^2} = \frac{\sqrt{N}}{2}$

The random walk on a line can be seen as the Markov process:
$P_{i,i+1} = p = 1 - P_{i,i-1}$
Where P is the transition and p is the probability of transition.

2.1 Alternative Approach

Let Z_i be the random variable denoting the i^{th} step, which is ± 1 each with probability $\frac{1}{2}$, and let $Z_{(n)} = \sum_{i=1}^{n} Z_i$. We can immediately notice that:

$$\mathbb{E}[Z_{(n)}] = \mathbb{E}[\sum_{i=1}^{n} Z_i] = \sum_{i=1}^{n} \mathbb{E}[Z_i] = 0$$

This means that for all n, the expected position of the particle after n steps is the origin. Another important quantity is the *distance* of the particle from the origin after n steps, which is exactly $|Z_{(n)}|$. Computing $\mathbb{E}[|Z_{(n)}|]$ exactly is quite challenging, but let's build some intuition for the correct answer. As $|Z_{(n)}| = \sqrt{Z_{(n)}^2}$, we might expect $\mathbb{E}[|Z_{(n)}|]$ to behave like $\sqrt{\mathbb{E}[Z_{(n)}^2]}$, which is not so bad to compute:

$$\mathbb{E}[Z_{(n)}^2] = \mathbb{E}[\sum_{i=1}^{n}\sum_{j=1}^{n} Z_i Z_j] = \sum_{i=1}^{n} \mathbb{E}[Z_i^2] + \sum_{i=1}^{n}\sum_{j=i+1}^{n} 2\mathbb{E}[Z_i Z_j]$$

We now observe that $Z_i^2 = 1$ always, for all i, and that $Z_i Z_j$ is still ± 1 each with probability $\frac{1}{2}$. Therefore we may evaluate the sum above to get $\mathbb{E}[Z_{(n)}^2] = n$. By the reasoning in the previous paragraph, we might expect $\mathbb{E}[|Z_{(n)}|] = O(\sqrt{n})$. This is true, but will not be proved.

Lastly, we might want to discuss the time it takes in order to reach distance k from the original. Specifically, let N_k be the smallest n such that $|Z_{(n)}| = k$. We'll conclude by computing $\mathbb{E}[N_k]$: Define $f_k(i)$ to be the expected number of steps to reach distance k from the origin if the particle is currently distance i from the origin. When $0 < i < k$, the particle takes a step left or right with equal probability. When $i = 0$, the particle goes either to 1 or -1, both of which are distance 1 from the origin. When $i = k$, we are already distance k from the origin. Putting this together, we get the following recurrence:

$$f_k(i) = 1 + \frac{1}{2} f_k(i-1) + \frac{1}{2} f_k(i+1),\ 0 < i < k$$
$$f_k(k) = 0$$
$$f_k(0) = 1 + f_k(1)$$

We can rearrange the top equation (first multiply by 2, then subtract one term from each side) to yield:

$$f_k(i) - f_k(i+1) = 2 + f_k(i-1) - f_k(i),\ 0 < i < k$$
$$f_k(0) - f_k(1) = 1$$

It is easy to see that this recurrence solves to $f_k(i) - f_k(i+1) = 2i + 1$ (subsitution method works). Using this, we can now solve for $f_k(0)$:

$$f_k(0) = \sum_{i=0}^{k-1} f_k(i) - f_k(i+1) = \sum_{i=1}^{k-1} 2i + 1 = k^2$$

Because the random walk starts at the origin, and $f_k(0)$ is the expected number of steps to reach a distance k away from the origin, we have $\mathbb{E}[N_k] = f_k(0) = k^2$.

If you use this approach, you can replace $O(n^2)$ with n^2 and use Markov's inequality instead of Chebyshev's in the section below (and also remove all discussion about standard deviations).

3 2-SAT

The Satisfiability problem asks whether an expression of binary variables and operations AND and OR, has an assignment of values which causes the expression to be true. In many cases, these expressions are written in conjunctive normal form (CNF) in which we have clauses in which variables are OR'ed together. The clauses are then AND'ed together. The general k-SAT problem has k variables in every clause. For example, here is a satisfiable instance of a 3-SAT problem:
$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x_3} \vee x_4) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee x_3 \vee \bar{x_4})$
This is satisfied by: $x_2 = 1, x_3 = 0, x_4 = 0$, among other possible solutions.

It is well-known that 3-SAT is NP-Complete, but 2-SAT is in P. We will now investigate a Monte Carlo algorithm which solves 2-SAT problems in $O(n^3)$ time. This is not the fastest algorithm; however, it is an interesting one.

2-SAT-Random-Walk

```
1  Arbitrarily assign all variables
2  while the expression is not satisfied
3       do
4          Pick an arbitrary unsatisfied clause
5          Choose a literal in that clause at random and flip the value of the chosen literal
```

Now we would like to analyze how long this takes to terminate. We can imagine progress being measured by the number of variables that have the correct assignment. Now when we flip the value of a variable, we have either taken an incorrect assignment and made it correct, or we have taken a correct assignment and made it incorrect. Thus, the number of correct assignments always either increases or decreases by one during every iteration. Now we can see a parallel to a random walk on a line. Finally, since we always choose an unsatisfied clause, at least one of those two literals must be incorrect, thus we have at least a 50% chance of increasing the number of variables which are correct. Thus our analogy is complete and we can thus ask what the expected number of steps will be before a satisfying assignment is chosen by asking how long it takes this random walk to go a distance of n away from zero. From our earlier analysis, this will reach a correct assignment after an expected $O(n^2)$ iterations, if one exists. To be a correct Monte Carlo algorithm, we actually need the probability of outputting the correct answer to be greater than $2/3$. To resolve this, we can look back at the random walk analysis and ask how many steps would be needed to have a dispersion of N with probability $2/3$. For convenience, instead of integrating our distribution above, we will use Chebyshev's inequality. This states:

$$\mathbb{P}(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

for random variable X, with mean μ, standard deviation σ and constant k. Thus, if we want a probability of $2/3$ we simply run for an extra factor of α such that the new expected mean is more than $3/2$ standard deviations away from n. Since σ scales linearly with n we only need a constant factor increase in running time to acheive the desired probability.

To find the running time, we note that checking the clauses and updating the literal both potentially take $O(n)$ time, so we end up with an $O(n^3)$ time algorithm overall. There exist linear time algorithms for solving 2-SAT which relies on analysis of strongly connected components in representative graphs. See Aspvall, Plass, and Tarjan 1979.

4 Random Transposition Shuffle

In recitation, we will consider yet another shuffling algorithm. Take a deck of cards, choose two cards from the deck uniformly at random (note these may be the same card) and swap their position

in the deck. It is easy to see this can be modeled as a lazy, strongly-connected Markov chain with nodes representing each of the permutations of the deck. We will bound the mixing time of this algorithm by a card-marking argument. Call the cards drawn at time t, R_t and L_t. We will mark R_t iff:

1. R_t is unmarked and
2. Either L_t is marked or $L_t = R_t$

Once all cards are marked, the deck has reached a uniform distribution over the possible permutations. We prove this by induction on t. Let $V_t \subset [n]$ be the set of cards marked at or before time t and let $U_t \subset [n]$ be the positions those cards occupy. We let our inductive hypothesis be, given t, V_t, U_t all possible permutations of cards in V_t on positions U_t are equally likely. The base case, $t = 1$ is trivially true since there is only zero or one element. For time $t+1$ if we do not mark R_{t+1} then $V_{t+1} = V_t$ and we have three cases:

1. If L_{t+1} and R_{t+1} are unmarked, then V_{t+1} and U_{t+1} do not change.
2. If both L_{t+1} and R_{t+1} are marked, then $U_{t+1} = U_t$ and we applied a uniform random transposition to the cards in V_t keeping all of the permutations equiprobable.
3. If L_{t+1} is unmarked and R_{t+1} was marked previously, then we update U_t by deleting the position occupied by R_t and add the position occupied by L_t. Since we chose R_t uniformly at random from V_t all permutations of V_t on U_{t+1} are equally likely.

If R_{t+1} is marked, then we know that $L_{t+1} \in V_{t+1} = V_t \cup \{R_{t+1}\}$ chosen uniformly at random. U_t has the position occupied by R_{t+1} added to it. Since V_t was uniformly distributed over the positions U_t all we have to do to maintain that equiprobability for the new sets is ensure a uniformly random element from V_{t+1} is inserted into the new position in U_{t+1} and the new element in V_{t+1} has an equal chance of being inserted into any position in U_{t+1}. Notice this is exactly the condition we have, and thus the inductive invariant is maintained.

Now we can consider the expected time needed to mark all cards in the set. You may notice an analogy to the stamp collecting problem. We can sum up the expected time of marking each of the kth cards. The probability of marking the kth card is the probability of choosing a marked card and either an unmarked card or the same card out of the deck. This leads to the summation:

$$\mathbb{E}(\tau) = \sum_{k=0}^{n-1} \frac{n^2}{(k+1)(n-k)}$$

Using partial fraction decomposition, we get:

$$\mathbb{E}(\tau) = \frac{n^2}{n+1} \sum_{k=0}^{n-1} \left(\frac{1}{(k+1)} + \frac{1}{(n-k)} \right)$$

Simplifying and noting the harmonic series, we can conclude:

$$\mathbb{E}(\tau) = 2n(\log n + O(1))$$