# Main class 13: DTMC-based Algorithms, absorbing Markov chains.

Izzy Grosof

October 28, 2024

## 1 Takeaways and Summary

- There are important algorithms based on DTMCs, including the Markov Chain Monte Carlo (MCMC) algorithm, and the PageRank algorithm.
- MCMC is used to search over a search space that's too large to explore exhaustively, and find elements with a high score. It works by constructing a DTMC whose stationary distribution is proportional to the score.
- The PageRank algorithm assigns an importance score to each website, corresponding to the stationary probability of a DTMC called the "random surfer" model. A search engine can then rank the sites according to the score.
- An "absorbing" state of a Markov chain is a state where the only transition is to itself, with probability 1.
- If a Markov chain has at least one absorbing state, it is called an absorbing Markov chain.
- Absorbing Markov chains are reducible.
- We focus on the case where all non-absorbing states $i$ have an absorbing state $j$ which is accessible: $i \to j$. The non-absorbing states are all transient, with finite return probability.
- We can sort the absorbing states separately from the transient states, then write the transition matrix $P$ as a block matrix:

  $$P = \begin{pmatrix} I & 0 \\ R & Q \end{pmatrix}$$

  $Q$ covers transient-to-transient transitions, while $R$ covers transient-to-absorbing transitions.
- For transient states, the $n$-step transition probability just depends on the $Q$ matrix: $P^n_{ij} = Q^n_{ij}$.
- An important concept for absorbing Markov chains is the "fundamental matrix" $N_{ij}$, which counts the expected number of visits to a transient state $j$, given that $X_0 = i$. It can be calculated in two ways:

  $$N_{ij} = \sum_{k=0}^{\infty} Q^n_{ij}, \quad N = (I - Q)^{-1}.$$

  To clarify, in the formula $(I - Q)^{-1}$, the $I$ refers to the identity matrix with the same dimensions as the matrix $Q$, and the $-1$ refers to the matrix inverse operation.

  Note that this sum is finite, because the state $j$ is transient.

- Another important concept is $B_{i\ell}$, the probability of ending up in an absorbing state $\ell$, given that $X_0 = i$. It is defined as

$$B_{i\ell} = \lim_{n\to\infty} P^n_{i\ell}.$$

Note: This is similar to a limiting probability, but it's allowed to depend on the initial state.

- We can calculate $B_{i\ell}$ by conditioning on the last transient state $j$ visited before $\ell$:

$$B_{i\ell} = \sum_j N_{ij} R_{j\ell}, \quad B = NR.$$

# 2 Recap: Local balance

1. Have a Markov chain
2. Guess that local balance might be present.
3. Use some of the local balance equations to argue that if there is a statioanry distribution that satisfies the local balance equations, it must be a specific distribution.
4. Check whether the rest of the local balance equations also hold.
5. If so, great! You've found the stationary distribution! If not, no local balance, use stationary equations instead.

# 3 DTMC-based algorithms (10 minutes)

## 3.1 Markov Chain Monte Carlo

Topic of class project. See lab notes, will post full assignment today. Will be due 11/15.

This is an important algorithm using all the time in real systems.

We have a bunch of things. We have a score indicating how good they are. In project, our things are decryption functions, and our scores are how English-like the decryption is. We want to find things with high scores. Too many things to check directly. Use DTMCs.

From importance score to stationary distribution.

Given an importance score $s(j)$, desired stationary distribution is $\pi_j = \frac{s(j)}{\sum_{j'\in F} s(j')}$. This allows us to find things with high scores.

Given desired stationary distribution $\pi$, construct $P$ s.t. $\pi = \pi P$. Simulate $X$ according to $P$. Approximate $\pi_j$ as $\frac{1}{n}\sum_{k=1}^{n} \mathbb{1}\{X_n = j\}$.

Straightforward way to approximate stationary probability: Simulate $\{X_i, i = 1, 2, 3, \ldots n\}$, for large $n$.

$$\pi_j \approx \frac{1}{n}\sum_{k=1}^{n} \mathbb{1}\{X_n = j\}$$

Important: If state space is large or infinite, we usually don't need $n$ to be as big as the size of the state space to get a good approximation. In project, state space is about $10^{28}$ decryption functions, but $10^5$ iterations will be enough.

## 3.2 PageRank Algorithm

From stationary distribution to importance score.

Search engine: Suppose there's lots of websites, and some websites have links to other websites. You want to figure out which websites are "important", from just this link information. Can't just count links – links from important websites mean you're important too. On the other hand, linking to important websites doesn't mean you're important.

Idea behind PageRank: imagine there was a person surfing the web randomly, according the following update rule:

Each time step, with probability $p$, goes to a uniformly random page on the whole internet. Otherwise, clicks on a uniformly random link on this page.

This is a DTMC. Each page has a stationary probability $\pi_j$. That stationary probability is a good importance score for the importance of the website $j$, and it serves as a useful way to order search results – put websites with high scores at the top of the results. It satisfies our design goal: Being linked from high-importance sites increases your importance.

Note: Simulating this "random surfer" isn't a good way to calculate the importance score – too slow. Other fast algorithms.

This was developed by the founders of Google (Sergey Brin and Larry Page), they patented it, it was the beginning of the Google search algorithm. The patent expired in 2019, so feel free to use it.

# 4 Absorbing Markov Chains (15 minutes)

State $i$ is absorbing if $P_{ii} = 1$. Once the state reaches $i$, stops changing. Reducible (not irreducible).

Example: Condition of a car.

Sold $\leftarrow$ Working $\leftrightarrow$ Damaged $\rightarrow$ Totalled

If it's totalled, it's done forever. If it's sold, it's not our problem. Sold and totalled are absorbing states, working and damaged are not.

Note that in this situation, every non-absorbing state can reach an absorbing state. We focus on this case. The absorbing states are transient – return probability less than 1.

Split into absorbing states and transient states.

$P_{ij}$ can be split into four parts, of which 2 are interesting.

$$P = \begin{pmatrix} I & 0 \\ R & Q \end{pmatrix}$$

Transitions within transient states: $Q_{ij}$. Transitions from transient states to absorbing states: $R_{ij}$.

Chance of moving from one transient state to another in $n$ steps: Can ignore the rest of the matrix.

If $i, j$ are transient states,

$$P^n_{ij} = Q^n_{ij}$$

Fundamental matrix $N_{ij}$: Count expected number of visits.

$$N_{ij} = \mathbb{E}[\# \; visits \; to \; j \mid X_0 = i]$$

$$N_{ij} = \sum_{k=0}^{\infty} Q^k_{ij}$$

Note: This is finite.

Can also calculate using linear algebra:

$$N = (I - Q)^{-1}$$

$B_{i,\ell}$: Probability of ever ending up in a given absorbing state $\ell$, starting from $i$. Similar to a limiting probability, but it's allowed to depend on the initial state $i$.

$$B_{i\ell} := \lim_{n \to \infty} P^n_{i\ell}$$

To calculate, look at the expected number of visits to each transient state $j$, times the probability of moving from $j$ to $\ell$:

$$B_{i\ell} = \sum_j N_{ij} R_{j\ell}$$
$$B = NR$$