PRINCETON COS 521: ADVANCED ALGORITHM DESIGN

Random walks, Markov chains, and how to analyse them

Today we study random walks on graphs. When the graph is allowed to be directed and weighted, such a walk is also called a *Markov Chain*. These are ubiquitous in modeling many real-life settings.

Example 1 (Drunkard's walk)**.** *There is a sequence of $2n + 1$ pubs on a street. A drunkard starts at the middle house. At every time step, if he is at pub number i, then with probability $1/2$ he goes to pub number $i - 1$ and with probability $1/2$ to pub $i + 1$. How many time steps does it take him to reach either the first or the last pub?*

Thinking a bit, we quickly realize that the first m steps correspond to m coin tosses, and the distance from the starting point is simply the difference between the number of heads and the number of tails. We need this difference to be n. Recall that the number of heads is distributed like a normal distribution with mean $m/2$ and standard deviation $\sqrt{m}/2$. Thus m needs to be of the order of n^2 before there is a good chance of this random variable taking the value $m + n/2$.

Thus being drunk slows down the poor guy by a quadratic factor.

Example 2 (Exercise)**.** *Suppose the drunkard does his random walk in a city that's designed like a grid. At each step he goes North/South/East/West by one block with probability $1/4$. How many steps does it take him to get to his intended address, which is n blocks north and n blocks east away?*

Random walks in space are sometimes called *Brownian motion*, after botanist Robert Brown, who in 1826 peered at a drop of water using a microscope and observed tiny particles (such as pollen grains and other impurities) in it performing strange random-looking movements. He probably saw motion similar to the one in the figure. Explaining this movement was a big open problem. During his "miraculous year"of 1905 (when he solved 3 famous open problems in physics) Einstein explained Brownian motion as a random walk in space caused by the little momentum being imparted to the pollen in random directions by the (invisible) molecules of water. This theoretical prediction was soon experimentally confirmed and seen as a "proof"of the existence of molecules. Today random walks and brownian motion are used to model the movements of many systems, including stock prices.

Fun fact: 1D random walks return to the origin infinitely often. That is, no matter where we start, eventually we'll reach the origin (or, in fact, any specific point) as the number of steps approaches infinity. The same is true for 2D random walks. For 3D random walks, there's always a non-zero probability of returning to the origin (for instance, you could go immediately left then right), but there's also a non-zero probability that you'll wander around forever and

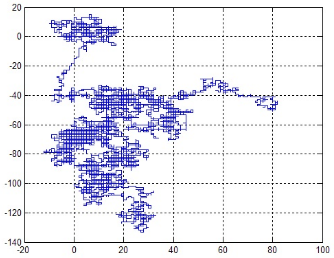

Figure 1: A 2D Random Walk

never return. The probability of returning approaches zero as the dimension grows.

Example 3 (Random walks on graph)**.** *We can consider a random walk on a d-regular graph $G = (V, E)$ instead of in physical space. The particle starts at some vertex v_0 and at each step, if it is at a vertex u, it picks a random edge of u with probability $1/d$ and then moves to the other vertex in that edge. There is also a* lazy *version of this walk where he stays at u with probability $1/2$ and moves to a random neighbor with probability $1/2d$.*

Thus the drunkard's walk can be viewed as a random walk on a line graph.

One can similarly consider random walks on directed graph (randomly pick an outgoing edge out of u to leave from) and walks on weighted graph (pick an edge with probability proportional to its weight). Walks on directed weighted graphs are called Markov Chains.

In a random walk, the next step does not depend upon the previous history of steps, only on the current position/state of the moving particle. In general, the term *markovian* refers to systems with a "memoryless"property. In an earlier lecture we encountered Markov Decision Processes, which also had this memoryless property.

Application: Bigram and trigram models

Markovian models are ubiquitous in applications. (Remember for example the Markov Decision Processes we encountered earlier.) Language recognition systems work by constantly predicting what's coming next. Having heard the first i words they try to generate a prediction of the $i + 1$th word[1]. This is a very complicated piece of software, but one underlying idea is to model language generation as a Markov chain. (This is not an exact model; language is known to not be Markovian, at least in the simple way described below.)

The simplest idea would be to model this as a markov chain on the words of a dictionary. Recall that everyday English has about $5,000$ words. A simple markovian model consists of thinking of a piece of

[1] You can see this while typing in the text box on smartphones, which always display their guesses of the next word you are going to type. This lets you save time by clicking the correct guess.

text as a random walk on a space with 5000 states (= words). A state corresponds to the last word that was just seen. For each word pair w_1, w_2 there is a probability p_{w_1,w_2} of going from w_1 to w_2. According to this Markovian model, the probability of generating a sentence with the words w_1, w_2, w_3, w_4 is $q_{w_1} p_{w_1 w_2} p_{w_2 w_3} p_{w_3 w_4}$ where q_{w_1} is the probability that the first word is w_1.

To actually fit such a model to real-life text data, we have to estimate $5,000$ probabilities q_{w_1} for all words and $(5,000)^2$ probabilities $p_{w_1 w_2}$ for all word pairs. Here

$$p_{w_1 w_2} = \Pr[w_2 | w_1] = \frac{\Pr[w_2 w_1]}{\Pr[w_1]},$$

namely, the probability that word w_2 is the next word given that the last word was w_1.

One can derive empirical values of these probabilities using a sufficiently large text corpus. (Realize that we have to estimate 25 million numbers, which requires either a very large text corpus or using some shortcuts.)

An even better model in practice is a *trigram model* which uses the previous two words to predict the next word. This involves a markov chain containing one state for every *pair* of words. Thus the model is specified by $(5,000)^3$ numbers of the form $\Pr[w_3 | w_2 w_1]$. Fitting such a model is beyond the reach of current computers but we won't discuss the shortcuts that need to be taken.

Example 4 (Checking the randomness of a person)**.** *Suppose you ask your friend to write down a sequence of 200 random bits. Then how "random"is their output? In other words, is their brain capable of actually generating 200 random bits? If yes, then after seeing the first* 199 *bits you should not have a better than* 50-50 *chance of predicting the* 200*th bit.*

In practice, people tend to skew their distributions because of a mistaken notion of what a random string should look like. For instance, suppose the last three bits were 0*. Then they might think that a fourth* 0 *would not look too random, so they may make that bit a* 1*. This is of course wrong since the next bit is then biased towards being* 1*. These sort of patterns will tend to make their output not random —meaning given the last n bits you may have a good chance of predicting the n+1'st.*

This naturally leads to a Markovian model to try to predict the next bit. Its state space could be, say, the set of all sequences of 3 *bits, and the current state denotes the last three bits output by your friend. Transitions are defined by the next bit they output. Thus if the current state is* 011 *and they output a* 0 *then the state becomes* 110*.*

This model has 8 *states, and* 16 *transitions, and given a long enough sequence from your friend you can estimate the transition probabilities, which gives you a way to learn his/her secret nonrandom patterns. Then you* *have a better than 50-50 chance of predicting the next bit.*

Recasting a random walk as linear algebra

A *Markov chain* is a discrete-time stochastic process on n states defined in terms of a transition probability matrix (M) with rows i and columns j.

$$M = \left(M_{ij}\right)$$

where M_{ij} corresponds to the probability that the state at time step $t + 1$ will be j, given that the state at time t is i. This process is *memoryless* in the sense that this transition probability does not depend upon the history of previous transitions.

Therefore, each row in the matrix M is a distribution, implying $M_{ij} \geq 0 \forall i, j \in S$ and $\sum_j M_{ij} = 1$. The bigram or trigram models are examples of Markov chains.

Using a slight twist in the viewpoint we can use linear algebra to analyse random walks. Instead of thinking of the drunkard as being at a specific point in the state space, we think of the vector that specifies his probability of being at point $i \in S$. Then the randomness goes away and this vector evolves according to deterministic rules. Let us understand this evolution.

Let the initial distribution be given by the row vector $x \in \Re^n$, $x_i \geq 0$ and $\sum_i x_i = 1$. After one step, the probability of being at space i is $\sum_j x_j M_{ji}$, which corresponds to a new distribution xM. It is easy to see that xM is again a distribution.

Sometimes it is useful to think of x as describing the amount of *probability fluid* sitting at each node, such that the sum of the amounts is 1. After one step, the fluid sitting at node i distributes to its neighbors, such that M_{ij} fraction goes to j.

Suppose we take two steps in this Markov chain. The memoryless property implies that the probability of going from i to j is $\sum_k M_{ik} M_{kj}$, which is just the (i, j)th entry of the matrix M^2. In general taking t steps in the Markov chain corresponds to the matrix M^t, and the state at the end is xM^t. Thus the

Definition 1. *A distribution π for the Markov chain M is a* stationary distribution *if $\pi M = \pi$.*

Example 5 (Drunkard's walk on n-cycle)**.** *Consider a Markov chain defined by the following random walk on the nodes of an n-cycle. At each step, stay at the same node with probability* $1/2$*. Go left with probability* $1/4$ *and right with probability* $1/4$*.*

The uniform distribution, which assigns probability $1/n$ *to each node, is a stationary distribution for this chain, since it is unchanged after applying one step of the chain.*

Definition 2. *A Markov chain M is* ergodic *if there exists a unique stationary distribution π and for every (initial) distribution x the limit* $\lim_{t\to\infty} xM^t = \pi$.

In other words, no matter what initial distribution you choose, if you let it evolve long enough the distribution converges to the stationary distribution. Some basic questions are when stationary distributions exist, whether or not they are unique, and how fast the Markov chain converges to the stationary distribution.

Does Definition 1 remind you of something? Almost all of you know about eigenvalues, and you can see that the definition requires π to be an eigenvector which has all nonnegative coordinates and whose corresponding eigenvalue is 1.

In today's lecture we will be interested in Markov chains corresponding to undirected d-regular graphs, where the math is easier because the underlying matrix is *symmetric*: $M_{ij} = M_{ji}$.

Eigenvalues. Recall that if $M \in \Re^{n\times n}$ is a square symmetric matrix of n rows and columns then an **eigenvalue** of M is a scalar $\lambda \in \Re$ such that exists a vector $x \in \Re^n$ for which $M \cdot x = \lambda \cdot x$. The vector x is called the **eigenvector** corresponding to the eigenvalue λ. M has n real eigenvalues denoted $\lambda_1 \le ... \le \lambda_n$.[2] (The multiset of eigenvalues is called the *spectrum*.) The eigenvectors associated with these eigenvalues form an orthogonal basis for the vector space $\Re^n$ (for any two such vectors the inner product is zero and all vectors are linear independent). The word *eigenvector* comes from German, and it means "one's own vector. "The eigenvectors are n prefered directions $u_1, u_2, \ldots, u_n$ for the matrix, such that applying the matrix on these directions amounts to simple scaling by the corresponding eigenvalue. Furthermore these eigenvectors span $\Re^n$ so every vector x can be written as a linear combination of these.

Example 6. *We show that every eigenvalue λ of M is at most* 1*. Suppose $\vec{e}$ is the corresponding eigenvector. Say the largest coordinate is i. Then* $\lambda e_i = \sum_{j:\{i,j\}\in E} \frac{1}{d} e_j$ *by definition. If $\lambda > 1$ then at least one of the neighbors must have $e_j > e_i$, which is a contradiction. By similar argument we conclude that every eigenvalue of M is at most -1 in absolute value.*

Mixing Time

Informally, the *mixing time* of a Markov chain is the time it takes to reach "nearly stationary" distribution from any arbitrary starting distribution.

Definition 3. *The mixing time of an ergodic Markov chain M is t if for every starting distribution x, the distribution xM^t satisfies* $\|xM^t - \pi\|_1 \le$

[2] Not every matrix has n real eigenvalues, but M does because it's doubly stochastic - we won't prove this.

$1/4$. *(Here $\|\cdot\|_1$ denotes the ℓ_1 norm and the constant "$1/4$" is arbitrary.)*

Example 7 (Mixing time of drunkard's walk on a cycle)**.** *Let us consider the mixing time of the walk in Example 5. Suppose the initial distribution concentrates all probability at state* 0*. Then* $2t$ *steps correspond to about* t *random steps (= coin tosses) since with probability* $1/2$ *the drunk does not move. Thus the location of the drunk is*

$$\text{(\#(Heads) − \#(Tails))} \pmod{n}.$$

As argued earlier, it takes $\Omega(n^2)$ *steps for the walk to reach the other half of the circle with any reasonable probability, which implies that the mixing time is* $\Omega(n^2)$*. We will soon see that this lowerbound is fairly tight; the walk takes about* $O(n^2 \log n)$ *steps to mix well.*

Upper bounding the mixing time (undirected d-regular graphs)

For simplicity we restrict attention to random walks on regular graphs. Let M be a Markov chain on a d-regular undirected graph with an adjacency matrix A. Then, clearly $M = \frac{1}{d}A$.

Clearly, $\frac{1}{n}\vec{1}$ is a stationary distribution, which means it is an eigenvector of M. What is the mixing time? In other words if we start in the initial distribution x then how fast does xM^t converge to $\vec{1}$?

First, let's identify two hurdles that would prevent such convergence, and in fact prevent the graph from having a unique stationary distribution. (a) *Being disconnected:* if the walk starts in a vertex in one connected component, it never visits another component, and vice versa. So two walks starting in the two components cannot converge to the same distribution, no longer how long we run them. (b) *Being bipartite:* This means the graph consists of two sets A, B such that there are no edges within A and within B; all edges go between A and B. Then the walk starting in A will bounce back and forth between A and B and thus not converge.

The following shows how to capture these two conditions via simple properties of the eigenvalues.

Lemma 1. *Let* G *be an undirected* d*-regular graph with adjacency matrix* A*. Let* $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n$ *be the eigenvalues of* A*. Then,*

1. $\lambda_1 = d$ *and the corresponding eigenvector is* $\mathbf{1}$*.*

2. $\lambda_2 < d$ *if and only if the graph is connected.*

3. $\lambda_n > -d$ *if and only if the graph is not bipartite.*

Proof. Let v be an eigenvector of A with eigenvalue λ. Then, $Av = \lambda v$. Let i be the coordinate of v with the largest possible magnitude.

Then, notice that $\lambda v_i = (Av)_i = \sum_j A_{i,j} v_j = \sum_{j \sim i} v_j \leq \sum_{j \sim i} |v_j| \leq d \max_i |v_i|$. Here, $j \sim i$ denotes that j is a neighbor of i in G. So $|\lambda| \leq d$. Thus, all eigenvalues of A are in $[-d, d]$.

Further, notice that since $(A\mathbf{1})_i = \sum_{j \sim i} 1 = d$ so $\mathbf{1}$ is an eigenvector with eigenvalue d. This completes the proof of the first bullet point.

Further, suppose G is connected and v is any eigenvector of A with eigenvalue d. We will show that v is parallel to $\mathbf{1}$. This shows that a connected G can only have one eigenvalue equal to d. To see why, first normalize v so that it's maximum magnitude of any coordinate is 1. WLOG assume that this is the first coordinate. Now, by multiplying v by -1 if needed, we can assume that $v_1 = 1$ and $|v_i| \leq 1$ for all i.

$dv_1 = (Av)_1 = \sum_{j \sim 1} v_j$. Since every coordinate of v is at most 1 in magnitude and 1 has exactly d neighbors, the equality here can hold only if $v_i = 1$ for every neighbor j of 1. We have thus proven that if a vertex i is such that $v_i = 1$ then for all its neighbors j, $v_j = 1$. We can now "propagate" this argument starting from vertex 1, we infer about its neighbors, their neighbors and so on. Since the graph is connected, we must end up setting $v_u = 1$ for all u.

Next, suppose that the graph is disconnected with U, U' be two connected components. Let $\mathbf{1}_U$ and $\mathbf{1}_{U'}$ be vectors with entry 1 at $u \in U$ ($u \in U'$ respectively) and 0 otherwise. Then, notice that since $(Av)_i = \sum_{j \sim i} v_i$ for any v and all neighbors of $i \in U$ are in U itself, we must have that $A\mathbf{1}_U = d\mathbf{1}_U$ and similarly $A\mathbf{1}_{U'} = d\mathbf{1}_{U'}$. Thus, $\mathbf{1}_U$ and $\mathbf{1}_{U'}$ are both eigenvectors of A with eigenvalue d.

We have now completed the proof of bullet point 2.

Finally, let's consider the last bullet point – the proof of which is analogous to the above. Suppose G is bipartite with bipartition U, V. Then, by direct computation, we can check that the vector $\mathbf{1}_U - \mathbf{1}_V$ is an eigenvector of A with eigenvalue $-d$.

On the other hand, by running a propagation argument as above, and assuming that vertex 1 is in U, we learn that 1-step neighbors of 1 must have value -1, 2 step neighbors $+1$, 3 step neighbors -1 and so on. This scheme is consistent if and only if no vertex can be reached by both an odd and an even length path from 1, which is true if and only if G does not have an odd length cycle. You may recall from your undergraduate algorithms class that a graph is bipartite if and only if it has no odd length cycle. So, our scheme succeeds if and only if the graph is bipartite.

□

Note that if x is a distribution, x can be written as

$$x = \vec{1}\frac{1}{n} + \sum_{i=2}^{n} \alpha_i e_i$$

where e_i are the eigenvectors of M which form an orthogonal basis and 1 is the first eigenvector with eigenvalue 1. (Clearly, x can be written as a combination of the eigenvectors; the observation here is that the coefficient in front of the first eigenvector $\vec{1}$ is $\vec{1} \cdot x / \|\vec{1}\|_2^2$ which is $\frac{1}{n}\sum_i x_i = \frac{1}{n}$.)

$$\begin{aligned}
M^t x &= M^{t-1}(Mx) \\
&= M^{t-1}(\frac{1}{n}\vec{1} + \sum_{i=2}^{n} \alpha_i \lambda_i e_i) \\
&= M^{t-2}(M(\frac{1}{n}\vec{1} + \sum_{i=2}^{n} \alpha_i \lambda_i e_i)) \\
&\dots \\
&= \frac{1}{n}\vec{1} + \sum_{i=2}^{n} \alpha_i \lambda_i^t e_i
\end{aligned}$$

Also

$$\| \sum_{i=2}^{n} \alpha_i \lambda_i^t e_i \|_2 \le \lambda_{max}^t$$

where λ_{max} is the second largest eigenvalue of M in absolute value. This calculation uses the fact that $\sum_i \alpha_i e_i$ is a unit vector and hence has ℓ_2 norm 1. We are also using the fact that the total ℓ_2 norm of any distribution is $\sum_i x_i^2 \le \sum_i x_i = 1$.

Thus we have proved $\|M^t x - \frac{1}{n}1\|_2 \le \lambda_{max}^t$. Mixing times were defined using ℓ_1 distance, but Cauchy Schwartz inequality relates the ℓ_2 and ℓ_1 distances: $\|p\|_1 \le \sqrt{n}\|p\|_2$. So if $\lambda_{max}^t < 1/n^2$ say, then the walk will have mixed. So we have shown:

Theorem 1. *The mixing time is at most $O(\frac{\log n}{1-|\lambda_{\max}}|)$.*

Note also that if we let the Markov chain run for $O(k \log n / \lambda_{\max})$ steps then the distance to uniform distribution drops to $\exp(-k)$. This is why we were not very fussy about the constant $1/4$ in the definition of the mixing time earlier.

Remark: What if $\lambda_{\max}$ is 1 (i.e., -1 is an eigenvalue)? This breaks the proof and in fact the walk may not be ergodic. However, we can get around this problem by modifying the random walk to be *lazy*, by adding a self-loop at each node that ensures that the walk stays at a node with probability $1/2$. Then the matrix describing the new walk is $\frac{1}{2}(I + M)$, and its eigenvalues are $\frac{1}{2}(1 + \lambda_i)$. Now all eigenvalues are less than 1 in absolute value. This is a general technique for making walks *ergodic*.

Example 8 (Exercise)**.** *Compute the eigenvalues of the drunkard's walk on the n-cycle and show that its mixing time is $O(n^2 \log n)$.*

Application 2: Metropolis Algorithm and Multivariate Statistics

Multivariate statistics —useful in a variety of areas including statistical physics and machine learning—involves computing properties of a distribution for which only the *density* function is known. Say the distribution is defined on $\{0,1\}^n$ and we have a function $\mu(x)$ that is nonnegative and computable in polynomial time given $x \in \{0,1\}^n$. (For instance, statistical physicists are interested in *Ising models*, where $\mu(x)$ has the form $\exp(\sum_{ij} a_{ij} x_i x_j)$.) Then we wish to sample from the distribution where probability of getting x is *proportional* to $\mu(x)$. Since probabilities must sum to 1, we conclude that this probability is $\mu(x)/N$ where $N = \sum_{x \in \{0,1\}^n} f(x)$ is the so-called *partition function*.

The main problem problem here is that partition function is in general NP-hard to compute, meaning ability to compute it even approximately allows us to solve SAT. Let's see this. Suppose the formula has n variables and m clauses. For any assignment x define $\mu(x) = 2^{2nf_x}$ where f_x = number of clauses satisfied by x. Clearly, $\mu(x)$ is computable in polynomial time given x. If the formula has a satisfiable assignment, then $N > 2^{2nm}$ whereas if the formula is unsatisfiable then $N < 2^n \times 2^{2n(m-1)} < 2^{2nm}$. In particular, the mass $\mu(x)$ of a satisfying assignment exceeds the mass of all unsatisfying assignments. So the ability to compute a partition function —or even sample from the distribution—would yield a satisfying assignment with high probability.

The Metropolis-Hastings algorithm (named after its inventors) is a heuristic for solving the sampling problem. It works in many real-life settings even though the general problem is NP-hard. Define the following random walk on $\{0,1\}^n$. *At every step the walk is at some $x \in \{0,1\}^n$. (At the beginning use an arbitrary x.) At every step, toss a coin. If it comes up heads, stay at x. (In other words, there is a self-loop of probability at least $1/2$.) If the coin came up tails, then randomly pick a neighbor x' of x. Move to x' with probability $\min\{1, \frac{\mu(x')}{\mu(x)}\}$. (In other words, if $\mu(x') \geq \mu(x)$, definitely move. Otherwise move with probability given by their ratio.)*

CLAIM: *If all $\mu(x) > 0$ then the stationary distribution of this Markov chain is exactly $\mu(x)/N$, the desired distribution*

Proof. The markov chain defined by this random walk is ergodic since $\mu(x) > 0$ implies it is connected, and the self-loops imply it mixes. Thus it suffices to show that the (unique) stationary distribution has the form $\mu(x)/K$ for some scale factor K, and then it follows that K is the partition function. To do so it suffices to verify that such a distribution is stationary, i.e., in one step the probability flowing out of a vertex equals its inflow. For any x, lets L be the neighbors with a *lower* μ value and H be the neighbors with value at least as high. Then the outflow of probability per step is

$$\frac{\mu(x)}{2K}(\sum_{x'\in L}\frac{\mu(x')}{\mu(x)} + \sum_{x'\in H} 1),$$

whereas the inflow is

$$\frac{1}{2}(\sum_{x'\in L}\frac{\mu(x')}{K}\cdot 1 + \sum_{x'\in H}\frac{\mu(x')}{K}\frac{\mu(x)}{\mu(x')}),$$

and the two are the same. □

Analysis of Mixing Time for General Markov Chains

We did not do this in class; this is extra reading for those who are interested.

In the class we only analysed random walks on d-regular graphs and showed that they converge exponentially fast with rate given by the second largest eigenvalue of the transition matrix. Here, we prove the same fact for general ergodic Markov chains.

Theorem 2. *The following are necessary and sufficient conditions for ergodicity:*

1. *connectivity:* $\forall i,j : M^t(i,j) > 0$ *for some* t.
2. *aperiodicity:* $\forall i : gcd\{t : M^t(i,j) > 0\} = 1$.

Remark 1. *Clearly, these conditions are necessary. If the Markov chain is disconnected it cannot have a unique stationary distribution —there is a different stationary distribution for each connected component. Similarly, a bipartite graph does not have a unique distribution: if the initial distribution places all probability on one side of the bipartite graph, then the distribution at time t oscillates between the two sides depending on whether t is odd or even. Note that in a bipartite graph* $gcd\{t : M^t(i,j) > 0\} \geq 2$. *The sufficiency of these conditions is proved using eigenvalue techniques (for inspiration see the analysis of mixing time later on).*

Both conditions are easily satisfied in practice. In particular, any Markov chain can be made aperiodic by adding self-loops assigned probability $1/2$.

Definition 4. *An ergodic Markov chain is* reversible *if the stationary distribution* π *satisfies for all i, j,* $\pi_i P_{ij} = \pi_j P_{ji}$.

We need a lemma first.

Lemma 2. *Let M be the transition matrix of an ergodic Markov chain with stationary distribution* π *and eigenvalues* $\lambda_1(=1) \geq \lambda_2 \geq \ldots \geq \lambda_n$, *corresponding to eigenvectors* $v_1(=\pi), v_2, \ldots v_n$. *Then for any* $k \geq 2$,

$$v_k\vec{1} = 0.$$

Proof. We have $v_k M = \lambda_k v_k$. Mulitplying by $\vec{1}$ and noting that $M\vec{1} = \vec{1}$, we get

$$v_k \vec{1} = \lambda_k v_k \vec{1}.$$

Since the Markov chain is ergodic, $\lambda_k \neq 1$, so $v_k \vec{1} = 0$ as required. □

We are now ready to prove the main result concerning the exponentially fast convergence of a general ergodic Markov chain:

Theorem 3. *In the setup of the lemma above, let $\lambda = max\{|\lambda_2|, |\lambda_n|\}$. Then for any initial distribution x, we have*

$$||xM^t - \pi||_2 \leq \lambda^t ||x||_2.$$

Proof. Write x in terms of $v_1, v_2, \ldots, v_n$ as

$$x = \alpha_1 \pi + \sum_{i=2}^{n} \alpha_i v_i.$$

Multiplying the above equation by $\vec{1}$, we get $\alpha_1 = 1$ (since $x\vec{1} = \pi\vec{1} = 1$). Therefore $xM^t = \pi + \sum_{i=2}^{n} \alpha_i \lambda_i^t v_i$, and hence

$$\begin{aligned} ||xM^t - \pi||_2 &\leq || \sum_{i=2}^{n} \alpha_i \lambda_i^t v_i ||_2 && (1) \\ &\leq \lambda^t \sqrt{\alpha_2^2 + \cdots + \alpha_n^2} && (2) \\ &\leq \lambda^t ||x||_2, && (3) \end{aligned}$$

as needed. □