Spectral Graph Theory Lecture 10

Random Walks on Graphs

Daniel A. Spielman October 1, 2018

10.1 Overview

We will examine how the eigenvalues of a graph govern the convergence of a random walk on the graph.

10.2 Random Walks

In this lecture, we will consider random walks on undirected graphs. Let's begin with the definitions. Let $G = (V, E, w)$ be a weighted undirected graph. A random walk on a graph is a process that begins at some vertex, and at each time step moves to another vertex. When the graph is unweighted, the vertex the walk moves to is chosen uniformly at random among the neighbors of the present vertex. When the graph is weighted, it moves to a neighbor with probability proportional to the weight of the corresponding edge. While the transcript (the list of vertices in the order they are visited) of a particular random walk is sometimes of interest, it is often more productive to reason about the expected behavior of a random walk. To this end, we will investigate the probability distribution over vertices after a certain number of steps.

We will let the vector $\boldsymbol{p}_t \in \mathbb{R}^V$ denote the probability distribution at time t. I will write $\boldsymbol{p}_t(a)$ to indicate the value of $\boldsymbol{p}_t$ at a vertex a–that is the probability of being at vertex a at time t. A probability vector $\boldsymbol{p}$ is a vector such that $\boldsymbol{p}(a) \geq 0$, for all $a \in V$, and

$$\sum_u \boldsymbol{p}(a) = 1.$$

Our initial probability distribution, $\boldsymbol{p}_0$, will typically be concentrated one vertex. That is, there will be some vertex a for which $\boldsymbol{p}_0(a) = 1$. In this case, we say that the walk starts at a.

To derive a $\boldsymbol{p}_{t+1}$ from $\boldsymbol{p}_t$, note that the probability of being at a vertex a at time $t+1$ is the sum over the neighbors b of a of the probability that the walk was at b at time t, times the probability it moved from b to a in time $t+1$. We can state this algebraically as

$$\boldsymbol{p}_{t+1}(a) = \sum_{b:(a,b)\in E} \frac{w(a,b)}{\boldsymbol{d}(b)} \boldsymbol{p}_t(b), \tag{10.1}$$

where $\boldsymbol{d}(b) = \sum_a w(a,b)$ is the weighted degree of vertex b.

We will often consider lazy random walks, which are the variant of random walks that stay put with probability 1/2 at each time step, and walk to a random neighbor the other half of the time. These evolve according to the equation

$$\boldsymbol{p}_{t+1}(a) = (1/2)\boldsymbol{p}_t(a) + (1/2)\sum_{b:(a,b)\in E} \frac{w(a,b)}{\boldsymbol{d}(b)}\boldsymbol{p}_t(a). \tag{10.2}$$

10.3 Diffusion

There are a few types of diffusion that people study in a graph, but the most common is closely related to random walks. In a diffusion process, we imagine that we have some substance that can occupy the vertices, such as a gas or fluid. At each time step, some of the substance diffuses out of each vertex. If we say that half the substance stays at a vertex at each time step, and the other half is distributed among its neighboring vertices, then the distribution of the substance will evolve according to equation (10.2). That is, probability mass obeys this diffusion equation.

I remark that often people consider finer time steps in which smaller fractions of the mass leave the vertices. In the limit, this results in continuous random walks that are modeled by the matrix exponential. These are in many ways more natural than discrete time random walks. But, I do not think we will discuss them in this course.

10.4 Matrix form

The right way to understand the behavior of random walks is through linear algebra.

Equation (10.2) is equivalent to:

$$\boldsymbol{p}_{t+1} = (1/2)\left(I + \boldsymbol{M}\boldsymbol{D}^{-1}\right)\boldsymbol{p}_t. \tag{10.3}$$

You can verify this by checking that it is correct for any entry $\boldsymbol{p}_{t+1}(u)$, and you should do this yourself. It will prevent much confusion later.

For the rest of the course, I will let $\boldsymbol{W}_G$ denote the *lazy walk matrix* of the graph G, where

$$\boldsymbol{W}_G \stackrel{\text{def}}{=} (1/2)\left(\boldsymbol{I} + \boldsymbol{M}_G\boldsymbol{D}_G^{-1}\right). \tag{10.4}$$

This is the one asymmetric matrix that we will deal with in this course. It is related to the *normalized Laplacian*, which is symmetric and which is defined by

$$\boldsymbol{N} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{M}\boldsymbol{D}^{-1/2}.$$

Thus, the normalized Laplacian is positive semidefinite, and has the same rank as the ordinary (sometimes called "combinatorial") Laplacian. There are many advantages of working with the normalized Laplacian: the mean of its eigenvalues is 1, so they are always on a degree-independent

scale. One can prove that $\nu_n \leq 2$, with equality if and only if the graph is bipartite. I recommend proving $\nu_n \leq 2$ by showing that

$$\boldsymbol{L} \succcurlyeq -\boldsymbol{M},$$

which follows from consideration of the quadratic form $\sum_{(a,b)\in E} w_{a,b}(\boldsymbol{x}(a) - \boldsymbol{x}(b))^2$.

The walk matrix is similar to a symmetric matrix that is related to the normalized Laplacian:

$$\begin{aligned} W &= \boldsymbol{I} - \frac{1}{2}\left(\boldsymbol{I} - \boldsymbol{M}\boldsymbol{D}^{-1}\right) \\ &= \boldsymbol{I} - \frac{1}{2}\boldsymbol{D}^{1/2}\left(\boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{M}\boldsymbol{D}^{-1/2}\right)\boldsymbol{D}^{-1/2} \\ &= \boldsymbol{I} - \frac{1}{2}\boldsymbol{D}^{1/2}\boldsymbol{N}\boldsymbol{D}^{-1/2}, \end{aligned}$$

So, we know that $\boldsymbol{W}$ is diagonalizable, and that for every eigenvector $\boldsymbol{\psi}_i$ of $\boldsymbol{N}$ with eigenvalue ν_i, the vector $\boldsymbol{D}^{1/2}\boldsymbol{\psi}_i$ is a right-eigenvector of $\boldsymbol{W}$ of eigenvalue $1 - \nu_i/2$:

$$\begin{aligned} \boldsymbol{W}\left(\boldsymbol{D}^{1/2}\boldsymbol{\psi}_i\right) &= \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{D}^{1/2}\boldsymbol{N}\boldsymbol{D}^{-1/2}\right)\boldsymbol{D}^{1/2}\boldsymbol{\psi}_i \\ &= \boldsymbol{D}^{1/2}\boldsymbol{\psi}_i - \frac{1}{2}\boldsymbol{D}^{1/2}\boldsymbol{N}\boldsymbol{\psi}_i \\ &= \boldsymbol{D}^{1/2}\boldsymbol{\psi}_i - \frac{\nu_i}{2}\boldsymbol{D}^{1/2}\boldsymbol{\psi}_i \\ &= (1 - \nu_i/2)\boldsymbol{D}^{1/2}\boldsymbol{\psi}_i. \end{aligned}$$

The key thing to remember in the asymmetric case is that the eigenvectors of $\boldsymbol{W}$ are not necessarily orthogonal.

You may be wonder why I have decided to consider only lazy walks, rather than the more natural walk given by $\boldsymbol{M}\boldsymbol{D}^{-1}$. There are two equivalent reasons. The first is that all the eigenvalues of $\boldsymbol{W}$ are between 1 and 0. The second reason is explained in the next section.

For the rest of the semester, we will let the eigenvalues of $\boldsymbol{W}$ be:

$$1 = \omega_1 \geq \omega_2 \geq \cdots \geq \omega_n \geq 0, \quad \text{where} \quad \omega_i = (1 - \nu_i/2).$$

Yes, I know that ω is not a greek equivalent of w, but it sure looks like it.

10.5 The stable distribution

Regardless of the starting distribution, the lazy random walk on a connected graph always converges to one distribution: the *stable distribution*. This is the other reason that we forced our random walk to be lazy. Without laziness[1], there can be graphs on which the random walks never converge. For example, consider a non-lazy random walk on a bipartite graph. Every-other step will bring it to

[1]Strictly speaking, any nonzero probability of staying put at any vertex in a connected graph will guarantee convergence. We don't really need a half probability at every vertex.

the other side of the graph. So, if the walk starts on one side of the graph, its limiting distribution at time t will depend upon the parity of t.

In the stable distribution, every vertex is visited with probability proportional to its weighted degree. We denote the vector encoding this distribution by $\boldsymbol{\pi}$, where

$$\boldsymbol{\pi} = \boldsymbol{d}/(\mathbf{1}^T \boldsymbol{d})$$

and we recall that $\boldsymbol{d}$ is the vector of degrees. We can see that $\boldsymbol{\pi}$ is a right-eigenvector of $\boldsymbol{W}$ of eigenvalue 1:

$$\boldsymbol{M}\boldsymbol{D}^{-1}\boldsymbol{\pi} = \boldsymbol{M}\boldsymbol{D}^{-1}\ \boldsymbol{d}/(\mathbf{1}^T\boldsymbol{d}) = \boldsymbol{M}\mathbf{1}/(\mathbf{1}^T\boldsymbol{d}) = \boldsymbol{d}/(\mathbf{1}^T\boldsymbol{d}) = \boldsymbol{\pi},$$

so

$$\boldsymbol{W}\boldsymbol{\pi} = (1/2)\boldsymbol{I}\boldsymbol{\pi} + (1/2)\boldsymbol{M}\boldsymbol{D}^{-1}\boldsymbol{\pi} = (1/2)\boldsymbol{\pi} + (1/2)\boldsymbol{\pi} = \boldsymbol{\pi}.$$

This agrees with the translation we have established between eigenvectors of $\boldsymbol{W}$ and eigenvectors of $\boldsymbol{N}$. For a connected graph the nullspace of $\boldsymbol{N}$ is spanned by $\boldsymbol{d}^{1/2}$, and $\boldsymbol{\pi}$ is a multiple of $\boldsymbol{D}^{1/2}\boldsymbol{d}^{1/2}$.

To see that the walk converges to $\boldsymbol{\pi}$, we expand $\boldsymbol{D}^{-1/2}$ times the initial distribution in the eigenvectors $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n$ of $\boldsymbol{N}$. Let

$$\boldsymbol{D}^{-1/2}\boldsymbol{p}_0 = \sum_i c_i \boldsymbol{\psi}_i.$$

Note that

$$c_1 = \boldsymbol{\psi}_1^T(\boldsymbol{D}^{-1/2}\boldsymbol{p}_0) = \frac{(\boldsymbol{d}^{1/2})^T}{\|\boldsymbol{d}^{1/2}\|}(\boldsymbol{D}^{-1/2}\boldsymbol{p}_0) = \frac{\mathbf{1}^T\boldsymbol{p}_0}{\|\boldsymbol{d}^{1/2}\|} = \frac{1}{\|\boldsymbol{d}^{1/2}\|},$$

as $\boldsymbol{p}_0$ is a probability vector. We have

$$\begin{aligned}
\boldsymbol{p}_t &= W^t\boldsymbol{p}_0 \\
&= (\boldsymbol{D}^{1/2}(\boldsymbol{I} - \boldsymbol{N}/2)\boldsymbol{D}^{-1/2})^t\boldsymbol{p}_0 \\
&= (\boldsymbol{D}^{1/2}(\boldsymbol{I} - \boldsymbol{N}/2)^t\boldsymbol{D}^{-1/2})\boldsymbol{p}_0 \\
&= \boldsymbol{D}^{1/2}(\boldsymbol{I} - \boldsymbol{N}/2)^t \sum_i c_i\boldsymbol{\psi}_i \\
&= \boldsymbol{D}^{1/2}\sum_i (1 - \nu_i/2)^t c_i\boldsymbol{\psi}_i \\
&= \boldsymbol{D}^{1/2}c_1\boldsymbol{\psi}_1 + \boldsymbol{D}^{1/2}\sum_{i\geq 2}(1 - \nu_i/2)^t c_i\boldsymbol{\psi}_i.
\end{aligned}$$

As $0 < \nu_i \leq 2$ for $i \geq 2$, the right-hand term must go to zero. On the other hand, $\boldsymbol{\psi}_1 = \boldsymbol{d}^{1/2}/\|\boldsymbol{d}^{1/2}\|$, so

$$\boldsymbol{D}^{1/2}c_1\boldsymbol{\psi}_1 = \boldsymbol{D}^{1/2}\left(\frac{1}{\|\boldsymbol{d}^{1/2}\|}\right)\frac{\boldsymbol{d}^{1/2}}{\|\boldsymbol{d}^{1/2}\|} = \frac{\boldsymbol{d}}{\|\boldsymbol{d}^{1/2}\|^2} = \frac{\boldsymbol{d}}{\sum_j \boldsymbol{d}(j)} = \boldsymbol{\pi}.$$

This is a perfect example of one of the main uses of spectral theory: to understand what happens when we repeatedly apply an operator.

10.6 The Rate of Convergence

The rate of convergence of a lazy random walk to the stable distribution is dictated by ω_2. There are many ways of saying this. We will do so point-wise. Assume that the random walk starts at some vertex $a \in V$. For every vertex b, we will bound how far $\boldsymbol{p}_t(b)$ can be from $\boldsymbol{\pi}(b)$.

Theorem 10.6.1. *For all a, b and t, if $\boldsymbol{p}_0 = \boldsymbol{\delta}_a$, then*

$$|\boldsymbol{p}_t(b) - \boldsymbol{\pi}(b)| \leq \sqrt{\frac{\boldsymbol{d}(b)}{\boldsymbol{d}(a)}}\omega_2^t.$$

Proof. Observe that

$$\boldsymbol{p}_t(b) = \boldsymbol{\delta}_b^T \boldsymbol{p}_t.$$

From the analysis in the previous section, we know

$$\boldsymbol{p}_t(b) = \boldsymbol{\delta}_b^T \boldsymbol{p} = \boldsymbol{\pi}(b) + \boldsymbol{\delta}_b^T \boldsymbol{D}^{1/2} \sum_{i \geq 2} \omega_i^t c_i \boldsymbol{\psi}_i.$$

We need merely prove an upper bound on the magnitude of the right-hand term. To this end, recall that

$$c_i = \boldsymbol{\psi}_i^T \boldsymbol{D}^{-1/2} \boldsymbol{\delta}_a.$$

So,

$$\boldsymbol{\delta}_b^T \boldsymbol{D}^{1/2} \sum_{i \geq 2} \omega_i^t c_i \boldsymbol{\psi}_i = \sqrt{\frac{\boldsymbol{d}(b)}{\boldsymbol{d}(a)}} \boldsymbol{\delta}_b^T \sum_{i \geq 2} \omega_i^t \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T \boldsymbol{\delta}_a.$$

Analyzing the right-hand part of this last expression, we find

$$\begin{aligned}
\left| \boldsymbol{\delta}_b^T \sum_{i \geq 2} \omega_i^t \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T \boldsymbol{\delta}_a \right| &= \left| \sum_{i \geq 2} \omega_i^t \left(\boldsymbol{\delta}_b^T \boldsymbol{\psi}_i \right) \left(\boldsymbol{\psi}_i^T \boldsymbol{\delta}_a \right) \right| \\
&\leq \sum_{i \geq 2} \omega_i^t \left| \boldsymbol{\delta}_b^T \boldsymbol{\psi}_i \right| \left| \boldsymbol{\psi}_i^T \boldsymbol{\delta}_a \right| \\
&\leq \omega_2^t \sum_{i \geq 2} \left| \boldsymbol{\delta}_b^T \boldsymbol{\psi}_i \right| \left| \boldsymbol{\psi}_i^T \boldsymbol{\delta}_a \right|.
\end{aligned}$$

To prove an upper bound on this last term, let $\boldsymbol{\Psi}$ be the matrix having the eigenvectors $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n$ in its columns. This is an orthonormal matrix, and so its rows must be orthonormal as well. Thus,

$$\begin{aligned}
\sum_{i \geq 2} \left| \boldsymbol{\delta}_b^T \boldsymbol{\psi}_i \right| \left| \boldsymbol{\psi}_i^T \boldsymbol{\delta}_a \right| &\leq \sum_{i \geq 1} \left| \boldsymbol{\delta}_b^T \boldsymbol{\psi}_i \right| \left| \boldsymbol{\psi}_i^T \boldsymbol{\delta}_a \right| \\
&\leq \sqrt{\sum_{i \geq 1} \left(\boldsymbol{\delta}_b^T \boldsymbol{\psi}_i \right)^2} \sqrt{\sum_{i \geq 1} \left(\boldsymbol{\delta}_a^T \boldsymbol{\psi}_i \right)^2} \\
&= \| \boldsymbol{\Psi}(b,:) \| \, \| \boldsymbol{\Psi}(a,:) \| \\
&= 1.
\end{aligned}$$

□

As $\omega_2 = 1 - \nu_2/2$, and

$$\omega_2^t = (1 - \nu_2/2)^t \approx e^{-t\nu_2/2},$$

we should expect random walks to converge once t reaches the order of $(\log n)/\nu_2$ or $1/\nu_2$.

10.7 Examples

We should now do some examples. I'd like to understand each in two ways: by examining ν_2 for each graph and by thinking about how a random walk on each graph should behave. While we have explicitly worked out λ_2 for many graphs, we have not done this for ν_2. The following lemma will allow us to relate bounds on λ_2 into bounds on ν_2:

Lemma 10.7.1. *Let $\boldsymbol{L}$ be the Laplacian matrix of a graph, with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, and let $\boldsymbol{N}$ be its normalized Laplacian, with eigenvalues $\nu_1 \leq \nu_2 \leq \cdots \leq \nu_2$. Then, for all i*

$$\frac{\lambda_i}{d_{min}} \geq \nu_i \geq \frac{\lambda_i}{d_{max}},$$

where d_{min} and d_{max} are the minimum and maximum degrees of vertices in the graph.

Proof. The Courant-Fischer theorem tells us that

$$\nu_i = \min_{\dim(S)=i} \max_{\boldsymbol{x} \in S} \frac{\boldsymbol{x}^T \boldsymbol{N} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}.$$

As the change of variables $\boldsymbol{y} = \boldsymbol{D}^{-1/2}\boldsymbol{x}$ is non-singular, this equals

$$\min_{\dim(T)=i} \max_{\boldsymbol{y} \in T} \frac{\boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y}}.$$

So,

$$\min_{\dim(T)=i} \max_{\boldsymbol{y} \in T} \frac{\boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y}} \geq \min_{\dim(T)=i} \max_{\boldsymbol{y} \in T} \frac{\boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}}{d_{max} \boldsymbol{y}^T \boldsymbol{y}} = \frac{1}{d_{max}} \min_{\dim(T)=i} \max_{\boldsymbol{y} \in T} \frac{\boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{y}} = \frac{\lambda_i}{d_{max}}.$$

The other bound may be proved similarly. □

10.7.1 The Path

As every vertex in the path on n vertices has degree 1 or 2, ν_2 is approximately λ_2, which is approximately c/n^2 for some constant c.

To understand the random walk on the path, think about what happens when the walk starts in the middle. Ignoring the steps on which it stays put, it will either move to the left or the right with probability 1/2. So, the position of the walk after t steps is distributed as the sum of t random variables taking values in $\{1, -1\}$. Recall that the standard deviation of such a sum is $\sqrt{t}$. So, we need to have $\sqrt{t}$ comparable to $n/4$ for there to be a reasonable chance that the walk is on the left or right $n/4$ vertices.

10.7.2 The Complete Binary Tree

As with the path, ν_2 for the tree is within a constant of λ_2 for the tree, and so is approximately c/n for some constant c. To understand the random walk on T_n, first note that whenever it is at a vertex, it is twice as likely to step towards a leaf as it is to step towards the root. So, if the walk starts at a leaf, there is no way the walk can mix until it reaches the root. The height of the walk is like a sum of ± 1 random variables, except that they are twice as likely to be -1 as they are to be 1, and that their sum never goes below 0. One can show that we need to wait approximately n steps before such a walk will hit the root. Once it does hit the root, the walk mixes rapidly.

10.7.3 The Dumbbell

Now, let's consider another one of my favorite graphs, the dumbbell. The dumbell graph D_n consists of two complete graphs on n vertices, joined by one edge. So, there are $2n$ vertices in total. The isoperimetric number of this graph is

$$\theta_{D_n} \sim \frac{1}{n}.$$

Using the test vector that is 1 on one complete graph and -1 on the other, we can show that

$$\lambda_2(D_n) \lessapprox 1/n.$$

Lemma 10.7.1 then tells us that

$$\nu_2(D_n) \lessapprox 1/n^2.$$

To prove that this bound is almost tight, we use the following lemma.

Lemma 10.7.2. *Let G be an unweighted graph of diameter at most r. Then,*

$$\lambda_2(G) \geq \frac{2}{r(n-1)}.$$

Proof. For every pair of vertices (a, b), let $P(a, b)$ be a path in G of length at most r. We have

$$\boldsymbol{L}_{(a,b)} \preccurlyeq r \cdot \boldsymbol{L}_{P(a,b)} \preccurlyeq r\boldsymbol{L}_G.$$

So,

$$K_n \preccurlyeq r\binom{n}{2} G,$$

and

$$n \leq r\binom{n}{2}\lambda_2(G),$$

from which the lemma follows. □

The diameter of D_n is 3, so we have $\lambda_2(D_n) \geq 2/3(n-1)$. As every vertex of D_n has degree at least $n-1$, we may conclude $\nu_2(D_n) \gtrapprox 2/3(n-1)^2$.

To understand the random walk on this graph, consider starting it at some vertex that is not attached to the bridge edge. After the first step the walk will be well mixed on the vertices in the side on which it starts. Because of this, the chance that it finds the edge going to the other side is only around $1/n^2$: there is only a $1/n$ chance of being at the vertex attached to the bridge edge, and only a $1/n$ chance of choosing that edge when at that vertex. So, we must wait some multiple of n^2 steps before there is a reasonable chance that the walk reaches the other side of the graph.

10.7.4 The Bolas Graph

I define the bolas[2] graph B_n to be a graph containing two n-cliques connected by a path of length n. The bolas graph has a value of ν_2 that is almost as small as possible. Equivalently, random walks on a bolas graph mix almost as slowly as possible.

The analysis of the random walk on a bolas is similar to that on a dumbbell, except that when the walk is on the first vertex of the path the chance that it gets to the other end before moving back to the clique at which we started is only $1/n$. So, we must wait around n^3 steps before there is a reasonable chance of getting to the other side.

Next lecture, we will learn that we can upper bound ν_2 with a test vector using the fact that

$$\nu_2 = \min_{\boldsymbol{x} \perp \boldsymbol{d}} \frac{\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{D} \boldsymbol{x}}.$$

To prove an upper bound on ν_2, form a test vector that is $n/2$ on one clique, $-n/2$ on the other, and increases by 1 along the path. We can use the symmetry of the construction to show that this vector is orthogonal to $\boldsymbol{d}$. The numerator of the generalized Rayleigh quotient is n, and the denominator is the sum of the squares of the entries of the vectors times the degrees of the vertices, which is some constant times n^4. This tells us that ν_2 is at most some constant over n^3.

To see that ν_2 must be at least some constant over n^3, and in fact that this must hold for every graph, apply Lemmas 10.7.1 and 10.7.2.

10.8 Final Notes

The procedure we have described—repeatedly multiplying a vector by $\boldsymbol{W}$ and showing that the result approximates $\boldsymbol{\pi}$—is known in Numerical Linear Algebra as the power method. It is one of the common ways of approximately computing eigenvectors.

In the proof of Theorem 10.6.1, we were a little loose with some of the terms. The slack comes from two sources. First, we upper bounded ω_i by ω_2 for all i, while many of the ω_i are probably significantly less than ω_2. This phenomenon is often called "eigenvalue decay", and it holds in many graphs. This sloppiness essentially costs us a multiplicative factor of $\log n$ in the number of

[2] A bolas is a hunting weapon consisting of two balls or rocks tied together with a cord.

steps t we need to achieve the claimed bound. You will note that in the examples above, the time to approximate convergence is typically on the order of $1/\nu_2$, not $(\log n)/\nu_2$. This is because of eigenvalue decay.

The second source of slack appeared when we upper bounded the absolute value of a sum by the sum of the absolute value. I am not sure how much this cost us.