Exploring Networks

Russ Thompson

1 Introduction

Graphs are a way of realizing the relationships between things. Often they are called networks, and in this guise they are probably familiar to you. The interactions of proteins in your body and the relationships between you and your friends give rise to such networks. In its most basic incarnation a graph is a collection of vertices, V, and a collection of edges, E. Each edge connects a pair of vertices (u, v) and represents a connection between these vertices. Several examples of graphs are shown in Figure 1.

Graphs have been studied by mathematicians for a very long time. Perhaps the most famous early example of a mathematical problem involving graphs is Euler's Königsberg bridge problem [2]. In 1735, when Euler presented this problem to the St. Peterburg Academy, Königsberg (now called Kaliningrad) was a Prussian town situated around the Pregel river. The city also encompassed two islands in the river, and there were seven bridges connecting these islands to the mainland (see Figure 2). Euler's question was whether or not it was possible to walk across each of the bridges without walking across any bridge more than once. This simple question is surprisingly tricky, and a full understanding of the problem tells us something beyond the scope of a handful of bridges. For now we let the reader ponder this question.

DRAFT

Question 1 (Euler's Question)**.** *Is it possible for someone to take a stroll in Königsberg and cross each bridge exactly once? Is the situation any different if we add or subtract islands and/or bridges (i.e. vertices or edges)?*

An important lesson of the Königsberg bridge problem is that graphs help us get rid of extraneous information. Instead of a complicated map we can restrict ourselves to a very simple object without losing any important information. Euler's question also introduces us to thinking about graphs geometrically. A graph can be thought of as a space just like the one we live in except that movement is restricted. We are only allowed to stand

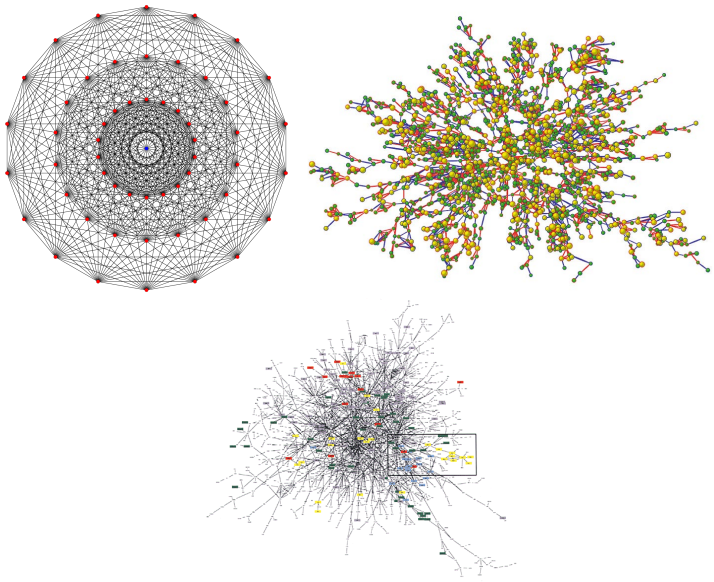


Figure 1: Graphs of the group $E7$, individuals in the Framingham heart study [4], and protein interaction network for yeast [16].

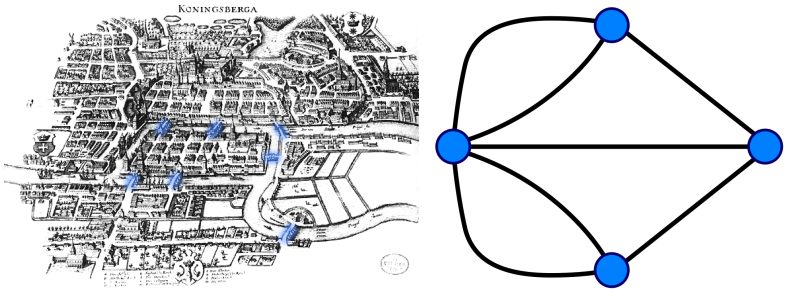


Figure 2: Königsberg and its a graph of the city obtained by treating bridges as edges.

at vertices, and edges tell us which vertices we can step to next. When we think about graphs in this way we will often refer to *paths* which are just sequences of edges that share endpoints. A path describes both the order in which we stop at vertices, and the edges we traverse to visit these vertices. A path has a natural *length* which is to the number of edges it traverses. Of course we can also think of our edges as particular lengths like sections of road between intersections, and when we do this we will refer to the length of an edge as its *weight*. Sometimes we will also care about the *degree* of a vertex which is the number of edges incident to that vertex. We will denote this number by $\deg(v)$ for a given vertex v.

While mathematicians have been studying graphs for the almost three centuries since Euler formulated his question, it was not until the 20th century that graphs become important objects of study in other disciplines. One of the first hints of this was psychologist Stanley Milgram's experiment [11]. Milgram's experiment was based on an idea that had been circulating since the early 20th century that most people were socially connected through a small number of other individuals; colloquially this was known as "six degrees of separation." In his experiment Milgram gave letters to people who were then to mail the letters to personal acquaintances with the ultimate goal of having the letters reach a friend of Milgram's who lived in New York. Milgram found that those the letters which made it to his friend passed on average through three hands. This experiment led the way in the study of what would later be called *small-world networks*. These networks are those in which, despite a large number of vertices, any pair of vertices can be joined by a short path.

Milgram's experiment also highlights the notion of a social network. You are part of many social networks, from the digital variety like Facebook or Twitter to more mundane networks like your family tree. However, not all networks are social. Computer networks are an obvious example but road systems, food webs and protein interactions are also important examples. Many of these are or are thought to be small-world networks [12]. That so many subjects can be described as graphs hints at the importance of understanding graphs in full generality: if mathematics can tell us something about graphs, then it is telling us something wherever we can use graphs to model reality.

Much of the early studies of graphs treated them as concrete objects, and attempts to understand their structure. More recent approaches treat them as underlying objects where the primary interest is in a process on the graph. We can contrast these views as studying a social network as a geometric object, e.g. whether it is a small world or not, or studying how

a rumor spreads through the network. Both viewpoints have much to offer, and are interrelated: the best way to know the geometry of a graph is often to stumble through it, much like how you learn to navigate in a new town. Our own approach to graph theory will model this. First we will explore several related topics that are concerned with graphs as objects, and later we will turn our attention to some subjects that are concerned with processes on a graph. As we proceed it will become clear how closely related these two views of graphs are.

2 Graphs as Geometry

In this section we will deal with problems from the earlier days of graph theory. These problems are concerned with the structure of graphs. Often this is done by identifying particular subgraphs. As an example, you can think of the roads in your hometown as a subgraph of the entire road network of the country. This is similar in spirit to why we study prime factorizations of integers or why we care to factor polynomials. As graphs can be quite large and complicated, being able to understand them in terms of smaller graphs is very useful.

Before we go any further, there are a few technical assumptions we need to make. First we will assume that our graphs do not have loops, i.e. there are no edges whose two endpoints are the same vertex. We make this assumption to simplify our discussions; adding loops to our graphs usually doesn't provide any extra information, but forces us to talk about them anyway. We will also assume that our graphs are *connected* which means that any pair of vertices can be connected by some path in the graph. Disconnected graphs can be thought of as some number of entirely separate graphs called *connected components*, and for most of our discussion we can deal with connected components separately to answer the questions at hand.

2.1 Eulerian Paths

Now we return to Euler walking the bridges of Königsberg. After enough pacing through the streets (or pondering his map), Euler realized that it was impossible to cross each bridge exactly once, and then he began to wonder why exactly this was true. We can't promise to follow Euler's thoughts exactly, but we can end at the same conclusion. Our first step is to define precisely what we are talking about. To make the problem more realistic we will suppose that Euler wants to take his stroll starting from his house and

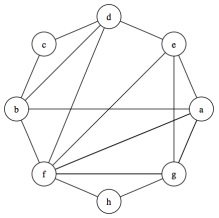


Figure 3: Can you find an Eulerian cycle in this graph?.

return there at its end. Such a path is called a cycle, and we are looking for a particular kind of cycle.

Definition 2 (Eulerian Cycle)**.** An *Eulerian cycle* (or an Eulerian path) is a cycle (or a path) which traverses each edge in the graph exactly once.

There is an observation about paths so plain that we might overlook it: whenever a path steps onto a vertex, it must also leave that vertex, unless the path has ended. For an Eulerian *cycle*, this would mean that the path has returned to its starting location. We can also see that traversing each edge in a graph requires that the path also visits each vertex. So for a graph to have an Eulerian cycle each vertex must have an entry edge and an exit edge for each time the cycle visits it. Furthermore, no edge can be used for both entry and exit, or be used for either more than once. This means that each vertex must have an even number of edges incident to it.

Question 3. *Try to find an Eulerian cycle in the graph in Figure 3. Is there more than one?*

In fact *any* graph whose vertices have only even degree has an Eulerian cycle, which gives us the following theorem [19].

Theorem 4. *A graph has an Eulerian circuit if and only if each vertex has even degree. Furthermore, there is a Eulerian path between two vertices if and only if they are the only vertices of odd degree.*

2.2 Hamiltonian Cycles

If we go forward another century we come to another important problem from the early history of graph theory. This question started as the Icosian game invented by the Irish mathematician William Hamilton in 1857 [2].

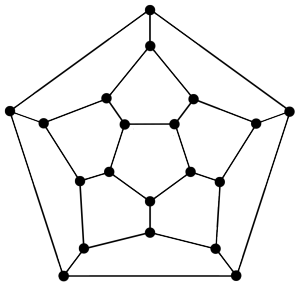


Figure 4: The Icosian graph.

The goal is to take a graph, called the Icosian graph (see Figure 4), and find a cycle which visits each *vertex* exactly once without crossing any edge more than once. You might want to try solving the puzzle yourself; a solution appears at the end of this section if you get stuck.

We can generalize the Icosian game to arbitrary graphs, and ask when they have cycles that visit each vertex. As with Euler's question we should give ourselves a definition to work with.

Definition 5 (Hamiltonian Cycle)**.** A *Hamiltonian cycle* is a cycle which visits every vertex of a graph and visits no edge more than once in a graph. Graphs with Hamiltonian cycles are called Hamiltonian.

We should also note that Hamiltonian cycles are closely related to Eulerian cycles; the only difference is that Hamiltonian cycles care about visiting vertices while Eulerian cycles care about visiting edges. It is often the case in graph theory that a question posed about the edges of a graph can be reformulated into an interesting question about vertices and vice-versa.

A closely related problem to the existence of a Hamiltonian cycle in a graph is the *traveling salesman problem*. Consider a salesman going from town to town trying to sell his goods, then returning home at the end of the day. Trying to save time and gas the salesman wants to do make his visits in the most efficient way possible. In Figure 5 you can see an optimal route through the largest German cities. You can easily think of each of these cities as being connected by road, where each edge has a cost corresponding to the length of the road. We will say the cost of a path in this graph is the sum of the costs of the edges in the path. The optimal route is already a Hamiltonian cycle for the road network, but there is something else special about it as well.



Figure 5: An optimal cycle through the 15 largest German cities.

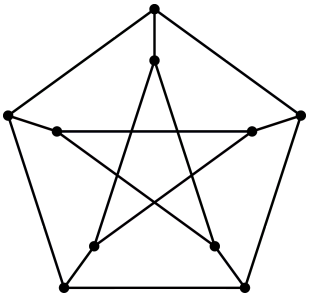


Figure 6: The Petersen graph. The vertices are those edge crossings with dots.

Fact 6. *The optimal path for a traveling salesman is the lowest cost Hamiltonian cycle in the graph (if it exists).*

The "if it exists" part of this fact is quite important. There are graphs which don't have Hamiltonian cycles just like there are graphs without Eulerian circuits. An interesting example of a graph without a Hamiltonian cycle is the Petersen graph (see Figure 6). A few attempts to find a Hamiltonian cycle should convince you that none exist in this graph. However the Petersen graph does contain a Hamiltonian *path*, that is a path which visits each vertex without crossing any edge more than once. Can you find such a path?

The traveling salesman problem, or rather efficient ways of finding nearly optimal solutions to it, has many important applications. One of the most often cited is in the construction of microchips [5]. To make a microchip, large silicon wafers are sent through an assembly line where a robotic arm needs to solder a large number of circuits in a short amount of time. After

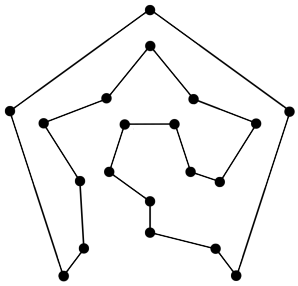


Figure 7: A Hamiltonian cycle in the Icosian graph.

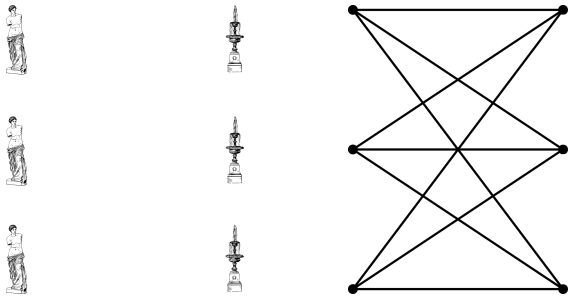


Figure 8: Fancygrass's garden and a graphical representation of it.

this the wafer is cut into the tiny chips found inside devices like computers or cellphones. The spots that need soldering become our vertices and each is connected to the others by an edge whose cost is the distance between them. An optimal solution to TSP would make the assembly line faster and save wear on the robotic arm. This example tells us that basic graph theory isn't just an abode of interesting mathematics, but is also home to useful techniques to solve industrial problems.

2.3 Planarity

Suppose that noted landscape designer S.P. Fancygrass has asked you for help. In her latest project there are to be three statues and three fountains (see Figure 8), and she wants there to be a stone path from each statue to each fountain. You start to question her need for a mathematical consultant, but then she tells you that she doesn't want the paths to cross.

Question 7. *Is there a way to redraw the edges of the graph in Figure 8 so*

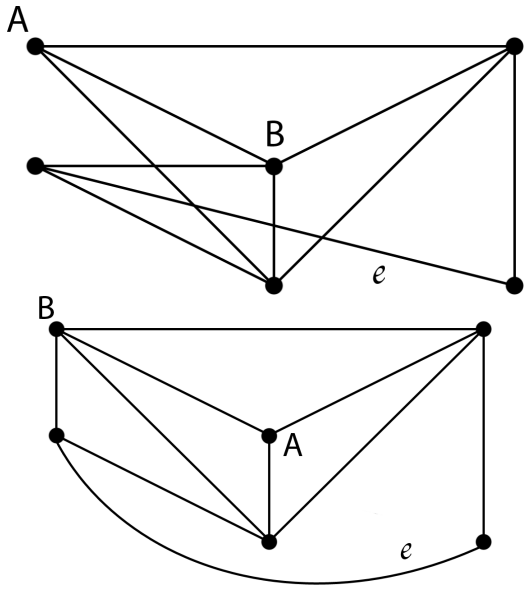


Figure 9: Two versions of the same graph. To make the first version planar the locations of the vertices A and B were changed, and the edge e was moved.

that they do not cross each other?

After pondering this conundrum for a while you can see there is simply no way to place the edges without them crossing. The only solution is for Fancygrass to build some bridges. You've stumbled upon a geometric property of graphs:

Definition 8. A graph is *planar* if it can be drawn in the plane without its edges crossing each other. While we normally draw our graphs with straight edges, when talking about planarity it's useful to allow edges to be curved.

The graph representing Fancygrass's garden is actually what is called a complete bipartite graph. A *bipartite* graph is one where there are two disjoint sets of vertices and each edge connects a vertex in one of these sets to a vertex in the other set. A bipartite graph is called *complete* when all possible edges which could go between the two sets are present. We use the notation $K_{n,m}$ to signify the complete bipartite graph where one of the sets has size n and the other has size m. The graph representing Fancygrass's garden is $K_{3,3}$.

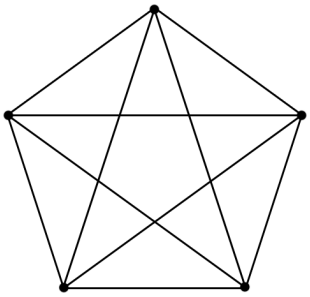


Figure 10: The complete graph on five vertices, K_5.

Another non-planar graph is the complete graph on five vertices (see Figure 10). We say a graph is *complete* when all possible edges are present. We denote a complete graph with n vertices as K_n. In fact K_5 and $K_{3,3}$ are the smallest non-planar graphs. While it would be very tedious to check by hand that this is true, doing a few examples is enlightening.

Question 9. *Suppose you make a new graph by removing an edge from either K_5 or $K_{3,3}$. Can you draw this graph without edges intersecting?*

It turns out that $K_{3,3}$ and K_5 are the atomic building blocks of non-planar graphs. The following theorem tells us precisely when we will not be able to draw a graph in the plane. First we need to introduce a notion similar to that of subgraphs. A *minor* of a graph is a new graph that can be formed by shrinking edges so that their endpoints merge into a single vertex.

Theorem 10 (Kuratowski's Thoerem)**.** *A graph is planar if and only if neither K_5 or $K_{3,3}$ are minors [19].*

This Kuratowski's theorem was an early hint at a much larger theorem about graphs called the *forbidden minor theorem*. The theorem states that there are many properties of graphs which are determined solely based on whether or not a given graph has certain minors. This is a very powerful result, but it is incredibly difficult to prove. In fact the full proof took the mathematicians Neil Robertson and Paul Seymour twenty-two papers to accomplish [8].

As an aside, we should point out that many planar graphs actually have interesting realizations as three dimensional objects. Let's consider a cube. There are eight vertices corresponding to the corners of the cube, twelve

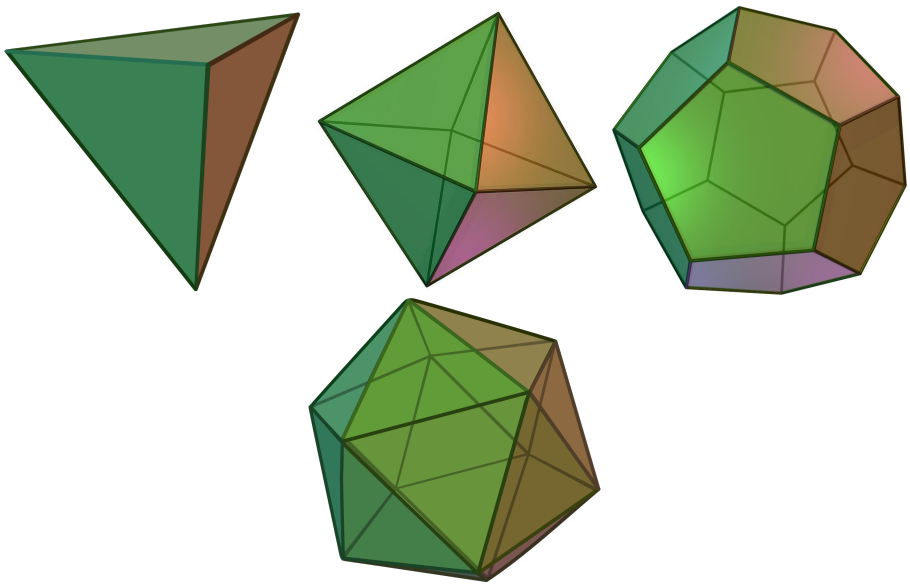


Figure 11: A tetrahedron, an octahedron, a dodecahedron, and an icosahedron.

edges connecting these vertices, and six square faces. If you add the number of vertices and faces then subtract the number of edges you get two. We'll denote sums like this by $V + F - E$. This works for other polyhedra as well. Can you verify that $V + F - E = 2$ for the other polyhedra in Figure 11?

What does this have to do with graphs? First let's think about polygons, which are really simple planar graphs. A polygon has the same number of vertices and edges, and if we count the inside and the outside as faces, then $V + F - E = 2$. For planar graphs (drawn with no edges intersecting), each region surrounded by edges will count as a face. (see Figure 12). The region outside the graph is surrounded by edges as well, so we will count it as a face as well. Once we do this we can check that $V + F - E = 2$ for any planar graph.

Theorem 11. *For any planar graph*

$$V - E + F = 2.$$

That Theorem 11 also seems to hold for polyhedra is no coincidence. Having $V - E + F = 2$ essentially tells us that polyhedra are made up of flat objects, namely polygons glued together along their edges. This means

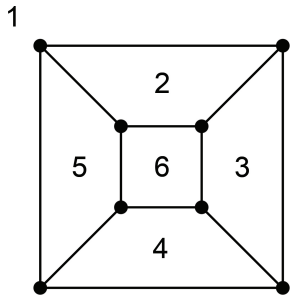


Figure 12: A planar graph with the faces enumerated. Can you see how this graph is related to a cube?

there should be a way to flatten a polyhedron into planar graphs. To do this suppose the faces of a polyhedron are transparent while the edges and vertices are opaque. We can shine a light through the polyhedron, and by looking at the shadow it casts get a graph. Rotating the polyhedron changes the appearance of the graph, and if we line it up just right the shadow graph will be planar. Figure 13 shows an example of this with a tetrahedron, and Figure 14 shows the planar graphs corresponding to to the polyhedra in Figure 11.

2.4 Friends and Strangers

In this section we will use the structure of graphs to solve a counting problem. This problem actually originated in the 1950s, and has since been enshrined in mathematical folklore [7]. Suppose you want to throw a party, and in a fit of whimsy you decide there must be at least three mutual friends or at least three mutual strangers present. To make this interesting, let's suppose you don't know whether or not your friends know each other, so you can't do something clever like inviting three people who already know each other.

Question 12. *How many people must you invite to guarantee at least three mutual friends or three mutual strangers are present?*

The answer to this question is surprisingly small. We'll model the party using the complete graph K_n where n is the number of people at the party. Next we find a *coloring* of the graph, which is just a way of distinguishing kinds of edges. If two people are friends then we will color the edge between them blue, and if they are strangers we color the edge red. Now finding

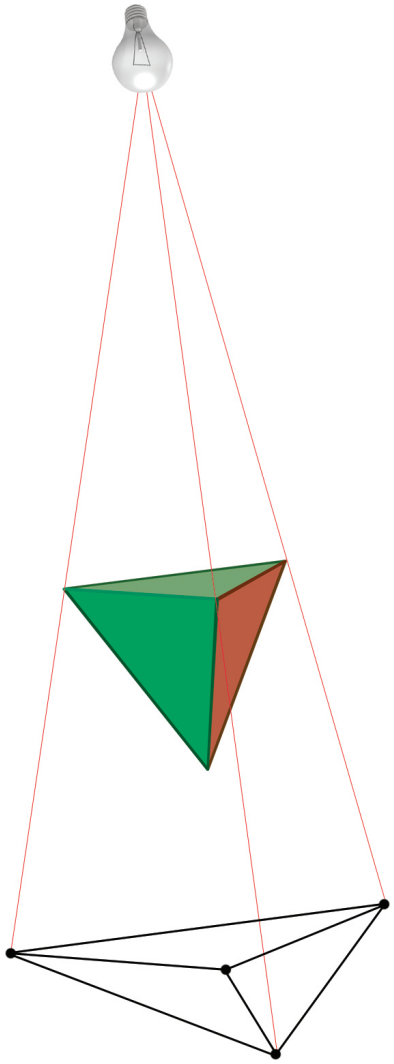


Figure 13: The shadow of a tetrahedron is a planar graph.

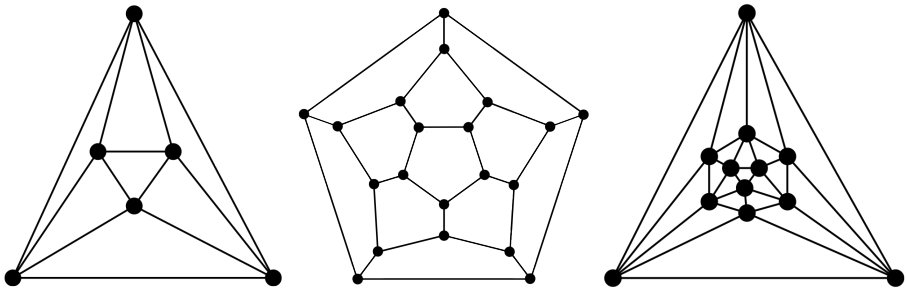


Figure 14: Planar graphs corresponding to the tetrahedron, cube, octahedron, and dodecahedron. The planar graph of the dodecahedron is the same as the graph used for the Icosian game.

three mutual friends or three mutual strangers is the same as finding a *monochromatic* copy of K_3 in this coloring of K_n.

Question 13. *How many edges are there in K_n? Drawing K_3, K_4 and K_5 might help you figure out the pattern.*

Question 14. *Can you find colorings of K_4 and K_5 which do not contain monochromatic colorings of K_3?*

What happens when we try to color K_6 without creating a monochromatic copy of K_3? To start our coloring let's pick a vertex. This vertex has five edges incident to it. While we don't know exactly how they are colored, we can tell that at least 3 of these edges will have the same color. We'll say that these three edges are red. At this point our coloring resembles the one in Figure 15.

Next we consider what happens between the vertices at the end of these red edges. These vertices can't be connected by red edges because this would make a red copy of K_3. Also the edges between these three vertices can't all be blue, as this would make a blue copy of K_3. What if we let two of them be blue (see Figure 16). This is a problem. None of these edges can be red, but they can't all be blue either. This means that there's no way to color these edges without creating a monochromatic copy of K_3 in K_6,.

The entire argument we just went through doesn't depend on which edges or vertices we consider; it is entirely determined by the use of 2 colors and the structure of K_6. So inviting six people is sufficient to guarantee at least three mutual friends or at least three mutual strangers. Of course if you invite more than six people to your party the graph of the party will

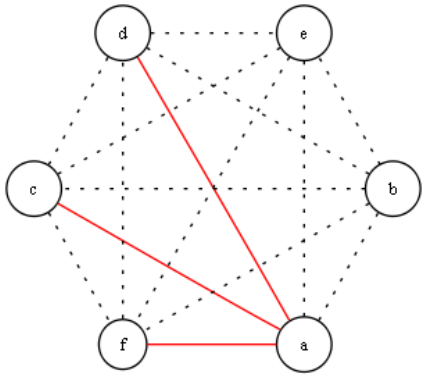


Figure 15: Three edges around a are chosen to be red.

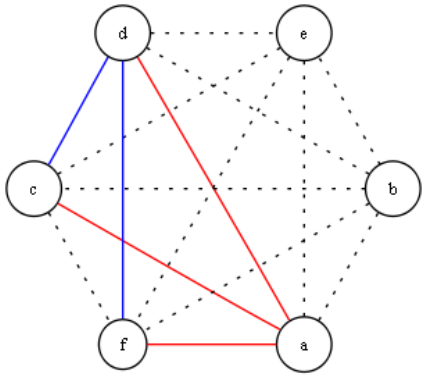


Figure 16: How can we color the edge (c, f) without creating a monochromatic K_3?

still contain a two-coloring of K_6, so you will still have at least the desired number of mutual friends or mutual strangers present.

We can ask how large a party one might need in order to have 4 mutual friends or 4 mutual strangers, or even n mutual friends and n mutual strangers. The size of the smallest number for which this is possible is called the Ramsey number, $R(n, n)$ [7].

A similar technique to the one used above allows us to calculate that $R(4, 4) = 18$. In general, however, $R(n, n)$ is difficult to calculate. Even computers aren't much help. There are 2^m two-colorings of K_m and each coloring has to be checked to see if it has n mutual friends or mutual strangers. To find $R(n, n)$ you'd have to find the smallest complete graph such that any two-coloring contains a monochromatic copy of K_n. Joel Spencer, speaking of the famous Hungarian mathematician Paul Erdös, offered an apt description of the difficulty of this task [17]:

> Erdös asks us to imagine an alien force, vastly more powerful than us, landing on Earth and demanding the value of $R(5, 5)$ or they will destroy our planet. In that case, he claims, we should marshal all our computers and all our mathematicians and attempt to find the value. But suppose, instead, that they ask for $R(6, 6)$. In that case, he believes, we should attempt to destroy the aliens.

3 Processes on Graphs

Much of the recent research on graphs has been motivated by their relationship to complex real world systems. While it is often interesting to examine the structure of the graphs that arise from computer, social, or biological networks, many important applications relate to processes on graphs. By a process we mean something that happens over time on the graph. This may be something like the appearance of new edges or vertices in a computer network as new machines or connections are added, or the spread of a virus through either social or computer networks.

This latter application has been very fruitful. A good model for the spread of a virus in a network helps us pinpoint where to apply vaccines. In particular, people with many social contacts as well as many possible contacts with infected individuals (i.e. high degree vertices near infected vertices) have been shown to be excellent targets for vaccination. Graph theory can also tell us what percentage of the population needs to receive a vaccine to cut off an epidemic before it can takeoff [9].

Beyond viruses, many other things can spread through social networks. Recent analysis of decades' worth of data assembled in the Framingham Heart Study has shown that things like happiness, smoking, or obesity can spread through social networks[18, 4]. One of the most intriguing results from this study is that the spread of these things is really determined by the social network. Each individual's ability to influence others is limited to friends of friends of friends, that is, vertices which are at most three edges away.

In pure mathematics the study of processes on graphs has been of much value as well. Random walks on graphs have helped shine light on the geometry of graphs, while random walks and other random processes on graphs provide many interesting and surprising examples in the study of probability. Later in this section we will dip our toes into the study of random walks on graphs.

All the processes we'll talk about will happen in discrete time steps. For some phenomena this isn't an unrealistic assumption; for instance if we were modeling a species that only reproduces during a mating season. For models like the spread of flu through a population, continuous time models are more appropriate. We avoid talking about these sorts of models because they have a higher technical overhead without behaving very differently from discrete time models. In fact many continuous time models can be approximated by discrete time models with very small time steps.

3.1 Fads and Epidemics

You're probably familiar with some trend or fad that has swept through your social network. Fads tend to start with a few early adopters whose friends or acquaintances join in, and from there the trend snowballs. We might also think in terms of innovation. In this context we can imagine that early adopters are using some new technology, and over time others in their social network adopt this technology. Either way there are often communication issues between those who have adopted and those who haven't, and we use this insight to help model how trends spread. The model of influence we present was introduced by the computer scientist John Kleinberg [10].

For this model we will use a coloring of our graph. We will color the vertices of our graph red, R, and blue, B, where red will signify people who have adopted the trend. The initial red vertices will be called early adopters. Each edge of the graph (u, v) represents a friendship and comes with a *pay-off function*, $p(u, v)$, whose value is between 0 and 1. The trend is also given a *strength* t between 0 and 1. This lets us explicitly write the

pay-off function.

Definition 15 (Pay-off Function)**.** The pay-off function is defined as

$$p(u,v) = \begin{cases} t & : u \text{ and } v \text{ are both red} \\ 1-t & : u \text{ and } v \text{ are both blue} \\ 0 & : u \text{ and } v \text{ are different colors} \end{cases}$$

A vertex will change its color when it receives more benefit by being the new color. To make this decision a vertex v needs to keep track of its *net pay-off* function $p_v(C)$ for each color C. When calculating $p_v(C)$ we assume that v is color C. Explicitly,

$$p_v(C) = \sum_{u:\ u \sim v} p(v,u),$$

where $u \sim v$ mean that the edge (u,v) is in our graph.

We'll let the trend spread in discrete time steps. We assume that the vertices update simultaneously, so that a vertex's new color depends only on its neighbors' colors during the prior time step. At each time step every vertex v picks its color depending on which net pay-off function is larger, $p_v(B)$ or $p_v(R)$. In case the two net pay-off functions are tied v chooses to become red. You should be aware that this setup makes even very strong trends likely to die out if there are few early adopters or if the early adopters are friends with many non-early adopters. Figure 17 provides an example of a trend spreading in a graph.

Question 16. *What might happen if the example in Figure 17 started with the early adopter in a different location?*

Question 17. *What happens to a trend with strength $t = 0.5$ on the graph in Figure 18? Can you figure out what happens for $t < 0.5$ or for $t > 0.5$? Also test this using two neighboring vertices as early adopters instead. What changes?*

The last question demonstrates an important mathematical phenomenon at work in this model. For small strength values the trend won't spread at all, while for large strength values the trend will spread throughout the entire graph (though some vertices may alternate between red and blue). This change in behavior happens all of a sudden at a specific value of t called the *critical threshold*. For this model of social influence the critical threshold

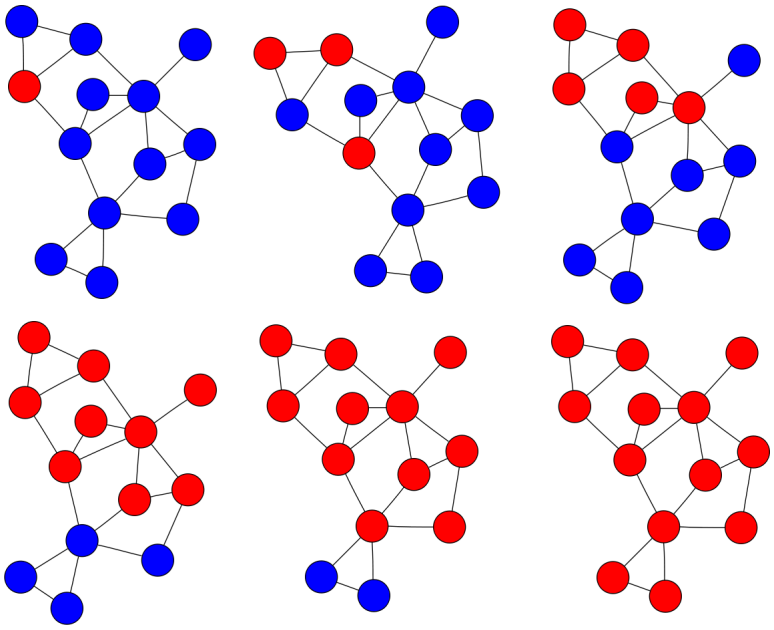


Figure 17: Example of a trend spreading in a graph with $t = 0.75$. While the early adopter abandons the trend after the first time step, the trend continues to spread until everyone has adopted.

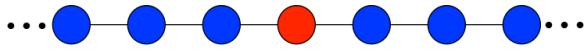


Figure 18: An interesting graph on which to test different trend strengths.

is at least 1/2 regardless of what the graph looks like [10]. Intuitively, this must be true because it wouldn't make sense for a trend that was weaker than the status quo to spread.

Sudden shifts from one type of behavior to another as we vary a parameter are called *critical phenomena* [8]. Critical phenomena arise in many different places in mathematics and the sciences. For instance the freezing and boiling points of water are critical thresholds related to the organization of water molecules.

What makes critical phenomena so interesting is that many different physical systems have similar critical phenomena. This is explained by a theory called *universality*, which connects the critical phenomenon of many complex real world systems to that of a simpler mathematical model [8]. Even in mathematics, we often find that simple models show the exact same critical behavior as complex models, and so we can understand a difficult problem by understanding an easier problem.

3.2 Random Walks On Infinite Graphs

> "A drunk man will find his way home, but a drunk bird may get lost forever."
>
> –Shizuo Kakutani [6]

A *random walk* on a graph is a path whose edges are chosen randomly. Often random walks are called drunkard's walks, as we can imagine the path being determined by a drunk man who stumbles from intersection to intersection in a vast city. At each intersection the drunk sees a number of streets and picks one at random (even the one he just stumbled down) to follow to another intersection.

No matter how large the city is, the drunkard will always return to his starting point if we give him enough time. The caveat here is that the city really needs to be two dimensional, hence Kakutani's quip. If instead of a drunk man we imagine a drunk bird flying through space, we would discover that given enough time the bird would eventually fly away from its starting point never to return. Our goal in this section is to make the behavior of man and bird mathematically concrete.

We will represent this infinite city or space using graphs. Unlike the graphs we have discussed before, these will have an infinite number of edges and vertices. Luckily, you are already well acquainted with some infinite graphs though you might not know it. Long ago you were probably introduced to the number line, and maybe you went so far as to mark off

the numbers "$\ldots, -1, 0, 1, \ldots$" on it. What the number line represents is a graph of the integers; each integer is a vertex, and we draw an edge between two integers if they differ by 1 or -1. In particular we can view each vertex as an integer n, and we move to the left or right by adding -1 or 1, respectively. We will call this graph $\mathbb{Z}$, which is the standard notation for the integers. Similarly we can think about the two-dimensional grid $\mathbb{Z}^2$, given by pairs of integers, as an infinite graph. The graph $\mathbb{Z}^2$ looks like an infinite sheet of graph paper. For the bird we will use the graph $\mathbb{Z}^3$ which consists of triplets of integers. This graph looks like a bunch of cubes stacked together. For each of these graphs vertices are connected by an edge when only one of their coordinates differ by ± 1. Often $\mathbb{Z}^d$ is called the d-dimensional integer lattice.

3.2.1 Transience and Recurrence

Before we can say what it means to always return home or to be lost forever, we need a better way to talk about random walks. We'll assume that home is always the origin, i.e. 0 in $\mathbb{Z}$ or $(0, 0)$ in $\mathbb{Z}^2$, and we'll generally denote this point by 0. To make things easier we'll assume that time only moves in discrete chunks so that we only need to talk about which vertex the drunkard is at time t where t is an integer. Finally we'll let $X(t)$ denote the position of the walker at time t. The walker has returned home when $X(t) = 0$.

How does the walker move? Each vertex in $\mathbb{Z}^d$ has 2^d edges incident, and the walker choses to walk down each edge with probability $1/2^d$. Figure 19 shows what the walker's path in $\mathbb{Z}^2$ looks like at different times. Random walks are examples of Markov Chains, which are random processes whose next state only depends on the current state. This is true because the drunkard's choice of which street to travel next doesn't depend on how he got to the intersection where he now stands.

We want to think about what happens to our walker over an infinite amount of time. Being lost forever means that the walker can only return home finitely many times. This leaves an infinite amount of time after that last return in which the walker never returns home again. We call this *transience*. If the walker will always return home given enough time, i.e. the walker returns home an infinite number of times, we say the walker is *recurrent*. Transience and recurrence are mutually exclusive. We want to think of these terms probabilistically. Suppose you were to look at all paths of length n in $\mathbb{Z}^2$. Since the walker picks edges with equal probability, each of these 4^n paths is equally likely to be the one the walker takes. If you were to look at these paths, you would find that most visit the origin many times.

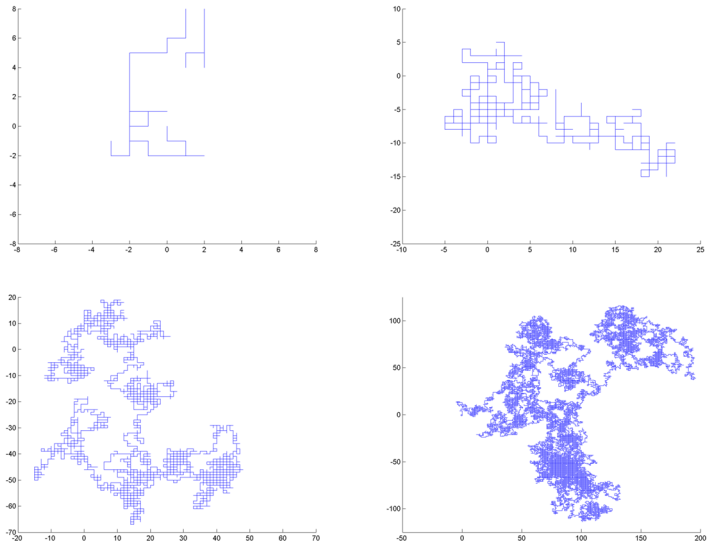


Figure 19: Random walks of 50, 500, 5000 and 50000 steps in $\mathbb{Z}^2$.

Something similar happens when we start talking about infinite paths, but we need a tool for describing this.

Visiting the origin infinitely often (recurrence) or finitely often (transience) are *tail events* of the random walk. An event is just something that can happen for the random walk like the fifth visit to the origin. You can imagine the body of the random walk as what has happened up to some finite time t and whatever will come after that as the tail. When we talk about tail events we are no longer talking about a specific path of the walker, but the ultimate behavior of the random walk. In particular, no matter how many times we start our walker at the origin, his tail behavior will always be the same. This means tail events can be tricky to deal with, but there is a powerful theorem that helps us make sense of them:

Theorem 18 (Kolmogorov's 0-1 Law)**.** *A tail event occurs with either probability 0 or 1 [6].*

This theorem tells us that if we have a tail event then it will either always happen or never happen. Which of these is the case depends solely on the graph so long as the walker doesn't have a bias for moving in a particular direction (for instance a walker that always goes to the right on $\mathbb{Z}$ never returns).

3.2.2 Probability of Return

We will determine whether or not a graph is transient using the probability that starting from the origin the walker will be back at the origin after taking n steps. We denote this *probability of return* by $\phi(n)$. The sum of the probabilities of return for all times n,

$$\sum_{n=1}^{\infty} \phi(n), \tag{1}$$

can be used to determine if the walk is transient or recurrent. This sum approximates the expected number of returns the walker will make to the origin, though we need a more precise understanding of how it relates to transience. This comes to us via the following theorem:

Theorem 19 (Borel-Cantelli Lemma)**.** *A random walk is transient if and only if $\sum_{n=1}^{\infty} \phi(n)$ is finite. Otherwise it is recurrent [6].*

Since the Borel-Cantelli lemma only cares if $\sum_{n=1}^{\infty} \phi(n)$ is finite, we never need to find its exact value. Furthermore, it also means that we don't need to know $\phi(n)$ exactly; we just need a reasonable approximation. Before going further there is another important fact we need.

Fact 20. *The sum*

$$\sum_{n=1}^{\infty} \frac{1}{n^{\alpha}}$$

is finite if and only if $\alpha > 1$.

For the graphs $\mathbb{Z}^d$ we actually don't need to worry about the probability of return for odd-numbered times. The random walks on these graphs are *periodic*: for even n the coordinates of $X(n)$ are even (recall that 0 is even), while for odd n they are odd. This means that $\phi(n) = 0$ when n is odd. From here on we'll only consider $\phi(2n)$ so that it is clear we are thinking of taking an even number of steps. Dealing with $2n$ instead of n isn't really a problem because the 2 only influences the probability of return by a cosntant multiple, i.e. we will still be showing that $\phi(2n) \approx n^{-\alpha}$.

Combining all these facts we have the following test for recurrence.

Fact 21. *A random walk is recurrent if and only if $\phi(n) \approx n^{-\alpha}$ for $\alpha \leq 1$. Otherwise it is transient.*

3.2.3 Estimating the Probability of Return

We start by estimating the probability of return on $\mathbb{Z}$. To do this we need to know how many paths there are in $\mathbb{Z}$ of length $2n$ that start at the origin, and how many of those paths end up at the origin. At every step the walker can go either left or right, so there are a total of 2^{2n} paths. For a path to return to the origin after $2n$ steps, half of these steps must be to the right while the other half are to the left.

To count these we can imagine having $2n$ blank spaces and choosing n of them to be filled with the letter L while the others are filled with R. Luckily for us there is an entire branch of mathematics called *combinatorics* dedicated to counting things. The number of ways to choose k things out of a set of m things is called a *combination*, denoted $C(m,k)$. These are also called binomial coefficients, and denoted $\binom{m}{k}$ in this context.

This means that

$$\phi(2n) = \frac{C(2n,n)}{2^{2n}}$$

on $\mathbb{Z}$, but we need to deal with that pesky combination. The combination $C(m,k)$ can be computed as

$$C(m,k) = \frac{m!}{k!(m-k)!},$$

where

$$m! = 1 \cdot 2 \cdots (m-1) \cdot m.$$

For our purposes we don't actually want to calculate the combination, and will instead use an approximation:

Fact 22 (Stirling's Approximation)**.**

$$n! \approx \sqrt{2\pi n}\ n^n e^{-n}. \tag{2}$$

Stirling's approximation tells us that

$$\begin{aligned} C(2n,n) &\approx \frac{\sqrt{4\pi n}(2n)^{2n}e^{-2n}}{(\sqrt{2\pi n}(n)^n e^{-n})^2} \\ &= \frac{2^{2n}}{\sqrt{4\pi n}}. \end{aligned}$$

Using this fact shows that

$$\phi(2n) \approx \frac{1}{\sqrt{\pi} n^{1/2}}.$$

Since this is an approximation, we can ignore the constant factor $1/\sqrt{\pi}$, and just say that

$$\phi(2n) \approx \frac{1}{n^{1/2}}.$$

Using Fact 21, we see that the random walk on $\mathbb{Z}$ is reucrrent because $\alpha = 1/2$.

On $\mathbb{Z}^2$ we pick up an extra pair of directions. Let's call the directions N, E, S, and W. For the walker to return home there must be an equal number of steps in the N and S directions and in the E and W directions. This means that n of the $2n$ steps must be in the N and E directions, and the other n must be S or W. This gives $C(2n, n)$ options. Next there are $C(n, i)$ ways to place i Ns, and $C(n, i)$ ways to place the i Ss needed to counteract these Ns. Thus there are $C(n, i)^2$ ways to place these Ns and Ss, and placing them also fixes the Es and Ws. Summing over the possible number of Ns gives

$$C(2n, n) \sum_{i=0}^{n} C(n, i)^2$$

paths which return to the origin after $2n$ steps. While this sum might seem problematic, we are in luck because it is actually a complicated way of writing $C(2n, n)$. There are $4^2 n$ total paths of length $2n$ since we have 4 choices of directions at each step. So on $\mathbb{Z}^2$ the probability of return is

$$\begin{aligned} \phi(2n) &= \frac{C(2n, n)^2}{4^{2n}} \\ &= \left(\frac{C(2n, n)}{2^{2n}} \right)^2 \\ &\approx \frac{1}{n}, \end{aligned}$$

where we have used our approximation of the probability of return for $\mathbb{Z}$ in the last step. Using Fact 21 we can see that $\mathbb{Z}^2$ is recurrent as well. Thus a drunk man will always return home.

Now we turn to our drunken bird. We add the directions U and D to those we had before. In order for the bird to return home in $2n$ steps, exactly

n of these steps most be distributed among the U, N, or E directions. As before there are $C(2n, n)$ ways to do this. Amongst these n steps i must be U, j must be N, and $n - i - j$ must be E. So there are $C(n, i)C(n - i, j)$ ways to distribute these 3 letters. Similarly, there are the same number of ways to distribute D, S, and W.

Since i and j can vary up to the point where their sum is at most n, we have that the number of possible paths of length $2n$ which return home is:

$$C(2n, n) \sum_{i+j \leq n} [C(n, i)C(n - i, j)]^2.$$

Finding a good approximation of this number is laborious, and we won't go into the details. What we ultimately find is that

$$\phi(2n) \approx \frac{1}{n^{3/2}}.$$

Finally the exponent of n in the denominator is larger than 1, and this means that $\mathbb{Z}^3$ is transient. Thus a drunk bird will eventually leave home forever.

You might have already guessed at the pattern to be found in the probabilities of return. Were we to consider $\mathbb{Z}^d$ for an arbitrary d, we would find the following fact:

Fact 23. *On a d dimensional grid the probability of return satisfies*

$$\phi(2n) \approx n^{-d/2}.$$

This means that only $\mathbb{Z}$ and $\mathbb{Z}^2$ are recurrent. Furthermore, it highlights a deep connection between the geometry of these graphs and the behavior of random walks on them. Even on infinite graphs far more complicated than the ones we have considered we would find that the probability of return is closely related to something akin to the dimension of the graph [15]. In fact the relationship between random walks and the geometry of graphs is still a lively area of mathematical research.

We can also think of random walks as occurring in much less orderly cities. Think of a city where many streets lead to a particular intersection; with so many ways to reach this place our drunkard will keep stumbling through it over and over again. Without knowing a thing about this intersection, the drunkard might begin to suspect that it is more important than other intersections. Why else would he keep ending up there? In the next section we'll explore how this intuition relates to searching a particular graph: the internet.

3.3 What Google Does

There are many different ways to search the internet, but we'll focus on the one that made Google famous. The first step in solving this problem is to think about what it means to find something on the internet. For any particular search term there may be thousands of sites containing relevant information, but we'd like to find the most relevant page to our search term. This means we need some way to rank webpages. Google's system of ranking webpages is based on an algorithm called *PageRank*. PageRank was invented by Google co-founders Larry Page and Sergey Brin while they were graduate students at Stanford University [3].

PageRank ranks a site's relevance based on both the number of sites linking to it and the relevance of those sites to the topic at hand. Intuitively, this is a great idea–if you want to know about Copernicus or ninja-puppies it's reasonable that a site linked by many other sites about Copernicus or ninja-puppies would have lots of great information. However, given the scale of the internet this is no easy task, and some mathematical trickery has to be used. We can think of the internet as a *directed* graph. A directed graph is one where the edges have orientations, i.e. they are arrows rather than lines. The vertices of this graph are webpages, and the edges correspond to links from one site to another.

Question 24. *How do we determine the relevance of a site to our search query using a graph of the internet?*

PageRank approaches the problem probabilistically [13]. Start at a random page mentioning Copernicus, and follow a randomly chosen link from that site to other sites mentioning Copernicus. You do the same thing at the new site. This means that you are doing a random walk through the internet. If you do this for a really long time, you'll start to see the same sites over and over again. This happens because the internet is finite, and we'll assume that every site links to at least one other site so you can't get stuck.

Suppose you spend exactly one minute at each site you visit. For any site v we'll let $R(v)$ denote the percentage of the total time you've spent viewing sites that has been spent viewing v. The function R is our way of ranking sites, as the more likely you are to visit a site the more relevant its information must be. The one issue we have with R at the moment is that it changes over time. If you've only clicked on 5 links your ranking is going to look very different than if you'd clicked on 5000 links. From now on let $R_n(v)$ represent the ranking of v after n steps of the random walk,

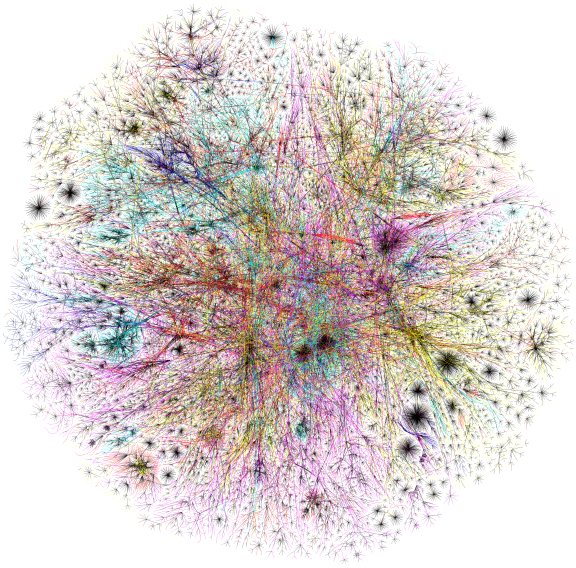


Figure 20: A graph of the internet circa 2005 [14].

or, equivalently, after visiting n pages.

When we start the random walk R_0 is zero for every site except for the site you are currently at, which has R_0 equal to one. We get R_1 by updating R_0 to reflect the links in the graph. To simplify things we'll assume that a site evenly spreads its relevance among the sites it links to. Explicitly, v contributes $R_0(v)/\deg(v)$ relevance to each site it links to, where $\deg(v)$ is the number of links leaving v. This allows us to calculate the updated relevance of a site by summing the contributions of each site that links to it:

$$R_1(v) = \sum_{u \text{ links to } v} \frac{R_0(u)}{\deg(u)}.$$

After taking n steps we calculate that

$$R_{n+1}(v) = \sum_{u \text{ links to } v} \frac{R_n(u)}{\deg(u)}.$$

New sites are created all the time, and we'd like to slip them into the graph without having to restart our computation of relevance. However, our current method for calculating R gives new sites zero relevance as no one is

linking to them. To compensate for this PageRank uses a dampening factor, d, between 0 and 1 to allot some relevance to new sites. Now we calculate relevance as

$$R_{n+1}(v) = \frac{1-d}{N} + d\left(\sum_{u \text{ links to } v} \frac{R_n(u)}{\deg(u)}\right), \tag{3}$$

where N is the total number of sites at time $n+1$.

Doing this computation for each site is still far too time consuming to be useful. What we need is a way to leap ahead and see what R_n would look like for very large n. We can do this because random walks are examples of Markov chains, which are random processes whose next state depends only on the current state. We can also treat R_{n+1} like a Markov chain, since it is defined in terms of R_n. Under very broad conditions the probability that a Markov chain is in a particular state goes to a fixed value for large n. We can find these values using matrices (see Chapter **??** for more on Markov chains and their relation to matrices).

We'll suppose that after running our random walk long enough the relevance of each site becomes fixed. This means we can talk about $R(v)$ again. Next we number all the sites from 1 to N, and put their ranks in a vector

$$\vec{r} = \langle R(1),\ R(2),\ \cdots, R(n)\rangle\,.$$

We make another vector $\vec{d}$ corresponding to the dampening term $(1-d)/N$ in our formula for R_{n+1}. We can then write the update formula (3) in terms of $\vec{r}$ as

$$\vec{r} = \vec{d} + \vec{r}M,$$

where M is an $N \times N$ matrix whose (i,j)th entry, m_{ij}, is defined as

$$m_{ij} = \begin{cases} \frac{1}{deg(i)} & \text{if } i \text{ links to } j \\ 0 & \text{otherwise.} \end{cases}$$

Once we have this matrix equation it is relatively easy to have a computer solve for $\vec{r}$ (actually the internet is so large that Google uses many computers working in tandem to do this sort of calculation). What is particularly nice about this equation is that it depends mostly on the matrix M, which is determined solely by the graph of the internet. One of the best ways to represent a graph in a computer is using the *adjacency matrix*. The adjacency matrix of a graph is defined in exactly the same way as M except

that all the non-zero entries are just one. This means that just having the graph of the internet stored in a computer has already gotten us pretty far along in creating M. This is how Google is able to perform so many searches so quickly.

PageRank has applications outside of searching the internet. One interesting application is to the study of food webs. A food web tracks which species prey on other species in an ecosystem, so it is just a directed graph with species linked to the species that they eat. PageRank can be used to determine how important a given species is to an ecosystem [1], by finding those species which serve directly or indirectly as food sources for many other species. For instance, plankton are eaten by fish, who are eaten by larger fish, and so on. Their removal from the ecosystem would endanger many other species, and even deprive us of much of the fish we consume. In this context, PageRank can help us decide which species are important in an ecosystem, and are thus the best targets for conservation efforts.

4 Conclusion

Hopefully these brief tastes of graph theory have given you a good idea of what the field encompasses and how it can be applied to the world at large. Of course, such a short treatment has barely begun to touch on all that graphs have to offer. In mathematics graphs play important roles in the field of combinatorics, as well as in the abstract study of symmetries, group theory. In fact all of the infinite graphs in Section 3.2 arise from groups. Graph theory is also nice because there are so many applications in both the sciences and business. Hamiltonian cycles and the traveling salesman problem have wide ranging applications. Processes on graphs help us to understand processes that are dependent on the interactions between people, computers or species. Many of these developments are recent, and so there are few books for the non-specialist reader. However, networks have a way of capturing the imagination, and so newspapers and magazines often publish articles on them. For the curious reader many of the books and articles cited in the bibliography make for good destinations on your journey through graph theory.

5 Bibliography

References

[1] Stefano Allesina and Mercedes Pascual. Googling food webs: Can an eigenvector measure species' importance for coextinctions? *PLoS Comput Biol*, 5(9):e1000494, 09 2009.

[2] Norman L. Biggs, E. Keith Lloyd, and Robin J. Wilson. *Graph theory. 1736–1936*. The Clarendon Press Oxford University Press, New York, second edition, 1986.

[3] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, pages 107–117, 1998.

[4] Nicholas A. Christakis and James H. Fowler. The spread of obesity in a large social network over 32 years. *N Engl J Med*, 357(4):370–379, 2007.

[5] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, third edition, 2009.

[6] Richard Durrett. *Probability: theory and examples*. Duxbury Press, Belmont, CA, second edition, 1996.

[7] Martin Gardner. *The colossal book of mathematics*. W. W. Norton & Co. Inc., New York, 2001. Classic puzzles, paradoxes, and problems, Number theory, algebra, geometry, probability, topology, game theory, infinity and other topics of recreational mathematics.

[8] Timothy Gowers, June Barrow-Green, and Imre Leader, editors. *The Princeton companion to mathematics*. Princeton University Press, Princeton, NJ, 2008.

[9] Matt J Keeling and Ken T.D Eames. Networks and epidemic models. *J R Soc Interface*, 2005.

[10] J. Kleinberg. *Cascading Behavior in Networks: Algorithmic and Economic Issues*. Cambridge University Press, 2007.

[11] Stanley Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[12] Mark Newman, Albert-László Barabási, and Duncan J. Watts, editors. *The structure and dynamics of networks*. Princeton Studies in Complexity. Princeton University Press, Princeton, NJ, 2006.

[13] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web, 1999.

[14] Opte Project. Internet map, 2005.

[15] Laurent Saloff-Coste. Probability on groups: random walks and invariant diffusions. *Notices Amer. Math. Soc.*, 48(9):968–977, 2001.

[16] Benno Schwikowski1, Peter Uetz, and Stanley Fields. A network of proteinprotein interactions in yeast, 2000.

[17] Joel Spencer. *Ten lectures on the probabilistic method*, volume 64 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 1994.

[18] Clive Thompson. Are your friends making you fat? *New York Times*, 2009.

[19] Robin J. Wilson. *Introduction to graph theory*. Longman, New York, third edition, 1985.

Index

adjacency matrix, 29

binomial coefficient, 24
bipartite graph, 9
    complete, 9
Borel-Cantelli Lemma, 23

combination, 24
combinatorics, 24, 30
connected component, 4
critical phenomenon, 20
critical threshold, 18

degree, 3
drunkard's walk, *see* random walk

Eulerian cycle, 5

food web, 30
forbidden minor theorem, 10

graph
    bipartite, 9
    coloring, 12, 17
    complete, 10, 12
    connected, 4
    cycle, 5
    directed, 27
    Hamiltonian, 6
    length, 3
    minor, 10
    path, 3
    planar, 9
    weight, 3
group theory, 30

Hamiltonian cycle, 5

Icosian graph, 6

integer lattice, 21

Kolmogorov's 0-1 Law, 22

Markov Chain, 21
Markov chain, 29

PageRank, 27
pay-off function, 17, 18
Petersen graph, 7
polyhedra, 11
probability of return, 23

Ramsey Number, 16
random walk, 20, 27
    recurrent, 21
    transient, 21

six degrees of separation, 3
small-world network, 3
Stirling's Approximation, 24

tail event, 22
traveling salesman problem, 6

universality, 20