Abstract

Real world social networks seem to possess a property of navigability, where short paths through the network can be found with only local information. If a network is given a notion of distance by associating each node with a point on a lattice, under certain conditions it is possible to show that a specific power law distribution maximizes this navigability property. This paper presents a simple algorithm, inspired by everyday use of real social networks, and demonstrates via computer simulation that it produces this optimal distribution and the efficient network path finding associated with it.

Introduction

In the 1960s, psychologist Stanley Milgram conducted a now famous series of experiments which created the first awareness of the "small world phenomenon." His participants were an everyday group of people from Wichita, Kansas. Each subject was presented with a package. They were instructed to forward their packages to those they knew on a first name basis, with the ultimate goal of passing to a woman in Cambridge, Massachusetts. Initially the success rate was quite low, but the packages that reached the target did so quite quickly, typically in under four steps. In later variations of the experiment, the rate of success was pushed as high as 97%, while the number of steps required remained low and seemed to average out about 6.

It is an established fact that on a random graph, short paths between any two nodes exist with high probability. Thus, there is nothing striking about the fact that short paths between individuals in a social network exist; what is striking is that individuals are able to find these short paths. Generally, it is safe to assume that people know their friends, perhaps know some of their friends' friends, and that this is the extent of their knowledge of their social network. Thus, what is remarkable about the participants in Milgram's experiments is that they achieved very efficient routing with only local information and local decision making.

Navigability is this property of a network that allows efficient routing with local information, and how it might arise is not fully understood. In an attempt to understand how the navigability property might arise, it is worth examining the strategies used by participants in the small world experiments. One important factor to consider is the notions of distance (independent of network structure) that participants use in deciding where to forward a package. The most important example of this seems to be geographic distance.   When trying to forward a package to someone in Massachusetts, there is a sense in which someone who is geographically closer to the target is likely to be closer in the social network. This intuition is supported by a study of small world experiment decision making conducted by Killworth and Bernard. Their finding was that the most heavily used choices of a starter (the first person in a chain in a small world experiment) was generally selected on the basis of their location, with occupation coming in a close second. This supports the notion that notions of distance independent of network structure are an important factor in the observed navigability of social networks

One way to endow a social network with the necessary notion of distance is to consider the nodes of the network as points on a lattice. Jon Kleinberg demonstrated that if a network imposed on a lattice, where each node is connected to its immediate neighbors and a fixed number of more distant ones, there exist a family of distributions for which efficient routing becomes possible, and an optimal distribution for which routing time grows radically slower than the lattice size as it increases. Particularly, if any two nodes are joined with probability proportional to the distance between them to the power of the negative of the dimension of the lattice, the expected number of steps required for a message to be routed between two random nodes grows proportionally to the log of the lattice size squared, and grows as a fractional power of n for exponents close to the optimal power.

The intuition for such an optimal exponent is simple. If the exponent is very small, the probability of a link very far away will decay too slowly with distance, and many links will go excessively far away. Thus it is easy to go a long way, but there is not much predictability about how far. On the other hand, a very large power law exponent will mean that long links will be very rare. Thus, links will become predictable but may not take you anywhere. It is sensible that there would be a happy medium in between which would produce optimal routing speed.

One interesting consequence of this particular optimal distribution is that the distribution of long range links over distance is the same regardless of the dimension of the lattice. This is because in d dimensions, the number of targets at length l from any given node is roughly proportional to l^d-1, and each is joined to that node with probability proportional to l^d. Thus, regardless of the dimension, the probability of a nodes link being of length l is proportional to 1/l.

The lattice model is one of several conceptions of a static social network with corresponding optimal distributions that produce good routing times. However, what is less well understood is how these distributions might arise. It is certainly the case that real social networks are being constantly modified by their constituents, and any networks structure that allows navigability must somehow arise from the everyday use of the network. If there existed a simple, decentralized algorithm for network revision that produced an optimal distribution, it would provide some hope that people's everyday activity might be produce networks that are quickly navigable.

This paper presents an algorithm, inspired by the dynamics of web-surfing, which on lattice networks produces Kleinberg's optimal 1/ l link length distribution in one, two, three, and likely higher dimensions. The algorithm focuses on a "web-surfer" who starting from some initial node (which can be visualized as their "homepage") tries to navigate to some distant target node by taking whatever available network edges (think "hyperlinks") allow them the most forward progress in terms of the notion of distance provided by the lattice. If the surfer travels too long without success, it becomes frustrated, stops surfing and "bookmarks" its current location by rewriting the long range link of its source node to its current position.

The Algorithm

The algorithm begins with some initial lattice network with pre-existing long range links. A variety of initial distributions were used, which will be discussed at length later. First a random node is selected on the lattice, with all nodes having equal probability. A desired integer target distance t is selected between one and the maximum possible distance on the lattice[1] according to a given target distance distribution, and an integer frustration threshold s is then selected between 1 and t according to another predetermined distribution. The algorithm then creates a "surfer" which attempts to navigate between the source node and a target selected from the set of nodes having distance t from the source node, with all such nodes having equal probability.

[1] This maximum distrance is (n*d)/2 where n is the width of the lattice and d is its dimension.

This surfer uses a simple greedy algorithm, and only has knowledge of the network structure at its current location. In each step, it follows the long range link out of its current location if that long range link takes it closer to its target. If not, it takes a step of lattice distance 1 to whatever neighboring node is closer to its target. If multiple neighboring nodes are closer, the surfer chooses one randomly with all eligible nodes being equally likely. The surfer repeats this stepping process until it either reaches its target or it has taken a number of steps equal to its frustration threshold. In the former case, nothing is changed, but in the latter the long range link of the surfer's source node is rewritten to its final position.

After every such trip by a "websurfer" the algorithm evaluates whether the network has reached a stable state. As is discussed in the results section, the algorithm produces a power law distribution as a steady state, and thus estimates of the power law exponent provided a natural way to gauge the stability of the network. The maximum likelihood method of parameter estimation was used, and because the normalization of any power law on a finite lattice is dependent on the size of that lattice, the likelihood function of any given set of data was dependent on the lattice size. This unfortunately made a closed form estimator for the power law exponent impossible. The only input of the data into the estimation was the sum of the logs of the link lengths of each node, and thus the stability of this sum was taken as a stand-in for the stability of the estimated exponent and thus of the distribution. Based on observation of the fluctuations of this sum, a convergence criterion was selected which halted rewriting after this sum changed by less than 2% over a period of ten rewriting per node of the lattice. For standards of precision tighter than 2%, on small graphs the fluctuation created by writing was great enough that the algorithm never terminated. Ten rewrites per node was selected as an amount that should engender noticeable change on any lattice not in equilibrium.

Once the criterion for stability had been satisfied, a numerical procedure was used to estimate the power law exponent using the sum of log link lengths. Estimation of power law exponents on lattices in one, two, and three dimensions are discussed in greater detail in Appendix A. The link length distribution was also recorded, along with a variety of other data collected during the rewriting process which is discussed later in this paper.

Results

For the majority of simulations conducted, both the target distance *t* and the frustration threshold *s* were uniformly distributed over their respective ranges. Other distributions were tried, and a wide variety produced similar results. These other distributions and their effects on the output of the algorithm will be discussed at length later; the following discussion is for uniformly distributed targets and frustration thresholds.

One might also expect that the initial distribution of links in the network would have some effect on the stable distribution produced by rewriting, but this did not seem to be the case. A variety of initial distributions produced the same power law link length distribution after rewriting. The steady state link length distributions from a variety of initial distributions are show in Figure 0; note that they are indistinguishable. The following data and discussion is for networks rewritten from a state where each node has a self loop, which can be thought of a having a power law distribution with infinite exponent.

The algorithm succeeded in producing power law distributed link length distributions on one, two and three dimensional lattices. Figure 1 depicts cumulative link length distributions for one, two, and three dimensional lattices of width 20, 200, and 2000 respectively, with power law fits for each. The computing time required to run the algorithm grows very rapidly with lattice width in higher dimension, which effectively prevented running simulations in four or more dimensions. Nevertheless, the consistency of results in dimensions as high as three suggests that the algorithm should produce a power law distribution regardless of the dimension of the lattice forming the substrate for the initial network.

Although results were fundamentally the same in all dimensions, there were other differences in the behavior of the algorithm across dimensions that are worth noting. One phenomenon that appeared in higher dimensions was an increase in the severity of finite size effects. This can be attributed to the fact that all lattices used were square (or a squares higher dimensional equivalent). For a fixed node, a square lattice in higher dimensions has significantly less nodes at distance *l* than an infinite lattice when *l* approaches the maximum possible distance. This can be visualized by thinking about a square, and that if the node in question is at the center of the square, there are only nodes at the max distance are squeezed into the corner of the square and are necessarily less numerous. This phenomenon produces an overabundance of mid range links and then a rapid drop off at extreme distances, which is apparent in two dimensions but begins to become pronounced in three.

Another notable difference between one and higher dimensional results was that the stable link length distributions for the one dimensional case were more "noisy," in that estimates of the power law distribution exhibited more fluctuation even when in their final steady state. This can be attributed to the higher ratio of max distance to number of nodes in the one dimensional case. For example, for a one dimensional lattice of width 10000, the stable link length distribution had 10000 observations of link length spread over the interval [1,5000], where 5000 is the maximum possible length. However, a two dimensional square lattice of 10000 nodes has width of only 100, and spreads 10000 observed link lengths over the interval [1,100]. Thus, the one dimensional link lengths distributions in some sense always contain lesss observations, and are more noisy. This is of some consequence to the interpretation of the algorithms results, as will be addressed later.

It should also be noted that one dimensional lattices of increasing size took an increasing number of rewrites per node to reach their stable state. Conversely, in higher dimensions, the number of required rewrites per node seemed to taper of to a constant value, or perhaps even be in a slow decline with lattice size. This can be attributed to the fact that navigation is much easier in one dimension, and the algorithm's surfer tended to reach its destination with much greater frequency on one dimensional lattices. The frequency of success (in the steady state) also increased on larger lattices. Thus, for every rewrite cycle, the algorithm did much less actual rewriting of links in the one dimensional case, and did even less for bigger lattices. This implies more rewrites per node to engender the same actual change in the network, and the difference in behavior between one and higher dimensional networks.

These differences across dimension aside, the algorithm not only seemed to produce power law link length distributions in all dimensions, but appeared to approaching the optimal exponent power law distribution on large lattices. The estimated power law exponents for the stable distributions were generally close to but slightly less than one, and slowly converged to one as the lattice size increased. Figure 2 shows the estimated power law exponents for a sequence of stable distributions on increasingly large one dimensional lattices, while Figure 3 shows the cumulative link length distributions with power law fits for those same stable distributions . Results were similar in two and three dimensions. Note the slow converge to the optimal exponent of 1, likely due to the finite size effects previously discussed.

This convergence of the stable distributions to the optimal one suggests that mean routing times on those stable distributions should eventually calm down to log^2 growth, and this seems to be the case. Growth of routing times with lattice size for the stable distributions on one and two dimensional lattices are visualized in Figure 4. The graphs clearly suggest that routing times grow at a rate less than log^3, and the flattening of the log^2 quotient in both graphs suggests that routing times on very large networks are indeed growing proportional to log^2. A log-log plot of lattice size against routing provides a useful opportunity to distinguish between poly-logarithmic growth and growth proportional to small powers of n, which is the other possibility predicted by Kleinberg. Such a log-log plot is shown in Figure 5. In the two dimensional case, there is a slight but clearly visible downward curve to the data, which suggests that the data is not growing proportionally to any power of n. This is not visible in the one dimensional case, but given the greater noisiness of the stable distribution in one dimension and other information regarding the routing time growth, it is likely that such a fine feature could exist but be obscured by the greater level of noise present in the one-dimensional data.

Returning to the target and frustration threshold distributions, experimentation with various distributions showed that the results were not totally dependent on the choice of distribution. In general, it seemed that any distribution which was not highly skewed towards small values would produce a power law link length distribution on at least part of the range of possible link lengths, with the period of power law behavior increasing with link size. Such distributions also produced routing times that grew slowly with increasing lattice size, although not all were log^2, and it is reasonable to assume that many were increasing proportionally to small powers of n. Some distributions actually outperformed the original algorithm; in particular, a linearly decreasing density, either as the target density, the frustration threshold density, or both, was a particularly strong performer.

Intuition for Results

The success of the algorithm in producing power law distributions and the unimportance of the particular target, frustration threshold, and initial network distributions all suggest that the dynamic of pursuing a distant target in a finite and variable number of steps is what produces the positive results. In order to develop an analytic intuition for why this occurs, it is helpful to approach the question first in a simplified case. Thus, consider the previously described algorithm on an extremely large, one-dimensional lattice where all edges are directed and pointing in the same direction. Within this framework, there is a simple argument to show that the distribution of the algorithms failures is log-normal with increasing mean and variance that converges to a power law in the limit of an infinite number of rewrites.

To construct this argument, it is important to think of any given step a surfer takes as a random variable with a fixed, but not necessarily known probability distribution. This implies that the distance at which a nodes next long range link will be rewritten to is also a random variable, as the sum of a random number of random variables, where this sum is less than the randomly determined target distance. It is also necessary to assume that all nodes are acted on simultaneously by the algorithm using the structure of the network from the previous rewrite, and that all nodes have their surfers fail in the same phase. If the first assumption is granted, the second is much more reasonable, given that if the algorithm acts on all nodes at once, each succeeds or fails independently of the others, and the distribution of those failures is thus not dependent on the number of other nodes whose surfers fail.

An important concept for working in the above framework is the notion of some distance *d* for which any step a surfer takes will almost surely be shorter than. Here almost surely is taken to mean that for any level of certainty required, there is some finite value *d* for which a walker will take a step of length less than *d* with that level of certainty. For such any such fixed *d* and any fixed target distribution not excessively skewed towards small values, the probability of a surfer failing further than *d* from its target approaches the total probability of failure on lattice sizes sufficiently large.

This means that if a surfer fails to reach its target on a very large lattice, the steps it took before failing were nearly identically distributed. This is because when a surfers target is *d* away, the probability of the surfer taking the long range link at its current location is nearly one, given that by assumption it points towards the target and that it has a vanishingly small probability of overshooting the target. Thus, on a very large lattice, each step taken by a surfer is distributed the same as the long range link at its current position, and these long range links are also by assumption all identically distributed.

If X(t) is a random variable representing the length of any given nodes long range link after t phases of rewriting, the probability of a surfer failing at a given distance from its start point is equal to the probability that the sum of random number of independent X(t) is equal to that distance, and that this distance is less than the randomly determined target threshold. If this logic is carried back to the very first phase of rewriting, the probability of failing at a given distance in the ith phase of rewriting is equal to the probability that the sum of S random variables representing the initial link length distribution equals that distance, where S is the product of i randomly distributed frustration thresholds and this sum is less than a set of constraints derived from the target and frustration thresholds of previous rewriting phases. If i becomes large, the value S approaches a log-normal distribution with mean and variance that increases with the number of phases of rewriting, and converges to a power law distribution in the limit.