Some Results on the SmallWorld Model

André Allavena – Cornell University
andre@cs.cornell.edu

Abstract

Jon Kleinberg published simulations exhibiting an interesting asymmetric behaviour in networks close to being small-world. We prove this behaviour and fully characterise the phenomena.

Newman et al. had a slightly different model, with hierarchies modelling the distance between nodes. Their simulations show that two hierarchies is the optimum. We analyse the concept of hierarchies in the Kleinberg model, better suited to computer networks, and we prove that in fact one hierarchy is the optimum.

1 Introduction and related work

Stanley Milgram run a famous experiment in 1967 [5] suggesting that the human social network exhibits "small-world" properties. Not only were people connected by small chains of acquaintances, but by making only local decisions, they were capable of routing a message from a source to some destination unknown to them. This last property is of high relevance to computer networks, in the context of decentralised search and routing.

Watts and Strogatz [6] proposed the first model trying to capture this phenomena: a grid with random long distance edges. This network had a small diameter. However, Jon Kleinberg [3] proved that local routing does not succeed quickly. In this model the long range links were distributed uniformly over the grid. Having the distribution of the long range links be inversely proportional to a power r of the distance between the end points of the edges was the way to fix it. Jon Kleinberg [3] proved that if and only if $r = 2$ in the plane (or r=dimension of the grid in the general case) would a local routing succeed: the expected time to reach the destination is polylogarithmic in the size of the grid. If $r \neq 2$, the expected time is polynomial.

In Nature [2] Kleinberg published simulations of the effect of that parameter r. They didn't match his lower bounds on the expected time to reach the destination. Here we provide a complete characterisation, having matching lower and upper bounds on the expected time it takes for an algorithm with only local knowledge to reach the destination, and matching his simulations.

Newman et al. [1] have simulations in a different model with hierarchies modelling the distance between nodes, and other parameters trying to match Millgram's experiment. Their simulations show that two hierarchies is the optimum. We analyse the concept of hierarchies in the model of [4], better adapted to computer networks, and prove that in fact one hierarchy is the optimum.

2 Grid Model

We consider the exact same model as in [3]. We have a square grid of size n, wrapped around the edges. Each node has a link to each of its four neighbours and one long range link, distributed according to the distribution $1/d^r$ where $r \geq 0$ is a parameter, and d is the lattice distance. The probability of an edge between u and v is then $\Pr[u \to v] = \frac{d(u,v)^{-r}}{C}$ where $C = \sum_{\text{all } v \neq u} \frac{1}{d(u,v)^r}$

Algorithm for upper bound (greedy) From the current node: follow the long distance link if it brings us closer to the destination, else follow any local link. Iterate until reaching the destination.

Algorithms for lower bound: We consider algorithms which have knowledge of the grid structure, all the local links, the source and the destination. In phase i, the message has been seen at i different locations. The algorithm knows this set of i nodes, their local and long range links. It then selects one of those edges (call the endpoint v), onto which the message is sent. In step $i+1$, the algorithm adds v and its links to its knowledge.

Results Jon Kleinberg showed in [3] that $r = 2$ is the only case when the expected time to reach the destination is polylogarithmic. We prove that this expectation is $\Theta(n^{\frac{r-2}{r-3}})$ when $r < 2$, and $\Theta(n^{r-2)})$ when $2 < r \leq 3$, improving his lower bounds and matching his simulations.

Proofs The distance between almost all pairs of points in $\Theta(n)$. In the following proofs, we evaluate the expected time E it takes to reach any destination from any source. However, for convenience, we will sometime neglect a small number of pairs, without having any consequence on the overall expectation though, due to the small size of the number of the disregarded pairs.

We will make extensive use of the following lemma described p15-16 in Motwani and Raghavan (Randomized Algorithm). $E(T) \leq \int_0^n \frac{\mathrm{dx}}{f(x)}$ with $E(T)$ the expected time a particle takes to reach its destination, x the distance to destination, and $f(x)$ upper bound on the expected length of a step The converse lemma holds as well, the proof follows the exact same steps as described p16.

These lemmas are a variant of the well known formula *time = distance/speed*. Here we use them with f being constant, set to E_{min} or E_{max}, lower, respectively upper bound on the one-step expected decrease of the distance to destination: $E \leq distance/E_{max}$, and $E \geq distance/E_{min}$. There is technical glitch in the latter when getting close to the destination, the lower bound doesn't hold anymore. We get around it by breaking the march to destination in a logarithmic number of phases and summing them up. We also use the principle of differed decision, and generate the long range edges only when we use them, on an "on-demand" basis.

Case $r > 2$ Both bounds are trivial applications of the lemma. Lower bound: we underestimate the distance by assuming we always step in a straight line. Upper bound: a constant fraction of the expected length of a step ($\Theta(n^{3-r})$) goes into decreasing the distance to destination (geometric proof). We get $E = \Theta(\frac{n}{n^{3-r}}) = \Theta(n^{r-2})$.

Case $r < 2$**, Lower bound** This is really the most interesting proof of the four cases. Let $\varepsilon = 2 - r > 0$. The normative constant of the probability distribution is $C = \Theta(n^\varepsilon/\varepsilon)$. Assume we are at a distance d from the destination. The expected distance we travel in the next step is

$$E = \sum_{v \ | \ d(u,v)>d} 1\frac{1}{Cd(u,v)^r} + \sum_{v \ | \ d(u,v)\leq d} d(u,v)\frac{1}{Cd(u,v)^r} \leq 1 + \frac{3\varepsilon}{n^\varepsilon}\sum_{j=1}^{d} j\frac{j}{j^r} \leq 1 + \frac{4\varepsilon}{(1+\varepsilon)}\frac{d^{1+\varepsilon}}{n^\varepsilon}$$

Let $d_0 = n^{\frac{\varepsilon}{1+\varepsilon}}$. For $d \leq 2d_0$, the distance one expects to travel is constant (independent of n). We consider a starting node u at a distance at least $2d_0$ from the destination v, and define $\mathcal{A} = \{\text{node } x \mid d(x,v) \leq d_0\}$ and $\mathcal{B} = \{\text{node } x \mid x \notin \mathcal{A} \wedge d(x,v) \leq 2d_0\}$ There are two possibilities: either getting into $\mathcal{B}$ and walking to the destination, which will take $\Omega(n^{\frac{\varepsilon}{1+\varepsilon}})$ by a direct application of the lemma; or jump directly into $\mathcal{A}$. But having a long range link reaching $\mathcal{A}$ is quite unlikely: we are at least d_0 away from $\mathcal{A}$, so $p \leq d_0^2\frac{d_0^{-r}}{C} = n^{-\frac{\varepsilon}{1+\varepsilon}}$ The expected time it takes to find a node with such a link is $1/p$. Conclusion: $E = \Omega(n^{\frac{\varepsilon}{1+\varepsilon}})$.

Case $r < 2$**, Upper bound** The proof is the same as [3] giving an upper bound of $\mathcal{O}(\log n)$ for $r = 2$. The idea is to consider concentric circles around the destination, and the time it takes to go from one circle into the next one. Adjusting the parameters to get the best upper bound, one then considers circles of radius powers of n^α with $\alpha \gtrsim \frac{\varepsilon}{1+\varepsilon} = \frac{2-r}{3-r}$. And we get $E = \mathcal{O}(n^{\frac{\varepsilon}{1+\varepsilon}} \ln n)$

3 Tree Model and multiple hierarchies

In this model a tree helps generate the distances between nodes: we have a tree representing a hierarchical structure – our nodes are only at leaf position in the tree – and the distance between two nodes is the height of the least common ancestor in the tree. The edges are generated according to this distance: the probability of an edge between u and v is proportional to $d(u,v)^{-\alpha}$. This is the same model as described in [4]. Choosing $\alpha = 1$ is the only value giving a logarithmic time to reach destinations. Similarly to the grid case, but with elementary arithmetic, we get $E(T) = \Theta(n^{|\alpha-1|}$.

In [1] Newman et al. run simulations studying the effect of having multiple hierarchies. The number of links is constant, they get evenly distributed between 1, or 2, or more hierarchies. Their simulations show that having two hierarchies is the optimum with respect to the time it takes to reach the destination. However, there is a built-in bias towards short paths (there is a probability of dropping the message at each relay), and the number of edges is large compared to the number of nodes. In fact, two hierarchies is only better because of these two artefacts. In the general case, one hierarchy is better, as the time scales in $1/k$ where k is the number of hierarchies. We have proofs and simulations backing this claim, and will present here an intuitive version of the proof.

Intuitive approach The hierarchies are represented by uncorrelated b-arry trees. Again, only their leaves correspond to real nodes. A given node corresponds to exactly one leaf in each of the trees. The distance between u and v is the minimum of the distances over the different hierarchies. But instead of considering the edge $e(u \to w)$ and distance as the minimum of the distances in the different trees, one can consider the edge as k different edges each with a single distance (no minimum). Here is how: we are at node u, have an edge to w, and the destination is v. There are k distances between w an v. Instead, split w in k different nodes w_i, each with a single distance to d. Note this doesn't mean adding nodes. Think of creating aliases, and switching from an array (the hierarchy) to another one at each step. If this isn't clear: look at it this way: align the arrays so that the destination is at the same position (address) in all of them. One (real) edge to node w gives you k addresses, one in each array. Look at these addresses in array 1. This corresponds to k edges. This is the core of the proof. Mathematical preciseness would further go against readability.

Now it becomes clear that having more hierarchies is better, since each edge acts like k different edges, right? Well no! Out of those k edges, only one is generated according to the nice $b^{-\alpha d}$ distribution (Take $\alpha = 1$ to have logarithmic time to reach the destination.) The $k-1$ others are generated uniformly at random – since there is no correlation between the hierarchies –thus are not directive enough to be useful in reaching the destination. Even worse, because the current position and the destination in a different tree will almost certainly be very far apart (they are two random positions) we won't change tree after the first few steps, wasting the (real) edges generated acording to the other trees. Assuming an even repartition among the trees, we only use 1/k th of the edges.

References

[1] P. Dodds D. Watts and M. Newman. Identity and search in social networks. *Science*, 296, 2002.

[2] J. Kleinberg. Navigation in a small world. *Nature*, page 406, 2000.

[3] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *32nd ACM Symposium on Theory of Computing*, 2000.

[4] J. Kleinberg. Small-world phenomena and the dynamics of information. In *NIPS, 14*, 2001.

[5] S. Milgram. The small world problem. *Psychology Today*, 1, 1967.

[6] D.J. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393, 1998.