The Role of Homophily and Popularity in Informed Decentralized Search

Florian Geigl and Denis Helic

Knowledge Technologies Institute
Inffeldgasse 13/5. floor, 8010 Graz, Austria
{florian.geigl,dhelic}@tugraz.at
http://kti.tugraz.at/

Abstract. When searching for specific nodes in a network an agent hops from one node to another by traversing network links. If the network is large, the agent typically possesses partial background knowledge or certain intuitions about the network. This background knowledge steers the agent's decisions when selecting the link to traverse next. In previous research two types of background knowledge have been applied to design and evaluate search algorithms: homophily (node similarity) and node popularity (typically represented by the degree of the node). In this paper we present a method for evaluating the relative importance of those two features for an efficient network search. Our method is based on a probabilistic model that represents those two features as a mixture distribution, i.e. as a convex combination of link selection probabilities based on the candidate node popularity and similarity to a given target node in the network. We also demonstrate this method by analyzing four networks, including social as well as information networks. Finally, we analyze strategies for dynamically adapting the mixture distribution during navigation. The goal of our analysis is to shed more light into appropriate configurations of the background knowledge for efficient search in various networks. The preliminary results provide promising insights into the influence of structural features on network search efficiency.

Keywords: Networks, Decentralized Search, Homophily, Search Algorithms

1 Introduction

Nowadays, in many aspects of our daily life we, knowingly or unknowingly, deal with networks. For instance, we participate in large offline and online social networks, we often deal with information networks such as Wikipedia and the Web, or with file sharing peer-to-peer networks. All of these networks are constantly growing and sometimes consist of billions of connected nodes. In many situations we, or some other kind of autonomous agents, have to search for specific nodes in those networks. Apart from being large, these networks also constantly change. For example, friendship connections in a social network are created and removed, or new Web sites are created and connected to old ones. Therefore, search in such large and ever changing networks is a complex and hard task. The task becomes even harder in networks without a global index such as the Google index for the Web. For example, peer-to-peer networks do not rely on the existence of a global index but base their search strategies on the local proximity of a candidate node to a specific target. More recent examples include autonomous swarms of drones building up ad hoc networks. In those systems, information has to be passed from one drone to another, while the network is constantly changing and therefore no central search or routing engine can be built. In such decentralized networks the search performance heavily relies on finding resources within as few hops as possible.

In previous research several decentralized search algorithms have been designed. Among others, these include:

- Kleinberg's algorithm (based on homophily): Informed greedy search works well on small world networks with a clustering exponent of 2. In his experiments the clustering exponent determined the probability of a connection between two nodes as a function of their similarity. The simulation was steered by homophily, i.e. the geographic distance on a network model was used. [4], [5], [6].
- Adamic's algorithm (based on popularity): Power-law graphs can be locally searched by moving to the neighbor node with the highest degree. The costs of the search scale sub-linearly with the graph size. This property makes this algorithm interesting when dealing with large networks [1].
- Jensen's algorithm (based on a fixed combination of homophily and popularity): Using homophily and degree disparity as background information, the authors construct a simple algorithm for decentralized search. In a couple of synthetic and real-world networks they were able to outperform Adamic's as well as Kleinberg's algorithm [9].

All of these algorithms perform remarkably well and are typical heuristic algorithms based on intuitive assumptions on the nature of features needed to efficiently search in networks. In this paper we set out to analyze these assumptions in greater detail, with a final goal of learning more about the influence of those two features (homophily and popularity) on the search performance. Thus, we aim to answer the following research questions:

1. How much information is provided by node homophily and node popularity and how well can this information be used for efficient search in networks?
2. How should we mix node homophily and node popularity to maximize the search performance? Can an adaptive approach to information mixture outperform a static mixture that remains constant throughout navigation.

To answer these questions we simulate a decentralized search approach, informed by a combination of two different local features, which serve as proxies for node popularity and node homophily – degree and cosine similarity to the target node. Additionally, we introduce and analyze adaptive mixing strategies. To evaluate the navigation performance we create a framework, which allows us to examine the impact of different mixtures. Our preliminary results show that in the majority of analyzed networks homophily provides, on average, more important and more useful information for efficient search. However, the optimal results are only achieved in situations where small amount of popularity information is also available to the search agent.

Thus, our work makes the following contributions:

1. *Methodological*: We provide a framework for analyzing various features to inform search in networks. The framework allows to investigate a wide range of feature combinations and to assess their search performance by simulating an agent navigating through a network.
2. *Empirical*: We apply our framework on real networks and learn about the importance of homophily and node popularity as features for informing network search.

Outline: The rest of the paper is organized as follows. We summarize investigations done by other researchers in the field in Section 2. Thereafter we shortly describe the methodology used in the paper in Section 3 followed by the experiments in Section 4 and their results in Section 5. A brief discussion of the results can be found in Section 6. Finally, we wrap up the paper with an outlook for the future work in Section 7.

2 Related Work

The research on decentralized search in networks was initiated by the famous Milgram experiment in the 1960s [10]. In this experiment individuals had to forward a letter to an unknown person – a stockbroker from Boston. The participants were only allowed to send the letter to somebody whom they know by the first name, i.e. to a friend who possibly can reduce the distance or even knows the target person. Remarkably, the average number of hops for the successful letter chains was less then six. Thus, a famous outcome of the experiment was the so-called "Six degrees of separation" phenomenon, which states that people in a large society such as the USA, are connected by friendship chains not longer than six. Later, Leskovec and Horvitz [7] analyzed a messaging network, namely MSN Messenger. Their result were similar as Milgram's – the average length of connection chains between all pairs of the users is only 6.6.

Another interesting result from the Milgram's experiment is the fact, that although they posses only local knowledge of the network, humans are capable of finding those short connection chains even in a large social network. Various researchers put more effort into investigation of this phenomenon and developed so-called decentralized navigation methods.

For example, Kleinberg [4] showed that in small world networks (with small average shortest path and highly connected groups of similar nodes) homophily (node similarity) can be exploited to efficiently find random target nodes. He observed that small world networks having a clustering exponent of 2 are navigable using a homophily feature. In his research the homophily feature was the geographic distance on the lattice of the Watts and Strogatz model as defined in [11].

Later, Adamic [1] showed that networks whose degree distribution follows a power-law distribution can be efficiently navigated with knowledge about node popularity (degrees) only. The algorithm moved in each step to the unvisited neighbor node with the highest degree. In situation where all neighbors where already visited a random neighbor was chosen. Compared to a random walk, which moves in each step to a random node, the algorithm was able to find nodes within just a few hops. Additionally, the average number of hops scales sub-linearly in the number of network nodes.

Simsek and Jensen [9] combined homophily and popularity in their algorithm called expected-value navigation (EVN). For EVN degree and attribute values of neighbors of the current as well as neighbors of the target node were needed. Additionally, they assumed that already visited nodes can be identified as such. In synthetic networks EVN was able to outperform both previously named algorithm.

3 Methodology

In this paper we set out to assess the importance and the role of homophily and popularity for informing decentralized search in networks. We base our approach on simulations of an agent that navigates through a network in search for a given target node. At each simulation step the agent needs to select one node from the list of neighboring candidate nodes. The selection is based on the available information about the candidate nodes and follows different navigation models. In detail, the simulation is based on three elements:

1. The available information, which we represent in the form of probability distributions over two features (homophily and popularity) and a particular mixture distribution of the corresponding probability distributions.
2. A background knowledge model, which defines the mixture distribution and its adaptation during the simulation.
3. A navigation model, which defines the candidate selection strategy.

Next we describe those three elements in more details.

3.1 Features & Mixture

To create probability distribution we use homophily and popularity as basic measures. We assume that the networks are undirected and without multiple links.

Degree: In our paper we use the degree as representation of popularity. Having a network with $\boldsymbol{A}$ as adjacency matrix (with $A_{ij} = 1$ if nodes i and j are connected by a link and $A_{ij} = 0$ otherwise), the degree is defined as follows:

$$k_i = \sum_{j=1}^{n} A_{ij} \quad . \tag{1}$$

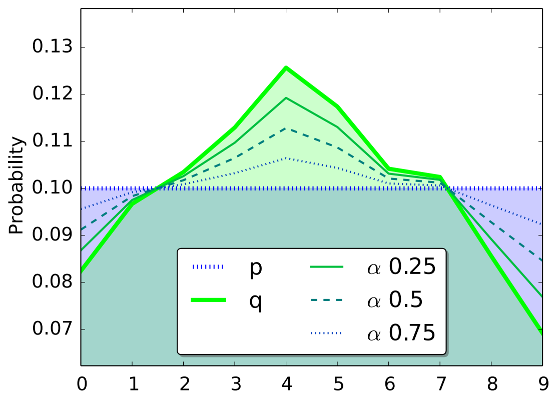


Fig. 1: **Convex combination:** The plot shows two discrete probability distribution p and q. p is uniform distributed, whereas q is non uniform. Varying α (0.25, 0.5, 0.75) one can observe the resulting mixture distributions.

Cosine Similarity: This feature serves as a proxy for homophily. Equation 2 defines the number of common neighbors of nodes i and j. Using this definition we compute the cosine similarity σ_{ij} of two nodes as defined in Equation 3. Cosine similarity results in a non-zero value between two nodes if they share at least one common neighbor.

$$n_{ij} = \sum_k A_{ik} A_{kj} \quad . \tag{2}$$

$$\sigma(i,j) = \frac{n_{ij}}{\sqrt{k_i k_j}} \quad . \tag{3}$$

Mixture Distribution: For the purpose of creating mixtures of these two features we convert them to probability distributions. Therefore, we calculate a probability mass function by dividing each feature value by the sum of all values. Thus, we need only local information for normalization. Precisely for each feature we divide the value of a node by the sum of all values corresponding to the same feature (all neighbors). Hence we get for each feature a probability distribution. For example, the degree normalization we define as following:

$$q_i = \frac{k_i}{\sum_{i=1}^{n} k_i} \quad . \tag{4}$$

Afterwards we apply a convex combination to generate a mixture distribution. Equation 5 defines the steps necessary to calculate the mixtures, where w_l is the weight of a probability distribution l.

$$f_i = \sum_{l=1}^{n} w_l p_i \quad \text{where} \quad \sum_{l=1}^{n} w_l = 1 \quad . \tag{5}$$

We denote the probability distribution of cosine similarity and degree as p and q. Furthermore we set $w_1 = \alpha$ and $w_2 = 1 - \alpha$ since we only have two distributions. This allows us to configure the mixture distribution over α only. Equation 6 defines the mixture distribution of p and q using α.

$$f_i = \alpha p_i + (1 - \alpha) q_i \quad . \tag{6}$$

For instance, Figure 1 visualizes different mixture distributions of p and q. Note that setting α to 0 results in a mixture distribution equal to q. On the opposite if α is set to 1, the resulting mixture distribution equals p.

3.2 Background Knowledge Models

Static Mixture: In our first analysis we use a constant α to create mixture distributions as defined in section 3.1. Recollect α defines the impact of cosine similarity p and degree p onto the mixture distribution. From now on we refer to this model as *static mixture*.

Static Switch: Inspired by human navigation strategies we generate a background knowledge model, which imitate human behavior. Scientists have discovered that humans have two phases when navigating in a network from one random node to another [3], [12].

In the first phase, also known as zoom out phase, humans try to reach a popular and thus highly connected node. As a consequence one characteristic of this phase is that nodes having a higher degree than the current one are chosen. Adamic presented such an algorithm in her work and showed that power-law graphs can be efficiently searched with it [1]. Setting α to 1 for the mixture distribution, allows us to mimic this strategy.

Contrary, the second phase of humans, the zoom in phase, is steered by the similarity of neighbors to the target node. Kleinberg mimicked this behavior and proved that small networks with a clustering exponent of 2 can be searched by doing so [4]. We simulate this behavior using the cosine similarity. By setting α to 0, we generate a mixture distribution equal to p. Thus the agent is only influenced by the similarity.

Recollect that the cosine similarity is always 0 if no neighbor has at least one common neighbor with the target node. Hence p can be a discrete, uniform distribution. This case increases the uncertainty of the mixture distribution if α is greater than 0. One can observe this effect in Figure 1. To tackle this problem, we imitate human behavior again and switch from the zoom out to the zoom in phase at a fixed point. We simulate this by initializing α to 0 and change the value to 1 at a certain point. We name this model *static switch*.

Dynamic Switch: However, the last method lacks of a dynamic transition point, since network size as well as the degree distribution influence the position of the optimal transition point. Thus we initialize the simulation with a certain α and set α to $1 - \alpha$ as soon as p is not uniformly distributed any more. In other words this model switches the weights of p and q in the mixture distribution the moment it reaches a part of the network near to the target node. We call this background knowledge model *dynamic switch*.

3.3 Navigation Models

Greedy Search: The simplest node selection mechanism is a greedy selection. In this strategy the algorithm traverse the network by moving always to the node with the highest probability of all unvisited neighbors. We reference this strategy as *greedy search*.

Stochastic Search: However, the *greedy search* model never selects nodes with a slightly lower probability in the mixture distribution than the maximum. To tackle this we select the next node by drawing randomly from the mixture distribution. We refer to this as *stochastic search*. Consequently this model results in a random walk if the mixture is uniformly distributed. On the other side it selects the same node as *greedy search* if one neighbor has a probability of 1 in the mixture distribution.

Softmax Search: Nevertheless, the randomness of mixture distributions can vary. To either increase or decrease the uncertainty of a distribution we use a softmax function. The softmax function we apply onto our mixture distributions is defined as following:

$$g_i = \frac{e^{\gamma f_i}}{\sum_i e^{\gamma f_i}} \quad . \tag{7}$$

This function increases the randomness of a distribution if γ is smaller than 1, whereas it decreases the uncertainty using values greater than 1. Figure 2 demonstrates the impact of various values for γ. Consequently we have the ability of continuous varying our navigation model. For example, a high γ results in *greedy search*, $\gamma = 1$ in *stochastic search*, and setting γ to zero produces a random walk. We refer to this model as *softmax search* with γ as a parameter.

4 Experiments

4.1 Dataset

For our experiments we use four networks:

Wikipedia for Schools is a subset of Wikipedia articles especially designed for the education of school children. It consists of 4.6 thousand articles (vertices) connected by 119.8 thousand hyper links (edges). The network is directed and belongs to the category of information networks. A power-law distribution provides the best fit for the degree distribution.

Furthermore we use a subset of *Facebook* created out of so called ego-networks where user represents vertices. Additionally, friendship between two users creates an edge in the network between those users. This dataset contains 3.9 thousand users, 88.1 thousand friendships and can be categorized as social network. Its degree distribution follows a log-normal distribution.

As a second social network we test our algorithm on a subset of *Twitter* consisting of 76.2 thousand users and 126.9 thousand edges. Contrary to Facebook, this dataset is directed. This is due the fact that in this social network user A

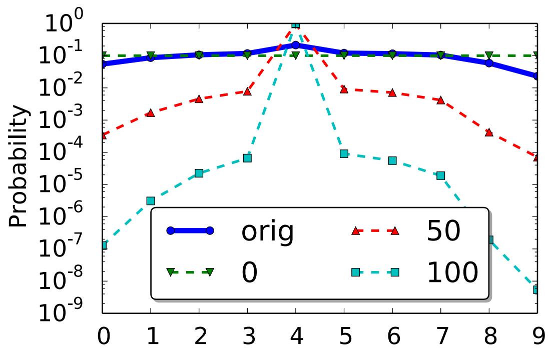


Fig. 2: **Softmax function:** One can see the impact of different values for δ of the softmax function defined in Equation 7. Values smaller than 1 have a flatting effect, whereas values higher than 1 concentrate more probability mass around modes – thus reducing the uncertainty. $\delta = 0$ results in a uniform distribution. Setting $\delta = 1$ does not modify the distribution.

can follow user B without the requirement that B follows A. The best fit for the degree distribution is a log-normal distribution.

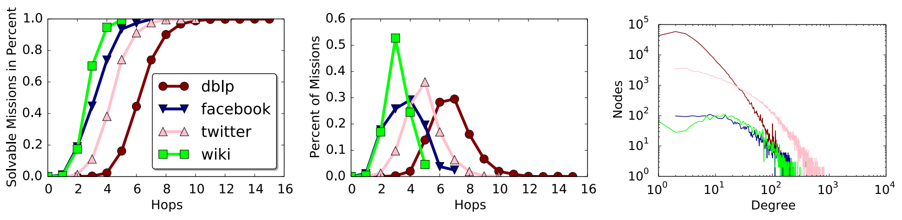
The last network is a co-authorship network, namely *DBLP*, which consists of 317 thousand authors (vertices) and 1 million co-authorship connections (edges). The degree distribution of the network can be fit best with a log-normal distribution.

We calculated the best fit for the degree distribution of these networks with the method proposed by Clauset [2]. The plot containing the degree distributions of all four networks can be found in Figure 3c.

4.2 Experimental Set-up

For each network we generate as many missions as there are nodes in the network. Each mission consists of two randomly chosen nodes reachable from each other. The hop plot and a plot of the shortest path lengths for all missions are shown in Figure 3.

The task of the algorithm is to navigate from one node to the other within as few hops as possible. We consider paths longer than 20 hops or containing revisits of nodes as unsuccessful. Furthermore our framework tries to avoid already visited nodes in each navigation by removing them from the candidate nodes. In our experiments we combine each navigation model with each background knowledge model once. This results in 9 experiments for each network. As evaluation metrics we gauge the success rate and stretch. The success rate is the fraction of successful solved missions, whereas the stretch is the length of the produced paths divided by their corresponding shortest paths lengths. In other words the stretch defines the factor of how much longer produced paths are compared to their shortest path.



(a) **Hop Plot** (b) **Shortest Distances** (c) **Degree Distributions**

Fig. 3: **Properties of Networks & Missions :** Figure 3a shows the hop plots of all missions. Lengths of shortest distances of all missions for each network are plotted in Figure 3b. Notice how long shortest distances of missions in the *DBLP* are compared to *Wiki for Schools*. Figure 3c shows the degree distributions of the networks.

5 Results

Figure 4 shows all results of our experiments. We exclude plots presenting the stretch, since they only emphasize the outcome of the success rate. Rows in Figure 4 are ordered by navigation models, whereas columns refer to the used background knowledge model.

The most interesting result is, that the cosine similarity p seems to be more important for an efficient navigation than the degree information q. Independent of the navigation model and network a fixed value for α near 0.9 seems to be a good choice. As imagined the optimal amount of hops after witch to switch from 0 to 1 for α depends on the network. For instance, the *static switch* model applied onto the *DBLP* - which is a sparse network with a high diameter - benefits from more steps to nodes with higher degrees during the first phase of the navigation. Additionally, for the *DBLP*, this holds for all navigation models. On the other side in *Wikipedia for Schools* the best performance is reached by switching to $\alpha = 0$ before the first hop is made. However, the *dynamic switch* model in combination with a low value for α outperforms all other strategies. This applies to all networks independent of the navigation model. This emphasizes, that in the first steps the uncertainty of mixture distributions with α greater than 0 is increased by the cosine similarity. Additionally, in the zoom in phase the degree information q likely steers in the opposite direction than the cosine similarity q. Consequently the zoom in phase also benefits from the *dynamic switch* model. This is due to the circumstance, that in power-law or log-normal networks a random selected node is likely to have low degree. Thus most missions have a low degree target node.

Anyhow, the difference between *greedy search*, *stochastic search*, and *softmax search* shows how determined the mixture distributions are. Using *softmax search* we can continuously adapted the uncertainty of the mixture distribution. In the last row of Figure 4 the parameter γ of the softmax function is set to 50.

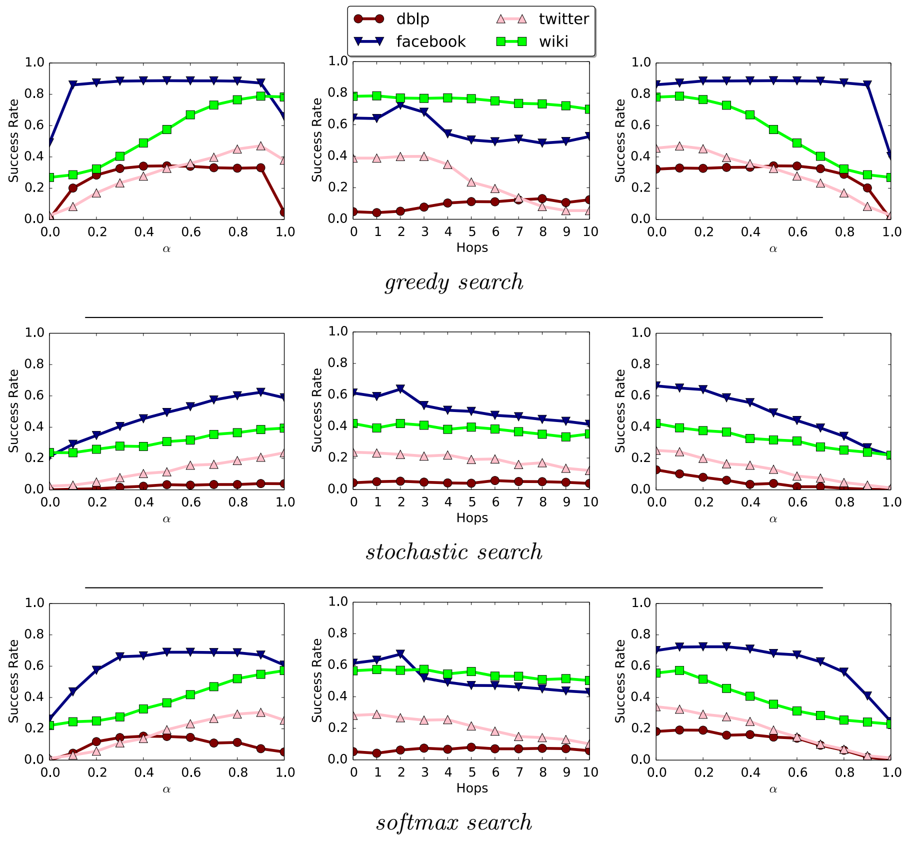


Fig. 4: **Results:** The figure shows the results of our simulations. Each row corresponds to one navigation model. We use the success rate as evaluation metric. The **left** column contains the results produced by the *static mixture* model. In this plot the x-axis defines the α parameter as specified in section 3.1. The **center** column α shows the outcome of the *static switch* model, where the transition point is on the x-axis. In the **right** column results of the *dynamic switch* model can be found. The initial value of α corresponds to the x-axis.

Consequently navigation behavior between *greedy search* and *stochastic search* is obtained.

Additionally to the success rate we gauge the normalized entropy of the mixture distributions. A normalized entropy of 1 is produced by uniform mixture distributions. On the other side, mixture distributions containing a value of 1, induce a normalized entropy of 0. Figure 5 shows the average normalized entropy for value of α between 0 and 1. Moreover on the left side γ is set to 1, whereas on the right a value of 50 is used. Notice the high normalized entropy of the *DBLP* at $\alpha = 1$ on both plots. This may be due the low density of the network

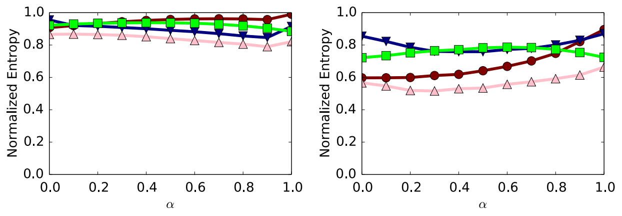


Fig. 5: **Normalized entropy of mixtures:** These plots depict the average normalized entropy of the mixture distributions produced during the simulation for each network. The left plot contains the gauged normalized entropy of unmodified mixture distributions. Contrary, the on the right the softmax function of Equation 7 is applied - using 50 as γ - onto the mixture distributions. Definitely the normalized entropy on the right plot is lower than on the left. This is a consequence of the reduced uncertainty of the mixture distributions through the applied softmax function.

which consequently results in situations where only a small amount of candidate nodes are available. Moreover this nodes are likely to have a high difference in the probability distribution. This is especially the case for cosine similarity.

6 Discussion

In our experiments the cosine similarity plays a more important role (best $\alpha = 0.9$) than the degree information. We believe that this is caused by the properties of the networks we use. All of them are small and have low diameters (3 - 8). Due to these properties the probability of having a neighbor which has a common neighbor with the target node is higher than it would be in larger networks with bigger diameters. Additionally, even if the cosine similarity of all neighbors is zero, the resulting navigation behavior - a random walk - has a high probability of moving to high degree nodes. For example, the famous page-rank algorithm takes advantage of this effect to identify popular nodes in a network [8]. Thus, a uniform mixture distribution favours high degree nodes.

Therefore, we strongly believe that simulations steered by cosine similarity - in power-law or log-normal networks - naturally include a zoom out phase. Nevertheless, additional information of the degree of neighbors may cut down the hops needed for the zoom out phase. Moreover, this intuition is supported by the observation, evident in our experiments where the *dynamic switch* model in combination with a low initial α outperformed all other models.

7 Future Work

One limitation of our research is that the cosine similarity is not a complete local measure of homophily. It includes information about the neighbors of both the current and the target node. Hence we are working on a metric more suitable as a proxy for local homophily. For example, a measure which can only determine if the target is in it's neighborhood or in the neighborhood thereof. This would also allow us to adjust the scope of local knowledge. Furthermore, we plan on applying our framework onto synthetic - especially non power-law/log-normal - networks in future work.

Acknowledgments This research was funded in part by the FWF Austrian Science Fund research projects P24866.

References

1. L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman. Search in power-law networks. *Phys. Rev. E*, 64:046135, Sep 2001.
2. A. Clauset, C. R. Shalizi, and M. E. Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.
3. D. Helic. Analyzing user click paths in a wikipedia navigation game. In *MIPRO, 2012 Proceedings of the 35th International Convention*, pages 374–379. IEEE, 2012.
4. J. Kleinberg. Small-world phenomena and the dynamics of information. *Advances in neural information processing systems*, 1:431–438, 2002.
5. J. M. Kleinberg. Navigation in a small world. *Nature*, 406(6798):845, Aug 2000.
6. J. M. Kleinberg. Small-world phenomena and the dynamics of information. In *Advances in Neural Information Processing Systems (NIPS) 14*, page 2001, Cambridge, MA, USA, 2001. MIT Press.
7. J. Leskovec and E. Horvitz. Planetary-scale views on a large instant-messaging network. In *Proceedings of the 17th international conference on World Wide Web*, pages 915–924. ACM, 2008.
8. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1999.
9. O. z. Simsek and D. Jensen. Decentralized search in networks using homophily and degree disparity. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 304–310, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.
10. J. Travers, S. Milgram, J. Travers, and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32:425–443, 1969.
11. D. J. Watts and S. H. Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.
12. R. West and J. Leskovec. Human wayfinding in information networks. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 619–628, New York, NY, USA, 2012. ACM.