

Chapter 2

Link Prediction Using Thresholding Nodes Based on Their Degree

Abstract In this chapter, we propose MIDT, a degree threshold-based similarity measure, for link prediction which exploits the power law degree distribution which social networks typically follow. We show that, in power law networks, the number of high-degree common neighbors is insignificant compared to the low-degree common neighbors. We use this property to assign a zero weight to the high-degree common neighbors and a higher weight to the low-degree neighbors in computing similarity between nodes. Experiments on standard benchmark datasets show the superiority of MIDT similarity measure. Specifically, MIDT shows an improvement of upto 4 % in terms of AUC when compared to the state-of-the-art link prediction similarity measures.

Keywords Power law degree distribution · Markov inequality · Degree thresholding · Clique-based approach · Area under the curve (AUC)

2.1 Introduction

Most networks can be approximated to follow the power law degree distribution. Accordingly, the probability of encountering a high-degree node is very small. Similarly, the frequency of distinct terms in large-scale collection of documents follows the Zipfian distribution which is a simple form of the power law degree distribution. In Information Retrieval, in the well-known technique called *stopping*, which is used for indexing document collection, the terms with high-frequency are often removed or not considered for indexing purpose as the high-frequency terms often lack good discriminating capability.

Analogously, it is possible that *high-degree nodes lack good discriminating capability; each high-degree node is connected to many other nodes*. Hence, we will need a threshold value on the degree in order to classify a node into low-degree and high-degree clusters. We use an appropriate threshold to split the set of nodes based on their degree into low-degree and high-degree node sets. Once, we classify the nodes into

Material in this chapter appeared in [6].

low-degree and high-degree node sets, then, we give different weights to common nodes in different clusters while computing the similarity between a pair of unconnected nodes. We emphasize the role of low-degree common nodes in terms of their contribution to the similarity and ignore the contribution of high-degree common nodes. We formally show that the number of high-degree common nodes between any pair of nodes is very small. Hence, we justify the proposed scheme formally which de-emphasizes the role of high-degree common nodes in computing similarity between two nodes. The modified similarity measure shows an improved performance on the benchmark datasets.

2.2 Markov Inequality for Determining Threshold

In general, given G_t , it is nontrivial to obtain a threshold for classifying nodes into low-degree and high-degree clusters. We use Markov inequality to derive a suitable value for this threshold. Markov inequality can be stated as follows: if X is a nonnegative random variable and there exists some positive constant $r > 0$,

$$P(X \geq r) \leq \frac{E[X]}{r}$$

We use the above inequality to obtain a threshold value (T) for dividing the set of nodes into low-degree and high-degree clusters. In this case, we take degree as the non-negative random variable (degree of a node is always nonnegative) and T will be positive. We can rewrite for the required inequality as:

$$P(\text{degree} \geq T) \leq \frac{E[\text{degree}]}{T} \Rightarrow T \leq \frac{E[\text{degree}]}{P(\text{degree} \geq T)}. \quad (2.1)$$

Thus, from Eq. 2.1 we can calculate the required threshold based on the number of high-degree nodes that we can ignore. We use Eq. 2.1 to bound the threshold value. For conducting experiments, we set $P(\text{degree} \geq T)$ to 0.1. Using the calculated threshold T , we divide the node set V into low-degree and high-degree node sets. We justify our de-emphasis of the high-degree common neighbors using the theorem below.

Notation

- x_y —degree of node y
- p_k —probability of existence of a k degree node in the graph
- n —number of nodes in the graph
- n_L —number of low-degree nodes, n_H —Number of high-degree nodes
- L —low-degree cluster $\{y|x_y < T\}$, H —High-degree cluster $\{y|x_y \geq T\}$
- L_{avg} —average degree of a node in L , H_{avg} —Average degree of a node in H

- K_L —expected number of low-degree common neighbors between any pair of nodes
- K_H —expected number of high-degree common neighbors between any pair of nodes
- T —degree Threshold, max —Maximum degree of a node in the network

Theorem 2.1 *For any pair of nodes a and b , the expected number of high-degree common neighbors (K_H) is very small when compared to expected number of low-degree common neighbors (K_L), i.e., $K_H \ll K_L$.*

Proof Consider the possibility that both $a, b \in L$. The probability that a common neighbor $z \in L$ is given by

$$P(z \in L) = \frac{n_L}{n} \times p_{L_{avg}} \times \frac{n_L}{n} \times p_{L_{avg}} \times \frac{n_L}{n} \times p_{L_{avg}} \times \frac{n_L}{n} \times p_{L_{avg}} \quad (2.2)$$

where $\frac{n_L}{n}$ accounts for the selection of a node from L and $p_{L_{avg}}$ accounts for the average probability of existence of a low-degree node. Note that the first four terms correspond to the probability of existence of a link between a and z and the last four terms correspond to the probability of existence of a link between b and z .

The above equation can be simplified to the following form:

$$P(z \in L) = \frac{n_L}{n} \times p_{L_{avg}} \times \frac{n_L}{n} \times p_{L_{avg}} \times \left(\frac{n_L}{n} \times p_{L_{avg}} \right)^2 \quad (2.3)$$

In a similar way the probability that a common neighbor $z \in H$ is given by

$$P(z \in H) = \frac{n_L}{n} \times p_{L_{avg}} \times \frac{n_L}{n} \times p_{L_{avg}} \times \left(\frac{n_H}{n} \times p_{H_{avg}} \right)^2 \quad (2.4)$$

So,

$$K_L = n \times P(z \in L) = n \times \frac{n_L}{n} \times p_{L_{avg}} \times \frac{n_L}{n} \times p_{L_{avg}} \times \left(\frac{n_L}{n} \times p_{L_{avg}} \right)^2 \quad (2.5)$$

and

$$K_H = n \times P(z \in H) = n \times \frac{n_L}{n} \times p_{L_{avg}} \times \frac{n_L}{n} \times p_{L_{avg}} \times \left(\frac{n_H}{n} \times p_{H_{avg}} \right)^2 \quad (2.6)$$

Thus, the ratio of K_H to K_L , from Eqs. 2.5 and 2.6 is given by

$$\frac{K_H}{K_L} = \left(\frac{n_H}{n_L} \right)^2 \times \left(\frac{p_{H_{avg}}}{p_{L_{avg}}} \right)^2 \quad (2.7)$$

a and b can be assigned to L and H in 3 other ways as follows:

1. $a \in L$ and $b \in H$
2. $a \in H$ and $b \in L$
3. $a \in H$ and $b \in H$

Note that in all these 3 cases also, the value of $\frac{K_H}{K_L}$ is the same as the one given in Eq. 2.7. Using power law degree distribution property of the network,

$$p_{L_{avg}} = C \times (L_{avg})^{-\alpha} \quad (2.8)$$

$$\text{and, } p_{H_{avg}} = C \times (H_{avg})^{-\alpha}$$

From Eqs. 2.7 and 2.8, we have

$$\frac{K_H}{K_L} = \left(\frac{n_H}{n_L} \right)^2 \times \left(\frac{L_{avg}}{H_{avg}} \right)^{2\alpha} \quad (2.9)$$

Consider $\frac{n_H}{n_L}$ which can be simplified as,

$$\frac{n_H}{n_L} = \frac{n \times \sum_{i=T}^{\max} p_i}{n \times \sum_{i=1}^{T-1} p_i} \quad (2.10)$$

Note that

$$2^{-\alpha} \leq \sum_{i=1}^T i^{-\alpha} \leq (T-1) \times 2^{-\alpha} \quad (2.11)$$

Using power law we can substitute $i^{-\alpha}$ for p_i and cancelling out n , we get

$$\frac{n_H}{n_L} = \frac{\sum_{i=T}^{\max} i^{-\alpha}}{\sum_{i=1}^{T-1} i^{-\alpha}} \leq \frac{(\max - T) \times T^{-\alpha}}{2^{-\alpha}} = (\max - T) \times \left(\frac{2}{T} \right)^{\alpha} \quad (2.12)$$

Also by noting that $L_{avg} < T$ and $H_{avg} > T$ we can bound $\frac{L_{avg}}{H_{avg}}$ as follows,

$$\frac{L_{avg}}{H_{avg}} = \frac{T - \delta}{T + \delta} \quad \text{for some } 1 < \delta < T \quad (2.13)$$

$$\frac{L_{avg}}{H_{avg}} = \left(1 - \frac{2\delta}{T + \delta} \right) < e^{-\left(\frac{2\delta}{T + \delta} \right)} \quad (2.14)$$

So from 2.9, 2.12 and 2.14, we get

$$\frac{K_H}{K_L} \leq (\max - T)^2 \times \left(\frac{2}{T} \right)^{2\alpha} \times e^{-\left(\frac{2\delta}{T + \delta} \right)^{2\alpha}} \quad (2.15)$$

which is a very small quantity and it tends to 0 as T tends to a large value which happens when the graph is large. It is intuitively clear that $L_{avg} < H_{avg}$. Further, because of the power law and selection of an appropriate threshold value we can make $\frac{n_H}{n_L}$ as small as possible. For example, by selecting the value of T to be less than or equal to $\frac{max}{2}$ we get $\frac{n_H}{n_L}$ to range between 0.003 and 0.04 for several benchmark datasets and for the same threshold the value of $\frac{L_{avg}}{H_{avg}}$ ranges from 0.07 to 0.14. So, the value of $\frac{K_H}{K_L}$ ranges from $0.000009 * (0.0049)^\alpha$ to $0.019 * (0.0016)^\alpha$. Typically, the value of α lies between 2 and 3. So, $\frac{K_H}{K_L}$ can be very small. Thus,

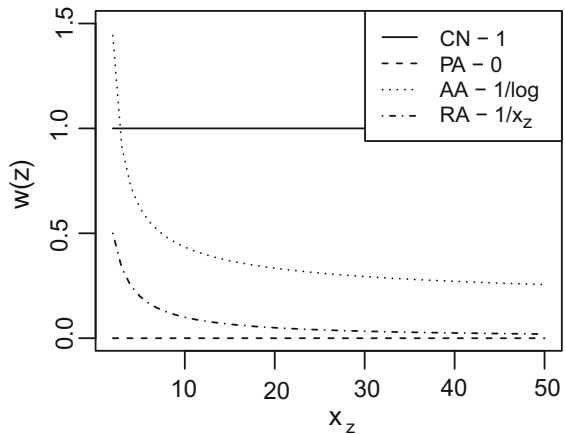
$$K_H \ll K_L \quad \square$$

2.3 Markov Inequality-Based Degree Thresholding Similarity Measure (MIDT)

We present the proposed similarity measure (MIDT) which is based on degree thresholding of nodes. The proposed similarity measure assigns a higher weight to the low-degree common neighbors and a zero weight to the high-degree common neighbors. Specifically, this idea is similar to *stopping*, which is used for indexing document collection; the terms with high-frequency are often removed or not considered for indexing purpose as the high-frequency terms often lack good discriminating capability. Let us represent the weight assigned to a common node z by $w(z)$. As discussed before, we can observe that PA, CN, AA, and RA maintain weight $w(z)$ of 0, 1, $\frac{1}{\log(x_z)}$ and $\frac{1}{x_z}$, respectively. We show the variation of weight assigned to a common node z using various similarity measures in Fig. 2.1.

Observe from Fig. 2.1 that the weight assigned to a common node z by AA and RA fall very steeply with degree and become negligible for high-degree common

Fig. 2.1 Weight of a common node z using various similarity measures



nodes. Now we present **MIDT** similarity measure. Let $N(a)$ and $N(b)$ represent the set of neighbors of nodes a and b , respectively. Further, let $R = N(a) \cap N(b)$.

$$MIDT(a, b) = \sum_{z \in R \wedge x_z < T} \frac{1}{\sqrt{x_z}} \quad (2.16)$$

Next, we present the outline of the link prediction algorithm using MIDT similarity measure.

Algorithm 2: Outline of Link Prediction Algorithm using MIDT

input : Input Graph $G_t = (V, E_t)$
 lp is the number of likely edges we want to predict
output: Predicted Edges Based on Similarity

```

1 for  $(a, b) \notin E_t$  do
2
3 end
4 Sort the node pairs in descending order based on the computed MIDT score.
5 Output the the top  $lp$  links.
```

$$MIDT(a, b) = \sum_{z \in R \wedge x_z < T} \frac{1}{\sqrt{x_z}}$$

In general, we can write the similarity score function as a combination of two monotonically nonincreasing functions *low* and *high*, where *low* is applied on common neighbors having degree less than threshold and *high* is applied on common neighbors having degree greater than the threshold.

$$score(a, b) = \sum_{z \in R} (low(z) + high(z)) \quad (2.17)$$

where z is the common neighbor of a and b . MIDT uses $\frac{1}{\sqrt{x_z}}$ and 0 for functions *low* and *high* respectively. It is clear from our approach that we assign zero weight to high-degree common nodes.

2.4 Experimental Setup

For conducting our experiments, we used the datasets shown in Table 1.1. The datasets do not contain time stamps representing the time at which links are formed in the network. For evaluating the link prediction similarity measures on such graph datasets, we use the experimental setup as in [3]. We conduct experiments on sampled training and test graphs generated in the following ways:

1. Perform edge sampling to divide the dataset into two parts each having 50 % links. We use one part as the training graph (G_t) and the other part as the test graph ($G_{t'}$). We predict the edges of $G_{t'}$. Let us call this **50–50 edge sampling**. 50–50 edge sampling indicates that training graph G_t has 50 % links and test graph $G_{t'}$ has the remaining 50 % links. This simulates link prediction on dense networks.
2. Perform edge sampling to divide the dataset into two parts each having 80 and 20 % links. We use the larger part as the training graph (G_t) and the smaller part as the test graph ($G_{t'}$). We predict the edges of $G_{t'}$. Let us call this **80–20 edge sampling**. 80–20 edge sampling indicates that training graph G_t has 80 % links and test graph $G_{t'}$ has the remaining 20 % links. This simulates link prediction on dense networks.

We use randomly sampled G_t as the graph at the current time instance and predict the links of the test graph ($G_{t'}$) to validate the predictions. We repeat this process 10 times to reduce any statistical bias introduced due to sampling.

For evaluating the performance of the link prediction similarity measures, we present the results in terms of the Area Under Curve (AUC) metric. While accuracy, precision, recall, and top- k equivalents are commonly used in link prediction literature, they are unstable due to class imbalance that arises in the link prediction problem [2]. We use the robust AUC metric which is stable and measures the area under the ROC curve for evaluating link predictor performance.

AUC, in the context of link prediction, can be computed as explained. Consider n random experiments of picking a correctly classified edge and a misclassified edge, if n_1 is the number of times the correctly classified edge has a higher score than the misclassified edge and n_2 is the number of times both have the same score. Then, AUC score can be computed as:

$$\text{AUC} = \frac{n_1 + 0.5 \times n_2}{n}$$

For our experiment we choose n to be 10000. AUC score indicates the ranking of the edges predicted based on the link prediction similarity measure. Higher AUC value implies that the similarity measure is a better ranking algorithm and yields better recommendations. We discuss the results in the next section.

2.5 Results

On performing the experiments using the MIDT similarity measures we report the AUC results in Tables 2.1 and 2.2 on various datasets on predicting 50 and 20 % missing links respectively.

From Tables 2.1 and 2.2, we observe that MIDT performs better than PA, CN, AA, and RA. We show the best results on each dataset when performing 50–50 and 80–20 edge sampling in boldface. Note that G_t is more dense in the case of 80–20

Table 2.1 AUC results for 50–50 edge sampling

Dataset	PA	CN	AA	RA	MIDT
Amazon	11.49	29.91	69.81	68.47	71.37
CondMat	35.59	65.15	73.53	68.35	73.82
HepTh	49.67	59.46	68.12	66.21	68.82

Table 2.2 AUC results for 80–20 edge sampling

Dataset	PA	CN	AA	RA	MIDT
Amazon	8.8	29.73	45.97	43.64	48.52
CondMat	55.01	64.42	79.34	81.22	82.78
HepTh	43.34	57.06	74.29	72.64	78.07

edge sampling when compared to 50–50 edge sampling. Hence, we can note that the link prediction similarity measures show a better performance of AUC in the case of 80–20 edge sampling when compared to 50–50 edge sampling.

Further, note that in general the AUC results of various link prediction similarity measures show better performance on the CondMat dataset when compared to the Amazon and HepTh dataset. The performance of link prediction similarity measure deteriorates with sparsity of training graph (G_t) [1]; CC is a good indicator of the density of a graph.

We can observe that MIDT performs the best when compared to the existing similarity measures in both the cases of 50–50 and 80–20 edge sampling. Further, MIDT performs better in the case of 80–20 edge sampling when compared to 50–50 edge sampling. In other words, MIDT shows better performance when the training graph is dense. Note that the AUC performance has increased by up to 4 %.

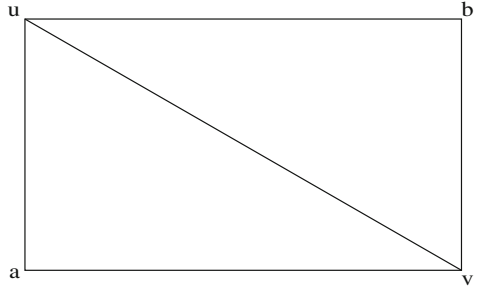
2.6 Clique-Based Approach to Link Prediction

Social networks follow the power law degree distribution [4]. We can exploit the power law degree distribution by ignoring the contribution of edges between a pair of high-degree nodes.¹ We consider the role of common edges between nodes where one of them is a low-degree node.

A clique in a graph G is any completely connected subgraph of G . In our method, we use cliques of size 3 to find out the common edges between two unconnected nodes. Figure 2.2 shows a common edge (u, v) shared by nodes a and b . By common edge (u, v) between two nodes a and b , we refer to the edge that is common to both the cliques auv and buv of size 3. Further, notice that there can be at most one edge common to two cliques of size 3. Note that we are considering cliques of size 3 to find the common edges and we ignore large size cliques as they are infrequent according to the power law distribution of cliques.

¹Material in this chapter appeared in [5].

Fig. 2.2 Common edge shared by nodes a and b



In the proposed algorithm, we make use of cliques for link prediction. The proposed similarity measures compute the similarity between two nodes by computing the similarity using the common node information and common edge information which we propose by making use of cliques of size 3. Specifically, for common edges shared we give higher weight to the edge if the endpoints of the common edge are low-degree nodes. Note that to define the threshold T , we use the same approach as explained before using Markov Inequality. If the endpoints of the common edge are high-degree nodes then the similarity contribution from the common edge is 0. Now, we present the proposed similarity measures which take into account the common edge information to the existing similarity measures. Let us call them **CN Using Cliques (CNC)**, **AA Using Cliques (AAC)**, and **RA Using Cliques (RAC)** which represent the extension for the existing similarity measures CN, AA, and RA, respectively.

$$CNC(a, b) = CN(a, b) + CE(a, b), \text{ where}$$

$$CE(a, b) = \begin{cases} 0 & \text{if } x_u > T \text{ and } x_v > T \\ \sum_{(u,v)} 2 & \text{else} \end{cases}$$

$$AAC(a, b) = AA(a, b) + AE(a, b), \text{ where}$$

$$AE(a, b) = \begin{cases} 0 & \text{if } x_u > T \text{ and } x_v > T \\ \sum_{(u,v)} 2/(\log(x_u) + \log(x_v)) & \text{else} \end{cases}$$

$$RAC(x, y) = RA(x, y) + RE(x, y), \text{ where}$$

$$RE(x, y) = \begin{cases} 0 & \text{if } degree(x) > T \text{ and } x_v > T \\ \sum_{(u,v)} 2/(x_u + x_v) & \text{else} \end{cases}$$

Here, (u, v) refers to the common edge shared by a cliques auv and buv . From our approach it is clear that we are ignoring the contribution of common edges between two high-degree nodes (in AE, RE, and CE) and giving more importance to common edges between two low-degree nodes (in AE and RE). Here, we choose two as a factor in the expression for CE, AE and RE as each common edge has two end vertices u and v whose contribution are weighed in a manner similar to that of CN, AA and RA respectively. In other words, as there can be one-degree nodes present in G_t the maximum value of CE, AE or RE will be 1. This avoids the domination of CE, AE or RE when compared to CN, AA and RA. In CE, we give equal score to all the edges whereas in AE and RE we give more importance to the edge if one of the end vertices u and v is of low degree which varies as $2/(x_u + x_v)$. The CE, AE, and RE score is zero if the end vertices u and v both have degrees greater than the threshold T .

The results can be found in [5]. From the results, we can conclude that clique-based similarity measures perform better than the original similarity measures in terms of AUC. Also common edges between high-degree nodes are not so useful in predicting new links. Thus, we completely ignore the contributions of common edges between high-degree nodes.

2.7 Summary

In this chapter, we proposed the MIDT similarity measure which is based on thresholding nodes based on their degree. Experiments show that MIDT performs better than PA, CN, AA, and RA by upto 4 % in terms of AUC. Further, it can decrease time when the contribution of high-degree neighbors is ignored. The MIDT framework shows that low-degree common neighbors matter the most for link prediction and the high-degree neighbors can be ignored in predicting new links. We can completely ignore or minimize the contributions of high-degree nodes by making use of a suitable nonlinear similarity measure to weigh their contributions accordingly. In the future, we would like to consider other tight bounds to learn the threshold value (T); Markov Inequality gives a loose upper bound. Further, we can concentrate on constructing suitable nonlinear similarity measures. In the next chapter, we present a generic form of the existing similarity measures like CN, AA, and RA which uses the degree distribution of the network explicitly.

We can conclude that edge-based similarity measures perform better than the existing similarity measures in terms of AUC. Also common edges between high-degree nodes are not so useful in predicting new links. Thus, we completely ignore the contributions of common edges between high-degree nodes. We would like to design better similarity schemes that exploit the power law distribution of clique sizes; we would like to consider the contributions of bigger size (more than size 3) cliques.

References

1. Feng, X., Zhao, J., Xu, K.: Link prediction in complex networks: a clustering perspective. *Eur. Phys. J. B* **85**(1), 3 (2012)
2. Lichtenwalter, R., Chawla, N.V.: Link prediction: fair and effective evaluation. In: Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012, pp. 376–383 (2012). doi:[10.1109/ASONAM.2012.68](https://doi.org/10.1109/ASONAM.2012.68)
3. Lü, L., Zhou, T.: Link prediction in complex networks: a survey. *Phys. A* **390**(6), 1150–1170 (2011)
4. Newman, M.E.J.: *Networks: An Introduction*. Oxford University Press Inc, New York (2010)
5. Virinchi, S., Mitra, P.: Link prediction using power law clique distribution and common edges distribution. In: Proceedings of the Pattern Recognition and Machine Intelligence—5th International Conference, PReMI 2013, pp. 739–744. Kolkata, India, 10–14 December 2013
6. Virinchi, S., Mitra, P.: Similarity measures for link prediction using power law degree distribution. In: Proceedings of the Neural Information Processing—20th International Conference, ICONIP 2013, Part II, pp. 257–264. Daegu, Korea, 3–7 November 2013

Link Prediction in Social Networks

Role of Power Law Distribution

Srinivas, V.; Mitra, P.

2016, IX, 67 p. 5 illus. in color., Softcover

ISBN: 978-3-319-28921-2