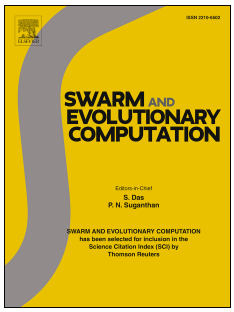
Accepted Manuscript



A gravitation-based link prediction approach in social networks

Esmaeil Bastami, Aminollah Mahabadi, Elias Taghizadeh

PII: S2210-6502(17)30470-4

DOI: 10.1016/j.swevo.2018.03.001

Reference: SWEVO 376

To appear in: *Swarm and Evolutionary Computation BASE DATA*

Received Date: 14 June 2017

Revised Date: 23 November 2017

Accepted Date: 2 March 2018

Please cite this article as: E. Bastami, A. Mahabadi, E. Taghizadeh, A gravitation-based link prediction approach in social networks, *Swarm and Evolutionary Computation BASE DATA* (2018), doi: 10.1016/j.swevo.2018.03.001.

A Gravitation-based Link Prediction Approach in Social Networks

Esmaeil Bastami[a], Aminollah Mahabadi[a,*], Elias Taghizadeh[a]

[a] *Computer Engineering Department, Shahed University, Tehran, Iran*

Abstract

Performance improvement of similarity based link prediction is an important task in social network analysis as an active research. The local, global and community information integrating can increase the prediction accuracy and the time consuming. We present a novel unsupervised gravitation-based link prediction approach to accuracy improvement of local and global predictions by integrating node features, community information and graph properties to distribute and reduce of the prediction space. The local prediction accuracy improves by proposed gravitation-based optimized subgraphs extraction from a community detection method and reduces the global prediction by search space distribution to network capacity increasing. The approach is demonstrated to be more accurate, adaptive and scalable than some existing similarity index-based methods with a significant reduction of the running time through experimental results. The accuracy improves with more existence powerful communities, triangle links and low diameter. As a trade-off between accuracy and execution time, the approach may also be applicable to large and complex networks.

Keywords: Link Prediction (LP), Social Networks (SN), Unsupervised Approach, Gravitation Search Algorithm (GSA), Community Detection.

1. Introduction

The rapid development of Social Networking revolutionary, changes our daily lives and global business, which has been addressed in recent research. A social network is a group of entities, as nodes, where do states cooperate and compete with each other as links. These entities can be cast as information, social connection and location [1]. Information entities, in online social networks, can be represented as nodes and links [2]. These links can be defined relationships among them [3]. Social links can be presented as social connections among users [1, 4]. Location links between users are indicated as location check-ins [4]. For example, Amazon and Google recommend potential goods to customers that predict the accurate link between customers and products based on customers interests. However, in some cases, all links of social networks are hidden. These cannot be observable due to privacy protection of communities or persons [5, 6] or mistakes hidden in crawling, storage or data transmission [7]. In other cases, many links that are nonexistent in the network can appear in the future [8, 9]. Therefore, predicting the missing links or potential links that will exist in the future may be an important and interesting problem in social networks that can discover important events for decision making [10]. This causes some problems to future networks in precision, accuracy and search time of link prediction (LP) to future networks in supporting new methods or algorithms [11].

1.1. Link Prediction

According to the heterogeneity of social networks, the LP problem has its own complexity [8]. Meanwhile, LP problems can be solved by a single link type or multiple link types prediction (e.g. social LP or co-author LP), simultaneously [1, 8, 12]. LP approaches are divided into centralized (as supervised), decentralized (as unsupervised agent-based) and distributed (unsupervised agent-based with negotiation protocol) categories based on implementation architecture. These approaches are based on unsupervised predictors [8], random walk [13, 14, 15], matrix factorization [16, 17, 18], supervised predictors [19, 20] and collective framework [4, 21]. Recently research has been shown increased interest in the search result of unsupervised link prediction that can be helpful in big data growth of Social Networks for large and complex networks [22, 23, 24]. The link prediction of the social networks problem is formally defined to predict unobservable links, missing links or links that will be formed in the future are

*Corresponding author. Tel.: +982151212078 Fax: +982151212020.
Email address: mahabadi@shahed.ac.ir (Aminollah Mahabadi)

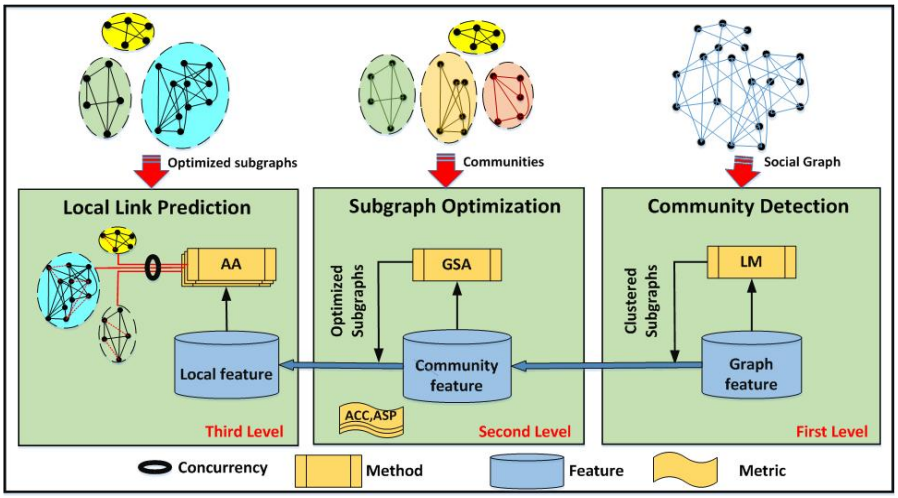


Figure 1: A practical overview of GLP/CGLP approach.

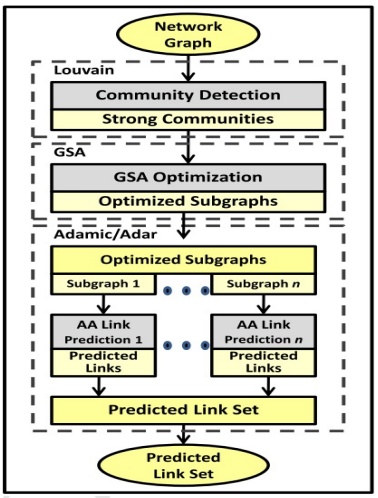


Figure 2: GLP/CGLP approach tasks.

based on a snapshot of the networks [7, 25]. A prediction can be made for predicting a new link to be continuously created or that has disappeared over time [12]. The LP concept has many applications in real-world social networks and many social services [26]. For example, the problem of predicting social links among users can be solved as a LP problem that recommends a user and his location [6]. Performance improvement of a similarity based link prediction is an important task in social network analysis as an active research for the network capacity increasing that the local, global and community information integrating can optimistically increase the prediction accuracy and pessimistically the time consuming. Due to the complexity and diversity of Social Networking, there are six critical sections to reduce the complexity of decision making for the speed accurate link prediction of large-scale graphs:

- *Sampling*: reducing the computational complexity of network graph by way of scale-down sampling of rich subgraphs to preserve graph properties for large-scale social networks;
- *Concurrency*: decreasing seriality prediction operations and increasing the parallelism by concurrent sampling to speed up the predictions and improve the scalability and time consumption in one or more levels;
- *Approach*: supervised or unsupervised modeling link prediction as centralized (discrete objects), decentralized (discrete agents), and distributed (correlated agents) to cope with the scalability through the prediction of space distribution and reduction;
- *Algorithm*: utilizing the combined local, quasi-local, community and global features to score/rank the potential edges as future links for improving the accuracy of the predictions;
- *Features*: adding information from nodes, edges and other graph attributes and increasing the accuracy of the prediction computation or complicated decision making needs preprocessing to obtain a smaller feature set to the dimension and size reduction; and
- *Metrics*: measuring the utility of the approach towards a set of applications.

In order to reduce the graph size and preserve graph properties at the same time many complex approaches are possible, while being very rigorous, can be extremely costly. Even when polynomial approximation algorithms are used, they are still too complex. However, the methods presented in most of the studies only improve the result of the prediction significantly for the network used in the study and is not reliable enough to cope with the huge search apace problem and difficulty to accurate prediction to large-scale networks toward real-time decision making. With these observations, we are more interested in combined techniques to make contributions in concurrency, scalability, algorithm, features and metrics aspects.

1.2. Indices Selection

Feature selection is a key task in the general link prediction process and usually considers the generic features which include node attributes, network attributes and topological or non-topological features. Given the potential

edge between nodes u and v of a social network, the m features $f = \{f_1, f_2, ..., f_m\}$ that we extract from nodes are usually guided by the problem domain (e.g. title words or salient keywords in papers for citation networks). Many works use topological features of network structures to link prediction task which consists of the interaction pattern and connection structure of the network. The structural similarity indices can be derived from the network structure and classified into four categories: *local features*, *global features*, *quasi-local features* and *community features*.

Local indices use only neighboring information of the u and v nodes such as the existence of many common neighbors. Main methods such as Common Neighbors (CN) [22], Jaccard Coefficient (JC) [27] and Adamic-Adar Index (AA) [28], successfully reduced the computational expense but suffered from relatively poor prediction performance. We use the AA method on *local prediction* but try to *improve the results* with *community features* for local computation reduction.

Global indices require global topological information between two nodes that are contrived to capture non-local effects of links, such as effective conductance, hitting time and commute time. These features are often related to the non-local community structure of the social network such as Katz Index (KI) [23] that use a length-weighted count of the number of paths between two vertices to propose fair prediction performance but suffer from high complex computations.

Quasi-local indices do not require global information but use additional topological information rather than local method to try to find a nice tradeoff between complexity and performance such as the Local Path Index (LPI) [24] and the Extended Jaccard Coefficient (EJC) [29]. Local indices require less information than global ones and provide competitive fast prediction with little accuracy [30]. A research work studied the Local Random Walk (LRW) and found that the limited step may attain a better prediction than the Global Random Walk (GRW) [31]. Another work presented a fast method for large-scale networks by using the MapReduce parallel computation model [32]. Some quasi-local methods have access to the whole network, but their algorithmic time complexity is still below the time complexity of global methods and depends on the length of the paths [33]. These methods try to devise features by truncation, distribution or concurrency approaches for complexity decomposition. We select the concurrency approach to speed up the proposed LP.

Community indices require a larger density of internal community links and sparse external links across communities to help improve prediction accuracy especially in large complex social networks. Since one social person may play different roles in different communities [34] the prediction in one domain can be inspired by the information about the others [35]. These show up as a block structure in adjacency matrices, when rows and columns are suitably reordered, using methods such as the Louvain method (LM) [36], the Cross-association model (CAM) [34], the Stochastic block model (SBM) [34], etc. The LM is a very fast and scalable method for large-scale networks, which exposes the rich block structure in a network. The LM is based on local information that has two simple steps: (1) each node is assigned to a community chosen in order to maximize network modularity; (2) makes a new network based on the nodes whose communities have previously been found and iterates until a significant improvement of the network modularity is obtained. The LM is useful for in large networks with different types and sizes and supports up to millions of nodes and billions of links on a standard PC. The communities entities (U and V) can be modeled as agents (A_U and A_V high level nodes) for communities relations to devise community indices. We extract *rich communities* (by the LM) for finding *optimized subgraphs* (by GSA) to *improve the local predictions* and *best links selection to predictions* (by AA).

1.3. Proposed Approach and Evaluation Overview

Our experiments show that when the Average Clustering Coefficient (ACC) is high and low Average Shortest Path (ASP) is low for all graphs the AA *performance* is better (as a motivation example see Table 2). This paper tries to find optimized subgraphs with the highest ACC or the lowest ASP by GSA optimization goal function (*min* ASP or *max* ACC) to local predictions of AA (see Figure 1). Our proposed Gravitation-based Link Prediction (GLP) predicts all relevant links in a three-level (Community Detection, Optimization Subgraph and Local Link Prediction) model (see Figure 2) to do *concurrent predictions* of the *distributed search space* by Concurrent GLP (CGLP) to the *search time reduction*. In experimental results, we observe that the external link prediction does not have significant improvement on the accuracy of our link prediction. Therefor the process ignorance (lines 14 to 24 in Algorithm 1) of our final method is shown in Figures 1 and 2.

In the first level, the process is started with the communities' detection of an input graph (by the louvain method) to extract the graph properties for the *search space distribution*. The graph features (such as ACC, ASP, Network Density (ND), and Global Clustering Coefficient (GCC) signals) leads to the *link prediction accuracy improving* of the third level. In the second level, the process is defined by an optimization method (GSA) to satisfy our objective function (*max* ACC). The process is repeated by merging the selected strong communities to find optimized subgraph(s) (used Newtonian gravity and the motion laws). These optimized subgraphs are used only

for link predictions to the *search space reduction*. In the third level, the prediction is done for *internal of subgraphs* (concurrent for all subgraphs) leads to the *search space distribution* and for *external between all pairs subgraphs* (concurrent between all subgraphs) that leads to the *search space reduction*. Finally, applying our LM (community detection) and GSA (extraction optimized subgraph) to local and global prediction leads to the *local prediction accuracy improvement* and the *search time reduction*.

We evaluate the approach with different datasets (*astro-ph, Blogs, grid, protein, Internet topology, us-air, NetFlix, CiteSeer, Cora, WebKb* and *Gnutella pere-to-peer*) on the final approach to only the internal predictions of all optimized subgraphs. Also, we validate and compare the system performance against the benchmark suits (AA, CN, JC and KI) and using main metrics as *Accuracy*, *Precision*, *Recall* and AUC. Also, the methods analyzed with time consuming, memory occupation and complexity analysis. This, together with our detailed feature ablation studies, suggest that features at multiple regularities are responsible for the accuracy.

1.4. Contributions and Organization

Our main idea behind of the proposed topological-based GLP model is *improving local link prediction accuracy of AA* by including *community information* to find *optimized subgraphs* by applying *Gravitation and motion laws* for presenting a *scalable method* to *search time reduction* through *search space distribution* and *concurrent prediction*. Here we focus on a community-based approach for social networks LP using GSA (a swarm intelligence algorithm based on the Newtonian gravity and the laws of motion) to *reduce* the *search time* and selecting parallelism for *prediction speed-up*. Inspired by success in recommender systems, we think the prediction accuracy can be considerably improved by designing hybrid algorithms to select the different categories' features [37] from optimized subgraphs to speed prediction by parallel operations. We take notice of the correlation between communities for finding optimized subgraphs to provide best links selection to accurate prediction and implementing a method based on structural information. This achieves higher prediction accuracy with low complexity.

The method is closely related globally to relations among all detected subgraphs and locally to the complexity of each subgraph. The algorithm locally improved the accuracy and globally reduced the search space based on community information by using Newton's law for network capacity increases. The key insight of our idea is *extracting optimized subgraphs from selected strong communities for local predictions and evaluating them to prevent irrelevant predictions* to reduce the search space for a reasonable predicting time with a new index, i.e. force between communities. To the best of our knowledge, this is the first GSA based LP approach which uses a multi-level framework. Our contributions ca be summarized from *method scalability* (by search space distribution & reduction), *parallel prediction operations* (by concurrent subgraphs generation & selection), *prediction accuracy and speed improvement* (by optimized subgraph(s) extraction & concurrent selection), *computational complexity reduction* (by concurrent subgraphs selection) and *local prediction improvement* (by communities information & Newton's law inclusion) to large-scale social networks. Also in CGLP, we reduced the proposed approach tasks to parallel internal prediction of the input optimized subgraphs for improving the local prediction methods, specially Adamic/Adar.

The remainder of this paper is organized as follows. Section 2 presents previous related works. The novel proposed GLP approach and its gravitational search method is explained in Section 3. Experimental results are provided and discussed in Section 4 using standard datasets and metrics. Finally, Section 5 concludes the paper and mentions some future directions.

2. Related Work

Link mining is a long-standing challenge in modern information science which tries to extract knowledge from a given graph and includes problems such as link-based object ranking, group detection and frequent subgraph discovery [22]. Among these, LP is like a particularly new and interesting one, as it directly makes rich existing graphs by predicting and adding new edges. The LP method requires a model to score each pair of nodes independently to estimate the likelihood of their link and predict future links with graph changes. This requires high precision predictions in order to obtain practical and usable results for graphs with millions of communities or nodes. The community in graphs is a densely connected community of nodes sparsely connected to other communities which include very important information for link prediction [38]. An LP that uses community information often increases the precision and accuracy of the results. The LP approach may be categorized into three different models: similarity based approaches, maximum likelihood methods, and probabilistic models [23]. There are two major classes of similarity indices: topological-based [24] and node attribute-based [39].

Many works use topological features of network structures to link prediction task which consists of interaction pattern and connection structure of the network. Large-scale graphs have become ubiquitous in social media and

computer-based prediction in the huge graphs, pose challenges in terms of algorithm design and resource usage efficiency when decided to make decision in a distributed environments. Moreover, link prediction algorithms for graph traversing have different requirements depending of the selected way [40]. Path-based algorithms perform traversals in different directions to build a large ranking of recommended vertices. Random walk-based algorithms to compute the final ranking, build an initial subgraph for several iterations on those vertices. The aims are search space reduction in graph traverse and dividing the search space into parallel search domains.

In recent years, more and more works have employed classical social theories, such as community, triadic closure, strong and weak ties, and structural balance, to solve the social network mining problems for large scale social networks [41]. The scalability and effectiveness are both important for massive real world social networks. A large number of similarity-based methods with different definitions of similarity have been proposed [42]. There is not just one best similarity index that is superior in all settings and various measures have been shown in previous research [6]. Also, a universal best set of predictors does not exist and topological similarity indices use information about the node's neighborhoods. We think that the more ?similar? two nodes? topological neighborhoods may be the more they exhibit a future link. Most existing structural methods simply summed paths up and neglected the heterogeneity in paths or clustering information between communities. If this is so, the possibility of different paths, even with the same score or length, indicate different similarities between two end nodes. This presents a major problem where a path containing small-degree nodes indicates a greater similarity, i.e. two nodes with a small-degree common neighbor may more likely to be similar.

Most of the existing explained methods do not consider using community clustering information. These densely connected groups of vertices may play significant future prediction roles for internal and external cluster links [3]. One research performed the performance of experiments was done on real networks with various clusters unveiled the relation between the network structure and the precision of link prediction methods [44] and showed improvement in the accuracy of link prediction. Another research also showed improvements in the accuracy of the similarity-based link prediction by including the clustering information [43]. The conclusion that the community detection results contain is the essential information for link prediction that has been proven in three related works [3, 41, 44].

Considering the strong community features of the real-world and applying them is getting richer, lowering complexity in social networks analysis, and applying approaches based on global information are important for accurate prediction and selecting an appropriate and fast search method which can improve prediction search time. These need new and combined methods to construct a prediction approach to support the complex relationship. Main factors of a trusted decision are the benefits of communities. The beliefs about the benefits one could get from communities versus that of nodes. Several different methods of including community information were proposed and illustrated that in most of these supervised cases, the inclusion of community information improved the accuracy of similarity-based link prediction methods [41]. They were based on topological information to propose light works without any time consumption on network behavior analysis to achieve a good performance.

The key factor for large-scale graph processing is the reduction of information and memory space usage. Topological based prediction methods can significantly reduce the information for processing and memory usage against using the semantic information. The use semantic information may enhance the system performance but it exponentially increases the computation time and the memory for large-scale networks. Unsupervised methods only use the structural or topological characteristics of the network graph to perform link prediction based on finding the similarity for each node's pair which have no link [23]. Determining the appropriate features and similarity is the main challenge in these approaches where the extraction of structural features is done by local and global approaches [45].

The local node structure is the basis for link prediction in local approaches and extracting the local features are remarkably fast. These have higher accuracy due to considering of the entire network information but they have higher computational costs [23]. Supervised link prediction methods attempt to use learning for predicting time consuming as the size of the networks grows. To achieve an accurate prediction, different similarity measures have been proposed. However, the main key question is how we can utilize the structural features to achieve a better prediction in a more reasonable time. An important problem related to mining large data sets, both in dimension and size, is selecting a subset of the original features [46]. Preprocessing to obtain a smaller set of representative features, retaining the optimal salient characteristics of the data, not only decrease the processing time but also leads to greater compactness of the models and better generalization. When class labels of the data are unavailable we use unsupervised feature selection.

In recent years, community-based link predictions has also been proposed to learning approach [47, 48, 49]. A multi-resolution community division based prediction approach proposed that the hierarchical communities property be considered to extract different community levels and used a model to compute node pairs' frequency in different resolution of the community generated the likelihood of missing links [50]. Also, there is a similarity based method

which explores herds in different communities to predict missing links [48]. Another method utilized edge centrality measure for defining the importance of a node neighbors by assigning weights to edges based on the locality of nodes in different communities to community-based link prediction [49]. One method explored communities to predict links and a sign of the links in signed networks [51]. An important work proposed the use of local node features and community features to show prescision improvement over state-of-the-art works [52]. It integrates signals from node features, the existing local link neighborhood of a node pair, community-level link density, and global graph properties that uses a stacked two-level learning paradigm to precision improvement.

Most of described prediction methods have four major disadvantages: First, described works are very complex and do not mention the community information for accuracy. Second, they do not focus on search methods to reduce the complex networks' searching time. Third, they do not propose a suitable decentralized distributed and intelligence architecture for a candidate scalability method. Fourth, recently high time consuming works used the semantic information and implemented on learning based method. Finally, the prediction of a single link type can not apply to multiple aligned networks to cope with the heterogeneity problem [1, 4].

3. Proposed GLP Approach

In this section we define the problem definition (Section 3.1), describe the gravitation model and its parameters and metrics (Section 3.2), propose GSA mapping (Section 3.3) with a speed search (Section 3.4) and present the Gravitation-based link prediction (Section 3.5).

3.1. Problem Definition

Social network analysis should model the network and define several metrics for analyzing experimental results. A social network can be modeled as a graph $G = (V, E)$ within the time period of $G[t_{i-1}, t_i]$, where $E = \cup_i E_i$ and $V = \cup_i V_i$ are the set of nodes and the set of links, respectively. The network might change in the next time period $G[t_i, t_{i+1}]$. A link prediction method defines how to predict the evolution of links and describe the difference between $V[t_{i-1}, t_i]$ and $V[t_i, t_{i+1}]$. This can be presented to the existing network: (1) only appearance of links, (2) only disappearance of links, and (3) both of them simultaneously. The focus of our method is only on predicting appearance. We represent an interaction between nodes u and v as $e = (u, v) \in E$. Each nonexistent link modeled as $(u, v) \in U - E$ where $u, v \in V$ and U represents the universal set. Each link prediction algorithm assigns a score value ($S_{u,v}$) to each nonexistent link to qualify its likelihood of existence. Similarity between nodes u and v is defined by this score and a higher score means a higher probability of the connection between u and v.

In our approach, a social network G is divided into *communities* $C = (v, e)$, where $v \subseteq V$ and $e \subseteq E$ to *optimized subgraphs ($OS_G s$)* generation $S = (\grave{v}, \grave{e})$, where $\grave{v} \subseteq V$ and $\grave{e} \subseteq E$ that $\grave{v} \geq v$ and $\grave{e} \geq e$. However, in worst case that the number of optimized subgraphs $S$is equal to number of communities C. In order for a set of potential links L to be predicted, they can be extracted only from the optimized subgraph S_G. The social LP model M (for their labels) can map links in L (the potential social link set) to their labels in $\{1, 0\}$, $f_M : L \rightarrow \{1, 0\}$, where if link $l \in L$ is predicted to be existent, then $f_M(l) = 1$; otherwise if $(f_M(l) = 0)$ then it should be ignored to make a prediction. Our link predictor assigns a score, $X_i^{t_k}$, to each nonexistent link, L_i in an optimized subgraph S_{G_i}. This value can be viewed as a measure of similarity between two nodes of the optimized subgraph; a higher value means a higher score. A set of links $L_i^{t_k} = \{l_1^{t_K}, l_2^{t_k}, \ldots, l_n^{t_K}\}$ that predicted has the same forces $F_i^{t_k} = \{f_1^i, f_2^i, \ldots, f_n^i\}$ (generated by a link selection approach t_k and a threshold γ_G) has been that is between the two merged communities for a subgraph generation. We denote $L_{f^i}^{t_k} > \gamma_G$ as the set of links with each link $L_i^{t_k}$ whose force is $f^i > \gamma_G$. If no link has $f^i > \gamma_G$, then $L_{f^i > \gamma_G}^{t_k} = \emptyset$. Each link force f^i is assigned based on the computed force. The threshold γ_G is obtained based on the threshold set that is extracted from forces between all pairs of communities as $F = \{f_1, f_2, \ldots, f_p\}$. Based on the maximum of all the forces this threshold value is set as $\gamma_G = {}^{max_{1 \leq i \leq p} f_i}/_3$.

3.2. Gravitational Search Algorithm

Gravitational Search Algorithm (GSA) is a global optimization algorithm and has been accepted by the scientific community recently which has never been applied to social networks and its introduction for local prediction improvement [53]. In GSA, agents are considered as objects and their performance is measured by their masses, and all these objects attract each other by the gravity force, while this force causes a global movement of all objects towards the objects with heavier masses. The GSA has been applied in various applications [54]. This excludes the distance between masses in its formula, whereas the both integral parts of the gravity law. In GSA, agents are considered to be objects and their performance has evaluated, with their masses, that each agent has associated with four specifications (agent position, its inertial mass, active gravitational mass and passive gravitational mass). The position of agents provides the solution for the problem while the fitness function is used to calculate the

gravitational and inertial masses. A time function named as Kbest agent is used to attract other agents and the Kbest function value decreases with time linearly and at last only one agent will be there with heavy mass that represents final optimum solution.

Newton's Law of Gravitation shows that the gravitational force between two agents is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. Newton's law argues that when a force, $F = G(^{(M_1 M_2)}/_{R^2})$, is applied to an agent, its acceleration, $a = {}^{F}/_{M}$, depends on force and its mass, M. M_1 and M_2 are the mass of two agents, R is the distance between them, F is the magnitude of the gravitational force and G is gravitational constant. The GSA step by step procedure is given as a search space Identification, Initial population Generate, fitness function evaluation for each agent in the population and updates the gravitational constant value that the steps are modeled as follows:

- **Step 1 (Initialization):** The position initialization of N agents is randomly as (1) where x_i^d represents the positions of the i^{th} agent in the d^{th} dimension, while n is the space dimension.

$$X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^n), for\ i = 1, 2, \ldots, N \tag{1}$$

- **Step 2 (Fitness evolution and best fitness computation):** For *min* or *max* problems, the best and worst fitness of all agents are performed by evaluating the fitness of each iteration as (2) and (3) where $best(t)$ and $worst(t)$ are the best and worst fitness at iteration t and $fit_j(t)$ is the fitness value of the j^{th} agent at iteration t (for maximization problems).

$$best(t) = \mathbf{max} fit_j(t), (j \in 1, \ldots, N) \tag{2}$$

$$worst(t) = \mathbf{min} fit_j(t), (j \in 1, \ldots, N) \tag{3}$$

- **Step 3 (Gravitational constant computation):** G is computed at iteration t as (4) where G_0 and α are initialized at the beginning and will be reduced with time to control the search accuracy and T is the total number of iterations..

$$G(t) = G_0 \exp^{\frac{-\alpha t}{T}} \tag{4}$$

- **Step 4 (Masses of the agents' calculation):** For each agent, inertia masses and gravitational are calculated at iteration t as (5) where M_{pi} and $M_{\alpha i}$ are the passive and active gravitational masses, respectively, and M_{ii} is the inertia mass of the i^{th} agent where fit_i is the fitness function value of the ith agent. For the optimization problem seeking minimal value, $best = min fit_j$, $worst = max fit_j$.

$$M_{\alpha i} = M_{pi} = M_{ii} = M_i, (i = 1, 2, \ldots, N) \tag{5}$$

$$m_i(t) = {}^{(fit_i(t) - worst(t))}/_{(best(t) - worst(t))} \tag{6}$$

$$M_i(t) = {}^{m_i(t)}/_{\sum_{j=1}^{N} m_j(t)} \tag{7}$$

- **Step 5 (Accelerations of agents' calculation):** The acceleration of i^{th} agent at iteration t is computed as (8) and the total force acting on i^{th} agent is calculated as (9). The force $F_{ij}^d(t)$ acting on agent I from agent J at d^{th} dimension and i^{th} iteration is computed as (10) where $Kbest$ is the set of first K agents the best fitness value and biggest mass that will decrease linearly with time. Finally, there will be only one agent applying force to the others. The Euclidian distance between two agents i and j at iteration t is $R_{ij}(t)$. The computed gravitational constant at the same iteration is $G(t)$ while ε is a small constant.

$$a_i^d(t) = F_i^d(t)/M_i(t) \tag{8}$$

$$F_i^d(t) = \sum_{j \in Kbest}^{(j \neq i)} rand_j F_{ij}^d(t) \tag{9}$$

$$F_{ij}^d(t) = {}^{G(t).(M_{pi}(t) \times (M_{\alpha i}(t)))}/_{((R_{ij}(t) + \varepsilon) \times (x_j^d(t) - x_i^d(t))} \tag{10}$$

- **Step 6 (Velocity and positions of agents updating):** Then, the searching strategy on this concept can be described by (11) and (12) where x_i^d represents the position of ith agent in d^{th} dimension, v_i^d is the velocity, a_i^d is the acceleration and $rand_i$ is a random number among [0,1].

$$v_i^d(t+1) = rand_i v_i^d(t) + a_i^d(t) \tag{11}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{12}$$

- **Step 7 (Repeat steps 2 to 6):** For maximum limit iterations, **Steps 2** to **6** are repeated. At the final iteration, the best fitness value is computed as the global fitness. The position of the corresponding agent at specified dimensions is computed as the global solution of that particular problem.

3.3. Applied GSA to Link prediction

The gravitation-based LP parameters, constant variables, fitness function and other requirements are modeled as below.

- **Initialization of Network Parameters:** The position of the agents i.e. ACC, ASP, Network density and etc. are randomly initialized considering the upper and lower bounds of Network Parameters and the position initialization is varied from conventional GSA to reduce the computational time and increase the prediction accuracy. The position initialization of N agents is randomly as (13) where $Init$ represents the positions of the i^{th} agent, X_{Upper} is the *max* ACC and X_{Lower} is the *min* ACC.

$$X_i = Init + (X_{Upper} - Init) \times rand(0, 1) + (X_{Lower} - Init) \times rand(0, 1) \tag{13}$$

- **Fitness Function:** After controlling the Network Parameters within its permissible limits, the ACC parameter is maximized and the fitness function i.e the mass value is being evaluated as bellow

  If $(NetworkParametr_i > NetworkParametr_i^{max})$ then $NetworkParametr_i = NetworkParametr_i^{max}$
  If $(NetworkParametr_i < NetworkParametr_i^{min})$ then $NetworkParametr_i = NetworkParametr_i^{min}$
  If $(NetworkParametr_i^{min} < NetworkParametr_i < NetworkParametr_i^{max})$ then $NetworkParametr_i = NetworkParametr_i$

  For our maximization problem, the best and worst fitness of all agents are performed by evaluating the fitness to each iteration as (2) and (3) where $best(t)$ and $worst(t)$ are the best and worst fitness at iteration t and $fit_j(t)$ is the fitness value of the j^{th} agent at iteration t.

- **Gravitational constant, best and worst fitness value:** For all agents the best and worst fitness value is being determined and Gravitational constant is computed for each iteration. The appropriate values of $G(t)$ are $G_0 = 100$, $\alpha = 10$ and $T = 100$.

- **Gravitational and Inertial Masses:** The gravitational and inertial masses are updated based on the GSA formula given above.

- **Total Force:** The force and total force computed as (14) and (15).

$$F_{ij}^d(t) = {}^{G(t).(M_{pi}(t) \times (M_{\alpha i}(t))}/_{((R_{ij}(t)+\varepsilon) \times (x_j^d(t) - x_i^d(t)), \varepsilon = 0.1} \tag{14}$$

$$TF_i^d(t) = \sum_{j=1}^{N} rand_j F_{ij}^d(t) \tag{15}$$

- **Acceleration and Velocity:** The acceleration and velocity of all Network Parameters are calculated as (16) and (17).

$$a_i^d(t) = (\sum_{j=1}^{N} rand_j F_{ij}^d(t))/(\frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)}) \tag{16}$$

$$v_i^d(t+1) = rand_i v_i^d(t) + a_i^d(t) \tag{17}$$

3.4. GSA Search Speed

To overcome convergence speed and solution quality improvement in GSA performance, a modification of mass assignment local procedure needs to be used as an optimization tool. The GSA bias toward the center of the search space is a serious barrier while the range around X_g could be the most promising area for finding the optimal solution at the exact center of the search space. A mass bounded to the range of $[L_M, U_M]$ is assigned to every agent while considering the fitness of each agent g, the linear time-invariant increasing function as (18) that maps the fitness to the mass g: $R \to R$, $f(X_i) \mapsto g(f(X_i))$, $\forall X_j \in X$ defined on the fitness set of agent X_i whose value is non-negative for $f(X_i)$ and S is linearly decreased to 1 at the last iteration [55].

$$M_i = g(f(X_i)) = L_M + (U_M - L_M)\frac{f(X_i) - maxf(X_j)}{minf(X_j) - maxf(X_j)}, j \in \{1, ..., S\} \tag{18}$$

3.5. Gravitation-based link prediction

We try to maximize the clustering performance by hierarchically merging similar feature pairs, as quantified by some index that unsupervised dimensionality reduction based on information content of features. We use feature similarity for the redundancy reduction that partitions the original feature set into some distinct subsets. The features within a cluster are highly similar, while those in different clusters are dissimilar. A single features from each such community is selected (by LM) to constitute the resulting reduced subset and a similarity measure is used in subset optimization (by GSA) for quantifying redundancy in a set. The task of feature selection involves two steps: portioning the original feature set into a number of homogeneous subsets (communities) and selecting a representative feature from each such community to subset optimization construction. It has been observed that the external link prediction does not have significant improvement of the accuracy of GLP link prediction results in presented datasets. This process is deleted (lines 14 to 24 in Algorithm 1) from the method to improve local Adamic/Adar prediction speed. The final practical approach has been proposed in Figure 1 and Figure 2. The total model has been defined with external prediction for large-scale networks with strong communities and significant connections to a practical scalable framework (see Algorithm 1).

The proposed GLP (lines 8 to 26 in Algorithm 1) starts with an input graph, output predicted links, functions definitions (lines 1 to 6 in Algorithm 1) and variable initialization (line 7 in Algorithm 1). This is divided into four levels: (1) *Community Detection.* Due to an increase in the usability and serviceability of large networks an accurate rapid and scalable community detection method for good is we needed clustering for finding accurate and swift prediction results. The communities of Network (N) are detected $(C_i s(i = 1 \dots k))$ by LM (line 9 in Algorithm 1). (2) *Subgraph Optimization.* Next, optimized subgraphs $OS_{G_j}(j = 1...n)$ from the all communities $(C_i s)$ by GSA (line 10 in Algorithm 1) are computed. (3) *Link Prediction.* First, we predict local internal links $(L_i(i = 1...n))$ of

Algorithm 1: Gravitation-based link prediction approach.

Input: Graph Z; {*A Social Network Graph* $Z : G = (V, E)$}
Output: $L = \{l_1, \ldots, l_m\}$; {*All predicted links*}
1 **CommunityDetection**(G); {Detect all Communities of an input graph ($C = c_1, ..., c_k$)}
2 **AAL**($Subgraph_i$); {Return internal link prediction of an input subgraph}
3 **MRK**($LinkList, FO$); {Return a marked link list with an input FO value}
4 **PBS**($Subgraph_i, Subgraph_j$); {Return external Link prediction between two input subgraphs}
5 **LinkSelection**($List, class$); {Return a selected list form the input list based on the input class value}
6 **ComputeForce**($Subgraph_i, Subgraph_j$); {Return force between two input subgraphs}
7 **Empty** CommunityList, ForceList, PredictedLinkSet, FinalPredictedLinkSet; OptimizedSubgraphList;
$\gamma_G \leftarrow 0, FC \leftarrow 0; FO \leftarrow 0$; {*Initialize Lists & variables*}
8 **Procedure** GLP(Z); {Take graph Z:G =(V,E) as input}
9 **CommunityDetection**(Z); {Add all detect Communities to CommunityList ($C = c_1, ..., c_k$)}
10 *OptimizedSubgraphList* ←**GSA**(C); {Add Optimized subgraphs to OptimizedSubgraphList ($OS = OS_1, ..., OS_l$)}
11 **for** *($j = 1 \rightarrow l$)* **do**
12 PredictedLinkSet ← **AAL(***$Subgraph_j$**)**; {Internal predictions of all optimized subgraphs in parallel}
13 **end**
14 **for** *(each pairs of subgraphs (X,Y) in OptimizedSubgraphList)* **do**
15 $FO_{xy} \leftarrow$ **ComputeForce**(X, Y); {Compute *Force* of X and Y (FO_{xy}) for external prediction}
16 ForceList ← FO_{xy}; {Add f_{xy} to Force List}
17 **end**
18 **for** *(each pairs of subgraphs (X,Y) in OptimizedSubgraphList)* **do**
19 **if** *($FO_{xy} \geq \gamma_G$)* {*The force between (X,Y) is greater than the threshold*} **then**
20 $X \leftarrow FirstSubgraph; Y \leftarrow SecondSubgraph$; {Mark subgraphs as (X,Y) for external link predictions}
21 $PredictedLinkSet \leftarrow$ **PBS(**X, Y**)**; {External predictions between all relevant optimized subgraphs}
22 **MRK(**$PredictedLinkSet, FO_{xy}$**)**; {Mark a predicted list with its class force between the subgraphs}
23 **end**
24 **end**
25 FinalPredictedLinkSet ← **LinkSelection(***PredictedLinkSet, class*γ_G**)**; {Select all predicted links higher than a class}
26 **return**(FinalPredictedLinkSet); {Return all predicted links $L = \{l_1, \ldots, l_m\}$}

optimized subgraphs OS_{G_j} by $AA_j(j = 1...l)$ (lines 11 to 13 in Algorithm 1), concurrently. Second, external links ($L'_i(i = 1...m)$) between all relevant optimized subgraphs $OS_{G_j}, j = 1...k$ by PBS (lines 18 to 23 in Algorithm 1) are predicted from a computed threshold force between the subgraphs (lines 14 to 17 in Algorithm 1). All predicted links are marked by its strength (lines 22 in Algorithm 1)and exercised to final acceptance (line 25 in Algorithm 1). (4) *Link Classification.* The final selected link list for relevant selection or classification is processed from all marked links and the final predicted links (line 25 in Algorithm 1) is returned (line 26 in Algorithm 1).

4. Experimental Result

The experiments aim at investigating the proposed GLP and CGLP methods. First, the benchmarks and setups used in the experiments was presented in Section 4.1. Second, the used dataset and manually constructed dataset for system validation were explained in Section 4.2. Third, the basic methods for comparing results are described in Section 4.3. After that, this is followed by the evaluation metrics in Section 4.4 and validation method in 4.5. Finally, the main experimental evaluation results of performance and complexity were described in Section 4.6.

4.1. Experimental Setup

The machine for this experiment was a Core i5 3GHZ INTEL system with 16G RAM. The proposed gravitation method was implemented in a multi-threading parallel environment in the INTEL system. Experimental results of the CGLP and the other methods (GLP, AA, JC, CN and KI) were tested in four cores and one core of the machine, respectively. We used the Gephi version 0.9.2 open source software and Java language were used to experiment with different data sets and implemented our proposed method of simulation [56]. There was no gravition-based Gephi plugin therefore, we developed a new. Also, we were obliged to convert the format of data sets that was downloaded directly from the internet to pre-process this data. The datasets have to be changed according to our format to meet some of our requirements. We use NetBeans 8.0 software to process the GML and GEXF formats of the data set to store in our format [57]. It was experimented on different datasets and real world networks (see Table 1).

Table 1: The statistical information of experimental Datasets.

Dataset	NN	NE	NC	MO	DA	AD	ACC	NT
astro-ph	18771	198050	321	0.619	014	21.09	0.677	1351441
Blogs	1224	19025	010	0.426	008	31.38	0.213	101043
grid	4941	6594	039	0.932	046	2.661	0.040	651
protein	1870	2277	199	0.839	019	2.402	0.033	222
Internet topology	34761	171403	085	0.852	010	9.861	0.537	554749
us-air	1574	28236	010	0.352	008	35.87	0.469	245172
Gnutella pere-to-peer	62586	147892	033	0.361	009	7.352	0.006	2024
Citeseer	3312	4732	467	0.887	008	1.421	0.080	1223
Cora	2708	5429	098	0.805	015	3.891	0.293	1630
WebKB	0877	1608	043	0.757	010	3.372	0.332	389
Facebook	46952	876993	2007	0.673	018	37.32	0.128	122852
Netflix	978148	100480507	018	0.255	005	5.711	0.001	19850

Numbers of Nodes (NN), Numbers of Edges (NE), Numbers of Communities (NC), MOdularity (MO)
DiAmeter (DA), Average Degree (AD), Average Clustering Coefficient (ACC), Number of Triangles (NT)

4.2. Dataset

To test the accuracy of an algorithm in experiments, the observed link E, is divided into two parts on time stamps available: the training set, E^{Train}, is treated as known information (at t_i time), while the test set, E^{Test}, is treated as unknown information (at t_{i+1} time) for prediction. Obviously, $E = E^{Train} \bigcup E^{Test}$ and $E^{Train} \bigcap E^{Test} = \oslash$. All real world social networks datasets and its used information in our experiments are shown are shown in Table 1. This includes its nodes and links at t_i and t_{i+1} times with its properties. In these experiments, we first work to prove the validity of the model and then evaluate the system with standard dataset. The standard used datasets are *astro-ph*, *Blogs*, *grid*, *protein*, *Internet topology*, *Gnutella pere-to-peer*, *us-air*, *NetFlix*, *CiteSeer*, *Cora*, *Facebook* and *WebKb*. The statistical information presented in Table 1.

The *astro-ph* (Astro Physics) is derived from the *e-print arXiv* and covers scientific collaborations between authors papers submitted for the Astro Physics category [38, 58]. If an author i co-authored a paper with author j, the graph contains a undirected edge from i to j. If the paper is co-authored by k authors this generates a completely connected (sub) graph on k nodes. *Blogs-weblog* is an online users' diary where users keep their daily writing pertinent of any topic of his like/dislike; and a special genre of *blogs* that comments on politics are termed as *political blogs* [58]. The *grid* is an undirected, unweighed network representing the topology of the Western States Power Grid of the United States [59]. The *protein* dataset is a protein structure database that is modeled around the various experimentally determined protein structures [60] whose aim is to organize and annotate protein structures, providing biological community access to experimental data in a useful way. *Internet topology* is a network of connections between autonomous systems of the Internet [61]. These nodes are collections of connected IP routing prefixes controlled by independent network operators. *USair* is a network of the *US air transportation system* that deals with connections between US airports [60]. Each edge represents a connection from one airport to another and the weight of an edge shows that the number of flights on that connection in the given direction. The *Netflix* dataset consists of movies and users where that each user has rated at least one movie [62]. The *CiteSeer* dataset consists of scientific publications and the citation network consists of links [63]. The *Cora* dataset consists of scientific publications and the citation network consists of links [63]. The *WebKb* dataset consists of scientific publications and the citation network consists of links [63]. The *Gnutella pere-to-peer* dataset consists of Hosts and the connection of links [64]. Here we treat its links as undirected and its self-connections are omitted. The *Facebook* dataset consists of friends lists [65].

4.3. Methods for Comparisons

Several indexes have been presented and have shown to be useful for similarity index-based link prediction [6, 33]. They showed that neighborhood similarity measures for local, global and quasi-local methods, such as AA, CN, KI and JC provided a large factor improvement over randomly predicted links. Note, we proposed the GLP/CGLP for AA method improvement then we must compare it with AA and show this improvement. The important selected similarity index-based methods (CN, KI and JC) are our based methods for comparison.

4.4. Evaluation Metrics

Different metrics for the prediction of results can be applied to measure the performance of model M. These metrics include *Accuracy*, *Precision*, *Recall* and *ROC* and *AUC* Curves [23]. *Accuracy* (bias) is the number of correctly classified links in the link set divided by the total number of links as (20). *Precision* (spread) is the number of correctly classified positive links divided by the total number of links that are classified as positive as (21). *Recall* is the number of correctly classified positive links divided by the total number of actual positive links as (22). The *ROC* Curve is a plot of the *true positive rate* (TPR) against the *false positive rate* (FPR) as (23). The area under the receiver operating characteristic curve *(AUC)* evaluates the performance of the algorithm according to the whole list and can be interpreted as the probability that a randomly chosen missing link is given a higher score than a randomly chosen nonexistent link. We select a missing link and a nonexistent link to compare their scores each time, randomly. If among n independent comparisons, there are $\grave{n}$ times the missing link having a higher score and $\hat{n}$ times they have the same score this is formulated as (19) [23].

Table 2: Prediction methods accuracy (AUC) and time consuming (*Sec*) results.

	Accuracy					Time					
Dataset	GLP	AA	CN	JC	KI	CGLP	GLP	AA	CN	JC	KI
astro-ph	**0.9313**	0.8912	0.8653	0.8732	0.8695	**0011**	0025	0015	0012	0013	0012
Blogs	**0.9014**	0.8786	0.8679	0.8274	0.8483	**0002**	0005	0002	0002	0003	0002
grid	**0.9332**	0.8633	0.8456	0.8145	0.7984	**0005**	0012	0004	0004	0004	0003
protein	**0.8647**	0.8633	0.8646	0.8433	0.7786	**0003**	0006	0003	0003	0003	0003
Internet topology	**0.9230**	0.8960	0.8715	0.8518	0.8318	**0015**	0028	0016	0016	0017	0016
us-air	**0.9212**	0.8865	0.8457	0.8423	0.8562	**0002**	0004	0002	0002	0002	0002
Gnutella pere-to-peer	0.8214	0.8213	**0.8248**	0.7844	0.7895	**0005**	0015	0006	0005	0007	0005
Citeseer	0.8612	0.8534	0.8658	**0.8677**	0.8565	**0003**	0007	0004	0003	0004	0003
Cora	**0.9455**	0.9123	0.8975	0.8865	0.9078	**0002**	0008	0003	0003	0003	0003
WebKB	**0.9468**	0.9233	0.8897	0.9011	0.9236	**0002**	0005	0002	0002	0003	0002
Facebook	**0.7943**	0.7456	0.7433	0.7567	0.7259	**0043**	0070	0050	0044	0041	0039
Netflix	**0.7934**	0.7143	0.7098	0.7024	0.6932	**0652**	0892	0984	1156	0968	0932
Average	**0.8864**	0.8450	0.8409	0.8292	0.8232	062.0	088.5	090.0	104.3	089.0	085.5

These metrics are formulated based on *TP*, *TN*, *FP* and *FN* parameters. *True-Positives* (TP) are examples correctly labeled as positives. *True-Negatives* (TN) corresponds to negatives correctly labeled as negative. *False-Positives* (FP) refers to negative examples incorrectly labeled as positive. Finally, *False-Negatives* (FN) refers to positive examples incorrectly labeled as negative. We are not equally interested in good and bad links; the recommender should suggest good links, not to discourage the use of bad links. Three standard *AUC*, *ROC* and Precision metrics are sufficient for the evaluation of prediction algorithms which, in special cases, the *PR* is used instead of the *AUC*. *Community Density* (CD) is a ratio between existing links and all possible links as (24), where E is the number of community nodes and the *# of All Possible Links* is equal to *(E(E-1))/2*.

$$AUC = {}^{(\acute{n} + 0.5\hat{n})}/_{n} \tag{19}$$

$$Accuracy = {}^{(TP+TN)}/_{(TP+FN+FP+TN)} \tag{20}$$

$$Precision = {}^{TP}/_{(TP+FP)} \tag{21}$$

$$Recall = {}^{TP}/_{(TP+FN)} \tag{22}$$

$$TPR = {}^{TP}/_{(TP+FN)}, FPR = {}^{FP}/_{(TN+FP)} \tag{23}$$

$$CD = {}^{\#ofAllExistingLinks}/_{\#ofAllPossibleLinks} \tag{24}$$

4.5. Validation Method

We evaluate each of the baseline and new GLP methods on each of the proposed networks as follows. First, we randomly remove 5%-55% of its edges (for t_i time) with the goal of measuring how well the different methods can reconstruct the original network using the original networks as the ground truth data (for t_{i+1} time). We apply the given LP measures to the network created in this way and score the network, so that the higher the score, the more likely the nodes are to be linked. We predict x% of the highest-scoring node and vary x from 0% to 100% in increments of 5%. At each x, we count the number of *TP*, *TN*, *FP*, and *FN*, and we compute the other metrics. To account for randomness in the above description for each level of change we randomly remove the given percentage of edges from the original network five times and average the above statistics over the five runs. Our method validation result shows the best modeling for final results evaluation.

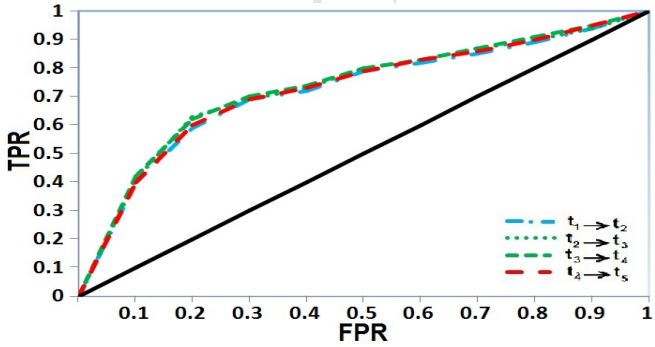


Figure 3: ROC curve of GLP predictor for periods $t_j \rightarrow t_{j+1}$ of validation dataset.

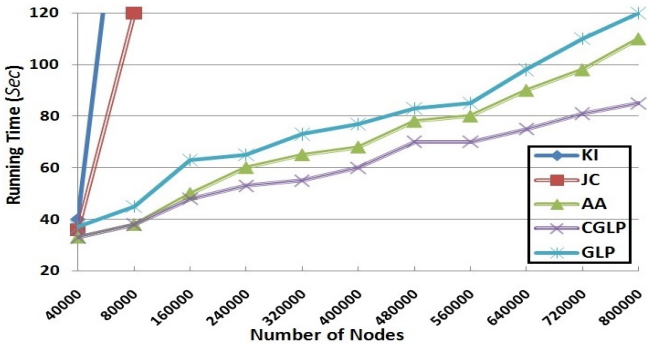


Figure 4: Running time of CGLP, GLP, AA, KI and JC methods based on the number of nodes.

4.6. Result Evaluation

In this section, the experimental results are presented in two different steps: the performance evaluation of proposed method is explained and the computational parallel tasks of GLP are analyzed with the formulation of the complexity of GLP.

4.6.1. Performance

GLP performs quiet well in different datasets and real world networks. The experimental results' evaluation of our implementation is explained with precision, accuracy, ROC and AUC. The ROC curve of our GLP shows that the *tpr* is larger than the fpr rate ($tpr > fpr$) (see Figure 3). We show that the AUC scores are greater than 0.76 for all selected time periods ($AUC_{t_1 \to t_2} = 0.73$, $AUC_{t_2 \to t_3} = 0.729$, $AUC_{t_3 \to t_4} = 0.73$, and $AUC_{t_4 \to t_5} = 0.726$ of our dataset in the validation set. However, the negative class (which does not exhibit future links) is often larger than the positive class (the number of new links), this given imbalance proposes that our gravitation-based predictor performs such as very good accuracy, compared to other works.

When analyzing the difference in the performance between GLP and baselines, we realized that it is the penalization on large-degree nodes and the consideration of long paths that explain the difference. The experimental results of our method shows its high accuracy and improvement (see Table 2 and Figure 4) on large networks (e.g. *astro-ph*, *Blogs*, *Internet topology* and *Facebook*) with complex connection (e.g edges more than 8 times the nodes). CGLP has a good balance on computational complexity and running time comparable to the main methods (see Column 7 in Table 2). A comparison of Column 2 and 3 in Table 2 shows improvement in accuracy of the motivated AA method in all cases as a main goal of GLP method. The results show improvement up to 13% accuracy over AA method (see Columns 2 and 3 in Table 2), up to 14% improvement in other cases (see Columns 2 and 4-6 in Table 2) and prediction space reduction to large-scale networks (e.g. *astro-ph*, *Blogs*, *Internet topology* and *Gnutella pere-to-peer*) and complex netwrok (e.g. *Netflix*). The GLP accuracy improves high clustering coefficient and triangle number (e.g. *astro-ph*, *Blogs*, *Internet topology*, *us-air* and *Facebook*). The CGLP timing shows significant improvement to large-scale and complex networks (e.g. *astro-ph*, *Blogs*, *Internet topology*, *Gnutella pere-to-peer* and *Netflix*) (see Column 6 in Table 2).

Our experiments show that the class inclusion of the community information improves the accuracy of the gravitation-based prediction method in all high ACC and NT cases (*astro-ph, Blogs, Internet topology* and *us-air*). The results show that gravitation-based link prediction has better performance in networks in which a predictor has used community information well. For example, the predictor had a good accuracy for the *Cora* and *WebKB* datasets, so the accuracy of algorithms in these datasets are better than the others. On the other hand, GLP performance is improved in networks with a higher density and a shorter graph diameter (see Table 1). The *Mass* and the *Distance* of each two communities are related to the density and the number of existing links between communities. So, the GLP performance gives better results in all datasets mostly because these datasets have many connected components and the number of features.

The GLP uses the structural attributes of the network more and in networks with a higher max degree it has better performance, although this is not a general rule. It is seen that the GLP and AA, which solely depend on link structure, improve as the graph becomes more dense (number of neighbors increases) or becomes more social (number of triangles increases) or becomes complex (number of node and links increases). *WebKb*, *Cora*, *Facebook* and *Citeseer* have very unstructured feature spaces. However, the high clustering coefficient of *WebKb*, *Cora* and *Facebook* balance this disadvantage and is a key reason for them good performance. Also, the very high triangles number of *astro-ph*, *Blogs*, *Internet topology*, *us-air* and *Facebook* is another key reason for them good performance. *Gnutella pere-to-peer*, *Citeseer* and *Netfliix* have very low ACC and perform poorly. However, the number of triangles and the complexity of *Netflix* balances this disadvantage and is a key reason for improving accuracy of GLP and significant improving time consumption of CGLP. We observe the variation of GLP over various datasets, across the most competitive algorithms. This variation is due to various graph properties like density, clustering coefficient, triangle number or network capacity.

4.6.2. Complexity

To compute the capability of our LP candidate to large graphs we must find ways of efficiently parallelizing the GLP approach. In that regard we have noticed that the LP may be a very appropriate problem for parallel computing, as it can be divided into independent tasks and subtasks. Since, in GLP, each link can be predicted independently from the rest, the tasks can be paralleled without dependency. In other words, there is no waiting time between internal and external link evaluations, which maximizes the use of CPU and memory. There are some domains of concurrency such as gravitation parameters computations, external predictions, internal predictions and link selection in our GLP approach. We can divide the GLP into two main concurrent parts for parallel tasks design as stated below:

- **Internal predictions:** Since each optimized subgraph internal link prediction can be computed independently from the other subgraphs, the tasks can be paralleled without dependency. We design each internal predictor as a task that can be run on different CPUs or Cores. There are no waiting times between individual internal link evaluations, which maximizes the use of CPU.
- **External predictions:** Also, each external link prediction of the optimized subgraphs can be computed independently from the other pairs. The tasks can be executed in a parallel form maximizes the use of CPU and memory. Calculating the computational time of all external predictions is difficult.

We model the internal and external predictions to the complexity evaluation of GLP. All internal predictions can be executed and evaluated concurrently. The maximum computational time of all GLP internal predictions is equal to the largest or strongest community (or optimized subgraph) predictions and is computed $O(v \times (v + e))$ by the AA method which is an optimized subgraph of input graph. Also, all the external predictions such as a football tournament, in which the team assignment problem is NP-hard can be scheduled and evaluated [66]. The modified canonical *tournament* pattern can be generated in polynomial time $O(C^2)$ where $C \ll N$ is the number of nodes (subgraphs) in the tournament, thus we can ignore it [67].

Since the time complexity to traverse the neighborhood of a node is simply k, the time complexity in calculating the gravitation-based approach is $\mathrm{O}(k^2C)$. Where C is the number of detected communities in the network and k is the average degree of network. So, the complexity is $\mathrm{O}(k^2C)$ and it can be ignored for $C \ll N$. Analogously, for the LP index, what we do not need to do is go one step further (called step 3) to check all neighbors of each of x's second-order neighbors, respectively. A node u does not appear neither in x's second-order neighborhood nor in the third-order neighborhood. Therefore, the time complexity in calculating the LP index is $\mathrm{O}(k^3N)$. We need to compute the complexity of the GLP method at many steps (see Figure 2). A detailed time complexity and memory consuming illustration for our link prediction is described as follows.

Firstly, we must compute the time complexity of the community detection algorithm that is seeking all strength communities. This is the best implemented version of the Louvain method. The time complexity is $\mathrm{O}(k|E|)$ or $\mathrm{O}(kV)$ where $|E| = V$ and k are the number of edges in the network and length of the k-paths, respectively [68]. *Secondly*, for subgraph optimization of all communities, the time complexity is $\mathrm{O}(k^2N)$. *Thirdly*, in external link prediction, the link prediction between all subgraphs is as a tournament ($\mathrm{O}(k^2N)$). *Fourthly*, in the internal link prediction, there is a AA link prediction for the internal prediction of each optimized subgraph, so the time complexity for homogenous network is $\mathrm{O}(2n^2)$ where ($n \subseteq N$) of the experimental graph is a subgraph of network graph. *Fifthly*, there is a is link selection for all predicted links so the time complexity is $\mathrm{O}(k^2N)$. The total complexity of GLP is $\mathrm{O}(kN^2)$ or $\mathrm{O}(N^2)$ which is equal to AA where AA is $\mathrm{O}(N^2)$ [69], KI is $\mathrm{O}(N^3)$ [70], CN is $\mathrm{O}(k^2N)$ [71], and JC is $\mathrm{O}(N^3)$ [69]. However, in our reduced approach ((see Figure 1)) we ignored the external link prediction between all optimized subgraphs and the time complexity can be reduced.

Besides time complexity, memory space is another limitation for the implementation of huge-size networks. In calculating gravitation and LP indices, the memory required is of the order $\mathrm{O}(kN)$. GLP keeps the graph nodes and edges in memory using an adjacency list representation, which requires $\mathrm{O}(N + V)$ space, where N is the total number of nodes and V is the total number of edges where the complete graph is equal to $\mathrm{O}(N + N^2)$ or $\mathrm{O}(N(N + 1))$. All other arrays (CommunityList, ForceList, PredictedLinkSet and FinalPredictedLinkSet and etc.) require $\mathrm{O}(4N)$ max space. Therefore, our GLP total space complexity is $\mathrm{O}(N + V + 4N)$=$\mathrm{O}(5N + V)$. In a word, compared with the applied GSA approach and other methods, the LP index is not only highly effective (i.e., accurate), but also highly efficient (i.e., required relatively less memory and CPU time).

5. Conclusion

In addition to *node features* for local prediction performance improvement in a novel *multi-level unsupervised approach* to link prediction this paper describes the use of *graph structure* and *community features*. The main goal aim of the proposed approach is to improve, *locally*, accuracy and *globally*, search speed based on *community detection* and *gravitation laws optimization* to network *capacity growing*. The predictions are divided *externally* between all generated optimized subgraphs and *internally* for each subgraph, into search space distribution. The experimental results on complicated networks show high *accuracy* and *speed* of CGLP and much less CPU time and memory space usage. This improves particulary on greater existence of powerful communities and triangles with a good balance on computational complexity and prediction time. In the proposed multi-level approach, community detection, subgraph optimization and the link prediction methods can be changed for a better performance by applying appropriate algorithms. However, we can not control and limit the amount of CPU time consumed of GLP. Our limitation is the search time of GLP on online graph distribution for community detection and subgraph optimization to make significant speed up. Further study on the real world for this evaluation would provide more insight into solving unsupervised big network problems and towards an online prediction in large-scale social networks.

[1] Zhang J., Kong X., Yu P., Predicting social links for new users across aligned heterogeneous social networks, ICDM, 2013.

[2] J. Leskovec, D. Huttenlocher, J. Kleinberg. Predicting Positive and Negative Links in Online Social Networks, In Proceedings of WWW'2010, ACM Press, New York, 2010.

[3] Liu Z., Zhang Q. M. , L? L. , Zhou T., Link prediction in complex networks: a local na?ve bayes model, EPL (Europhysics Letters), 2011, 96.

[4] Zhang JJ., Kong X.,Yu P., Transferring heterogeneous links across location-based social networks, WSDM, 2014.

[5] Backstrom L., Dwork C., Kleinberg J., Wherefore art thou R3579X?: Anonymized social networks, hidden patterns and structural steganography, WWW, 2007.

[6] Wang D., Pedreschi D., Song C., Giannotti F., Barabasi A., Human mobility social ties, and link prediction, KDD, 2011.

[7] Clauset A., Moore C., Newman M., Hierarchical structure and the prediction of missing links in networks, Nature, 2008, 453(7191).

[8] Liben-Nowell D., Kleinberg J., The link prediction problem for social networks, CIKM, 2003, 556-559.

[9] Wang H., et al. ,Nodes' Evolution Diversity and Link Prediction in Social Networks, IEEE Transactions on Knowledge and Data Engineering,2017, 29(10): 2263-2274.

[10] Hu W. , Wang H., et al., An event detection method for social networks based on hybrid link prediction and quantum swarm intelligent, World Wide Web-internet & Web Information Systems, 2017, 20(4): 775-795.

[11] Jia Y., Wang Y., Li J., Feng K., Cheng X., Li J., Structural-interaction link prediction in microblogs, WWW, 2013.

[12] Yang Y., Chawla N., Sun Y., Han J., Link prediction in heterogeneous networks: Influence and time matters, ICDM, 2012.

[13] Fouss F., Pirotte A., Renders J., Saerens M., Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation, TKDE, 2007.

[14] Konstas I., Stathopoulos V., Jose J. M., On social networks and collaborative recommendation, SIGIR, 2009.

[15] Tong H., Faloutsos C., Pan J., Fast random walk with restart and its applications, ICDM, 2006.

[16] Aditya K., Menon A., Elkan C., Link prediction via matrix factorization, ECML/PKDD, 2011.

[17] Tang J., Gao H., Hu X., Liu H., Exploiting homophily effect for trust prediction, WSDM, 2013.

[18] Dunlavy D., Kolda T., Acar E., Temporal link prediction using matrix and tensor factorizations, TKDD, 2011.

[19] Sun Y., Han J., Aggarwal C., Chawla N., When will it happen?: relationship prediction in heterogeneous information networks," WSDM, 2012.

[20] Yu X., Gu Q., Zhou M., Han J., Citation prediction in heterogeneous bibliographic networks, SDM, 2012.

[21] Domingos P., Richardson M., Markov logic: A unifying framework for statistical relational learning, ICML Workshop, 2004.

[22] Newman M. E. J., Clustering and preferential attachment in growing networks, Physics Review E, 2001, 64.

[23] Lu L., Zhou T., Link prediction in complex networks: A survey, Physica A: Statistical Mechanics and its Applications, 2011.

[24] Zhou T., Lu L., Zhang Y., Predicting missing links via local information, The European Physical Journal B, 2009.

[25] Bliss C. A., Frank M. R., Danforth C. M., Dodds P. S., An Evolutionary Algorithm Approach to Link Prediction in Dynamic Social Networks, Journal of Computational Science, 2014, 5(5): 750-764.

[26] Kong X., Zhang J., Yu P., Inferring anchor links across multiple heterogeneous social networks, CIKM, 2013.

[27] Jaccard P., ?tude comparative de la distribution florale dans une portion des Alpes et des Jura, Bulletin de la Soci?t? vaudoise des sciences naturelles, 1901, 37: 547-579.

[28] Adamic L. A., Adar E., Friends and neighbors on the web, Social networks, 2003, 25(3): 211-230.

[29] Chartsias A., Link prediction in large scale social networks using hadoop, PhD thesis, Technical University of Crete, Greece, 2010.

[30] Gibson, D., Kleinberg, J., Raghavan, P., Inferring Web communities from link topology. Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia, 1998.

[31] Liu W. P.,Lu L L., Link prediction based on local random walk, Eur. Phys. Lett., 2010, 89.

[32] Yu-xiao D., Qing K.E., WU B., Link prediction based on node similarity, Computer Science, 2011, 38(7): 162-164.

[33] Mart?nez V., Brezil F., Cubero JC., A Survey of Link Prediction in Complex Networks, ACM Computing Surveys (CSUR), 2016, 49(4): 69-99.

[34] S. Fortunato, Community Detection in Graphs, Physics Report, 2010,486.

[35] B. Cao, N. N. Liu, Q. Yang, Transfer learning for collective link prediction in multiple heterogenous domains, In Proceedings of the 27th International Conference on Machine Learning, 2010.

[36] Blondel V. D., Guillaume J., Lambiotte R., Lefebvre E., Using community information to improve the precision of link prediction methods, Proceedings of the 21st International Conference Companion on World Wide Web ACM, 2012, 607-608.

[37] E. Zheleva, L. Getoor, J. Golbeck, Ugur Kuter, Using friendship ties and family circles for link prediction, In Proceedings of the 2nd Workshop on Social Network Mining and Analysis, ACM Press, New York, 2008.

[38] J. Leskovec, J. Kleinberg and C. Faloutsos. Graph Evolution: Densification and Shrinking Diameters. ACM Transactions on Knowledge Discovery from Data (ACM TKDD), 2007, 1(1).

[39] Aiello LM., Barrat A., Schifanella R., Friendship prediction and homophily in social media, ACM Trans. Web. (TWEB), 2012, 6(2).

[40] Corbellini A, Godoy D, Mateos C, Schiaffino S, Zunino A., DPM: A novel distributed large-scale social graph processing framework for link prediction algorithms, Future Generation Computer Systems, 2017.

[41] Soundarajan S., Hopcroft J., Using community information to improve the precision of link prediction methods, Proceedings of the 21st international conference companion on World Wide Web, ser. Proceedings of WWW?12 Companion, 2012, 607–617.

[42] Pulipati S., Manjula R., Similarity Index based Link Prediction Algorithms in Social Networks: A Survey, Journal of Telecommunications and Information Technology, 2016, 2: 87-95.

[43] Soundarajan S., Hopcroft J., Fast unfolding of communities in large networks, Physica A: Statistical Mechanics and its Applications, 2012, 607-608.

[44] Feng X.,J. Zhao J., Xu K., Link prediction in complex networks: a clustering perspective, the European Physical Journal B, 2012,85: 1-9.

[45] Mitra, P., Murthy, C.A. and Pal, S.K., Unsupervised feature selection using feature similarity. IEEE transactions on pattern analysis and machine intelligence, 2002, 24(3):301-312.

[46] Fayyad U. M., Shapiro P. G., Smyth P. ,Data mining and knowledge discovery in databases: an overview, Communications of the ACM, (1996), 39(11).

[47] Ding J., Jiao L., Wu J., Liu F., Prediction of missing links based on community relevance and ruler inference, Knowledge-Based Systems, 2016, (98): 200-?215. doi:10.1016/j.knosys.2016.01.034

[48] Kuang R., Liu Q., Yu H., Community-based Link Prediction in Social Networks, Springer International Publishing, (2016):341-?348.

[49] Biswas A., Bhaskar B., Community-based link prediction, Multimedia Tools and Applications, (2017):1-21.

[50] Ding J., Jiao L., Wu J., Hou Y., Qi Y., Prediction of missing links based on multi-resolution community division, Physica A: Statistical Mechanics and its Applications, (2015) 417:76-?85.

[51] Shahriary S., Shahriari M, Noor R. A community-based approach for link prediction in signed social networks, Scientific Programming (2015) :5.

[52] De A., et al., Discriminative Link Prediction using Local, Community, and Global Signals," IEEE Transactions on Knowledge and Data Engineering, 2016, 28(8): 2057-2070.

[53] Rashedi E., Nezamabadi-pour H., Saryazdi S., GSA: A Gravitational Search Algorithm, Information Sciences, 2009, 179(1313):2232-2248.

[54] Sabri N. M., Puteh M., Mahmood M. R., A Review of Gravitational Search Algorithm, Int. J. Advanced Soft Computing Appllied, 2013, 5(3).

[55] Davarynejad M.,van den Berg J., Simulated big bounce: a continuous space global optimizer Technical report, Faculty of technology policy and management, Delft University of Technology, The Netherlands, 2012.

[56] https://gephi.github.io/users/download/.

[57] https://netbeans.org/downloads/6.7.1/.

[58] https://snap.stanford.edu/data/ca-AstroPh.html.

[59] https://networkdata.ics.uci.edu/index.php.

[60] http://konect.uni-koblenz.de/networks/.

[61] http://konect.uni-koblenz.de/networks/topology.

[62] http://www.netflixprize.com.

[63] www.cs.umd.edu/projects/linqs/projects/lbc.

[64] http://snap.stanford.edu/data/p2p-Gnutella31.html

[65] https://snap.stanford.edu/data/egonets-Facebook.html

[66] McAloon K., Tretkoff C., Wetzel G., Sport league scheduling, Annual ILOG Optimization Users Conference, 1997.

[67] Schaerf A., Scheduling sport tournaments using constraint logic programming, Proc. of the 12th European Conf on Artificial Intelligence (ECAI-96), 1996, 634-639.

[68] De Meo P., Ferrara E., Fiumara G., Provetti A., Generalized louvain method for community detection in large networks, Proceedings of the 11th International Conference on Intelligent Systems Design and Applications, 2011.

[69] Peng W., BaoWen Xu., YuRong Wu., XiaoYu Z., Link prediction in social network: the state-of-the-art, Sci. China Inf. Sci., 2015, 58: 1-38.

[70] Lu L., Jin C. H., Zhou T., Effective and efficient similarity index for link prediction of complex networks, arXiv:0905.3558, 2009.

[71] Rahman M., Hassan MR., Buyya R., Jaccard index based availability prediction in enterprise grids, International Conference on Computer Science, ICCS-2010, 2010, 2701-2710.

- This unsupervised scalable approach employs Newtonian's law for link prediction in social graphs.
- It outperforms community information to speed accurate link prediction for complex search space reduction.
- It presents a new gravitation-based metric to social networks link prediction.