The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

WSCAN-TFP: Weighted SCAN Clustering Algorithm for Team Formation Problem in Social Networks

Kalyani Selvarajah, Amangel Bhullar, Ziad Kobti, Mehdi Kargar
University of Windsor, Windsor, ON, Canada.
(selva111,bhull113,kobti,mkargar)@uwindsor.ca

Abstract

In this paper, we provide a novel approach for the Team Formation Problem (TFP) in social networks. With a given social network of experts and communication cost between them, we address the problem of finding a team with a set of required skills necessary to complete a project. An expert of this network is treated as a node and possesses a given set of skills. The basic idea of the Structural Clustering Algorithm for Networks (SCAN) is to detect clusters, hubs, and outliers in networks. To employ SCAN on TFP, we first find the pool of experts with required skills. Then we search for highly connected (core) expert among all experts network. We expand the cluster from core to neighborhood nodes, and it goes from densely connected to loosely connected nodes within a threshold range of communication cost. We solve this TFP by identifying experts while minimizing communication cost for the project with specific skills. We then measure the communication cost with the sum of a distance function. An enhanced variant of SCAN is the weighted structural clustering algorithm (WSCAN) which is implemented in this paper to solve the TFP with minimum communication cost. Our result with WSCAN performed approximately equal to Greedy algorithms while slightly worse than other Genetic, Cultural and Exact Algorithms. The run-time of WSCAN however, was better compared to the others.

Introduction

Team Formation Problem (TFP) deals with two main components: experts and projects, where each project requires a set of skills that an expert would possess. The goal of TFP is to assemble effective teams to complete a project successfully. In forming the team, we take into account the expected skills and their degree of collaboration among the group of experts. The authors of (Lappas, Liu, and Terzi 2009; Kargar and An 2011) evaluated communication cost based on any two experts based on previous experience and assigned the cost as an edge-weight between them. For example, if the weight is low, their relationship strength is high, and they are highly preferred to be on the same team. If they worked together in the past, we assume their relationship is strong and the probability to finish the project on time by this team is high.Researchers in TFP used various attributes to form an efficient team. However, they found that involving each function to minimize or maximize is an NP-hard problem. Therefore, in the paper (Lappas, Liu, and Terzi 2009) the authors used greedy algorithms to optimize communication cost. Later, (Kargar and An 2011) employed a new function to measure the communication cost between experts of a team and applied greedy algorithms to find an approximate answer.

With large networks, greedy algorithms may not consider a global optimum value, and there are high chances of increasing run-time. So, by considering these issues, we approach the TFP using the SCAN algorithm, proposed by (Xu et al. 2007), is designed to find clusters, hubs, and outliers in large networks. We define clusters based on the experts' past collaboration frequency. In other words, we cluster them together with their communication cost value. If the experts have less communication distance we consider them as a similar group and are clustered together. Then we continue the search until we find the experts with the required skills.

In this paper we will employ WSCAN to solve TFP. We are proposing a new term called collective expertise, defined as a phenomenon of occurrence of a certain level of expertise among a group of individuals who are possessing a set of skills necessary to complete a task as a team. Next, we compare our proposed algorithm with greedy, genetic, random, and exact algorithms. We conduct experiments using real-world networks with 50K nodes derived from the DBLP dataset. The rest of the paper is organized as follows: In the next section, related work is discussed then we present the problem statement. Next we employ WSCAN to find the best team of experts. Then a set of experiments has been presented with a real dataset. Finally, we conclude our discussion and findings.

Related Work

Xu in (Xu et al. 2007) proposed SCAN to solve the problem of ignoring important isolated nodes such as hubs by other clustering algorithms such as modularity-based algorithms while forming clusters within a graph. This algorithm is used in many applications to detect clusters as well as to find hubs and outliers present in a graph. Chertov et al. (Chertov, Kobti, and Goodwin 2010) introduced a weighted version of SCAN which allowed to overcome the limitations of the original SCAN. The original SCAN was implemented on the unweighted and undirected graph.

The authors of (Lappas, Liu, and Terzi 2009) proposed two communication cost functions and used Rarest First and Enhanced Steiner algorithm to discover a team of experts from a social network. Another method was proposed by Kargar and An (Kargar and An 2011) who introduced a team with a leader that minimize leader distance function and produce top-k team. Kargar et al.(Kargar, An, and Zihayat 2012) assumed every expert is associated with a cost in order to perform an assumed task in a given project. By using tradeoff parameter, they combined together two objective functions into one.

Awal et al.(Awal and Bharadwaj 2014) proposed to find a team of experts in a social network using collective intelligence index. They used a random expert to optimize communication cost and expert level with implementation of the general genetic algorithm (GA). Everyone applied various techniques in crossover and mutation to have a better team of experts. Recently, Selvarajah et al.(K.Selvarajah and Kobti 2017) applied CA, proposed by Reynolds (Reynolds 1994), to find a better optimal solution by extracting knowledge from the initial population and update to next population. It achieved a slight improvement over Genetic and Greedy Algorithms.

Problem Statement

Assembling a team while considering optimization of communication cost will be an effective solution for TFP. The general problem is to assign the experts to a team T from a set of experts e_i possessing a set of skills l_j to complete a project. However, to complete any project, we find a team through a specific requirement criteria R.

Set of experts can be defined as a set of individuals E; $e_i \in E$ where $e_i,\ i = 1, 2, \ldots, m$ possess a set of skills and their profile represented by the skill space L; $\{l_j \in L\}$.

Set of domain specific skills can be defined as a set of total number of abilities l_j, where $l_j,\ j = 1, 2, \ldots, n$ possessed by all experts available.

Set of project specific skills can be defined as a subset of abilities l_j , required to carry out a specific task with predefined criteria R, $\{R_k \subseteq l_j \in L\},\ k = 1, 2, \ldots, x$. Project specific skills, satisfying task requirement criteria to complete a task, is simply a subset of domain specific skills set.

In this paper, we focus on a social network modeled on weighted undirected graph G. An underlying social network connects the experts in E. Let $G = (E, D)$ be a graph with vertices (E) and edges (D) that are weighted W. Vertices indicate the set of experts and edges represent the previous collaboration between the connected experts.Terms such as node and expert can be used interchangeably in this work.

As we have already discussed, we assume that individuals are organized in an undirected and weighted graph. Every node of G corresponds to an individual in $e_i \in E$. The edge weight W gives communication cost between two experts. If two experts have frequent collaborations, the edge weight is small and conversely, if the weight is large that means rare collaborations occurred. For example, if two experts work on many projects in their past experience, their strength of connectivity is high, it means distance between them is low.

Suppose, each expert E_i has a set of skills $S(e_i) \subseteq L$. To be part of a team or to be member of a task or project team, every expert must have at least one skill from R, $\{ R \subseteq l_j\}$ and $\{l_j \in L\}$. Therefore, if at least one element of R is satisfied by any E ;$\{ e_i \in E, i = 1, 2, \ldots n\}$ from set of n experts. Then, she/he is a member of the team.

$E = \{e_1, e_2, \ldots, e_n\}$ specifies a set of n experts, and $L = \{l_1, l_2, \ldots, l_m\}$ specifies a set of m skills. Each expert e_i has a set of skills, specified as $S(e_i)$, and $S(e_i) \subseteq L$. If $l_j \in S(e_i)$, expert e_i posses skill l_j. A subset of experts $E' \subseteq E$ have skill l_j if at least one of them posses l_j. For each skill l_j, the set of all experts that posses skill l_j is specified as $E(l_j) = \{e_i | l_j \in S(e_i)\}$. A project $P = \{l_1, l_2, \ldots, l_t\}$ is composed of a set of R skills that are required to be completed by some experts.

Definition. ***(Team of Experts)*** *Given a set of experts E and a project P that needs a set of skills* $\{el_1, el_2, \ldots, el_m\}$*, a* team T of experts *for P is a set of R skill-expert pairs:* $T = \{\langle e_{l_1}\rangle, \langle e_{l_2}\rangle, \ldots, \langle e_{l_r}\rangle\}$*, where* e_{l_j} *is an expert that posses skill* l_j *for* $j = 1, 2, \ldots, r$.

Definition. ***(Sum of Distances)*** *Given a graph G and a team of experts* $\{ T = \langle e_{l_1}\rangle, \langle e_{l_2}\rangle, \ldots, \langle, e_{l_r}\rangle\}$*, the* sum of distances *of the team is defined as*

$$sumDistance = \sum_{i=1}^{x} \sum_{j=i+1}^{y} dist(e_{l_i}, e_{l_j})$$

where $dist(e_{l_i}, e_{l_j})$ *is the distance between* e_{l_i} *and* e_{l_j} *in G.*

Definition. ***(Communication Cost*** (CC)***)*** *can be defined a distance between two experts* e_i *and* e_j *on a graph G.*

In this paper, $CC(e_i, e_j)$ *and edge weight* $w(e_i, e_j)$ *are used interchangeably.*

Problem. ***(Team Discovery)*** *Given a project P, a set of experts E, and a social network that is modeled as graph G, the problem of team discovery in social networks is to find a team of experts T for P from G so that the communication cost of T is minimized.*

Algorithm

We are using WSCAN on social network as a graph to find the best team of experts.

Definition. ***(Vertex structure)*** *Let* $e_i \in E$ *, the structure of* e_i *is defined by its neighborhood, denoted by* $\tau(e_i)$.

$$\tau(e_i) = \{e_j \in E \vee (e_i, e_j) \in E\} \cup \{e_j\}$$

Definition. ***(Neighborhood*** (ϵ) ***)***

$$N_\epsilon = \{e_j \in \tau(e_i) | \sigma(e_i, e_j) \geq \epsilon\}$$

Definition. ***(Extended Structural Similarity)*** *Structural similarity of two vertices, suppose* e_i *and* e_j *will be large if they share a similar structure of neighbors that is frequent regime of working together and communication cost.*

$$\sigma(e_i, e_j) = \frac{|\tau(e_i) \cap \tau(e_j)|}{\sqrt{|\tau(e_i||\tau(e_j|)}} w(e_i, e_j)$$

Where w is weight of the edge between e_i and e_j. σ is inversely proportional to communication cost. If σ is high, communication cost (CC) will be low.

$$\sigma(e_i, e_j) \propto \frac{1}{CC(e_i, e_j)}$$

Relationship of CC and strong/weak bonding between experts can be express as, sum of CC_e of experts is inversely proportional to the frequent collaboration f_e of experts E.

$$CC(e_i, e_j) \propto \frac{1}{f(e_i, e_j)}$$

Therefore, less communication cost represents strong bond between e_i and e_j and, strong bonds between e_i and e_j gives high structural similarity.

$$\sigma(e_i, e_j) \propto f(e_i, e_j)$$

If two experts e_i and e_j collaborates together more frequently they are likely to have more structural similarity.

Definition. ***(Core)*** *Let $\mu \in N$, a vertex $e_i \in E$ is called a core with reference to ϵ and μ , if its ϵ - neighborhood contains at least μ vertices.*

$$Core_{\epsilon,\mu}\,(e_i) \leftrightarrow |N_\epsilon| \geq \mu$$

Where μ is number of neighborhood experts connected to core vertex (highly connected expert).

Experts Selection Strategy

Algorithm 1 is our solution to TFP using WSCAN approach. As a first step of our algorithm, we find out the pool of experts PoE from total n number of experts E. Next we find the core person out of PoE. WSCAN will choose core person based on μ number of neighborhood nodes that have ϵ and are connected to it. However, ϵ is a threshold that can be defined as the most communication cost feasible to have a team of experts possessing skills from a set of project specific skills R. We check if $e_(i, j)$ is a core node of cluster, that is, if it is highly connected. Then, the cluster expands around $Core_e(i, j)$. We want to make clusters based on structural similarity σ. However, σ is inversely proportional to CC. If $e_(i, j)$ is not core then it will assign a non member level to it and the loop ends here. This achieves a cluster with experts with at least one project specific skill R from L, where $R \subseteq L$ from pool of experts. Furthermore, we will check for skills R from $S_e(i, j)$. If two experts have the same set of skills to R, we choose based on least CC, and if CC is the same, we choose randomly. Further, if we find an expert for one skill from set R, then we have a potential team member. We keep this team member on the required team list and we remove that skill from our requirement list $ReqS$ because we no longer need to search for the same skill. Thereafter, we will check for the remaining skills based on minimum CC. If we cannot find within this value we will increase the distance. We continue the search for a potential team members until we find them by searching highly connected clusters, we then search for isolated nodes that are hubs and outliers. If two clusters are connected through a common node that is loosely connected, then it is a potential team member and it is connecting two clusters with other potential team members for different R skills; thus we are choosing it. Moreover, if this node is not joining two clusters and loosely connected to a single cluster, then it is an outlier. Finally, we are able to find team T for project P from pool of experts Pos.

Algorithm 1 WSCAN-TPF

```
Input: Graph $G = (< E, D >, \epsilon, \mu), W(e_i, e_j)$;
$E \in 1 \geq i, j \leq n \leftarrow$ number of n experts
$L \in 1 \geq i, j \leq m \leftarrow$ number of m domain specific skills
$R \in 1 \geq i, j \leq x \leftarrow$ number of x required project skills
Output: Best Team $T$

  Store Pool of experts $(PoE)$ from $E$
  for each unclassified vertex $e \in E$ do
    if $e$ is core $Core_{\epsilon,\mu}\,(e)$ then
      generate new clusterID;
      insert $e_i \in N_\epsilon(e)$ into queue $Q$;
    end if
    while $Q \neq 0$ do
      $e_j$ = first expert in $Q$
      if $e_i$ is unclassified or non-member then
        assign current clusterID to $e_i$;
      end if
      if $e_i$ is unclassified then
        insert $e_i$ into queue $Q$;
        remove $e_j$ from $Q$;
      else if $e$ is not core then
        labeled as non member label of $e$;
      end if
    end while
  end for
  for $CC \leq$ threshold do
    calculate distance from Core vertex $e_i$
    if $e_i$ have project specific skill from $Se_i$ then
      if check minimum $CC$ then
        if more than one with same $CC$ then
          Choose an random expert & Store in teamlist $T$
        end if
      end if
    end if
    remove skill already found from list = $ReqS$
    increase $CC$
  end for
  if $e_{i,j}$ is common to cluster 1 and cluster 2 then
    label it as a hub.
  end if
  if check high value of $CC$ then
    label it as a outlier
  end if
  if required skills remain then
    if checking for hubs with minimum cost then
      put in teamlist $T$
    end if
  end if
  return teamlist $T$
```

Experiment and results

To evaluate WSCAN for TFP in a social network, we compare our results with the Cultural algorithm (CA), the Genetic algorithm (GA), Greedy algorithm, Exact algorithm

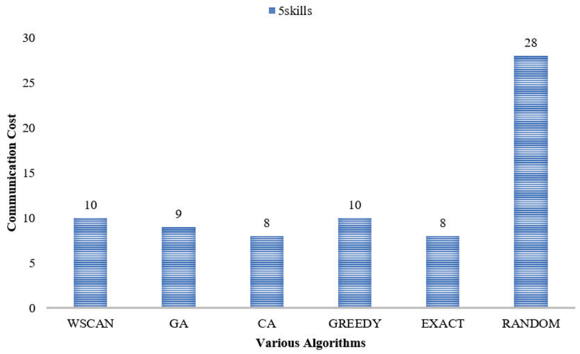

Figure 1: Comparison for various algorithms for TFP with weighted SCAN for the project require 5 skills.

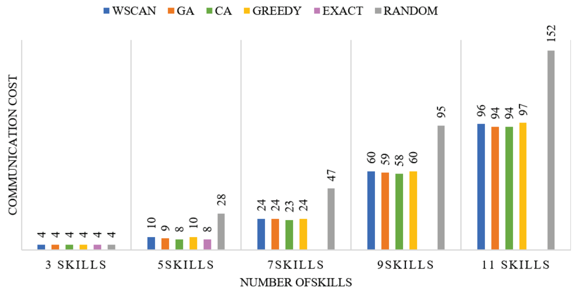

Figure 2: Comparison of the communication cost of a team of experts for various number of skills with different algorithms

and Random method. The experiments use the real data set of DBLP. Our experiments use 50K nodes from the DBLP dataset. For the application of WSCAN function, we use the sum of distance to calculate the weight between two experts.

We implemented CA, GA, Greedy Algorithm and Exact algorithm to find team of experts under the same condition as WSCAN. To have a baseline comparison, we developed random methods which always select the team of experts randomly from the set of the team which has the lowest communication cost. To test our algorithm on real networks, we use the DBLP[1] dataset. The basic concept of DBLP network is, when two authors publish any paper together, they will have the connection between them. We generate the 50K nodes of equal edge weight graph with 1.0 of weight on all edges. The SCAN requires threshold value to form structural similarity with neighborhood nodes.Therefore we tested the different number of skills to find them at most value as the threshold. From this experiment, we assign the threshold as 4.0 to find the nearest neighborhood.

The experiment always begins by calculating CC from CORE expert to the neighborhood. Therefore, we calculated the value of communication cost of the team of experts with required skills. The Figure 1 shows the comparison for the CC of a team for the required number of skills 5 with various algorithms. It shows approximately equal value with Greedy algorithms. However, with both CA and GA, WSCAN didn't perform well. Then we examine by varying the number of required skills for a specific project as shown in the Figure 2. However, we found that the result always follows the same findings as we saw in Figure 1. Importantly, the run time of the WSCAN was less than all the other algorithms executing on the same machine.

Conclusion and Future Work

In this paper, we examined the problem of finding a team of experts in a social network that covers the set of projects with specific skills while minimizing CC among team members. In the best case scenario, all skills of experts for a specific project lie within the first cluster and in the worst case scenario, most skills experts belongs to outliers with maximum CC. However, all the tested experiments with WSCAN fall into the threshold range and performed similar as Greedy algorithms while little worse than the CA and GA. Mainly, the run-time of WSCAN was better than all other compared algorithms. In future, we like to test TFP as hybrid of SCAN and CA to improve the performance as well as accuracy together. At the same time, to have the more realistic result, we like to test with personnel cost of experts and their workload.

References

Awal, G. K., and Bharadwaj, K. K. 2014. Team Formation in Social Networks based on Collective Intelligence – an Evolutionary Approach. *Appl Intell* 41(2):627–648.

Chertov, A.; Kobti, Z.; and Goodwin, S. D. 2010. Weighted scan for modeling cooperative group role dynamics. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, 17–22. IEEE.

Kargar, M.; An, A.; and Zihayat, M. 2012. Efficient Bi-objective Team Formation in Social Networks. In *ECML/PKDD*, 483–498.

Kargar, M., and An, A. 2011. Discovering top-k Teams of Experts with/without a Leader in Social Networks. In *CIKM*, 985–994.

K.Selvarajah, Moradian Zadeh, M., and Kobti, Z. 2017. A Knowledge-based Computational Algorithm for Discovering a Team of Experts in Social Networks.

Lappas, T.; Liu, L.; and Terzi, E. 2009. Finding a Team of Experts in Social Networks. In *KDD*, 467–476.

Reynolds, R. 1994. An Introduction to Cultural Algorithms. In *Proceedings of the third annual conference on evolutionary programming*.

Xu, X.; Yuruk, N.; Feng, Z.; and Schweiger, T. A. 2007. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 824–833. ACM.

[1] *http://dblp.uni-trier.de/xml/*