Research Journal of Applied Sciences, Engineering and Technology 7(7): 1294-1301, 2014
DOI:10.19026/rjaset.7.394
ISSN: 2040-7459; e-ISSN: 2040-7467
© 2014 Maxwell Scientific Publication Corp.
Submitted: April 12, 2013 Accepted: June 05, 2013 Published: February 20, 2014

Research Article

Mining of Network Communities by Spectral Characterization Using KD-tree

R. Renuga Devi and M. Hemalatha
Department of Computer Science, Karpagam University, Coimbatore, India

Abstract: Different methods and algorithms have been employed to carry out the task of community mining. Conversely, in the real world, many applications entail distributed and dynamically evolving networks, wherein resources and controls are not only decentralized, but also restructured frequently. This leads a problem of finding all communities from a given network, within that the links are dense, but between which they are sparse. This is referred as Network Community Mining Problem (NCMP). A network community contains a group of nodes connected based on certain relationships sometimes that refers to a special sort of network arrangement where the community mining is discovering all communities hidden in distributed networks based on their relevant local outlooks. To avoid the above mentioned problem, the existing work presented a novel model for characterizing network communities via introducing a stochastic process on networks and analyzing its dynamics based on the large deviation theory. By Using the fundamental properties of local mixing, then proposed an efficient implementation for that framework, called the LM (Network community mining based on Local Mixing properties) algorithm. There also some drawbacks are identified. Actual number of communities is estimated by using a recursive bisection approach, jointly with a predefined stopping criterion. The recursive bisection strategy does not optimize communication performance and the complexity of performing the partitioning. To solve these problems in proposed work a new community bipartition scheme is developed by using KD-Tree. Also, the stopping criterion is calculated automatically by efficiently determining the minimum Eigen-gap without explicitly computing eigenvalues. The experimental result shows that the proposed scheme is more effective and scalable when compared with the existing scheme.

Keywords: Community bipartition scheme, community mining, eigen-gap, eigen values, KD-tree, recursive bisection strategy

INTRODUCTION

A social network is a social structure made up of single person (or entire organizations) called" nodes," which are joined (connected) by one or more particular types of interdependency, such as friendship, kinship, based on common interest, or a few financial exchange. Social communities are provided by anytime anywhere to freely share knowledge, experiences, expert opinions, services and other several useful knowledge based resources. Techniques that can automatically discover such virtual communities will provide huge help in building and managing personalized and smart Web portals through analyzing and predicting the collective behaviors of users through mining their underlying community structures. In recent years, social network mining is a rising, thrilling area of research that has facing itself a long way to go, with the involvement of many research fields. The broad range of troubles and disputes in mining social networks requires powerful new systems and algorithms towards new boundaries in perceptive the stimulating experience of social networks. The beginning of online social networks has been one of the most stimulating events. Several popular online social networks such as Twitter, Face book and LinkedIn have become gradually more popular. Such social networks naturally contain an incredible amount of content and relation data which can be influenced for analysis.

Revising the dynamics of these social networks suits an interesting data mining charge. One of such applications is to perceive the up-and-coming communities. We suppose to utilize the methods of social network analysis and web mining to demonstrate the networks of the users and by this to determine the interest groups. The investigation consequences will then be utilized to build an automatic creation recommendation method that is based on the interest groups categorization method. In accessible method, network communities refer to clusters of vertices surrounded by which their concerning links are intense but among which they are light. A Network Community Mining Problem (or NCMP) is worried with the problem of discovering all such communities from a specified network. A large diversity of applications can be dignified as NCMPs, ranging from small or large

Corresponding Author: M. Hemalatha, Department of Computer Science, Karpagam University, Coimbatore, India

social and/or biological network analysis to recent web mining and investigating. Thus far many algorithms dealing with the NCMP have been expanded and most of them decrease into the categories of either optimization based or heuristic methods.

This study has exposed the association between network modularity and network's spectrum properties and suggested the perception of network's spectral signature. One can suppose a lot of significant information transmitted to community arrangement from a network's spectral signature, for instance the quality of modularity, the consistency and separatability of communities, the amount of communities and the hierarchical arrangement of communities. Based on the perception of spectral signature, this work has offered a theoretical framework for characterizing, analyzing and mining communities of a specified network through gathering its spectral signature and one of its meta-stable states. Utilizing the essential properties of meta-stability, that is locally identical and momentarily predetermined, a scalable accomplishment for this structure, called LM algorithm, has been suggested that is adjusted in the direction of realistic applications. Conversely there is main disadvantage of this technique is there is no best possible consequences with the recursive bisection strategy and as well the stopping criterion should be automatically calculated.

The contribution of the proposed system is as follows:

- Spectral signature quantity, to differentiate and analyze network communities and the utilizing the fundamental properties of meta-stability, that is nearby consistent and for the moment predetermined, a scalable completion for this structure, called LM algorithm.
- Methods for assembling an optimized KD-tree are explained. In a functioning, an optimized KD-tree procedure obtains input of a set of data points appropriate for large-scale computer revelation applications. The procedure separates the set of the data points into subsets of data points with nodes though generating hyper-planes.
- By computing the minimum Eigen-gap exclusive of unambiguously computing eigenvalues, regularly computing the stopping criterion.

LITERATURE REVIEW

Bo *et al*. (2007) proposed a new algorithm, called Finding and Extracting Communities (FEC), to mine signed social networks where both positive within-group relations and negative between-group relations are intense. FEC observed as mutually the sign and the density of relations as the clustering attributes, making it effective for not merely signed networks other than also conventional social networks collectively with only positive relations. Conversely, the associations between the entities of the Web are diverse and complex, which cannot be formed accurately just by positive links. Meenu and Rajeev (2011) developed a new algorithm BFC which utilizes statistical approach for community mining in Social networks. The algorithm earnings in breadth first way and incrementally take out communities from the Network. This algorithm is straightforward, quick and can be extent easily for large Social networks. However algorithm is not covenant with large and dynamic networks. Di *et al*. (2010) proposed GALS has been analyzed on both computer-generated and real-world networks and evaluated with some reasonable community mining algorithms. Deng *et al*. (2005) methodically examined the problem of mining hidden communities on heterogeneous social networks. Seeing as it is difficult for a user to understand the whole representation of numerous social networks, one may be unsure how a user is able to pretence high-quality queries.

Jiyang *et al*. (2010) Meerkat utilized recently urbanized layout, mining and event detection algorithms currently unavailable elsewhere and will keep on to contain cutting edge algorithms as they are enlarged. These new expansions will straight away exist to clients through Meerkat's web based user-interface, signifying there will be no necessitate running an update installer to facilitate taking delivery of new features when they are obtainable. Tushar *et al*. (2011) suggested CRA (Clustering Re-clustering Algorithm) which works in two phases. The first phase is supported on Breadth First Search algorithm that forms clusters on the foundation of the positive links only. The second phase acquires the output of first phase as its input and created clusters on the foundation of a robust criteria termed as contribution level. Bo *et al*. (2009) has addressed the problem of mining communities from a dispersed and dynamic network. It changes from the previous ones in that here introduced the concept of self-organizing agent networks and presented an Autonomy-Oriented Computing (AOC) approach to distributed and incremental mining of network communities. Aurel and Narsingh (2005) offered a greedy, best-first algorithm for the seed-growth version of community-mining. Opening with a set of seed nodes, this algorithm produced a community by frequently selecting some nodes from community's neighborhood and insertion them in the community.

Pranay and Malik (2012) expanded a spectral approach improved with iterative optimization. They have used algorithms to revise both communities and structural balance. David *et al*. (2011) noticed in extracting Direct Antagonistic Communities (DACs) contained by a rich trust network involving trusts and distrusts. Every DAC is created by two sub-communities with trust relationships among members of each sub-community but disbelieve relationships across the sub-communities. From these surveys there were several methods developed for the network community problem and also discussed. However, structuring communities within a social network assists to decrease the number of users that the matching system needs to consider and assists to overcome other difficulties from those social networks suffer, such as the nonappearance of user performance information about a new user. Also the main drawback and need of new system is that they depend on the recursive bisection strategy that will increase the complexity of the system.

SPECTRAL SIGNATURE BASED PARTITIONING

A Social Network contains of social structure of nodes fixed mutually with one or more type of association such as companionship, dislike, operate, economic replace, etc. A social network can be described as a graph *G* = (*V*, *E*), anywhere $V = \{V1, V2, V3.,,,,, V_n\}$ the set of vertices and E is the set of edges concerning pairs of vertices. Each edge symbolizes the social relationships between two nodes representing individuals. They have varied and multi-relational dataset symbolized by graphs. Normally these graphs are very great and both nodes and associates have attributes. Social networks require not to social in background. There are many real world illustrations of economic, biological, technological and business social networks. Let X = {X_t; t≥0} denote the agent positions and P {X_t = i, 1≤I≤n} be the probability that the agent hits the vertex i after precise t steps. For i_t∈v, we have:

$$P\{X_t = i_t \mid X_0 = i_0, X_1 = i_1, \dots X_{t-1} = i_{t-1}\}$$

$$= p\{X_t = i \mid X_{t-1} = i_{t-1}\}$$

At the present let us believe the dynamics of the on top of stochastic model. For a network with a community arrangement, its corresponding Markov chain should enclose some local mixing conditions depending on its total number of communities K. Previous to the chain attains its global mixing state, denoted as s_1; it should go during a sequence of local mixing states initial along the time measurement, denoted as …; s_3; s_2; s_1. In each of the local mixing states, vertices inside the same communities have just about identical row allocations. In that order, in a local mixing state, we should examine a meta-stable transition matrix. Specifically, the random walk will steadily continue in a meta-stable state for a period of time with a probability value equal to 1, according to the large-deviation theory. The social network is shown in Fig. 1.

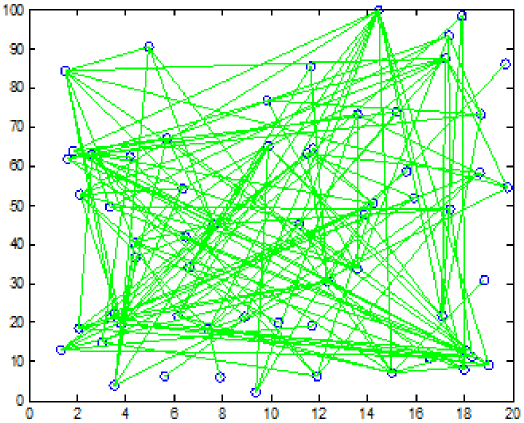

Fig. 1: Social network

In this study we are providing the input as dolphins' social network dataset. By characterizing the community structures, we have discovered the connection between the community structure of a network and the spectrum of its matching Markov generator. For any given network, devoid of clustering it by means of a particular algorithm, one can distinguish and analyze its communities by answering some difficulties transmitted to its topological structures during observing and inferring its spectral signature. We have founded a general framework for characterizing and analyzing communities for a given network. Now, we can additional enlarge it to a more general structure for mining communities with a hierarchical structure for a given network by understanding its spectral signature and one of its meta-stable states. Its foremost steps are summarized as follows:

```
Step 1: $Construct\ Transistion\ mat\ P$
Step 2: $Calculate\ spectrum\ of\ I - P(\lambda_1, \leq \cdots \leq \lambda n$
Step 3: $Calculate\ spectralsignature\ (CQ_1 \ldots\ldots CQ_{n-1})$
Step 4: $Find\ \min(CQ_k)\ 1 < k < n$
Step 5: $CQ_k = \frac{\lambda_K}{\lambda_{K+1}}$
Step 6: $K = \arg\min_k CQ_k$
Step 7: $Find\ similarity\left(P^{1/\lambda_{K+1}}\right)$
Step 8: $If\ (CQ_k < th)$
Step 9: $Mine\ K\ communities\ from\ \left(P^{1/\lambda_{K+1}}\right)$
Step 10: $Else\ goto\ step\ 2$
```

The LM algorithm: In every bipartition, determining the node with greatest degree needs O (n) time; computing an Ordering Time Distribution (OTD) needs O ((n + m) /λ_{K+1}) time; in-between an OTD by its mean needs O (n) time; and computing Q-value requires

$O(m)$ time (in functioning, Q-value is incrementally calculated, which obtains time much less than O (m). Consequently, the worst time of LM (Local Mixing) is equivalent to the total time necessitated by the first bipartition multiplied by the total number of recursively callings. For decision out all K communities, precisely 2 K - 1 recursive callings are entailed. Consequently, the worst time of LM is surrounded by O [$K^{(n+m)}$ /λ_{K+1}] Recall that λ_{K+1} is the thrashing time of the K^th^ meta-stable state and will be determined by the consistency of communities slightly than the scale of network. The algorithm is given below:

```
Step 1: $Select\ attractor\ (column\ selection)$
Step 2: $C = \text{argmax}_i\, \pi_i = \text{argmax}_i \left\{\frac{d_i}{\sum_k d_k}\right\} =$
        $\text{argmax}_i\{d_i\}$
Step 3: $Calculate\ OTD\ p_{i,c}^{(t)} = \frac{1}{d_i}\sum_{\langle i,j\rangle\in E} A_{ij}\cdot p_{i,c}^{(t-1)}$
Step 4: $Bipartition\ of\ p_{i,\ c}^{(t)}$
Step 5: $Stopping\ criterion\ Q = \sum_i e_{ij} - a_i^2$
Step 6: $If\ stopping\ criterion\ is\ satisfied$
Step 7: $Return\ result$
Step 8: $Else\ goto\ step\ 3$
```

The stopping criterion of the recursive bisections is: the local bipartition in difficulty will disintegrate the excellence of the global partition previously attained. Q-Function proposed by Newman is selected to assess the quality of partitions, which is defined as:

$$Q = \sum_i e_{ij} - a_i^2.$$

where e_{ij} indicates the fraction of all weighted edges in networks that link the vertices in community i to those in community j and:

$$a_i = \sum_j e_{ij}.$$

It is believed that improved partitions of a known network will be with greater Q-values. The LM algorithm result is shown in Fig. 2.

Finally, the foundations following spectral methods and the LM are entirely diverse from the perspective of practical computation. Spectral schemes partition graphs into K communities by first manipulative the negligible K eigenvectors of the Laplacian matrices, which normally takes O (n^2) time or ($K\cdot\frac{n+m}{\lambda_2-\lambda_2}$) for sparse networks by using some spectral systems, such as the Lanczos methods and then clustering n K-dimension vectors into K clusters with the K means method, which will cost O (InK^2) time, where I denotes the number of iterations required by K-means to converge. On the other hand, the LM discerns all K communities by first manipulating the OTD, single column distribution of a transition matrix, within a time of O ((n + m) /λ_{K+2}) and then divides it by its means with a time of O (n). The clustered results are shown in Fig. 3.

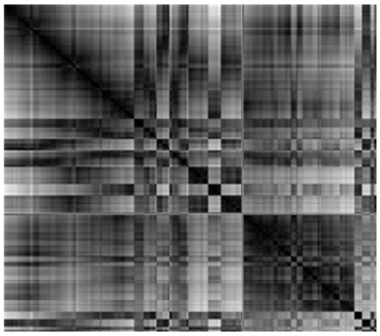

Fig. 2: LM algorithm

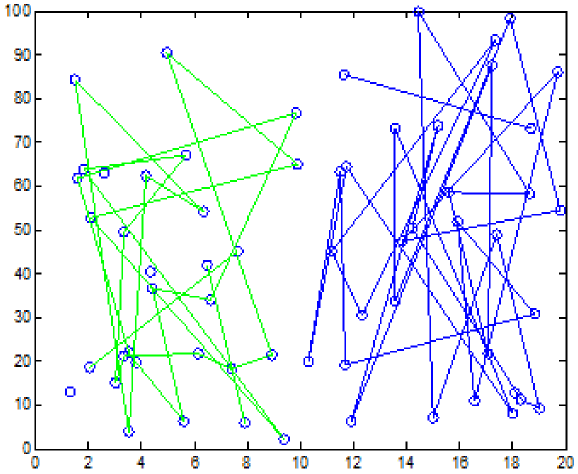

Fig. 3: Clustered results

KD-tree based partitioning: A KD-Tree is a multi-dimensional binary search tree generally assumed for organizing spatial data. It is helpful in numerous troubles like graph partitioning, *n*-body replications and database applications. KD-Tree is employed to optimize the detection of the closest centroid for each pattern. The vital initiative is to group patterns with related coordinates to perform group assignments, whenever possible, devoid of the explicit distance computations for each of the patterns. Through a pre-processing step the input patterns are prearranged in a KD-Tree. At each of iteration the tree is traversed and the patterns are assigned to their closest centroid. Construction and traversal of the tree are described in the next two sections. The stopping criterion is computed based on the Eigen gap.

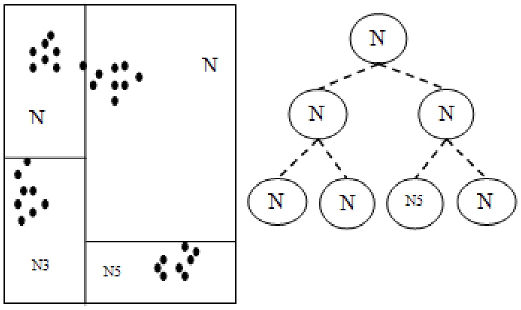

Fig. 4: An example of a KD-tree for a community set in a two dimensional space

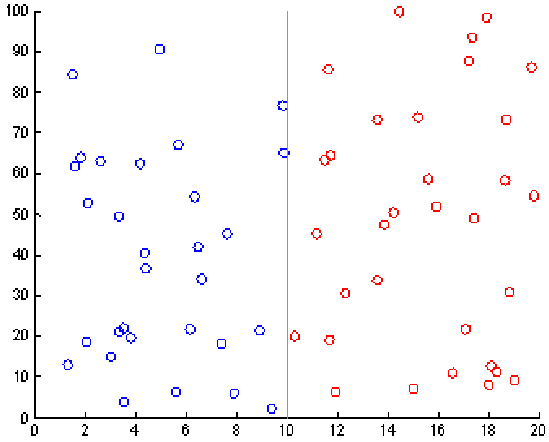

Fig. 5: Clustering result using LM with KD-tree

KD-tree construction: Every node of the tree is connected with a set of patterns. The root node of the tree is connected with all input patterns and a binary partition is recursively performed. At every node the community is partitioned in approximately two equal sized sets, which are dispensed to the left and right child nodes. The partitioning procedure is repeated until the full tree is built and each leaf node is associated to a single community pattern. Where the stopping criterion is fixed automatically based on the Eigen gap. To implement clustering constancy, we will maximize the eigengap $\Delta_K = \lambda_K (P(\theta)) - \lambda_{K+1}(P(\theta))$. To stimulate this preference, one be able to evoke the statement that a great eigengap in *P* constructs the subspace. A minimum leaf size (>1) can also be described which directs to an unfinished tree. The partitioning operation is achieved by choosing a dimension and a plot value. Community points are allocated to the left child if the organize in that dimension is smaller than the plot value, or else to the right child. The dimension for partitioning can be chosen with a threshold value. This way the same dimension is utilized to split the community sets of internal nodes which are at the same tree level.

The mean value can be the median or the midpoint. The median assurances equivalent sized partitions through some computational cost. The calculation of the midpoint is faster but may initiate imbalanced trees. An example of a KD-Tree for a community set in a two dimensional space is shown in Fig. 4.

Typical algorithms construct KD-trees by partitioning point sets. Partitioning stops after Eigen gap value obtain, with each point in its personal leaf cell. Every other KD-tree construction algorithms insert points incrementally and divide the suitable cell, though such trees can turn out to be seriously disturbed. The clustering result is shown in Fig. 5:

The cutting planes along any path from the root to another node describe a unique box-shaped region of space and every subsequent plane cuts this box into two boxes. Each box-shaped region is described by 2k planes, where *k* is the number of dimensions. Certainly, the `KD' in *KD*-tree is short for *k*-dimensional tree. In any exploration, performed using a KD-tree, we keep up the present region defined by the intersection of these half-spaces as we move down the tree. The step by procedure is given below:

Step 1: $Build\ the\ k-d\ Tree\ (ReferencePts, [\,]);$
Step 2: $Neighbour\ poini\ \left(D-dimensional\ Euclidean, 2-norm, distance\right)$
Step 3: $Initial\ value\ of\ centroid$
Step 4: $Calculate\ point\ to\ cluster\ centroid\ distances\ Euclidean\ d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$
Step 5: $Cl = clusters\ d(p,q)$
Step 6: $Plot\ all\ the\ points\ in\ the\ k-D\ tree$
Step 7: $Calculate\ eigen\_gap\ \Delta_K = \lambda_K\ (P(\theta)) - \lambda_{K+1}(P(\theta))$
Step 8: $If\ large\ eigengap\ for\ P(\theta) is\ met$
Step 9: $End\ process$
Step 10: $Else$
Step 11: $Goto\ step\ 2$

So as to speed up the pre-processing step, the construction of the initial KD-Tree can be distributed between the social networks. However, a sequential construction of a KD-Tree is normally achieved formerly for a given data set and can be utilized many times for sequential and for parallel algorithm executions. On behalf of especially large data sets and for essentially distributed data sets the most suitable parallel approach should be accepted. In any case of how the initial (global) tree is built, each process accepts a node of the level *log* (*p*), which describes the local data partition on which a complete local KD-Tree is built.

EXPERIMENTAL RESULTS AND DISCUSSION

There are two input dataset we have taken that can be referred as football, dolphin and IMDB. The football.gml file contains the network of American football games between Division IA colleges during regular season fall 2000, as used by M. Girvan and M. Newman. The network nodes have values that point out to which conferences they belong. If you make use of these data, please cite M. Girvan and M. E. J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99, 7821-7826 (2002). The IMDB is Co-appearance of actors in the same movie. In this study the input are dolphin dataset which is represented in Fig. 6. The file dolphins.gml contains an undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand, as compiled by Lusseau *et al*. (2003). These results of clustering are compared with the existing system based on some parameters. With the aim of analyzing and comparing the performance of the above talked about algorithms, the metrics like clustering accuracy, Time taken, True Positive Rate, False Positive rate were used.

This proposed method is evaluated by the following manner for this we have taken five parameters such as precision, recall, time taken, accuracy and F-measure. The precision rate is only considering the topmost results. The precision of proposed system would consequently point out the proportion of total number of properly classified positive states to the total number of predicted positive states and recall would indicate the proportion of correctly classified positive states to the total number of actual positive states. Thus, a high precision and high recall are desirable for browsing behavior detecting system. The F-measure is then defined as the weighted harmonic mean of precision and recall to address the difficulty which may be present in credit card fraud detection scenario. The accuracy rate measurement of a system is the rate of nearness of capacity of a quantity to that quantity's actual (true) value.

Recall rate: Figure 7 represents the recall rate comparison among the two different algorithms. This graph shows the recall rate of two different clustering methods based on two parameters of recall and number of network dataset. From the graph we can see that, when the number of number of Network datasets is improved the recall rate also improved in proposed system which is represented as blue line, but when the number of number of Network datasets is improved the recall rate is reduced in two existing systems which are represented as redline and green line. From this graph we can say that the recall rate of LM with KD-Tree is increased which will be the best for the community mining.

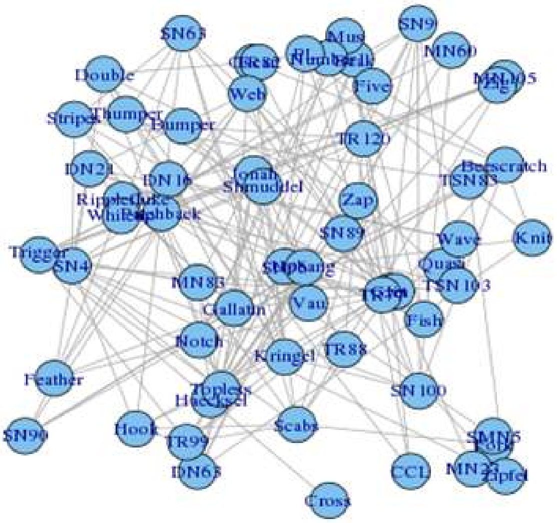

Fig. 6: Dolphin social network

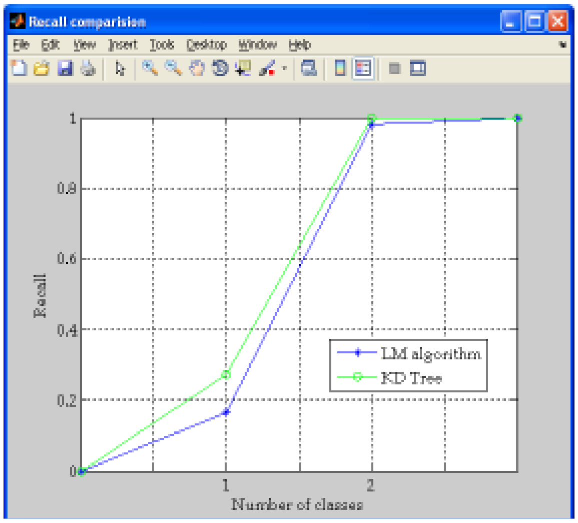

Fig. 7: Recall rate comparison

Precision rate: Figure 8 represents the precision rate comparison among the two different algorithms. This graph shows the precision rate of two different clustering methods based on two parameters of precision and number of network dataset. From the graph we can see that, when the number of number of Network datasets is improved the precision rate also improved in proposed system which is represented as blue line, but when the number of number of Network datasets is improved the precision rate is reduced in two existing systems which are represented as redline and green line. From this graph we can say that the precision rate of LM with KD-Tree is increased which will be the best for the community mining.

True positive and false positive rate: The following Fig. 9 shows mainly the true positive and false positive

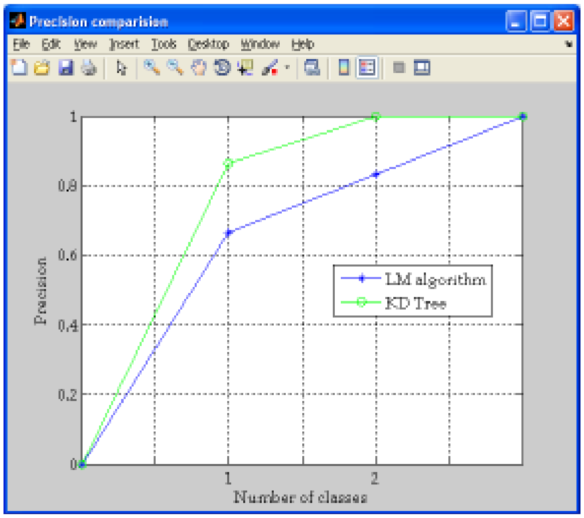

Fig. 8: Precision rate comparison

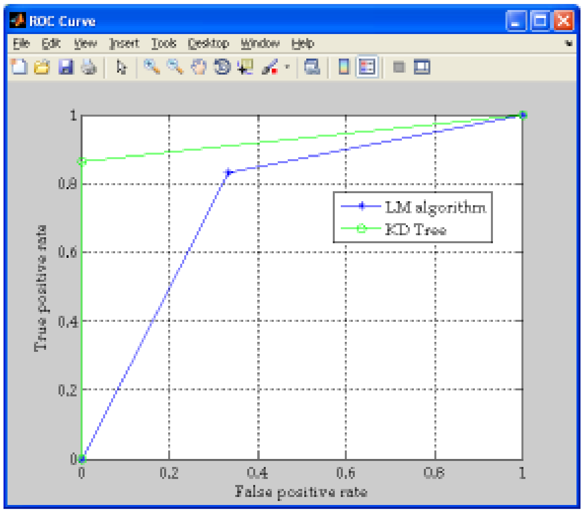

Fig. 9: True positive and false positive rate

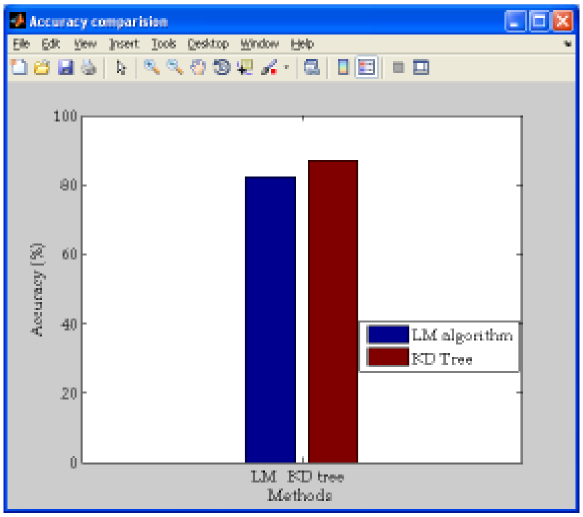

Fig. 10: Accuracy rate comparision

rate of KD-Tree and LM algorithm based on two parameters of true positive and false positive and number of data. From the graph we can see that, when the number of data is improved the true positive and false positive rate also improved in KD algorithm but when the number of number of data is improved the true positive and false positive rate is reduced in LM algorithm. From this graph we can say that the true positive and false positive rate of KD-Tree is increased which will be the best one. The true positive and false positive rate of proposed system 0.83 and the true positive and false positive rate of existing system are 0.82. Thus the proposed system is 0.01 better than the existing system.

Accuracy rate: Figure 10 shows mainly the accuracy rate of KD-Tree and LM algorithm based on two parameters of accuracy and number of Data. From the graph we can see that, when the number of data is improved the accuracy rate also improved in KD algorithm but when the number of number of Data is improved the accuracy rate is reduced in LM algorithm. From this graph we can say that the accuracy rate of KD-Tree is increased which will be the best one. The accuracy rate of proposed system 90% and the accuracy rate of existing system is 80%.

CONCLUSION

In this study we have done two main contributions. First is that the KD-Tree based clustering that produce good clustering results. Then, we established a new criterion to avoid the complexity of clustering process. The criterion optimizes the quality of the target clustering, while constraining the parameters Δk as little as possible in the process. This is accomplished by choosing the gap as the clustering quality and by adding the eigengap as regularization term. The best partition-based speed-up scheme we experienced is a bidirectional search, accelerated in both directions with KD-tree. The experimental results show that the proposed system is more effective and scalable in terms of accuracy, recall rate, precision rate and true positive and false positive rates when compared with the existing method. Thus the proposed system accuracy is 10% better than the existing system.

REFERENCES

Aurel, C. and D. Narsingh, 2005. A greedy community-mining algorithm based on clustering coefficient. Congr. Numer., 172: 161-176.

Bo, Y., K.C. William and L. Jiming, 2007. Community mining from signed social networks. IEEE T. Knowl. Data En., 19(10).

Bo, Y., L. Jiming and L. Dayou, 2009. An autonomy-oriented computing approach to community mining in distributed and dynamic networks. Auton. Agent Multi-Agent Syst., Springer Science+Business Media, LLC, DOI: 10.1007/s10458-009-9080-2.

David, L., S. Didi, Z. Kuan and L. Ee-Peng, 2011. Mining direct antagonistic communities in explicit trust networks. Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM '11), pp: 1013-1018.

Deng, C., S. Zheng, H. Xiaofei, Y. Xifeng and H. Jiawei, 2005. Mining hidden community in heterogeneous social networks. Proceedings of the 3rd International Workshop on Link Discovery (LinkKDD '05), pp: 58-65.

Di, J., H. Dongxiao, L. Dayou and B. Carlos, 2010. Genetic algorithm with local search for community mining in complex networks. Proceeding of the 22nd International Conference on Tools with Artificial Intelligence.

Jiyang, C., F. Justin, G. Randy, R. Reihaneh, S. Farzad, T. Mansoureh, V. Eric and Z. Osmar, 2010. Meerkat: Community mining with dynamic social networks. Proceeding of the IEEE International Conference.

Lusseau, D., K. Schneider, O.J. Boisseau, P. Haase, E. Slooten and S.M. Dawson, 2003. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behav. Ecol. Sociobiol., 54: 396-405.

Meenu, G. and Y. Rajeev, 2011. Statistical approach of social network in community mining. Int. J. Inform. Technol. Knowl. Manage., 4(1): 43-46.

Pranay, A. and M.I. Malik, 2012. Communities and balance in signed networks: A spectral approach. Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM, 2012), pp: 235-242.

Tushar, S., C. Ankit and P.K. Singh, 2011. Community mining in signed social networks: An automated approach. Proceeding of the International Conference on Computer Engineering and Applications (IPCSIT), Vol. 2.