Journal of Engineering and Applied Sciences 12 (21): 5452-5456, 2017
ISSN: 1816-949X
© Medwell Journals, 2017

Minimum Spanning Tree Based Community Detection for Biological Data Analysis

Maria Joseph and Sreeja Ashok
Department of Computer Science and IT, School of Arts and Sciences,
Amrita University, Kochi, India

Abstract: Bioinformatics is an important area in which computing techniques can be applied for efficient data analysis and for mining meaningful patterns. The organization, analysis and interpretation of data are the major challenges faced by biologists when dealing with large amount of heterogeneous and complex data. Unsupervised learning techniques are widely used for data reduction and pattern extraction for in-depth analysis and knowledge discovery. Graph clustering is a more suitable approach, since, the interactions of the biological components can be effectively demonstrated by networks. The complexity of the graphs can be reduced by extracting highly significant edges instead of focusing on all edges that represents the association between data objects. This study compares and evaluates the significance of different Minimum Spanning Tree algorithms (MST) as a preprocessing step for community detections in biological data. Multiple algorithms were reviewed and compared and the process performance is compared with benchmark community detection algorithms.

Key words: Clustering, biological data, minimum spanning tree, performance evaluation, preprocessing step, compares and evaluates

INTRODUCTION

Due to advancement of technology, large volumes of biological data are generated and it is available in various sources. These terra bytes of information have to be managed efficiently to make it useful. Biological data is also considered as highly complex due its heterogeneous nature. It includes information from different knowledge sources namely molecular and cell biology, genetics, structural biology, pharmacology, physiology, etc. Each of these domains has its own entity types, terminology and data needs. In addition, even though the results generated by the experimental procedure are related they are not identical. Several new dimensions are being given to biological data types by new analytical procedures and scientific advancements. So, the range of types of data varies from sequences to 3-dim structures, images, graph structures, data table, semi structured and unstructured text. Major information is easily available in public reference data bases, specialized private data sources and study of scientific research.

In order to retrieve efficient patterns from these biological data sources, clustering techniques are more appropriate and relevant (Abhishek, 2006). Process of clustering means the organization of data objects to a number of groups whose group members are similar in some way. Thus, clustering is a group of "similar" objects within a group and "dissimilar" from other cluster's objects (Jain, 2010; Tzanis *et al*., 2005; Ashok and Judy, 2015; Li and Zhu, 2013; Hruz *et al*., 2013). The problem of can be defined as: given a matrix d as input of order m×n where m is the count of genes and n is the sample count, algorithm of clustering processes the input and make a group of similar genes as a collection based on the similarity measures. Output will be a cluster set c = {ci, 1≤I≤k} where k is the cluster's count and ci's are the clusters, i.e., collection of genes.

Graph clustering techniques are widely used for biological data analysis for extracting meaningful patterns from large amount of heterogeneous information. Various graph clustering algorithms based on fundamental graph theories and algorithms have been emerging recently. These strategies have been providing efficient means to process and analyse complex and large quantities of information (Ashok and Judy, 2015, 2016; Sonumol *et al*., 2015; Speer *et al*., 2004; Schaeffer, 2007). The complexity of the graphs can be reduced by integrating a pre-processing step to capture the most significant information from the graph. This can be efficiently achieved by extracting a graph's minimum spanning tree. It's based on the principle that a removal of edges

Corresponding Author: Maria Joseph, Department of Computer Science and IT, School of Arts and Sciences, Amrita University, Kochi, India

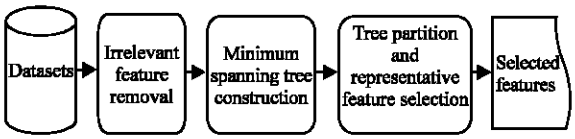

Fig. 1: MST process flow

into n distinct groups. In addition, the overhead of calculations based on graph and the complexities of dependencies related to geometrical shapes of the clusters can be avoided with the MST representation and it is one of the easiest and best-analysed optimization problems. Utilizing the minimum spanning tree has various advantages. It is effective and easy to compute and it directly catches similarity by amplifying the aggregated weight of the subsequent minimum spanning tree of the graph. If the edge weights in a graph denote distance measure, then the aggregate of the subsequent minimum spanning tree graph weight should be minimized. Moreover, cutting the minimum spanning tree edges partitions the graph into components based on the similarity or the distance measures used.

Given an undirected, connected graph of m weighted edges in which an o(m)-time depth-first search is used to find an arbitrary spanning tree, i.e., a tree which connects all vertices of g and uses only edges of g. The flow chart given in Fig. 1 illustrates the process flow of minimum spanning tree. First step in the process is to 7 collect relevant information from the available knowledge sources. Then the useful features will be extracted based on the significance of the problem under study and a tree will be constructed. This tree will be partitioned and only the representative feature will be selected. All the redundant features will be eliminated after evaluation in each iteration. The features finally extracted will reflect the required pattern of the particular biological area. Figure 1 represents the flow chart of MST process.

Literature review: In order to process the highly complex multi-dimensional data originated from microarrays, effective and efficient data algorithms are needed. When the information related to the connection among biological molecules is not available clustering huge biological data becomes a major challenge. A heuristic algorithm based on minimum spanning tree, B-MST is proposed by Pirim *et al.* (2015) to cluster gene expression data provided by an innovative objective function: the Tightness and Separation Index (TSI). The algorithm makes use of an objective function, TSI as a new tool to evaluate tightness and distinction concurrently, while taking into consideration the transitive distances on a binary graph to originate biologically worthy clusters (Pirim *et al.*, 2015). Elsayad developed an enhancement of CLUMP algorithm known as iCLUMP (improved Clustering algorithm through MST in Parallel) by making use of the data structure called cover tree. The result of execution showed that iCLUMP is more efficient than CLUMP in terms of complexity and runtime. Minimum spanning tree is also used for image segmentation in a study conducted by Chaudhari and Shah (2011). It is a critical and prominent area in image analysis and in the field of machine vision. In this study, minimum spanning tree based clustering approach for unsupervised object based segmentation is applied in the area of tumour. Prim's algorithm is applied for the segmentation of image. It was observed that the algorithm research faster when compared to standard algorithm.

Olman *et al.* (2009) developed a parallel clustering algorithm for pattern recognition and the performance was evaluated on a graph representation of the data. The algorithm identified clusters by finding of densely intra-connected sub graphs. The result indicated that the program was able to handle multi dimensional datasets more accurately (Olman *et al.*, 2009). Several data management challenges of complex biological data are being reviewed by Topaloglou *et al.* (2004). An OpenMP algorithm is proposed by Chapman and Kalyanaraman (2011) for clustering biological graphs by making use of adjacency lists, hash tables and union find data structures in parallel.

MATERIALS AND METHODS

Comparative analysis of Minimum Spanning Tree (MST) algorithms: In this study, different prominently used algorithms for finding minimum spanning tree namely Boruvka's algorithm, Prim's algorithm, Kruskal's algorithm, Reverse-delete algorithm and Linear-time randomized algorithm are compared and evaluated.

Boruvka's algorithm: Boruvka's algorithm is a greedy algorithm used to find a minimum spanning tree of a graph where all the weights of edges are different. It will first check every vertex and include the cheapest edge from that vertex to another in the graph. It will not consider the already included edges and will continue combining these clusters in a like pattern until a tree that is spanning all vertices is created. The Boruvka's algorithm works in the following manner:

- Input a connected, weighted graph
- All vertices needed to be initialized as individual component
- At first, initialize the MST as empty
- While there is more than one component, find the closest weight edge component to any other component and add it to the MST for each component
- At last the MST will be returned

Table 1: Application of minimum spanning tree algorithms in biological data analysis

Algorithms	Applications
Boruvkass	Study of genomic alterations in the area of cancer research
	Progression reconstruction from unsynchronized biological data
	Analysis of genetic data for generating patterns for disease prediction
	Creation of fast-shared memory algorithms that calculates the minimum spanning forest of sparse graphs
	Generating a cellular hierarchy from high-dimensional cytometry data by making use of SPADE
Prim's	Construction of polygenetic trees from networks Tomas
	Domain identification by clustering sequence alignments
	Analysis of genetic data for generating patterns for disease prediction
	Microarray's data analysis
	To relate evolutionary trees recreated from gene frequencies in blood groups
Kruskal's	Analysis of genetic data for generating patterns for disease prediction.
	Microarray's data analysis
	Filtering of biological networks

Prim's algorithm: Prim's algorithm is a greedy algorithm to find a minimum spanning tree. This algorithm will generate an edge subset that creates a tree which contains all vertices where edge weight's total is minimized. It will start from an arbitrary vertex and then continues by generating the tree with one at a time. It will add the cheapest connection among the tree and another vertex. The steps to be followed are as follows:

- Initially the MST will be empty
- There will be 2 sets of vertices. The vertices which are already included in the MST will be in the first set
- All other vertices will be in the other set. In each step, both of the sets will be considered and the minimum weighted edge will be picked and added to the MST

Kruskal's algorithm: Kruskal's algorithm is a greedy algorithm that search out a possible lowest weight edge that connects any two trees in the forest. It search out a minimum spanning tree by including arcs of increasing cost at each step, meaning that it search out an edge subset that generates a tree which contains all vertex where all of the edge weights in total of the tree is at the minimum. If it is a disconnected graph, then it searches out a minimum spanning forest. This algorithm follows a number of steps to execute:

- It will first sort all the edges in non-decreasing order
- Then the smallest edge will be picked up ensuring that there is no cycle is formed
- Repeat the above steps for v-1 edges

Reverse-delete algorithm: The reverse-delete algorithm is a graph algorithm that retrieves a minimum spanning tree of a graph that is connected with weights assigned to the edges. It is the reverse process of Kruskal's algorithm where the algorithm at first starts with the original graph and deletes edges from it. The main objective is to delete edge until the deletion does not lead to disconnection of graph. Reverse delete algorithm has less execution time among all algorithms.

Table 2: Comparison of different MST algorithms

Algorithms	Advantages
Boruvka's	Simple and efficient; oldest MST algorithm
	Union-find data structure permits fast implementation
	Running time is conservative upper bound of O(E logV) It is easily parallelized since the selection of cheapest outgoing edge for each node is independent of the selection made by other nodes
	Used in faster randomized algorithms that executes in linear time O(E)
Prim's	Simple and efficient; oldest MST algorithm
	Union-find data structure permits fast implementation
	Running time is conservative upper bound of O(E logV)
	It is easily parallelized since the selection of cheapest outgoing edge for each node is independent of the selection made by other nodes
	Used in faster randomized algorithms that executes in linear time O(E)
Kruskal's	Works for connected and weighted graphs
	Performs better for sparse graphs since it uses simpler data structures
	Solutions are built from the cheapest edge and then selecting next cheapest edge without forming a cycle
	Provides better performance if the edges are sorted in linear time
	Kruskal (sort)-cost dominates (E logE)
	Kruskal (partial sort)-cost depends on longest edge (E+X logV)

Linear-time randomized algorithm: It is a randomized algorithm to compute a weighted graph's minimum spanning forest that has no isolated vertices. This algorithm depends on methodology which is a combination of Boruvka's algorithm and an algorithm for analyzing a minimum spanning tree in linear time. Table 1 and 2 represent the summary of various biological applications of MST algorithms (Xu *et al*., 2001; Hepsiba, 2014; Qiu *et al*., 2011a, b; Johnson and Metaxas, 1992; Katajainen and Nevalainen, 1983; Xu *et al*., 2001; Wang *et al*., 2013; Sreeja and Krishnakumar, 2017) and Table 2 compares each algorithm with respect to different parameters.

RESULTS AND DICUSSION

The three main algorithms namely Prim's, Boruvka's and Kruskal's algorithm were analyzed using yeast dataset taken from UCI repository a research data repository for machine learning and intelligent systems. We have extracted a total of 503 data instances with

Table 3: Comparison of different MST algorithms on yeast dataset

Algorithms	Number of stages needed	Time needed to find the MST	Optimum weight (Sum of weights of the arcs)
Prim	502	22.15	29.72425
Kruskal	100469	39.10	29.72425
Boruvka	1	97.57	29.72425

Table 4: Performance comparison using modularity measure. Evaluation done using significant path identification with MST and without MST using different benchmark community detection algorithms

Community detection with MST				
Community detection algorithm	Prim	Kruskal	Boruvka	Community detection without MST
Fast greedy	0.01116596	0.7193676	0.7193676	0.01116596
Leading Eigenvector	0.00019832	0.7595891	0.6905128	0.00019832
Walktrap	0.05954813	0.6207346	0.7034671	0.05954813

3 classes for study. Each instance is explained by 9 attributes. The analysis was done using R programming language. Table 3 represents the comparison of three algorithms for the yeast data with respect to number of stages, time taken for execution and total sum of the weights of the arcs. The generated MST based on each algorithm is then given as input for community detection using three algorithms and the performance is compared using modularity value.

The MSTs generated are then given as input for cluster identification using standard benchmark algorithms like fast greedy, leading eigen vector and walktrap. The performance of the clustering results is evaluated using modularity value. Table 4 shows the performance comparison of clustering results with significant path identification using MST and without identifying the significant paths using MST. The results clearly show that the performance is better for communities derived using MST than using the fully connected graphs. Better performance is achieved for communities derived using leading eigenvector algorithm and using Prim's MST.

CONCLUSION

The exponent increase in the rate of data and the multifaceted nature and abundance of the data content is a major challenge in mining meaningful patterns and extracting valuable inputs for better decision making. Graph-based clustering techniques are extremely useful because many real world problems have a natural graph representation. Characteristic properties of graph models such as patterns of associations can influence the hidden framework's behavior and function. Digging for such patterns can provide insights into the root cause of the diseases. Such experiences can empower us to form better diagnosis and effective treatment plans. Minimum Spanning Tree (MST) algorithms are more efficient for producing the optimum cost tree of the dataset. Moreover identification of MSTs reduces the complexity of data representation and analysis. Community detection after complexity reduction increases the performance of the clustered outputs with respect to modularity measures.

ACKNOWLEDGEMENT

This research is supported by the DST Funded Project, (SR/CSI/81/2011) under Cognitive Science Research Initiative in the Department of Computer Science, School of Arts and Sciences, Amrita Vishwa Vidyapeetham, Kochi.

REFERENCES

Abhishek, K., 2006. Clustering genes and inferring gene regulatory networks using multiple information sources. Ph.D Thesis, Indian Institute of Technology Kanpur, Kanpur, India.

Ashok, S. and M. Judy, 2016. Exploring key gene interactions using particle swarm optimization. Intl. J. Pharma Bio Sci., 7: 734-741.

Ashok, S. and M.V. Judy, 2015. A novel iterative partitioning approach for building prime clusters. Intl. J. Adv. Intell. Paradigms, 7: 313-325.

Chapman, T. and A. Kalyanaraman, 2011. An OpenMP algorithm and implementation for clustering biological graphs. Proceedings of the 1st Workshop on Irregular Applications: Architectures and Algorithms, November 13, 2011, ACM, New York, USA., ISBN:978-1-4503-1121-2, pp: 3-10.

Chaudhari, D. and V. Shah, 2015. MRI brain image segmentation using MST. Intl. J. Adv. Res. Electr. Electron. Instrumentation Eng., 4: 5386-5390.

Hruz, T., M. Wyss, C. Lucas, O. Laule and V.P. Rohr *et al.*, 2013. A multilevel gamma-clustering layout algorithm for visualization of biological networks. Adv. Bioinf., 2013: 1-10.

Jain, A.K., 2010. Data clustering: 50 years beyond K-means. Pattern Recogn. Lett., 31: 651-666.

Johnson, D.B. and P. Metaxas, 1992. A parallel algorithm for computing minimum spanning trees. Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms and Architectures, June 29-July 1, 1992, ACM, San Diego, California, USA., ISBN:0-89791-483-X, pp: 363-372.

Katajainen, J. and O. Nevalainen, 1983. An alternative for the implementation of kruskal's minimal spanning tree algorithm. Sci. Comput. Program., 3: 205-216.

Li, X. and F. Zhu, 2013. On clustering algorithms for biological data. Eng., 5: 549-552.

Olman, V., F. Mao, H. Wu and Y. Xu, 2009. Parallel clustering algorithm for large data sets with applications in bioinformatics. IEEE. ACM. Trans. Comput. Biol. Bioinf., 6: 344-352.

Pirim, H., B. Eksioglu and A.D. Perkins, 2015. Clustering high throughput biological data with B-MST, a minimum spanning tree based heuristic. Comput. Biol. Med., 62: 94-102.

Pramanik, S., U.N. Chowdhury, B.K. Pramanik and N. Huda, 2010. A comparative study of bagging, boosting and C4.5: The recent improvements in decision tree learning algorithm. Asian J. Inform. Technol., 9: 300-306.

Qiu, P., A.J. Gentles and S.K. Plevritis, 2011. Discovering biological progression underlying microarray samples. PLoS. Comput. Biol., 7: 1-11.

Qiu, P., E.F. Simonds, S.C. Bendall, J.K.D. Gibbs and R.V. Bruggner *et al*., 2011. Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. Nat. Biotechnol., 29: 886-891.

Schaeffer, E.S., 2007. Graph clustering. Comput. Sci. Rev., 1: 27-64.

Sonumol, N.S., V.R. Uma, S. Ashok and M.V. Judy, 2015. Community detection in multidimensional genomic dataset. Intl. J. Artif. Intell., 13: 109-117.

Speer, N., C. Spieth and A. Zell, 2004. A memetic co-clustering algorithm for gene expression profiles and biological annotation. Proceedings of the Congress on Evolutionary Computation Vol. 2, June 19-23, 2004, IEEE, Portland, Oregon, USA., ISBN:0-7803-8515-2, pp: 1631-1638.

Sreeja, A. and U. Krishnakumar, 2017. Gene ontology based functional analysis and graph theory for partitioning gene interaction networks. Intl. J. Pharma Bio Sci., 8: 183-192.

Topaloglou, T., S.B. Davidson, H.V. Jagadish, V.M. Markowitz and E.W. Steeg *et al*., 2004. Biological data management: Research, practice and opportunities. Proceedings of the 13th International Conference on Very Large Data Bases Vol. 30, August 31-September 3, 2004, VLDB Endowment, Toronto, Canada, ISBN:0-12-088469-0, pp: 1233-1236.

Tzanis, G., C. Berberidis and I.P. Vlahavas, 2005. Biological Data Mining. In: Encyclopedia of Database Technologies and Applications, Laura, R.C. (Ed.). Idea Group, Hershey, London, Melbourne, pp: 35-41.

Wang, Z., D. Huang, H. Meng and C. Tang, 2013. A new fast algorithm for solving the minimum spanning tree problem based on DNA molecules computation. Biosyst., 114: 1-7.

Xu, Y., V. Olman and D. Xu, 2001. Minimum spanning trees for gene expression data clustering. Genome Inf., 12: 24-33.