**ABSTRACT**

Markov Cluster Algorithm (MCL) is a fast and scalable unsupervised graph clustering algorithm, based on stochastic flow simulation model. By random walks on the graph, the flow of the graph is estimated for the clusters. It simulates flow using two simple algebraic operations on matrices i.e. expansion and inflation. The expansion operator connects different regions of graph while inflation operator strengthens and weakens the flow. This procedure follows until a steady state is reached and all the values in column are same. This algorithm works efficiently for clusters with small diameters. It is suitable for fast and accurate grouping of protein structures in human body.

Restricted neighborhood search Algorithm (RNSC) is also graph clustering technique that uses stochastic local search. Cost functions are assigned to clusters of a graph to get an optimal cost clustering. It is implemented for un-weighted and undirected graphs and uses heuristic approach. A number of moves are considered in this algorithm. A global move results in an optimal change in cost, diversification move shuffles the clustering randomly and an intensification move is associated with a good cost chosen from restricted portion of the current clustering’s neighborhood. To escape cycling, a tabu list is created to avoid same moves in the algorithm. This algorithm is efficient as it reduces the run time of clustering algorithm. The paper proposes behavior analysis of both thealgorithms effectively which is implemented in Python programming language.

**ACKNOWLEDGEMENT**

An understanding such as this one is never the work of a single person and this research work is not an exception. We, at this very outset, thank all those who directly or indirectly helped me in the completion of my research work and encouraged us in moments of our despair.

We wish to express our appreciation to \*\*\*\*\*\*\*\*\*\*\*\*under whom we worked, interacted and whose insights and thoughts helped not only in completion of my research work but also in enhancing my knowledge. We’re highly indebted to his generous help and guidance throughout the development of our research work.

Finally, we would like to heartily thank all my class mates without their help & support our research work would not have been successfully completed.

And last but certainly not the least; we acknowledge our deep gratitude to all the persons concerned.

- NAME

- ROLL NO

B.Tech (Final Year)

Computer Science and Technol***ogy.***

**TABLE OF CONTENTS**

* 1. Abstract ……………………………………………………………………………..i
  2. Acknowledgement…………………………………………………….....................ii
  3. Introduction ………………………………………………………………………..1
  4. Literature Survey ………………………………………………………………….4
  5. Research methodology …………………………………………………………….6
  6. Work plan…………………………………………………………………………..9
  7. Proposed Outcome ………………………………………………………………..9
  8. Bibliography and References………………………………………………….......9
  9. Appendix………………………………………………………………………......11

**INTRODUCTION**

Markov cluster algorithm is an unsupervised clustering algorithm which simulates the flow of the graph. It was first implemented by Stijen van Dogen at the Center for Mathematics and Computer Science in Netherlands. A Markov model is a stochastic model used to model randomly changing systems where it is assumed that future states depend only on the present state and not on the sequence of events that preceded it. Clusters in a graph are characterized by the presence of edges between the nodes of the clusters and the number of longer paths in the clusters should be high.

The MCL algorithm finds the clusters in a graph by a mathematical bootstrapping procedure. This process computes the probabilities of random walks through graph, and uses two operators (i.e. expansion and inflation) that transform one set of probabilities into another. This is done by stochastic matrices which are also known as Markov matrices that uses the concept of random walks on a graph. Expansion process is done by taking the power of a stochastic matrix using normal matrix product (matrix squaring). Inflation step involves the inflating of the resulting matrix, followed by the scaling step such that the resulting matrix is stochastic again i.e. the matrix elements (on each column) correspond to the probability values.

Iterating expansion and inflation results in the separation of graph into different segments. There are no longer any paths between these segments and collection of resulting segments is interpreted as a clustering. Inflation parameter can be altered using parameter ‘r’. By increasing this parameter, the inflation operator can be strengthened which increases the granularity or tightness of the clusters. Expansion causes flow to dissipate within clusters and inflation eliminates flow within clusters. When the matrix reaches a steady state i.e. the matrix elements does not change with further expansion and inflation steps.

Restricted Neighborhood Clustering algorithm (RNSC) is a popular method to detect the complexes in protein-protein interaction networks. It explores the solution space for all possible clusters by minimizing cost function that reflects the number of inter-cluster and intra-cluster edges. This algorithm is a stochastic meta-heuristic technique and can be used in various search space representations. It is a cost based clustering and performs local search iteratively to obtain optimum clustering in an efficient way. The main goal of this algorithm is to find the best cost for the set of clusters from a graph by assigning some cost functions (i.e. naive cost function and scaled cost function). The memory requirement for this algorithm is O (n^2). The complexity of a move in Naive cost function is O (n), which is the size of the restricted neighborhood of a move M.

Naive cost function acts as a pre-processor to produce initial clustering result on a graph and the Scaled function tries to optimize the output from naïve function to reach a global optimal solution. The scaled cost function is applied to evaluate a low cost clustering result. A large number of data structures are used in this algorithm which decreases the runtime of this RNSC algorithm. It first computes all the possible moves and then makes the best possible move available as compared to the other algorithms where initially the move is made and then the cost is computed. A number of moves are being considered in RNSC. A global move is a move that results in a near optimal change in cost. A Diversification move randomly shuffles the clustering and can be achieved through different ways. An Intensification move is associated with a good cost chosen from the restricted part of the current clustering’s neighborhood.

Here the performance of MCL and RNSC is done on the real scale- free graphs. The comparative assessment is measured in terms of cost of clustering, cluster size, modularity index of clustering results and NMI value to find out the better and more efficient algorithm for analyzing the protein-protein interaction network.

**LITERATURE SURVEY**

The complete set of scale data on protein interactions are generally represented as large networks in graph, connecting thousands of proteins. Since proteins function in complexes, it is important to find out protein complexes from such graphs. A number of researches have been done in order to reduce the complexity of protein-protein interactions. MCL is one such algorithm that helps in partitioning protein interaction graphs. The process has been carried out for un-weighted networks of protein interactions from a set of 408 protein complexes, and evaluates the performance of MCL algorithm with Affinity Propagation (AP) algorithm. The performance is being measured on the basis of parameter space of each algorithm which is sampled in order to get the optimal values. The robustness of algorithms is evaluated by quantifying the complexity level since interactions are randomly added and removed from the network to simulate noise. One of the major challenges for clustering procedure is to deal with the high level of noise in networks, as this requires reducing the effect of noise by various means. For example we need to consider topological properties of the network, either during the clustering process or by modifying distance metric prior to clustering.

Affinity Propagation (AP) was proposed to identify the representative examples in the given dataset by exchanging real-valued messages between all the data points. These points are then grouped with their most representative exemplar to give the final set of clusters. AP was applied to many problems including face recognition and gene identification from putative exons using micro array data. In AP each node is assigned a “preference” that it should be selected as an exemplar. In case no prior knowledge is available on which node should be chosen as exemplar, then all the nodes must be assigned with the same preference value and the magnitude is used to control the granularity of clusters. For each node ‘i’ and each candidate exemplar ‘k’, AP computes “responsibility” r (i , k), which indicates how well suited is ‘k’ as an exemplar for node ‘i’, and the “availability” a(i, k) reflecting the evidence that node ‘i’ should choose ‘k’ as an exemplar. The matrix s (i, k) denotes the similarity between the two nodes ‘i’ and ‘k’, whereas the diagonal of the matrix consists of the preference values. Each node ‘i’ can then be assigned to the exemplar ‘k’ which maximizes the sum [a (i, k) + r (i, k)], and if ‘i=k’ then ‘i’ is an exemplar.

The analyses have shown that MCL procedure is more tolerant to noise and is more robust in nature than AP algorithm. AP displays much convergence problems on the majority of un-weighted graphs that have been tested. But in case of MCL procedure, it is able to identify meaningful clusters. Therefore MCL is considered as the best method for identifying protein complexes from network interactions. For physical interaction networks, MCL performs better than AP in terms of its ability to generate meaningful partitions.

Although both the algorithms run very fast on weighted consolidated protein networks, but as noise is added to this network, AP can fail to converge at some preference values and it can be difficult to determine which parameters lead to convergence. For example, AP did not converge at any of the preference value for the un-weighted networks with edges randomly removed. On weighted networks with 30% noise, the algorithm converged at preference values of 0.65 and 0.9 only. Thus for this application, one difficulty in using AP is to determine the appropriate interval and granularity level for searching Preference values.

MCL algorithm considers both edge weight and network connectivity (topology) information; whereas AP procedure can fail in situations where high weight edges connect two clusters. For example, two cliques, A and B are connected with single, high weight edge. If one of the nodes comprises this edge is an exemplar in clique A, the adjacent node in clique B may be incorrectly assigned to clique A by AP, despite being highly connected to members of B. This shows that MCL achieves its robust performance by considering network topology, and AP relies on the edge weight to capture information. In order to overcome this limitation, a modified distance metric is to be defined that captures both the tendency of two proteins to interact and graph topology, and re-run the AP on modified graph. The performance of AP is similar to that of MCL when the network is not concerned, because the random shuffling of edges distorts the topological information present in the edge weight. In the un-weighted network, where no topological information is available, AP is only able to cluster unperturbed networks with a few inter-complex edges.

MCL algorithm works well for the small clusters but in case of clusters with large diameters, there might be a possibility that the components are weakly connected resulting in overlapping clusters. In such cases, MCL is unable to detect and remove overlapping in clusters. In order to remove the problem of overlapping that can result in an inconsistent set of uni-class clusters, authors adopted and modified the existing MCL algorithm and proposed a new graph partitioning approach named as Modified-Markov Clustering algorithm (Modified-MCL). This algorithm checks the clusters without overlapping by an iterative process of expansion and inflation operations. The results obtained by modified-MCL, when tested with a hypothetical object-oriented system, were compared with the existing MCL algorithm and is observed that there is no overlapping of clusters.

In the proposed approach, a software system is partitioned into subsystem components. This is being done because it becomes difficult to extend and modify large software systems for the new applications. Thus the smaller subsystems are easier to understand by a software designer. The studies have suggested a modified algorithm that inherits a few features of MCL algorithm. The inputs for Modified-MCL algorithms is an undirected weighted graph G, inflation parameter ‘m’ and expand parameter ‘b’. From the input graph, an associate and degree matrix is generated, which are then combined to obtain a symmetric matrix. This matrix is now normalized using the min-max normalization technique. The expansion and inflation operators are applied to the normalized matrix. This process is repeated iteratively until a stable matrix is obtained, and used to form a cluster as shown in the flowchart below.

**Flowchart of MCL Algorithm**

INPUT: A, adjacency matrix. Initialize M to MG, the canonical transition matrix M: =MG : = (A+I)D-1

Expand: M: = M\*M

Inflate: M: = M^ r (r usually 2),

Renormalize the columns

Output clusters

Converged?

Pruning Step