Adaptive Link-Based Cluster Ensemble for an Efficient Data Clustering

[1]Mendu Madhuri,[2]Dr.G.V.S.N.R.V.Prasad
[1]M.techStudent, [2] CSE,Professor & Head of CSE Dept.
Gudlavalleru Engg. College

Abstract: Data clustering is a primary tool for understanding the structure of data sets. Its application domain includes machine learning, data mining, information retrieval, and pattern recognition etc. Clustering aims to categorize data into groups or clusters such that the data in the same cluster are more similar to each other than to those in different clusters. Although conventional algorithms includes k-means clustering and expectation maximization (EM) clustering, PAM etc and different clustering ensemble approaches were used for clustering process they have limitations in handling unrelated entries in dataset resulting in a detrimental performance. So previously a link-based algorithm which is a two stage process involving generation of a conventional matrix by discovering unknown entries through similarity between clusters in an ensemble, and then obtaining a weighted bipartite graph from this refined matrix. In Existing system the construction of the weighted bipartite graph generation is irrespective of the size of the matrix which takes more time. For an optimized performance, this paper proposes use of ACO (ant colony optimization) Algorithm to Solve Minimum-Weighted Bipartite Matching for a smaller refined matrix and Metropolis Algorithm for Maximum-Weighted Bipartite Matching for a larger refined matrix. This kind of an adaptive approach to varying matrix sizes rather than a single static approach to all matrix sizes determines the optimization parameter such as timescales involved in data clustering process. An implementation of the proposed system validates our claim.

Index Terms: Maximum-Weighted Bipartite Matching, Ant Colony Optimization, Graph Partitioning Technique

I. INTRODUCTION

Data clustering is one of the fundamental tools we have for understanding the structure of a data set. Clustering aims to categorize data into groups or clusters such that the data in the same cluster are more similar to each other than to those in different clusters. Clustering algorithms like k-means and PAM have been designed for numerical data. These cannot be directly applied for clustering of categorical data that domain values are discrete and have no ordering defined. Many categorical data clustering algorithms have been introduced in recent years, with applications to interesting domains such as protein interaction data. The conventional k-means with a simple matching dissimilarity measure and a frequency-based method to update centroids. A single-pass algorithm makes use of a prespecified similarity threshold to determine which of the existing clusters to data point under examination is assigned.

The concepts of evolutionary computing and genetic algorithm have also been adopted by a partitioning method for categorical data. Cobweb is a model-based method primarily exploited for categorical data sets. A large number of algorithms have been introduced for clustering categorical data. The No Free Lunch theorem suggests there is no single clustering algorithm that performs best for all data sets and can discover all types of cluster shapes and structures presented in data. It is difficult for users to decide which algorithm would be the proper alternative for a given set of data. Cluster ensembles have emerged as an effective solution that is able to overcome these limitations and improve the robustness as well as the quality of clustering results. The main objective of cluster ensembles is to mix different clustering choices in such a way as to achieve accuracy superior to that of any individual clustering. Samples of well-known ensemble strategies are:

a. The feature-based approach that transforms the problem of cluster ensembles to agglomeration categorical data
b. The direct approach that finds the ultimate partition through relabeling the bottom agglomeration results
c. Graph-based algorithms that use a graph partitioning methodology and
d. The pair wise-similarity approach that produces use of co-occurrence relations between knowledge points

The underlying ensemble-information matrix presents only cluster-data point relationships while completely ignores those among clusters. The performance of existing cluster ensemble techniques may consequently be degraded as many matrix entries are left unknown. A link-based similarity measure is exploited to estimate unknown values from a link network of clusters. The performance of existing cluster ensemble techniques could consequently be degraded as several matrix entries are left unknown as per the result. A link-based from similarity live exploited to estimate unknown values a link network of clusters. Referred link analysis bridges the gap between each task of information agglomeration.

II. RELATED WORK

Let X = {x1; . . . ; xN} be a set of N data points and π = { π_1; . . . ; π_{M}} be a cluster ensemble with M base clustering's, each of which is referred to as an *ensemble member*. Each base clustering returns a set of clusters $\prod_i = \{ C_1^i, C_2^i, \ldots\ldots, C_k^i \}$. Then$\bigcup_{j=1}^{k_i} C_j^i = X$. Where ki is the number of clusters in the ith clustering.

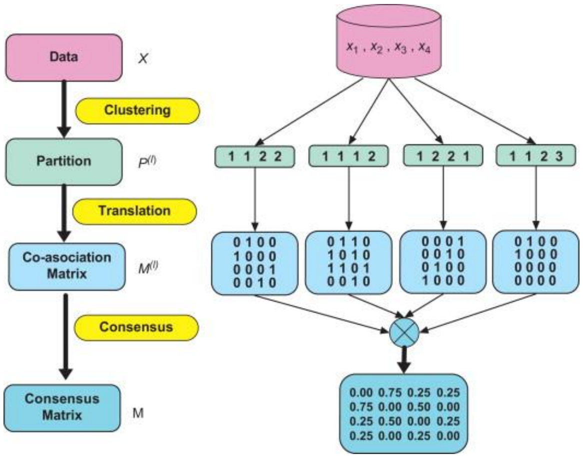


Figure 1: The basic process of cluster ensembles. It first applies multiple base clustering's to a data set X to obtain diverse clustering decisions (π_1 . . . π M). Then, these solutions are combined to establish the final clustering result (π *) using a consensus function.

As shown in the fig.1 the major problem is to find a new partition π *of a data set X that summarizes the information from the cluster ensemble π. Solutions achieved from different base clustering are aggregated to form a final partition. In the metalevel methodology involves in two major tasks:

a. Producing the final partition
b. Generating a cluster ensemble

Particularly for knowledge clump, the results obtained with any single algorithmic rule over several iterations were typically similar. Many heuristics have been planned to introduce artificial instabilities in clustering algorithms. While a large number of cluster ensemble techniques for numerical data have been put forward in the previous decade. The method introduced in creates an ensemble by applying a conventional clustering algorithm. The technique developed acquires a cluster ensemble without actually implementing any base clustering on the examined data set. Existing cluster ensemble methods to categorical data analysis rely on the typical pair wise-similarity and binary cluster-association matrices. The quality of the final clustering result may be degraded regardless of a consensus function. In spite of promising findings is based on the data point pair wise-similarity matrix that is highly expensive to obtain. A new link-based algorithm has been specifically to generate such measures in an accurate, inexpensive manner as shown in the fig.2.

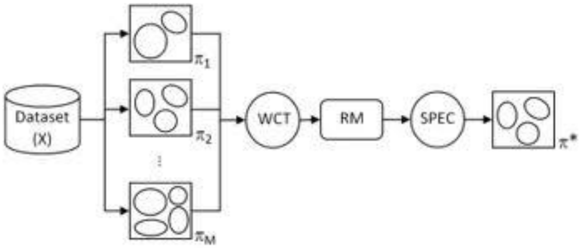


Figure 2: The link-based cluster ensemble framework

Cluster ensemble transforms the matter of categorical knowledge cluster-to-cluster ensembles by considering each categorical attribute price (or label) as a cluster in as ensemble. While single-attribute knowledge partitions might not be as correct as those obtained from the cluster of all knowledge attributes. To generate diversity among an ensemble is to take advantage of a number of various knowledge subsets.

III. EXISTING SYSTEM

Clustering is a data mining technique used to place data elements into related groups without advance knowledge of the group definitions. A popular conventional algorithm includes k-means clustering and expectation maximization (EM) clustering, PAM etc. However, these cannot be directly applied for clustering of categorical data, where domain values are discrete and have no ordering defined. Although, a large number of algorithms have been introduced for clustering categorical data, the "No Free Lunch" theorem suggests there is no single clustering algorithm that performs best for all data sets and can discover all types of cluster shapes and structures presented in data. Each algorithm has its own strengths and weaknesses. For a particular data set, different algorithms, or even the same algorithm with different parameters, usually provide distinct solutions. Therefore, it is difficult for users to decide which algorithm would be the proper alternative for a given set of data. Due to their inefficiency different clustering ensemble approaches (Homogeneous ensembles, Random-k, Data subspace/sampling, Heterogeneous ensembles, Mixed heuristics) to obtain data clusters were developed and used. Clustering ensembles combine multiple partitions of the given data into a single clustering solution of better quality. Works well for all datasets. Users need not choose the clustering filtration manually. Although results were satisfactory, the ensemble approaches generate a final data partition based on incomplete information without considering the unrelated entries resulting in a detrimental performance. So a better system is required that has all the benefits of an ensemble system and is better equipped to handle unrelated entries. The underlying ensemble-information matrix presents only cluster-data point relations, with many entries being left unknown. Ignoring dataset unsolvable entries during clustering degrades the quality of the clustering result. The new link-based algorithm is a two-stage process.

- It improves the conventional matrix by discovering unknown entries through similarity between clusters in an ensemble.
- Then to obtain the final clustering result, a graph partitioning technique is applied to a weighted bipartite graph that is formulated from the refined matrix.

An obtained clustering result suggests that the proposed link-based method usually achieves superior clustering results compared to those of the traditional categorical data algorithms and prior cluster ensemble techniques.

IV. PROPOSED SYSTEM

A weighted bipartite graph is formulated from the refined matrix obtained from link based cluster ensemble. The construction of the weighted bipartite graph is irrespective of the size of the matrix. For an optimized performance, we propose to use ACO (ant colony optimization) Algorithm to Solve Minimum-Weighted Bipartite Matching for a smaller refined matrix. Metropolis Algorithm for Maximum-Weighted Bipartite Matching for a larger refined matrix This kind of an adaptive approach to varying matrix sizes rather than a single static approach to all matrix sizes determines the optimization parameter such as timescales involved in data clustering process.

ANT COLONY OPTIMIZATION ALGORITHM

Combinatorial optimization problems are tantalize because they are often easy to state but very difficult to solve. To practically solve large instances one often has to use approximate methods, which return near-optimal solutions in a relatively short time. The algorithms of this type are loosely called heuristics. A metaheuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. A particularly successful metaheuristic is inspired by the behavior of real ants. A number of algorithmic approaches based on the very same ideas were developed and applied with considerable success to a variety of combinatorial optimization problems from academic as well as from real-world applications. The ACO metaheuristic has been proposed as a common framework for the existing applications and algorithmic variants of a variety of ant algorithms. The first algorithm to fall into the framework of the ACO metaheuristic was Ant System (AS). To be mentioned here is also the international workshop series ‘‘ANTS: From Ant Colonies to Artificial Ants’’ on ant algorithms. The ACO metaheuristic was inspired by the foraging behavior of real ants. It has a very wide applicability: it can be applied to any combinatorial optimization problem for which a solution construction procedure can be conceived. The ACO metaheuristic is based on a generic problem representation and the definition of the ants' behavior. The ants in ACO build solutions to the problem being solved by moving concurrently and asynchronously on an appropriately defined construction graph. The application of ACO is particularly interesting for

- ***NP-hard problems:*** It cannot be efficiently solved by more algorithms that are traditional.
- ***Dynamic shortest-path problems***
  Some properties of the problem's graph representation change over time concurrently with the optimization process
- ***Problems in spatial distribution***
  The problem of the computational architecture is spatially distributed

An ACO algorithm designed to help solve the routing problem in telecommunications networks. Network routing refers to the activities necessary to guide information in its travel from source to destination nodes. The ACO processing paradigm is a good match for the distributed and non-stationary nature of the problem. It presents a high level of redundancy and fault tolerance and can handle multiple objectives and constraints in a flexible way. Communications networks can be classified as either circuit-switched or packet switched. The routing table is a common component of all routing algorithms: it holds the information used by the algorithm to make the local forwarding decisions. One routing table is maintained by each node in the network: it tells the node's incoming data packets that among the outgoing links to use to continue their travel toward their destination node.

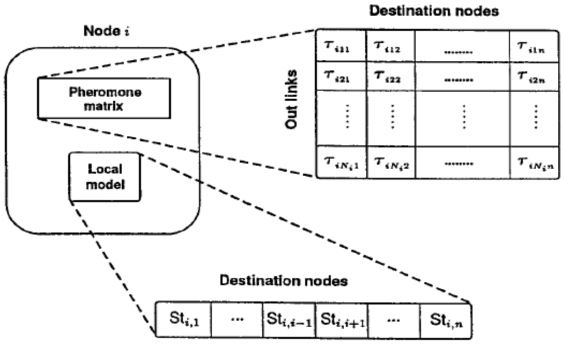
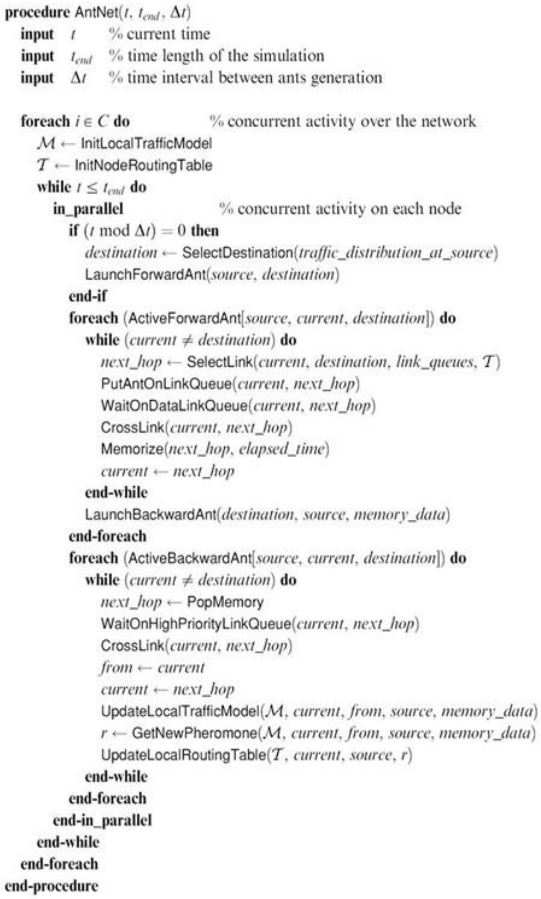


Figure 3: Data structures used by the artificial ants in AntNet.

AntNet a direct extension of the Simple Ant Colony Optimization algorithm. AntNet is even closer to the real ants' behavior that inspired the development of the ACO metaheuristic than the ACO algorithms for NP-hard problems. In AntNet, artificial ants move on the construction graph with the constraint of never using the set of links that do not belong to the network graph. AntNet is conveniently described in terms of two sets of artificial ants. AntNet called in the following forward and backward ants. Ants communicate in an indirect way through the information they concurrently read and write on the network nodes they visit. The AntNet algorithm, whose high-level description in pseudo-code is given in fig.4. The ant builds a path performing the following steps:



```
procedure AntNet($t$, $t_{end}$, $\Delta t$)
  input  $t$         % current time
  input  $t_{end}$   % time length of the simulation
  input  $\Delta t$  % time interval between ants generation

  foreach $i \in C$ do               % concurrent activity over the network
    $\mathcal{M}$ ← InitLocalTrafficModel
    $\mathcal{T}$ ← InitNodeRoutingTable
    while $t \leq t_{end}$ do
      in_parallel                    % concurrent activity on each node
        if ($t$ mod $\Delta t$) = 0 then
          destination ← SelectDestination(traffic_distribution_at_source)
          LaunchForwardAnt(source, destination)
        end-if
        foreach (ActiveForwardAnt[source, current, destination]) do
          while (current ≠ destination) do
            next_hop ← SelectLink(current, destination, link_queues, $\mathcal{T}$)
            PutAntOnLinkQueue(current, next_hop)
            WaitOnDataLinkQueue(current, next_hop)
            CrossLink(current, next_hop)
            Memorize(next_hop, elapsed_time)
            current ← next_hop
          end-while
          LaunchBackwardAnt(destination, source, memory_data)
        end-foreach
        foreach (ActiveBackwardAnt[source, current, destination]) do
          while (current ≠ destination) do
            next_hop ← PopMemory
            WaitOnHighPriorityLinkQueue(current, next_hop)
            CrossLink(current, next_hop)
            from ← current
            current ← next_hop
            UpdateLocalTrafficModel($\mathcal{M}$, current, from, source, memory_data)
            r ← GetNewPheromone($\mathcal{M}$, current, from, source, memory_data)
            UpdateLocalRoutingTable($\mathcal{T}$, current, source, r)
          end-while
        end-foreach
      end-in_parallel
    end-while
  end-foreach
end-procedure
```

Figure 4: The Ant Net Algorithm

METROPOLIS ALGORITHM

We first look at two important applications of the Metropolis Algorithm

- The Ising model

  This model is one of the most extensively studied systems in statistical physics. The model is a 2D or 3D regular array of spins and an associated energy *E* (s) for each configuration. We want to estimate a mean of some function *f*(s) because such quantities give us a first-principles estimate of some fundamental physical quantity.

- simulated annealing

  One approach is *hill climbing*. Given a set of possible changes to a tour like permuting the order of some visits that choose the change, which decreases the tour length as much as possible. The Metropolis Algorithm offers a possible method for jumping out of a local minimum. The tour's length plays the same role that energy plays in the Ising model.

  Suppose we have a posterior p(θ|y) that we want to sample from. Even though

- The posterior does not look like any distribution we know
- The posterior consists of more than 2 parameters
- Some (or all) of the full conditionals do not look like any distributions we know

The Metropolis-Hastings Algorithm follows the following steps:

Step.1: Choose a starting value$\theta^{(0)}$.

Step.2: At iteration t, draw a candidate θ^* from a jumping distribution $J_t^{(\theta^*|\theta^{(t-1)})}$.

Step.3: Compute an acceptance ratio (probability):

$$P = \frac{P(\theta^*|y)/\ J_t^{(\theta^*|\theta^{(t-1)})}}{P(\theta^{(t-1)}|y)/\ J_t^{(\theta^{(t-1)}|\theta^*)}}$$

Step.4: Accept θ^* as $\theta^{(t)}$) with probability min(r, 1). θ^* is not accepted then $\theta^{(t)} = \theta^{(t-1)}$

Step.5: Repeat steps 2-4 M times to get M draws from p(θ|y) with optional burn-in and/or thinning.

It is important to monitor the acceptance rate of Metropolis-Hastings algorithm. The chain is probably not mixing well if the obtained acceptance rate is too high. If acceptance rate is too low then algorithm is too inefficient.

V. EXPERIMENTAL ANALAYSIS

The quality of data partitions generated by *hill climbing* technique is assessed against those created by different categorical data clustering algorithms and cluster ensemble techniques. We Compare the results of our proposed Metropolis and ACO clustering algorithm with LCE based clustering algorithm. Both specifically developed for categorical data analysis and those state-of-the-art cluster ensemble techniques found in literature.

We import required data sets on different domains and specifically entered into data computational process model.

Each clustering method divides data points into a partition of K clusters it is then evaluated against the corresponding true partition using the following set of label-based evaluation data sets like accident, diabetes, marks and economy ratings.

Then we calculate each dataset time computation for data released into computational process model. This time computation can be calculated in both priori approach and proposed approach.

Data Set	Existing System	Proposed System
Accident	0.55	0.53
Diabetes	0.75	0.43
Economy Ratings	0.33	0.27
Marks	0.02	0.003

Table1: Comparison results on each data sets in terms of time complexity.

The table-1 shows the time for clustering using existing and proposed approaches.It also shows that the proposed system takes less time compared to the exixting system.

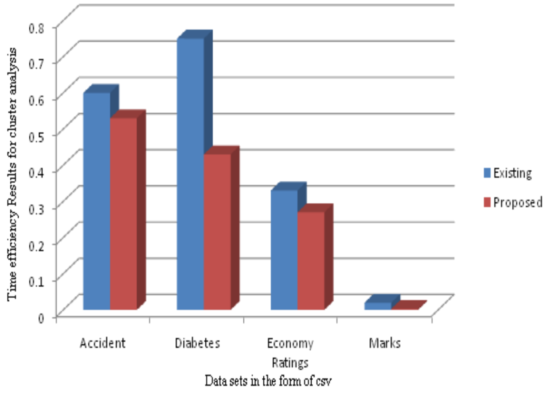


Figure 5: Comparison results on each data sets in terms of time complexity.

Analysis of these results time complexity in proposed approach was reduced when compared to prior approach.

Ant colony optimization clustering technique is an efficient method for classifying similar data items for cluster formation. Formation of centroid is the main process in cluster application. This process can be applied for solving relevant similarity results for each data set. Then combine all the similar results of each data item.

Established results are occupied and formed cluster centroid for each data set. The parameter analysis aims to provide a practical means by which users can make the best use of the link-based framework. This paper presents a unique, extremely effective link-based cluster ensemble approach to categorical knowledge agglomeration.

VI. CONCLUSION

We use an link-based algorithm; we observed the construction of the weighted bipartite graph generation is irrespective of the size of the matrix. For an optimized performance, we propose to use ACO (ant colony optimization) Algorithm to Solve

Minimum-Weighted Bipartite Matching for a smaller refined matrix and Metropolis Algorithm for Maximum-Weighted Bipartite Matching for a larger refined matrix. An implementation of the proposed system validates our claim. We propose an ACO algorithm for the minimum weighted bipartite matching for the small refined matrix. The ACO refers to Ant Colony Optimization that defined by the ACO metaheuristic was Ant System (AS). And for the Maximum weighted bipartite graph we use an metropolis algorithm where we use the a posterior p(θ|y) that we want to sample from. Our experimental results show efficient bipartite graph formation in each data set. Further improvement of cluster analysis in each data set can be developed in greedy heuristic algorithms to reduce computational overhead on each data set.

VII. REFERENCES

[1] Natthakan Iam-On, Tossapon Boongoen, Simon Garrett, and Chris Price, “A Link-Based Cluster Ensemble Approach for Categorical Data Clustering,” IEEE Transactions on Knowledge and Data Engineering, VOL. 24, NO. 3, PP: 413-425, MARCH 2012.

[2] D.S. Hochbaum and D.B. Shmoys, “A Best Possible Heuristic for the K-Center Problem,” Math. of Operational Research, vol. 10, no. 2, pp. 180-184, 1985.

[3] L. Kaufman and P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. Wiley Publishers, 1990.

[4] P. Zhang, X. Wang, and P.X. Song, “Clustering Categorical Data Based on Distance Vectors,” The J. Am. Statistical Assoc., vol. 101, no. 473, pp. 355-367, 2006.

[5] M.J. Zaki and M. Peters, “Clicks: Mining Subspace Clusters in Categorical Data via Kpartite Maximal Cliques,” Proc. Int’l Conf. Data Eng. (ICDE), pp. 355-356, 2005.

[6] V. Ganti, J. Gehrke, and R. Ramakrishnan, “CACTUS: Clustering Categorical Data Using Summaries,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD), pp. 73-83, 1999.

[7] D. Barbara, Y. Li, and J. Couto, “COOLCAT: An Entropy-Based Algorithm for Categorical Clustering,” Proc. Int’l Conf. Information and Knowledge Management (CIKM), pp. 582-589, 2002.

[8] Y. Yang, S. Guan, and J. You, “CLOPE: A Fast and Effective Clustering Algorithm for Transactional Data,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD), pp. 682- 687, 2002.

[9] A.K. Jain and R.C. Dubes, Algorithms for Clustering. Prentice-Hall, 1998.