 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Cooperative Activities of Public Network Information with Edge Clustering

[1]T.J. Sateesh Chandra, [2]K.John Paul
[1,2]Dept. of CSE, NOVA College of Engineering & Technology,Vijayawada, AP, India

Abstract
It is well known that actors in a network demonstrate correlated behaviors. In this work, we aim to predict the outcome of collective behavior given a social network and the behavioral information of some actors. In particular, we explore scalable learning of collective behavior when millions of actors are involved in the network. Our approach follows a social-dimension based learning framework. Social dimensions are extracted to represent the potential affiliations of actors before discriminative learning occurs. As existing approaches to extract social dimensions suffer from scalability, it is imperative to address the scalability issue. We propose an edge-centric clustering scheme to extract social dimensions and a scalable k-means variant to handle edge clustering. Essentially, each edge is treated as one data instance, and the connected nodes are the corresponding features. Then, the proposed k-means clustering algorithm can be applied to partition the edges into disjoint sets, with each set representing one possible affiliation. With this edge-centric view, we show that the extracted social dimensions are guaranteed to be sparse. This model, based on the sparse social dimensions, shows comparable prediction performance with earlier social dimension approaches. An incomparable advantage of our model is that it easily scales to handle networks with millions of actors while the earlier models fail. This scalable approach offers a viable solution to effective learning of online collective behavior on large scale.

Key words
Classification with Network Data, Collective Behavior, Community Detection, Social Dimensions

I. Introduction

We can generalize the social network advertising problem to the study of collective behavior. Different types of behavior can include a broad range of actions, such as joining a group, connecting to a person, clicking on an ad, becoming interested in certain topics, or dating certain types of people. Collective behavior refers to behaviors of individuals in a social network environment, but it is not simply the aggregation of individual behaviors. In a connected environment, individuals' behaviors tend to be interdependent, influenced by the behavior of friends. This naturally leads to behavior correlation between connected users. Such collective behavior correlation can also be explained by homophily,1a term coined in the 1950s to explain our tendency to link up with one another in ways that confirm rather than test our core beliefs. In other words, we are more likely to connect to others sharing certain similarities with us, and similar people tend to become friends, leading to similar behavior between connected egos in a social network. This phenomenon has been observed in both the real world and online environments. For example, if our friends buy something, there is a better-than-average chance that we'll buy it, too. Because a social network provides valuable information concerning actor behaviors, it is natural to ask how we can use the behavior correlation presented in a social network to predict collective behavior. That is, given a social network with behavior information of some actors, how can we infer the behavior outcome of the remaining actors within the same network? This problem assumes that we can observe the behaviors of some individuals so that social learning is attainable. The amount of information that we can collect in reality depends on tasks. For instance, if we want to know whether a user will click on an ad, we can collect this information when the ad is displayed to the user. To determine behavior concerning voting for a presidential candidate, we can collect some voluntary responses using online surveys. With such behavior information, we can unravel the collective behavior by exploiting the network connectivity between actors. (See the "Related Work in Collective Behavior Prediction "sidebar for previous work in this area.)

The advancement in computing and communication technologies enables people to get together and share information in innovative ways. Social networking sites (a recent phenomenon) empower people of different ages and backgrounds with new forms of collaboration, communication, and collective intelligence. Prodigious numbers of online volunteers collaboratively write encyclopedia articles of unprecedented scope and scale; online marketplaces recommend products by investigating user shopping behavior and interactions; and political movements also exploit new forms of engagement and collective action. In the same process, social media provides ample opportunities to study human interactions and collective behavior on an unprecedented scale. In this work, we study how networks in social media can help predict some human behaviors and individual preferences. In particular, given the behavior of some individuals in a network, how can we infer the behavior of other individuals in the same social network [1]? This study can help better understand behavioral patterns of users in social media for applications like social advertising and recommendation.

II. Existing System

In this section we discuss the Clustering algorithm of K-means Algorithm.

K-means clustering (MacQueen, 1967) is a method commonly used to automatically partition a data set into k groups. It proceeds by selecting k initial cluster

1. Each instance d_i is assigned to its closest cluster center.
2. Each cluster center C_j is updated to be the mean of its constituent instances

The algorithm converges when there is no further change in assignment of instances to clusters. In this work, we initialize the clusters using instances chosen at random from the data set. The data sets we used are composed solely of either numeric features or symbolic features. For numeric features, we use a Euclidean distance metric; for symbolic features, we compute the Hamming distance. The nal issue is how to choose k. For data sets where the optimal value of k is already known (i.e., all of the UCI data sets), we make use of it; for the real- world problem of ending lanes in GPS data, we use a wrapper search to locate the best value of k.We now proceed to a discussion of our medications to the k-means algorithm. In this work, we focus on background knowledge that can be expressed as a set of instance-level constraints on the clustering process. After a discussion of the kind of constraints we are using, we describe the constrained k-means clustering algorithm.

In the context of partitioning algorithms, instance-level constraints are a useful way to express a priori knowledge about which instances should or should not be grouped together. Consequently, we consider two types of pair wise constraints:

1. Must-link constraints specify that two instances have to be in the same cluster.
2. Cannot-link constraints specify that two in-stances must not be placed in the same cluster

The must-link constraints define a transitive binary relation over the instances. Consequently, when making use of a set of constraints (of both kinds), we take a transitive closure over the constraints. The full set of derived constraints is then presented to the clustering algorithm.

COP-KMEANS(data set D, must-link constraints $Con_= \subseteq D \times D$, cannot-link constraints $Con_{\neq} \subseteq D \times D$)

1. Let $C_1 \ldots C_k$ be the initial cluster centers.
2. For each point d_i in D, assign it to the closest cluster C_j such that VIOLATE-CONSTRAINTS(d_i, C_j, $Con_=$, $Con_{\neq}$) is false. If no such cluster exists, fail (return {}).
3. For each cluster C_i, update its center by averaging all of the points d_j that have been assigned to it.
4. Iterate between (2) and (3) until convergence.
5. Return $\{C_1 \ldots C_k\}$.

VIOLATE-CONSTRAINTS(data point d, cluster C, must-link constraints $Con_= \subseteq D \times D$, cannot-link constraints $Con_{\neq} \subseteq D \times D$)

1. For each $(d, d_=) \in Con_=$: If $d_= \notin C$, return true.
2. For each $(d, d_{\neq}) \in Con_{\neq}$: If $d_{\neq} \in C$, return true.
3. Otherwise, return false.

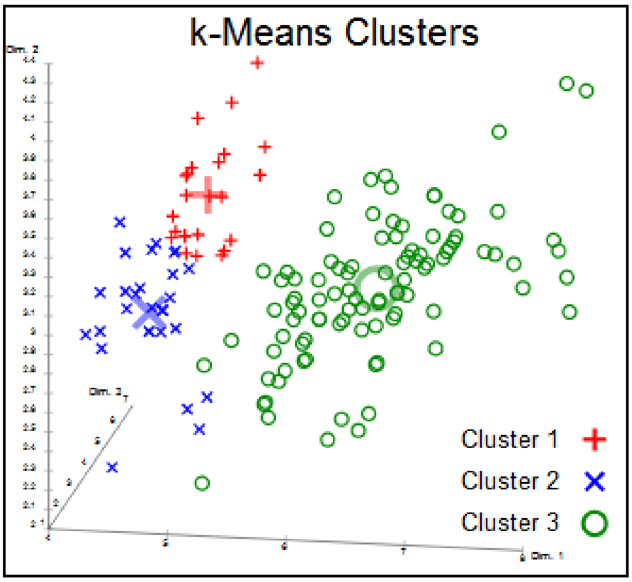

Fig. 1:

III. Proposed Work

In this section we Discuss the Edge Clustering Algorithm. In this section, we give a brief overview of our edge-clustering framework. We assume that the positions of the nodes in the input graph are already available. For some applications, node positions encode geographic information and any dramatic adjustment of node positions may cause confusion for users. For other applications, the positions of nodes can be computed by methods such as force-based models and thus a relatively good initial layout can be obtained. Therefore, we do not further change node positions and the original node layout is preserved. Our goal is to convert general straight line graphs into road-map-style graphs, and the basic idea of our method is to cluster the edges based on a control mesh that reflects the underlying graph structures.

Illustrates the framework of our approach. It consists of three major steps:

1. Control mesh generation
2. Edge clustering
3. Visualization

Control mesh generation has two components: graph analyzer and mesh generator. The node and edge information of the original graph is first sent to the analyzer to detect underlying edge distribution patterns. After that, some representative primary edge directions are output to the mesh generator, which then generates some mesh edges perpendicular to each selected primary direction. These mesh edges serve as basic control-mesh edges. By further adding more mesh nodes and triangulating the nodes and basic edges, the mesh generator completes the control mesh and sends it to the bundler. Based on the intersections between the original graph and the control mesh, the edge bundler sets some control points on the control-mesh edges and curves the original graph edges to pass through these control points to form edge clusters. In the edge smoother, some curved edges with too many zigzags are further fine tuned to become visually pleasing. Finally, in the visualize, an intuitive exploration interface is provided for users to interact with the edge-clustering results.

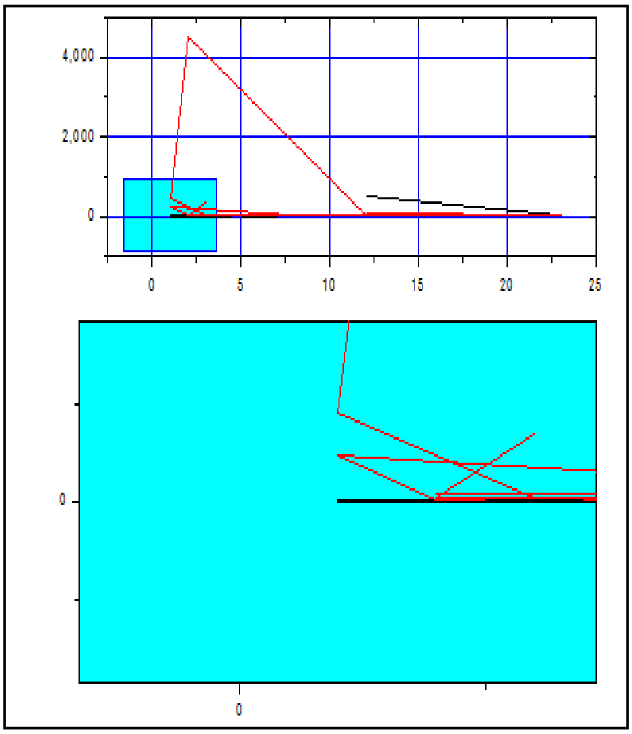

Fig. 2: Edge Clustering Algorithm Performance

IV. Results

In order to implement these two algorithms on the basis on time complexity

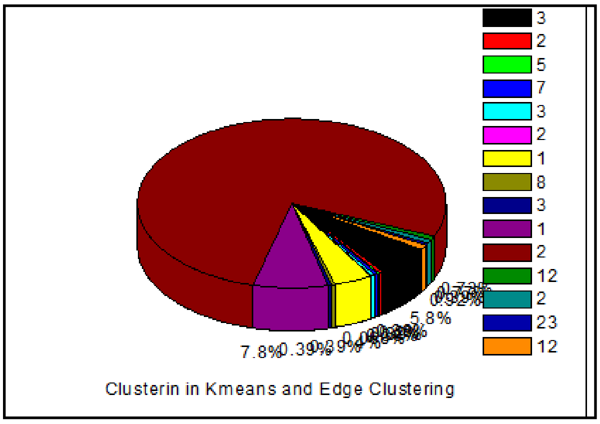

Fig. 3: Pie Chart Total Clusters in Kmeans and Edge Clustering

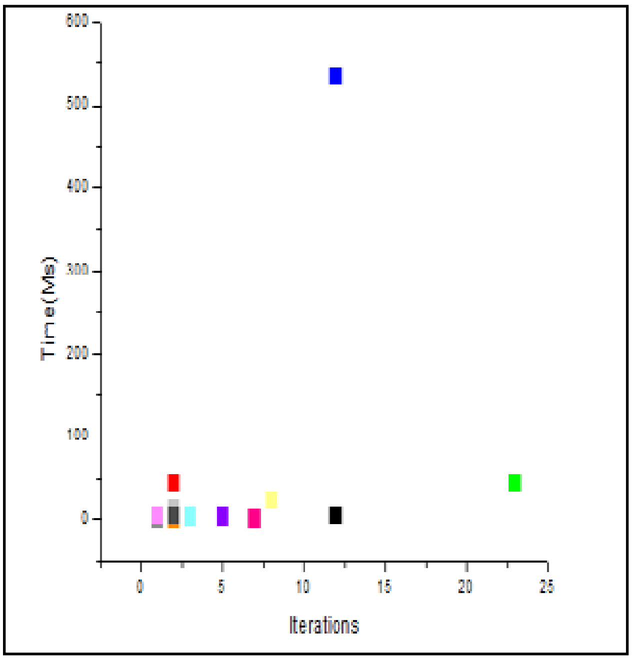

Fig. 4: Clusters in K Means Clusters

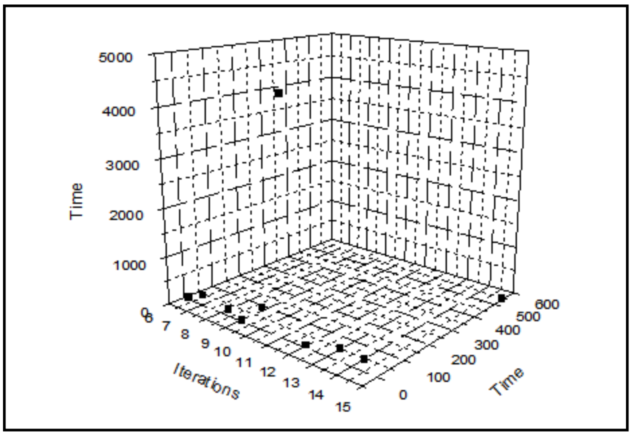

Fig. 5: Chart Clusters in Edge Clusters

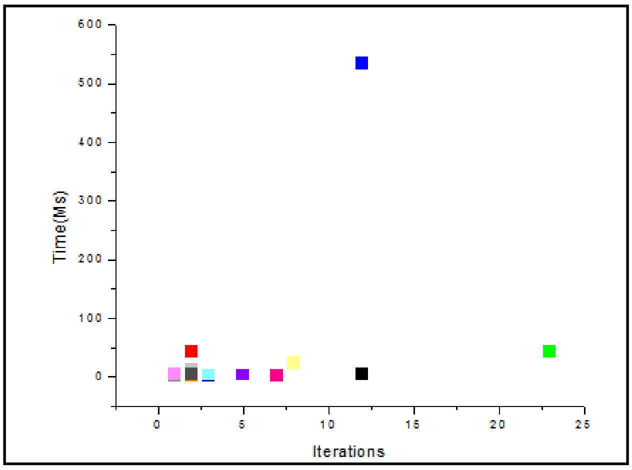

Fig. 6: Spatial Graph Clusters

V. Conclusions

In social media, multiple modes of actors can be involved in the same network, resulting in a multimode network. For instance, in YouTube, users, videos, tags, and comments are intertwined with each other in co-existence. Extending the edge-centric clustering scheme to address this object heterogeneity can be a promising future direction. Since the proposed Edge Cluster model is sensitive to the number of social dimensions as shown in the experiment, further research is needed to determine a suitable dimensionality automatically. It is also interesting to mine other behavioral features (e.g., user activities and temporal spatial information) from social media, and integrate them with social networking information to improve prediction performance.

References

[1] L. Tang, H. Liu,"Toward predicting collective behavior via social dimension extraction", IEEE Intelligent Systems, Vol. 25, pp. 19–25, 2010.

[2] "Relational learning via latent social dimensions", in KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, USA: 2009, pp. 817–826.

[3] M. Newman,"Finding community structure in networks using the eigenvectors of matrices", Physical Review E (Statistical, Nonlinear, and Soft Matter Physics), Vol. 74, No. 3, 2006.

[4] L. Tang, H. Liu,"Scalable learning of collective behavior based on sparse social dimensions", in CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management. New York, USA: 2009, pp. 1107–1116.

[5] P. Singla, M. Richardson,"Yes, there is a correlation: - from social networks to personal behavior on the web", in WWW '08: Proceeding of the 17th international conference on World Wide Web. New York, USA: 2008, pp. 655–664.

[6] M. McPherson, L. Smith-Lovin, J. M. Cook,"Birds of a feather: Homophily in social networks", Annual Review of Sociology, Vol. 27, pp. 415–444, 2001.

[7] A. T. Fiore, J. S. Donath,"Homophily in online dating: when do you like someone yourself?" in CHI '05: CHI '05 extended abstracts on Human factors in computing systems", New York, USA: 2005, pp. 1371–1374.

[8] H. W. Lauw, J. C. Shafer, R. Agrawal, A. Ntoulas, "Homophily in the digital world: A LiveJournal case study", IEEE Internet Computing, Vol. 14, pp. 15–23, 2010.

[9] S. A. Macskassy, F. Provost,"Classification in networked data: A toolkit and a univariate case study", J. Mach. Learn. Res., Vol. 8, pp. 935–983, 2007.
[10] X. Zhu,"Semi-supervised learning literature survey", 2006. [Online]Available:http://pages.cs.wisc.edu/jerryzhu/ pub/ssl survey 12 9 2006.pdf
[11] L. Getoor, B. Taskar, Eds.,"Introduction to Statistical Relational Learning", The MIT Press, 2007.

Mr.T.J.SATISH CHANDRA is a student of NOVA college of Engineering & Technology, Jangareddy gudem. Presently he is pursuing his M.TECH from this college and He received his M.Sc from Nagarjuna University In the year 2001.

Mr. K.JOHN PAUL, well known Author and excellent teacher Received B.Tech from jntu hydrabad, andM. Tech (CSE) from jntu kakinada is working as Associate Professor and HOD, Department of IT, Nova college of Engineering and Technology, he has 5 years of teaching experience in various engineering colleges. To his credit couple of publications both national and international conferences journals. His area of Interest includes Data Base management systems, computer networks, information security, flavors of Unix Operating systems and other advances in computer Applications.