Community Detection by modeling entity-annotated text: A Non-Negative Matrix Factorization Approach.

Pranav Nerurkar
Research Scholar,
Dept. of Computer Engineering,
Veermata Jijabai Technological Institute,
Matunga (W), Mumbai - 400019.
Email: pranavn91@gmail.com

Sunil Bhirud
Professor,
Dept. of Computer Engineering,
Veermata Jijabai Technological Institute,
Matunga (W), Mumbai - 400019.
Email: sgbhirud@vjti.org.in

***Abstract*—**

***Index Terms*—community detection, machine learning, clustering.**

I. Introduction

Exploratory Data Analysis [E.D.A.] is a domain on the intersection of fields such as machine learning, pattern recognition and information retrieval. The key goal of this domain is to generate effective summarization, visualization, discovery and retrieval from data thus reducing the exponential cost involved in storage. The main task performed in E.D.A. is clustering analysis. Cluster analysis is a type of unsupervised learning as ground truth labels are not present or rather are implicit and clustering is data driven. Clusters have no standard definition and hence there is subjectivity in the deciding what is a clusters and the best method to detect it. Distance based definition of clusters have been explored and have created a family of techniques such as partition based clustering [1] , hierarchical clustering [2], mixture model based clustering [3], fuzzy clustering [4] amongst others. In contrast to a line of previous work a second definition of clusters was proposed based on density this created popular algorithms such as DBSCAN, OPTICS [5] [6].

A second field is community detection which involves identification of latent groups of entities in data which correspond to autonomous regions known to have a higher degree of homogeneity within its members than with members of other such sub groupings. In network science such sub groups are called communities which are identified using network topology. A vast area of literature has uncovered several state of the art community detection algorithms that aim to find these communities in un-directed as well as directed graphs [9] [12] [13] [14] . Such literature is based on concepts related to Information Theory, Random walks on graphs or have used random walk literature to create an approach known as Map Equation [7] [8]. Apart from this community detection also developed a concept called modularity and a new family of algorithms were developed that detected communities in graphs by optimizing modularity in a greedy manner [10] [11] [15].

Community affiliation graph model was another concept that emerged and led to a new line of research on detecting communities by utilizing meta-data that is associated with the entities. Techniques were created that utilized the information about network topology along with meta data for obtaining generative models of networks [16] [17]. The approach used in this line of algorithms was to combine the Latent Dirichlet Allocation technique with information about network topology. However even with such methods there were drawbacks such as the limited applicability, as they cannot detect overlapping communities. A second drawback is that they assume soft node-community memberships, which are not appropriate for modeling communities because they do not allow a node to have high membership strength to multiple communities simultaneously. Finally, such methods have a large time complexity and can't be scaled to graphs having more than 1000 nodes.

Another approach, Non negative matrix factorization of the term document matrix was found to be effective in document clustering [18]. NMF was later extended to clustering by aiming to learn the adjacency matrix of a graph. NMF research did not pay attention to the interpretation of latent factors which are used to nd out the matrix. This led to development of BIGCLAM which aimed to learn latent factors which the authors argued represented strengths of community afliations of nodes. BIGCLAM and NMF both required community afliation knowledge to be known apriori for the nodes so that their membership strengths could be estimated. However, recent surveys on community detection argued that meta data isnt useful for community detection [19] [20]. While most existing literature [21] [22] [23] focuses either on using the meta data or entity annotation for improving the quality of community detection. To the best

our knowledge, in the research no mention could be found of using the attributes of a data point for calculating the latent features on the basis of which communities shall be detected. The work is based on BIGCLAM framework but the critical difference in this work is that attributes shall be used instead of community affiliations. The intuition here is that attributes are useful in determining the cluster affiliation, this is consistent with homophily seen in networks.

The paper is organized as follows. Section II briefly surveys related work. In Section III, the statistical model of the approach is defined, and in Section IV, the parameter fitting procedure is provided in detail. This is followed by describing experimental evaluation in Section V and the conclusion.

II. Related Work

Clustering approaches have their own biases in identifying clusters in data. Each type of algorithm has its own objective function and optimizing criteria and hence there exist multiple algorithms for clustering but none is considered a universal best t. For example, the objective function to be minimized in the k-partitioning algorithms is variance or SSE i.e. Sum of Squared Distance. $SSE = \sum_{k=1}^{k} \sum_{x_i \epsilon c_k} \|x_i - c_k\|^2$ where c_k = centroid of the cluster. In this case the clusters are convex but the process converges to a local optima as the objective function is non convex [1]. Another type is Hierarchical Clustering algorithms which are a category of algorithms having a completely unsupervised approach to clustering. They do not require the users to specify the number of clusters in advance and are broadly of two types: Agglomerative and divisive. To measure the dissimilarity between clusters obtained in hierarchical clustering, linkage methods were developed with several popular techniques being listed in the literature [1] [2].

Fuzzy clustering minimizes the objective function given as $\sum_{j=1}^{k} \sum_{x_i \in C_j} u_{ij}^m (x_i - u_j)^2$ with μ being the fuzzifier and m defining the level of cluster fuzziness. The MST clustering algorithm discussed in [24] is known to be capable of detecting clusters with irregular boundaries. Unlike traditional clustering algorithms, the MST clustering algorithm does not assume a spherical shaped clustering structure of the underlying data. Once the MST is built for a given input, there are two different ways to produce a group of clusters. If the number of clusters k is given in advance, the simplest way to obtain k clusters is to sort the edges of the minimum spanning tree in descending order of their weights, and remove the edges with the first k_1 heaviest weights. Undesired clustering structures and an unnecessarily large number of clusters are problems commonly faced during MST based clustering.

In [3] Mixture Model based clustering is discussed which unlike the traditional clustering algorithms doesn't rely on heuristics but assumes that the data has been generated from a mixture of multiple probability distributions (Gaussian or multinomial) whose parameters mean, co-variance matrix are to be estimated using the Expectation Maximization algorithm . Subspace clustering [1] [2] is based on key principle which is to discretize the data-space into grids and estimate the density by counting the number of points in a grid cell. Other methods in the literature are Affinity propagation [2] which is based on concept of message passing, Spectral clustering [1] in which the rst k eigenvectors $u_1, u_2, ..., u_k$ corresponding to the k smallest eigenvalues are computed to get matrix $U \in R^{n*k}$ which has $u_1, u_2, ..., u_k$ as columns. Then for $y_i \in R^k$ which is the i^{th} row of U, all rows are treated as points and clustered by k-means to get k clusters. DB-SCAN, OPTICS are based on the concept of density and treat clusters are dense regions connected by less dense regions.

Community detection is another field that deals with obtaining coarse grained descriptions of networks as real world graphs are too large to be analyzed efficiently this is done by utilizing network topology to detect communities of nodes. With development of LDA based models for topic modeling, unified models for joint modeling of underlying topics, author community and link formation were implemented viz. Topic Link LDA and Block LDA. Topic link LDA aims to quantify the effect of topic similarity and community similarity to the formation of a link [16]. Block LDA is a joint model of two components, with one that models links between pairs of entities represented as edges in a graph with a block structure, and the second that models text documents, through shared latent topics. There has also been limited work on combining graph and content information for community discovery leading to techniques such as CESNA, BIGCLAM. CESNA was for statistically modeling the interaction between the network structure and the node attributes where the authors argued that this would lead to more accurate community detection as well as improved robustness in the presence of noise in the network structure [22]. BIGCLAM is another approach that detects both 2-mode as well as cohesive communities which may overlap or be hierarchically nested and is based on affiliation graph models.

III. Mathematical Model

The stochastic generative model for generating communities is presented in this section in which the probability of two entities in data being present in the same community is dependent on the attributes or the annotated text data associated with these nodes. An efficient model fitting procedure is the presented which allows for detecting communities in the network. The current work is based on the assumption that attributes of the data are categorical. The aim is to build upon BIGCLAM, an affiliation model for community detection, however the objective is using attribute information in place of affiliation information.

Directed Attribute Affiliation Model: The community detection methods detect autonomous regions in the network

based on network topology alone while ignoring attribute level information. BIGCLAM however ignores importance of attribute of the nodes being responsible for community creation. Homophily is the tendency to be associated with others who share similar preferences and therefore attribute associated with the entity in a network play an important role in deciding communities. The hypothesis of correlation between attributes and communities is reasonable as its presence is also seen in empirical evidence provided in the literature [22]. Based on this reasoning, a simple conceptual model called Directed Attribute Affiliation Model is formulated. This builds on the family of affiliation network models, but in this work affiliation models are extended to consider attributes.

To represent node and attribute affiliation a bipartite affiliation graph is created where nodes are the data points and attributes to which they belong are shown as the top layer. Cluster affiliation can then be modeled using such a bipartite graph where un-directed edges are formed between nodes and attributes to denote that those nodes contain that attribute. The top layer represents attribute values and bottom layer represents nodes in the data as shown in Figure 1.

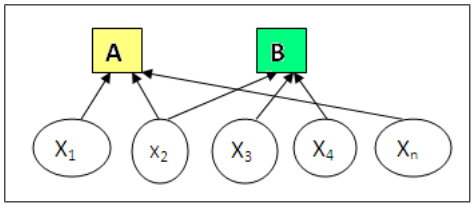

Fig. 1. Bipartite Affiliation Graph

A bipartite affiliation graph is denoted as $B(X, C, M)$, with X as the nodes, C as the attribute value and M denotes the directed edge from X to C if node X has attribute value C. The problem now is to create a set of communities $S = S_1, S_2, ..., S_k$ given $B(X, C, M)$. A parameter p_c is assigned to an attribute value $c \in C$. This is for calculating the probability that a node x_i has the attribute value c. This can also be called the probability that a node x_i belongs to the same community as another x_j having the value of a particular attribute as c. The $P_A(i, j)$ denotes that the nodes i, j belong to the same community A. This can be shown by the below equation.

$$P_A(i,j) = 1 - \prod_{c \in M_i \cap M_j} (1 - p_c) \quad (1)$$

Where,

- M_i = node i has membership to attribute value c.
- M_j = node j has membership to attribute value c.

In Eqn. 1, the value of $P_A(i, j)$ is set to ε, following the BIGCLAM procedure the value of ε can be set as $2|E|/|V|(|V| - 1)$ [21].

A. Calculate the latent weights of the attributes

Every attribute has its own importance or strength in determining the cluster to which the node should belong to, this is denoted here formally as F_{uC}. This is the strength that attribute C has for node u in determining its cluster. Considering this membership strength the Eqn. 1 can be modified as follows:

$$P_A(i,j) = 1 - exp(-F_{uC}.F_{vC}^T) \quad (2)$$

F_{uC} is the membership strength of a single attribute, similarly it is assumed that every node i has a attribute membership vector F_i which contains the membership strengths to all attributes in the data. The modified probability that nodes i, j now share a cluster is Eqn 2.

The intuition behind the above formula is simple, Consider a node having attribute values same as the attribute values of another node, in such a case the likelihood of both nodes belonging to a particular community increases. This means that for each attribute a pair of nodes shares we get an independent chance of grouping the nodes. Thus, naturally, the more attributes a pair of nodes shares, the higher the probability of sharing the same community and being connected.

If $M_u \cap M_v = 0$ then $P(u, v) =$ this is done to consider cases where nodes might not share attributes but still are connected. F_u is the vector that denotes the strengths of association of a node u with each community in the network. The task is to find the matrix of memberships F that maximizes the likelihood of generating the graph $G(V, E)$. The log-likelihood of this is Eqn. 5. The Gradient update algorithm is used to find the value of F as shown in Eqn. 6

$$l(F) = \sum_{u,v \in E} log(1 - \exp(-F_u.F_v^T)) - \sum_{u,v \notin E} (F_u.F_v^T) \quad (3)$$

$$\bigtriangledown l(F_u) = \sum_{v \in N(u)} F_v \frac{\exp(-F_u.F_v^T)}{1 - \exp(-F_u.F_v^T)} - \sum_{v \notin N(u)} F_v \quad (4)$$

Decide Community Affiliation: The membership strengths matrix F is computed from above and the next step is to determine a suitable threshold above which it is possible to determine whether the node i belongs to a community. This threshold is δ set at $\sqrt{log(1 - \varepsilon)}$ [21]. The initialization isnt done using locally minimal neighborhoods approach of BIGCLAM [21] as entity annotated attributes are used to get initial values of the membership strengths F_i. The value of $F_{i,k}$ is 0 if attribute k is present and 0 if absent.

Choosing the number of communities: This is done by procedure specified in [23] where the model is trained using an initial value of K. Then we detect K communities on the 80% of node pairs and then evaluate the likelihood on the hold out set. The K with the best hold out likelihood is used.

B. Model Representation

1) Artificial Neural Networks: Artificial neural networks are composed of input layers, hidden layers and output layers. The number of hidden layers ideally should be more as they can compute more nonlinear relationships between the inputs. However, it is important to avoid over-fitting while deciding the number of hidden layers and the number of hidden units. The input feature matrix X and the weight matrix of the W_{21} (θ_1) and the result are given to the sigmoid function to decide which neurons of the hidden layer 1 i.e. (a_1) are activated. The output of this hidden layer is the multiplied with the second weight matrix W_{31} (θ_2) and so on till the output layer. The back-propagation algorithm is then used to obtain the adjusted weights. The cost function of the network for the classification problem is provided in Eqn. (1) and the penalty term (regularization) is used to it for preventing over-fitting is provided in Eqn.(2).

2) Cost Function - Artificial Neural Networks: The cost function is that of the standard regularized logistic regression which is generalized for k outputs instead of one output. For our problem the output is a single value of 0/1 depending upon whose influence in a network is more. The cost function shall be minimized using the backpropagation algorithm to get the ideal value of the weights.

$$J(\Theta) = \frac{a}{b}\sum_{i=1}^{k}\sum_{k=1}^{k} *[-y_k^i * log((h_\Theta(x^i))_k) - (1-y_k^{(i)}) * log((h_\Theta(x^i))_k)] \tag{5}$$

$$Penalty = \frac{\lambda}{m} * [\sum_{j=1}^{J1}\sum_{k=1}^{K1} *(\Theta_{(j,k))}^{(1))})^2 + \sum_{j=1}^{J2}\sum_{k=1}^{K2} *(\Theta_{(j,k))}^{(2))})^2] \tag{6}$$

3) Gradient boosted trees: Gradient boosting combines several weak learners into a strong learner using multiple iterations. Gradient boosting is applied with decision trees of fixed size as the first level learners. The decision tree partitions the input space into disjoint regions and predicts a constant value in each region. The General idea for a gradient boosted tree is to fit a model to the data i.e F_1(x) = y . Fit the model to the residuals denoted by $h_(1)$(x) = y - F_1(x). Then obtain a new model F_2(x) = F_1(x) + h_1(x). The value of 'm' or the hyper parameter which gives the number of iterations of the residual correction procedure is obtained by cross validation.

4) Gradient Boosted trees: Model Update rule:

$$F_m(x) = F_{(m-1)}(x)) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \epsilon R_{jm}) \tag{7}$$

C. Influence Ranking Algorithm

The algorithm for comparison of influence between two users whose network features have been collected from Twitter is presented. The parameters of various models are set initially and these values are modified till the ideal values are obtained. The cross validation set is used for this purpose.

Algorithm 1 Influencer Ranking model

1: Set initial seed for random numbers
2: Set the training control values
3: Set the tuning grid for parameter search
4: **for** each parameter set **do** do
5: **for** each resampling iteration set **do** do
6: hold out specific samples
7: Pre process the data (Center and Scale)
8: Fit the model on the remaining samples
9: Predict the held out samples
10: **end for**
11: Calculate the average performance across held out predictions
12: **end for**
13: Determine the optimal parameter set
14: Fit the final model to all the training data using optimal parameter set

IV. Experimental Study

A. Experiment setting

The performance of the Influence Ranking algorithm in Section III-(C) is studied to understand the best set of features to measure influence.

1) Dataset: The dataset, provided by Peer-index, comprises a standard, pair-wise preference learning task. Each data point describes two individuals whose identities are anonymized and their future references in the paper are made using labels 'A' and 'B'. For each person, 8 pre-computed, non-negative numeric features based on twitter activity are provided. The binary label represents a human judgement about which one of the two individuals is more influential. The Ground truth labels provided are 0/1 to indicate which of the users is more influential. The test set has 5952 entries for which label has to be predicted.

TABLE I
Description of the dataset

Training set size	Test set size	Feature vector	Classification
5500	5952	22	Binary

B. Results

The accuracy and performance on test set was measured for the Three layered ANN trained using correlated and uncorrelated predictors.

TABLE II
MODELING INFLUENCE USING MULTILAYERED PERCEPTRON WITH BACKPROPAGATION

Sample size	Hidden units	Training Accuracy	Training Kappa
3521	16	72.29	44.55
3521	24	71.29	43.42
3521	32	71.55	43.04

For above experiment correlated features (≥ 0.75) were removed. Cross validation technique used was X-cross fold with X = 5 and 5 times repeat. Training set accuracy was used as selection criteria for evaluation and hidden units were decided as 16. The trained model gave an accuracy of 0.801 on the test set.

TABLE III
MODELING INFLUENCE USING MULTILAYERED PERCEPTRON WITH BACKPROPAGATION

Sample size	Hidden units	Training Accuracy	Training Kappa
3521	22	73.11	46.19
3521	33	72.96	45.89
3521	44	73.08	46.14

For above experiment correlated features were allowed for training. Cross validation technique used was X-cross fold with X = 5 and 5 times repeat. Training set accuracy was used as selection criteria for evaluation and hidden units were decided as 22. The trained model with 22 hidden nodes gave accuracy of 0.81 on test set.

Modeling influence using Decision trees with gradient boosting was performed. To tune the parameters of the trees such as Number of Trees N_{trees}, Shrinkage, Interaction depth and Minimum observations in nodes, a grid search was conducted. Training set Accuracy was the criteria to select the optimal model. Fig. 2 shows the performance of the algorithm during grid search and the accuracy is seen in Fig. 2 to degrade as the interaction depth increases beyond 5 and the shrinkage is increased beyond 0.1. The final parameter values for the optimal model obtained through grid search were N_{trees} = 110, interaction depth = 5, shrinkage = 0.1 and minimum observations in nodes = 20. The accuracy obtained on the test set was 0.8601.

Random forests (Rf) technique to model influence using the correlated features and removing correlation was performed. Advantage of random forests models are better accuracy but at the cost of interpret-ability of the model. The tuning parameter is Number of trees for these model whose value has to be determined empirically. The final model selected using training set accuracy had classification accuracy of 0.856 on the test set. The model with correlated features gave slightly better result 0.859. Removing correlation doesnt improve accuracy on this dataset for the ANN and Rf.

Improving accuracy using ensembling, boosting or bagging at the cost of interpret-ability is a disadvantage for implementation. Such techniques have improved accuracy but such models obtained had more theoretical importance than practical value as seen in the NetFlix competition. Table V contains the training set and test set accuracy of greedy ensembles of GBM Trees, Random forests and ANN obtained in Section IV-B. The test set accuracy of the models is used as the selection criteria and Extreme Gradient Boosting has the highest accuracy on the test set at 0.87. ROC curves to denote the performance of Ensembled models of ANN, GBM, Rf is shown in Fig 3 and Area under curve is used as the selection criteria to obtain the optimal model. The results are shown for each combination in Fig. 3. The correlation between the predictions of GBM and Rf is 0.88 and so their ensemble was discarded due to highly correlated predictions. The highest accuracy on the test set is seen for the ensemble of GBM and Rf as shown in Table V.

TABLE IV
ENSEMBLED PREDICTORS

Sr. No	Technique	Training Accuracy	Test set Accuracy
1	Boosting	0.7856	0.82
2	Greedy Ensemble (ANN,GBM,RF)	0.86	0.867
3	Greedy Ensemble (GBM,RF)	0.86	0.83
4	Averaging Predictors (GBM,RF)	0.79	0.866
5	Extreme Gradient Boosting	0.79	0.87

V. CONCLUSION

The framework for ranking influential nodes in a social network based on characteristics obtained from a nodes interaction on the social network has been presented in this paper. The influence is modeled using machine learning techniques unlike the conventional influence maximization approaches. Features of the individual nodes obtained from its interaction characteristics are used and the network architecture is not considered. The performance metrics of the algorithms used in the experiments is classification accuracy and using this objective criteria Extreme Gradient Boosted decision tree has shown the highest accuracy. The influence ranking algorithm in this paper is a suitable technique for computation of the influence score of various nodes and this has been validated by the extensive experiments performed on Twitter dataset.

REFERENCES

[1] Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S. and Bouras, A., 2014. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. IEEE transactions on emerging topics in computing, 2(3), pp.267-279.

[2] Jain, A.K., 2010. Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), pp.651-666.

[3] Bezdek, J.C., Ehrlich, R. and Full, W., 1984. FCM: The fuzzy c-means clustering algorithm. Computers and Geosciences, 10(2-3), pp.191-203.

[4] Fraley, C. and Raftery, A.E., 2002. Model-based clustering, discriminant analysis, and density estimation. Journal of the American statistical Association, 97(458), pp.611-631.

[5] Ester, M., Kriegel, H.P., Sander, J. and Xu, X., 1996, August. A density-based algorithm for discovering clusters in large spatial databases with noise. In KDD (Vol. 96, No. 34, pp. 226-231).

[6] Ankerst, M., Breunig, M.M., Kriegel, H.P. and Sander, J., 1999, June. OPTICS: ordering points to identify the clustering structure. In ACM Sigmod record (Vol. 28, No. 2, pp. 49-60). ACM.

[7] Rosvall, Martin, and Carl T. Bergstrom. "Maps of random walks on complex networks reveal community structure." Proceedings of the National Academy of Sciences 105.4 (2008): 1118-1123.

[8] Pons, Pascal, and Matthieu Latapy. "Computing communities in large networks using random walks." International Symposium on Computer and Information Sciences. Springer Berlin Heidelberg, 2005.

[9] Clauset, Aaron, Mark EJ Newman, and Cristopher Moore. "Finding community structure in very large networks." Physical review E 70, no. 6 (2004): 066111.

[10] M. E. J. Newman, Fast algorithm for detecting community structure in networks. Phys. Rev. E 69, 066133 (2004).

[11] Newman, Mark EJ, and Michelle Girvan. "Finding and evaluating community structure in networks." Physical review E 69.2 (2004): 026113.

[12] Raghavan, Usha Nandini, Rka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks." Physical review E 76.3 (2007): 036106.

[13] Blondel, and Vincent D., et al. "Fast unfolding of communities in large networks." Journal of statistical mechanics: theory and experiment 2008.10 (2008): P10008.

[14] Reichardt, Jrg, and Stefan Bornholdt. "Statistical mechanics of community detection." Physical Review E 74.1 (2006): 016110.

[15] Newman, Mark EJ. "Finding community structure in networks using the eigenvectors of matrices." Physical review E 74.3 (2006): 036104.

[16] Liu, Y., Niculescu-Mizil, A., and Gryc, W. (2009, June). Topic-link LDA: joint models of topic and author community. In proceedings of the 26th annual international conference on machine learning (pp. 665-672). ACM.

[17] Balasubramanyan, R., & Cohen, W. W. (2011, April). Block-LDA: Jointly modeling entity-annotated text and entity-entity links. In Proceedings of the 2011 SIAM International Conference on Data Mining (pp. 450-461). Society for Industrial and Applied Mathematics.

[18] Xu, W., Liu, X., & Gong, Y. Document clustering based on non-negative matrix factorization. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 267-273). ACM.

[19] Fortunato, S. and Hric, D., 2016. Community detection in networks: A user guide. Physics Reports, 659, pp.1-44.

[20] Fani, Hossein, and Ebrahim Bagheri. Community detection in social networks. Encyclopedia with Semantic Computing and Robotic Intelligence 1.01 (2017): 1630001.

[21] Yang, J. and Leskovec, J., 2013, February. Overlapping community detection at scale: a nonnegative matrix factorization approach. In Proceedings of the sixth ACM international conference on Web search and data mining (pp. 587-596). ACM.

[22] Yang, J., McAuley, J. and Leskovec, J., 2013, December. Community detection in networks with node attributes. In Data Mining (ICDM), 2013 IEEE 13th international conference on (pp. 1151-1156). IEEE.

[23] Yang, J., McAuley, J. and Leskovec, J., 2014, February. Detecting cohesive and 2-mode communities indirected and undirected networks. In Proceedings of the 7th ACM international conference on Web search and data mining (pp. 323-332). ACM.

[24] Grygorash, O., Zhou, Y., & Jorgensen, Z. (2006, November). Minimum spanning tree based clustering algorithms. In Tools with Articial Intelligence, 2006. ICTAI06. 18th IEEE International Conference on (pp. 73-81). IEEE.