International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

A Survey Paper on Data Clustering using Incremental Affine Propagation

Pratap Shinde[1], Madhav Ingle[2]

[1]Department of computer engineering of JSCOE Handewadi Road, Hadapsar, Pune-411028, India

[2]P.G.Coordinator, Computer dept. of JSCOE Handewadi Road, Hadapsar, Pune-411028, India

Abstract: ***Clustering domain is vital part of data mining domain and widely used in different applications. In this project we are focusing on affinity propagation (AP) clustering which is presented recently to overcome many clustering problems in different clustering applications. Many clustering applications are based on static data. AP clustering approach is supporting only static data applications, hence it becomes research problem that how to deal with incremental data using AP. To solve this problem, recently Incremental Affinity Propagation (IAP) is presented to overcome limitations. However IAP is still suffered from streaming data clustering support missing. In this project our main aim is to present extended IAP with support to streaming data clustering. This new approach is called as IAP for Streaming Data Clustering (IAPSDC). First we have to present two IAP clustering methods are presented such as K-medoids (IAPKM) as well as IAP clustering using Nearest Neighbor Assignment (IAPNA). For streaming data with IAP we are using our algorithm for clustering streaming data uses a subroutine called LSEARCH algorithm. The practical work for this project will conducted on real time datasets using Java platform. Though many clustering problems have been successfully using Affinity Propagation clustering, they do not deal with dynamic data. This paper gives insight of incremental clustering approach for a dynamic data. Firstly we discuss the affinity propagation clustering in an incremental space using K-medoids and nearest neighbour algorithm and then Incremental Affinity Propagation.***

Keywords: Incremental Affinity Propagation; Streaming Data Clustering; K-medoids; Nearest Neighbour Assignment.

1. Introduction

Clustering which is also well known as an unsupervised way of classifying the data is a very important part of the data mining. It aims at partioning the patterns into a group also called as cluster with data points having similarity with each other at its maximum compared to the data points in the other clusters. Clustering find a variety of applicatios in the pattern recognition, structure identification in an unstructured data. In 1955 the first clustering algorithm K-means was published. It has been 60 years since the K-mean algorithm for clustering have been proposed but K-mean is widely used even today. Thousands of different clustering algoithms have been proposed since then but the general purpose clustering algorithm is yet to be standardised. This is because the wide range of formats of the unstructured data.

Most of the clustering algorithms deals with the static data, however there was a need for a clustering algoithm that can process data like web pages, blogs, video surveillance which is dynamic in nature. This impose a challenge of rapidly processing large amount of data which is dynamically arriving. Also the storage devices cannot store such a large amount of data and remember that much data which was scanned earlier. Transient streams which cannot be stored on the host machine can only be scaned once so faster proessin of such data stream is required for effective clustering. The clustering algorithm must be able to detect the emerging clusters and also should be able to add the individual data points into a cluster which is having data points with maximum similarity. To handle the high-speed data processing requirements many traditional clustering methods like K-mean, K-medoids have been extended to work in the incremented environment.

K-mean clustering algorithm to process the time stamped parallel data stream[3] has been discussed previously. Affinity propagation clustering is emerging which is an "Exemplar"-based approach. It is given by the assignment of the data points to their nearest exemplar. In this paper we are extending the Affinity propagation approach to work in the streamed data environment. A new approach to handle the streamed dat is proposed to adjust the clustering results as the new object arrives. This reduces the time required to apply the clustering to the whole data set. So, an efficient approach is designed to work with the dynamic data.

2. Related Work

Clustering can be defined as grouping a set of objects into different classes (clusters) so that the similar objects in a perticular sense get added to the same class and the objects with dissimarities get in the different classes. Sometimes the clusterin can be used to form the natural clusters based on the natural hierarchy[4].

Affinity propagation find a wide range of applications in clustering the images of faces, detect genes in microarray data, identify representative sentences in this manuscript, and identify cities that are efficiently accessed by airline travel as mentioned by Brendan J. Frey and Delbert Dueck[5]. To work with the dynamic data or data streams many approaches were proposed in the literature. An incremental Affinity propagation algorithm was proposed aimed at streaming data by Xiangliang Zhang, Cyril Furtlehner[2] in 2008. The bipartite or factor graph is used to represent the massege passing between the different local functions[6][7]. The TimeSseries data streams like stock rate i.e. data items in the real number form were clustered by J. Beringer and E. Hullermeier[4]. Affinity propagation clustering does not need the number of clusters to be specified priviously as that was needed in the former approaches K-mean, instead it takes similarity value $s(k,k)$ as an input for each data point so that

the data point having maximum *s(k,k)* is chosen as an exemplar.

Table 1 presents the summary of the literature so far

K-mean	K-medoids	KNearest Neighbour	Affinity Propagation
k-means clustering partitions N observations into K clusters[9][10]	A medoid can be defined as the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal	In k-Nearest Neighbour classification, the final output of the clustering is a class membership	AP is a clustering technique based on massege passing between the objects of the data sets results into the set of exemplars[1]
Results of this clustering is data space is partitioned into Voronoi cells	k-medoid is a partitioning technique of clustering that clusters the data set of N objects into K clusters. K is known apriori	An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (K is a small positive integer)	An object is assigned to its nearest exemplar base on the similarity value corresponding to the object and exemplars from the exemplar set[1]

3. Problem Formulation of Data Clustering using Incremental Affine Propagation

Since the general problem of clustering is a NP-hard, the goal of the problem definition is to produce an algorith which gives solution to the near optimal solution.

Assume that *{Xt}, t = 1, 2, ... , T;* is a sequentially collected data set, where *Xt* is an $m_t \times d$ matrix, represents m_t is *d*-dimensional objects observed at time stamp *t*. While clustering a static data, time stamp is not considered, and all objects are assumed to be available at once. Therefore, the data set is represented as X_0. It is an $m_0 \times d$ matrix, represents m_0 *d*-dimensional objects. Traditional clustering algorithm is aimed to partition these objects into some groups (e.g. k) such that objects in the same groups or clusters are more similar than the objects in different cluster.

A data stream can be defined as an ordered and sequential data points those can be read only a small number of times. clustering a data stream is expected to be a single-pass algorithm. Only one object is monitored at each time step as per the assumptions, the original data set can be rewritten as *{Xt}, t = 1, 2... ,T*. At time step *t*, the set of all available objects is $Ut = Ut-1 \cup Xt$ the clustering result is c*t*, and the similarity matrix is *St*.

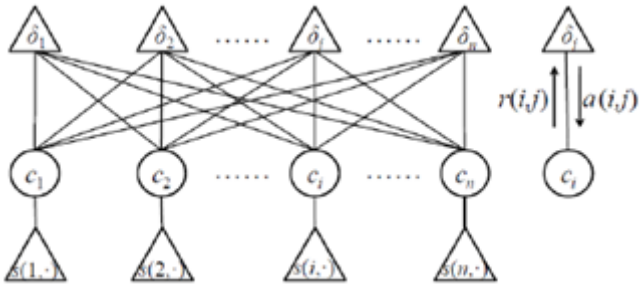

Figure 1: Factor graph of AP clustering. Triangle nodes represent function nodes, circle nodes represent variable nodes. Object function is the sum of all the triangle nodes

Exemplar based clustering is realised by identifying some special kind of objects, called as an exemplars[1]. The other remaining objects are then associated with it's nearest exemplar. The objective of the exemplar based clustering is to minimize the value of

$$z = \sum_{i=1}^{n} s(i, c_i)$$

where $s(i,ci)$ denotes similarity between x_i and its nearest exemplar x_{ci}. The exemplar stores the compressed information about the whole data set that is to be clustered. Finding the exemplars is a Hard combinational optimization problem. The constraint function can be defined as

$$z = \sum_{i=1}^{n} s(i, c_i) + \sum_{j=1}^{n} \delta_j(c)$$

where $c = (c1, c2,..., cn)$. $j(c)$ is constraint function defined as $\delta j(c) = -\infty$ if $cj \neq j$ but $\exists cj=j, \delta j(c)=0$ otherwise. .A valueof $ci = j$ for $i \neq$ j indicates that object *i* is assigned to a cluster with object *j* as its exemplar. A value of $cj = j$ indicates that object *j* is an exemplar. The introduction of penalty term $\delta j(c)$ is to avoid such a situation that object *i* chooses object *j* as its exemplar, but object *j* is not an exemplar at all. The unconstrained optimization problem can be visualized by a bipartite graph in Fig. 1. Triangle nodes represent function nodes, while circle nodes correspond to variable nodes. Object function is the sum of all the function nodes. In Fig. 1, there are two kinds of message passing on graph. They are responsibilities and availabilities. Responsibility *r(i, j)* is sent from variable node *ci* to function node δj . It indicates how strongly object *i* wants to choose candidate exemplar *j* as its exemplar. *r (i, j)* can be computed as follows:

$$r(i, j) \leftarrow s(i, j) - \max\{ a(i, j') + s(i, j')\}$$

Availability *a(i, j)*, sent from function node δ_j to variable node c_i, reflects the accumulated evidence for how well-suited it would be for point *i* to choose point *j* as its exemplar. It is computed as

$$a(i, j) \leftarrow min\left\{0, r(j, j) + \sum_{i'}^{n} max\{0, r(i', j)\}\right\}$$

$$a(i, k) \leftarrow \sum \max\{0, r(i', k)\}$$

The availability a(i ,k) is set to the self responsibility r(k,k) plus the sum of the positive responsibilities candidate exemplar k receives from other points. For a good exemplar only the positive portions of incoming responsibilities are added to explain some data points well regardless of how poorly it explains other data points Negative self responsibility r(k,k) indicates that point k is currently better suited as belonging to another exemplar rather than being an

exemplar itself, the availability of point k as an exemplar can be increased if some other points have positive responsibilities for point k being their exemplar. The total sum is thresholded to limit the influence of strong incoming positive responsibilities so that it cannot go above zero. The ―self-availability‖ a(k,k) is updated differently:

This message reflects that point k is an exemplar sent to candidate exemplar k from other points. The above update rules require only local and simple computations that are easily implemented in equation and messages need be exchanged between pairs of points with known similarities Availabilities and Responsibilities can be combined to identify exemplars at any point during affinity propagation. For point i, the value of k that maximizes a(i,k) + r(i,k) either identifies point i as an exemplar if k = i, or identifies the data point that is the exemplar for point i. After changes in the messages fall below a threshold , the message-passing procedure may be terminated after a fixed number of iterations. To avoid numerical oscillations that arise in some circumstances, it is important that they be damped when updating the messages. Each message is set to l times its value from the previous iteration plus 1 – λ times its prescribed updated value, where the damping factor l is between 0 and 1. Each iteration of affinity propagation consists of :

I. Updating all responsibilities given the availabilities.
II. Updating all availabilities given the responsibilities and
III. Combining availabilities and responsibilities to monitor the exemplar decisions and terminate the algorithm.

Disadvantages of Affinity Propagation

1) It is hard to know the value of the parameter preferences which can yield an optimal clustering solution.
2) When oscillations occur, AP cannot automatically eliminate them.

4. Incremental Affinity Propagation

A semi-supervised scheme called incremental affinity propagation clustering. Incremental Affinity Propagation integrates AP algorithm and semi-supervised learning. The labeled data information is coded into similarity matrix in some way. And the incremental study is performed for amplifying the prior knowledge. In the scheme, the pre-known information is represented by adjusting similarity matrix. An incremental study is applied to amplify the prior knowledge. To examine the effectiveness of this method, concentrate it to text clustering problem. In this method, the known class information is coded into the similarity matrix initially. And then after running AP for a certain number of iterations , the most convinced data are put into the "labeled data set" and reset the similarity matrix. This process is repeated until all the data are labeled. Compared with the method in , the dealing with constrained condition in this scheme is soft and objective. Furthermore, the introduction of incremental study amplifies pre-knowledge about the target data set and therefore leads to a more accurate result. Also, the parameter of "preference" in the method is self-adaptive mainly according to the target number of clusters. Focused on text clustering problem, Cosine coefficient method is used to compute the similarity between two different points (texts) in the specific I-APC algorithm.

In summary, the I-APC scheme can be described as follows:

1. Initialize: including the initializations of labeled data set *L*, the responsibility matrix and availability matrix, similarities between different points and self-similarities say, set as a common value as well.
2. If the size of *L* reaches a pre-given number *P*, goto step 5, else run AP.
3. Select the most convinced data point and then update *L*. Reset similarities between different points , and self-similarities
4. Go to step 2.
5. Output results, end.

The flow chart of the I-APC scheme is shown in Fig 2.

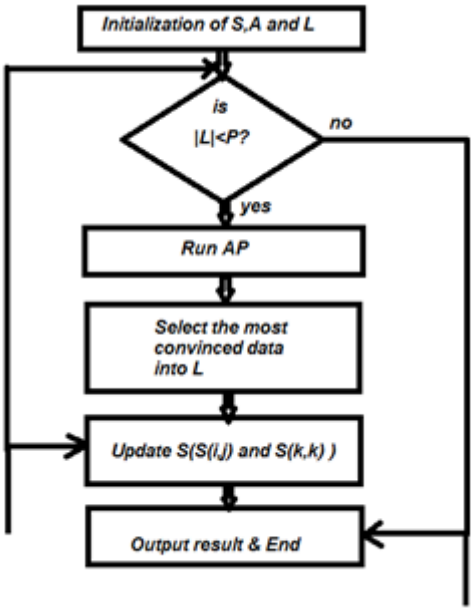

Figure 2: Flowchart of I-APC scheme

5. Conclusion

In this survey, various clustering approaches and algorithms in document clustering are described. . A new clustering algorithm which combines Affinity Propagation with semi-supervised learning ,i.e Seeds Affinity Propagation algorithm is present. In comparison with the classical clustering algorithm k-means, SAP not only improves the accuracy and reduces the computing complexity of text clustering and but also effectively avoids being trapped in local minimum and random initialization. Whereas Incremental Affinity Propagation integrates AP algorithm and semi-supervised learning. The labeled data information is coded into similarity matrix. The Adaptive Affinity Propagation algorithm first computes the range of preferences, and then searches the space to find the value of preference which can generate the optimal clustering results compare to AP approach which cannot yield optimal clustering results because it sets preferences as the median of the similarities.

The area of document clustering has many issues, which need to be solved. We hope , the paper gives interested readers a broad overview of the incremental affinity propagation clustering technique. As a future work, improvement over the existing systems with better results which offer new information representation capabilities with different techniques can be attempted.

References

[1] Leilei Sun, Chonghui Guo, "Incremental Affinity Propagation Clustering Based on Message Passing," IEEE Transactions On Knowledge And Data Engineering Vol:Pp No:99 Year 2014
[2] X. Zhang, C. Furtlehner, and M. Sebag, "Frugal and Online Affinity Propagation," Proc. Conf. francophone sur l'Apprentissage (CAP '08), 2008.
[3] J. Beringer and E. Hullermeier, "Online Clustering of Parallel Data Streams," Data and Knowledge Engineering, vol. 58, no. 2, pp. 180-204, Aug. 2006.
[4] J. Beringer and E. Hullermeier, "Online Clustering of Parallel Data Streams," Data and Knowledge Engineering, vol. 58, no. 2, pp. 180-204, Aug. 2006.
[5] B.J. Frey and D. Dueck, "Response to Comment on 'Clustering by Passing Messages Between Data Points'," Science, vol. 319, no. 5864, pp. 726a-726d, Feb. 2008.
[6] F.R. Kschischang, B.J. Frey, and H.A. Loeliger, "Factor Graphs and the Sum-product Algorithm," IEEE Trans. Information Theory, vol. 47, no. 2, pp. 498-519, Feb. 2001.
[7] J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Constructing Free-Energy Approximations and Generalized Belief Propagation Algorithms," IEEE Trans. Information Theory, vol. 51, no. 7, pp. 2282-2312, July 2005.
[8] L. Ott and F. Ramos, "Unsupervised Incremental Learning for Long-term Autonomy," Proc. 2012 IEEE Int. Conf. Robotics and Automation (ICRA '12), pp. 4022-4029, May 2012,
[9] Adil M. Bagirov, Julien Ugon, Dean Webb, " Fast modifiedglobal k-means algorithm for incremental cluster construction," Pattern Recognition 44 (2011) 866–876
[10] A.K. Jain, "Data Clustering: 50 Years Beyond K-means," Pattern Recognition Letters, vol. 31, no. 8, pp. 651-666, June 2009.