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

Streaming Data Clustering using Incremental Affine Propagation Clustering Approach

Pratap Shinde[1], M. D. Ingle[2]

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

[2]Professor, 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 an 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 propose an algorithm Incremental Affinity Propagation using Streaming Data Clustering (IAPSDC) using the same approach in streaming data clustering. IAPSDC when compared with previously put clustering schemes Incremental Affinity Propagation clustering using K-Medoids and Incremental Affinity Clustering using Nearest Neighbour Assignment (IAPKM and IAPNA) give a comparable result for a streaming data clustering.***

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

1. Introduction

Unlike classification and regression, which analyze class-labeled (training) data sets, clustering analyzes data objects without consulting class labels. In many cases, classlabeled data may simply not exist at the beginning. Clustering can be used to generate class labels for a group of data. The objects are clustered or grouped based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity. That is, clusters of objects are formed so that objects within a cluster have high similarity in comparison to one another, but are rather dissimilar to objects in other clusters. Each cluster so formed can be viewed as a class of objects, from which rules can be derived. Clustering can also facilitate taxonomy formation, that is, the organization of observations into a hierarchy of classes that group similar events together.

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.

Licensed Under Creative Commons Attribution CC BY

2. Literature Survey and Problem Definition

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.

B.J. Frey and D. Dueck in "Response to Comment on Clustering by Passing Messages Between Data Points" proposed Clustering data by identifying a subset of representative examples is important for processing sensory signals and detecting patterns in data. Such "exemplars" can be found by randomly choosing an initial subset of data points and then iteratively refining it, but this works well only if that initial choice is close to a good solution. We devised a method called "affinity propagation," which takes as input measures of similarity between pairs of data points. Real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges. Affinity Propagation suffers from its quadratic complexity in function of the number of data items.

C. Du, J. Yang, Q. Wu, and T. Zhang in "Face Recognition Using Message Passing based Clustering Method" proposed that the traditional subspace analysis methods are inefficient and tend to be affected by noise as they compare the test image to all training images, especifically when there are large numbers of training images. To solve such problem, a fast face recognition (FR) technique called APLDA was proposed by combining a novel clustering method affinity propagation (AP) with linear discriminant analysis (LDA). By using AP on the reduced features derived from LDA, a representative face image for each subject can be reached

a) Problem Formulation

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 $\{X_t\}$, $t = 1, 2, \ldots, T$; 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 $m_o \times d$ 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

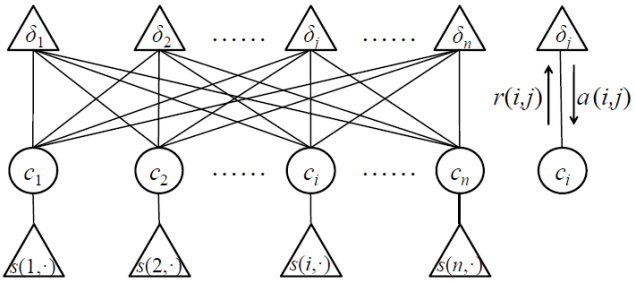


Figure 1: Factor graph of AP clustering. Triangle nodes represent function nodes, circle nodes represent variable nodes. Object function is the sum of all the triangle nodes.

$\{X_t\}$, $t = 1, 2 \ldots, T$. At time step t, the set of all available objects is $U_t = U_{t-1} \cup X_t$ the clustering result is c_t, and the similarity matrix is S_t.

b) Affinity Propagation Clustering

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

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

where $s(i, c_i)$ 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

$$z = \sum_{i=1}^{n} s\,(i, c_i) + \sum_{j=1}^{n} \delta_j\,(\mathrm{c}) \qquad (2)$$

where $c = (c_1, c_2, \ldots, c_n)$. $j(c)$ is constraint function defined as $\delta_j(c) = -\infty$ if $c_j \neq j$ but $\exists c_j = j, \delta_j(c) = 0$ otherwise. A value of $c_i = j$ for $i \neq$ j indicates that object i is assigned to a cluster with object j as its exemplar. A value of $c_j = j$ 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

Licensed Under Creative Commons Attribution CC BY

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 c_i to function node δ_j . It indicates how strongly object i wants to choose candidate exemplar j as its exemplar. $r(i, j)$ can be computed as follows:

$$r(i,j) \leftarrow s(i,j) - \max\{a(i,j') + s(i,j')\} \quad (3)$$

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:

$$a(i,j) \leftarrow \min\{0, r(j,j) + \sum_{i'}^{n} \max\{0, r(i',j)\}\} \quad (4)$$

Responsibilities and availabilities update as (3) and (4) till convergence, then the clustering result $c = (c_1, \ldots, c_n)$ can be obtained by

$$c_i = \arg\max_j \{a(i,j) + r(i,j)\} \quad (5)$$

the Sum of Similarities (SS) is defined as

$$SS = \sum_{i=1}^{n} s(i, c_i) \quad (6)$$

A larger SS indicates a better clustering performance.

3. Incremental AP for Streamed Data Clustering

In this section we propose a new algorithm named Incremental Affinity Propagation for Streamed Data Clustering. The traditional affinity propagation clustering and streamed data clustering is combined to get the benefits of AP in finding the initial exemplar set and later use that exemplar set for clustering the streamed data. AP can find a most liable exemplars automatically. IAPSDC works in two steps: In the first step IAP is provided with the initial batch of the objects as an input and then IAPSDC modify the clusters as the new streamed data is available.

In Fig.2 a schematic graph of IAPSDC is shown where triangular nodes are representing exemplars and circular nodes represent all the objects. Every node in Fig.2 computes according to the message it received, and then sends message to the relevant nodes. The computation of a circle node i is defined as:

$$r_{SDC}(i,j) = arg \max_{j \in E}\{s(i,j)\} \quad (7)$$

Where $E = \{e_1, e_{2,\ldots}, e_k\}$ is current exemplar set. Equation (7) states that circle node i decide which exemplar it is belonging to according to the message it receives, and then tells each exemplar node its choice. The *j-th* triangular node decides which object is the new exemplar of cluster j according to all the circle nodes choices. The formula for calculation is:

$$a_{SDC} = \arg \max_{q, c_q = e_j} \left\{ \sum_{i=1}^{n} f(c_i, q) s(i, c_i) \right\} \quad (8)$$

Where $f(i, j) = 1$, *if* $i = j$; $f(i, j) = 0$, otherwise.

$$\sum_{i=1}^{n} f(C_i, q) s(i, Ci) \quad (9)$$

is sum of all the similarities of cluster j.

Equation (8) tries each member of cluster j to be the exemplar of cluster j. The object is used as the exemplar of cluster j when the sum of similarities of that exemplar is maximized.

Algorithm 1 presents IAPSDC. Traditional AP clustering is implemented on the first batch of objects U_{t-1}, and the clustering result is c_{t-1}. When a new batch of objects S_t are arriving, assign each new object to the current exemplars.

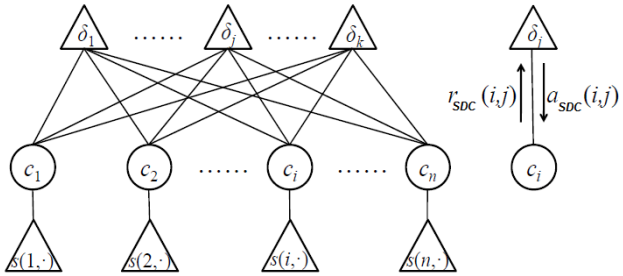


Figure 2: IAPSDC in message-passing manner. The message sent from circle node c_i to triangle node δ_j indicates which exemplar it belongs to, and message sent from triangle node δ_j to circle node indicates which object is the new exemplar of cluster j.

Renew available data set to U_t, and renew label vector c_{t-1} to c_t. Then Streamed Data Clustering is then implemented to modify the result of clustering till to the end of the stream S_t which is assumed to be finite in nature.

Algorithm 1 IAPSDC

Input: U_{t-1}, c_{t-1}, S_t,
Output: c_t ,
Steps:
1: Assign each new object to the current exemplars, and label vector of all the new objects is indicated by c^*_{t-1} ;
2: $U_t = U_{t-1} \cup S_t$, $c_t = [c_{t-1} \quad c^*_{t-1}]$;
3: Message-passing continues according to equation (3) and equation (4);
4: Repeat Step 3 till convergence, c_t is saved.

4. Experiments

We conducted all the experiments on the Intel(R) core(TM) i5, 4200M, with clock speed of 2.50 GHz machine having 4.00 GB RAM. Unlabeled 3 dimensional CAR, WINE, WDBC data sets were used for the experiments. Each data set is split into two parts, first part is used for initial clustering and next part is added later. The platform used for implementing the IAPKM is JAVA and SQL. The front end which includes the GUI and Data Set Manipulation is implemented in JAVA using Net Beans IDE ver 7.2.1. The back end which stores the user detailes and datasets is implemented using the MySQL GUI ver 11.23 of SQLyog Community which is GNU General Public License and is best open source license.

Dataset	*Number of initial objects*	*Number of New objects*	*Precision Coefficient*
CAR	50	10	0.017
WINE	50	20	0.015
WDBC	50	20	0.017

Licensed Under Creative Commons Attribution CC BY

5. Analysis and Results

Algorithm in this paper are based on similarity matrix S. The measure of similarity between two objects is also an important problem in data mining. In this paper, negative square root of Euclidean distance is adopted. A problem of Euclidean distance is that some features with large amplitude often cover the effect of other features. Therefore, a preprocessing is used to normalize the original data set

$$x_i^j = \frac{x_i^j - x_{min}^j}{X_{\max}^j - x_{min}^j} \quad (7)$$

Where x_i^j is the j^{th} feature of object *i* and $X_{\max}^j = \max(x_1^j, x_2^j, \ldots x_{Mt}^j)$, $X_{\min}^j = \min(x_1^j, x_2^j, \ldots x_{Mt}^j)$. s(i, j) gives similarity between object *i* and object *j*. It is defined as

$$s(i, j) = -\sqrt{\|x_i - x_j\|^2} \quad (8)$$

Another parameter needs to be specified is the preference *p*. Generally, larger preference p generates larger number of clusters. Frey et al. suggests it should be the median, or minimum value of similarities[5]. Therefore, the following procedure is employed to determine the value of p

$$p = \min_{i,j}[s(i,j)] - pc \cdot Mt \quad (9)$$

Where Mt is the number of current available objects. pc, short for preference coefficient, is a constant determined by the initial batch of objects. Varying pc and running traditional AP clustering on the first batch of objects, when the number of exemplars is proper, the corresponding value of pc is stored and used in the following incremental clustering.

Accuracy is a measure to state the effectiveness of the clustering algirithm. It is calculated as:

$$Accu = \frac{\sum_{i=1}^{n} \delta\left(c_i, \text{map}\left(\bar{c}_i\right)\right)}{n} \quad (10)$$

where c_i is the real label of object *i*, and where $\bar{c}_i$ is the real label of object *i*, and $\bar{c}_i$ is the obtained clustering label. $\delta(i, j) = 1$, if $i = j$; $\delta(i, j) = 0$, otherwise. Function map() matches true class label and the obtained cluster label.

Experiment results on the four unlabled data sets are shown in Table 2

Table 2: Comparison of Accuracy

Data Set	Method	First	Second
CAR	IAPSDC	0.74	0.74
	IAPKM	0.30	0.30
	IAPNA	0.61	0.61
WINE	IAPSDC	0.91	0.90
	IAPKM	0.91	0.91
	IAPNA	0.90	0.90
WDBC	IAPSDC	0.89	0.89
	IAPKM	0.89	0.89
	IAPNA	0.89	0.90

6. Conclusion

In this paper we have proposed a clustering algorithm IAPSDC to use Incremental Affinity Propagation for the streamed data. IAPSDC when compared to the AP, IAPKM and IAPNA give comparable results. Two popular unlabeled datasets are used to evaluate the IAPSDC. Results of the experiments show the effectiveness of the IAPSDC. The proposition IAPSDC is inspired by the combination of Incremental Affinity Propagation and Streamed Data Clustering. Affinity propagation is very effective in finding out the initial exemplar set which can later be used to cluster the streamed data points coming as an ordered sequential data. Streamed clustering is a branch of incremental data clustering. Some other incremental clustering problems are also of great importance.

Additionally, some other problems such as how to measure similarity between objects, and how to extract features from time series and labelled data set are also of great importance. However, that is not the focus of the paper. It can be a future scope of this paper.

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
[6] Passing Messages Between Data Points'," Science, vol. 319, no. 5864, pp. 726a-726d, Feb. 2008.
[7] 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.
[8] 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.
[9] 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,
[10] Adil M. Bagirov, Julien Ugon, Dean Webb, " Fast modifiedglobal k-means algorithmforincrementalclusterconstruction," Pattern Recognition 44 (2011) 866–876
[11] A.K. Jain, "Data Clustering: 50 Years Beyond K-means," Pattern Recognition Letters, vol. 31, no. 8, pp. 651-666, June 2009.

Licensed Under Creative Commons Attribution CC BY