Preeti Kashyap, Babita Ujjainiya / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 3, May-Jun 2013, pp.274-282

A Survey On Seeds Affinity Propagation

Preeti Kashyap, Babita Ujjainiya
(Department of Information Technology, SATI College, RGPV University,Vidisha (M.P), India)
(Ass. Prof In Department of Information Technology, SATI College, RGPV University, Vidisha (M.P), India)

ABSTRACT

Affinity propagation (AP) is a clustering method that can find data centers or clusters by sending messages between pairs of data points. Seed Affinity Propagation is a novel semi-supervised text clustering algorithm which is based on AP. AP algorithm couldn't cope up with part known data direct. Therefore, focusing on this issue a semi-supervised scheme called incremental affinity propagation clustering is present in the paper where pre-known information is represented by adjusting similarity matrix The standard affinity propagation clustering algorithm also suffers from a limitation that it is hard to know the value of the parameter "preference" which can yield an optimal clustering solution. This limitation can be overcome by a method named, adaptive affinity propagation. The method first finds out the range of "preference", then searches the space of "preference" to find a good value which can optimize the clustering result.

Keywords- Affinity propagation, Clustering, Incremental, Partition adaptive affinity propagation and Text clustering.

I. INTRODUCTION

Clustering is a process of organizing objects into groups whose members are similar in some way. Cluster analysis seeks to partition a given data set into groups based on specified features so that the data points within a group are more similar to each other than the points in different groups [1], Therefore, a cluster is a collection of objects that are similar among themselves and dissimilar to the objects belonging to other clusters. Text Clustering is to divide a set of text into cluster , so that text within each cluster are similar in content. Clustering is an area of research, finding its applications in many fields. One of the most popular clustering method is k-means clustering algorithm. Arbitrarily k- points are generated as initial centroids where k is a user specified parameter. Each point is then assigned to the cluster with the closest centroid then the centroid of each cluster is updated by taking the mean of the data points of each cluster. Some data points may move from one cluster to other cluster. Again calculate new centroids and assign the data points to the suitable clusters. Repeat the assignment and update the centroids, until convergence criteria are met. In this algorithm mostly Euclidean distance is used to find distance between data points and centroids. Standard K-Means method has two limitations: (1) the number of cluster needs to be specified first. (2) the clustering result is sensitive to the initial cluster centers .

Traditional approaches for clustering data are based on metric similarities, i.e. symmetric, nonnegative, and satisfying the triangle inequality measures. More recent approaches, like Affinity Propagation (AP) algorithm [2], can take as input also general non metric similarities. In the domain of image clustering, AP can use as input metric selected segments of images pairs [3]. AP has been used to solve a wide range of clustering problems [4] and individual preferences predictions [5].The clustering performance depends on the message updating frequency and similarity measure AP has been used in text clustering for its simplicity, good performance and general applicability. By using AP to preprocess text for text clustering. It was combined with a parallel strategy for e-learning resources clustering. But AP was used only as an unsupervised algorithm and did not consider any structural information derived from the specific documents. For text mining tasks, the vector space model (VSM), which treats a document as a bag of words and uses plain language words as features [6]. This model can represent the text mining problems directly and easily. With the increase of data set size, the vector space becomes sparse, high dimensional, and the computational complexity grows exponentially. In many practical applications, unsupervised learning is lacking relevant information whereas supervised learning needs an initial large number of class label information, which requires time and expensive human labor. [7], [8]. In recent years, semi-supervised learning has captured a great deal of attentions [9], [10]. Semi-supervised learning is a machine learning in which the model is constructed using both labeled and unlabeled data for training—typically a small amount of labeled data and a large amount of unlabeled data.

.

II . RELATED WORK

This section includes so far study on Affinity propagation.

2.1 Affinity Propagation

AP, a new and powerful technique for exemplar learning. It is a fast clustering algorithm especially in the case of large number of clusters and has some advantages: speed, general applicability and good performance. In brief , AP works based on similarities between pairs of data points and simultaneously considers all the data points as potential cluster centers called exemplar. It computes two kinds of messages exchanged between data points. The first one is called "responsibility" and the second one is "availability". Affinity propagation takes as input a collection of real-valued similarities between data points, where the similarity s(i,k) indicates how well the data point with index k is suited to be the exemplar for data point i. When the goal is to minimize squared error, each similarity is set to a negative squared error i.e Euclidean distance: For point's x_i and x_k,

$$s(i,k)=-||x_i-x_k||^2 \quad (1)$$

The two kinds of messages are exchanged between data points, and each takes into account a different kind of competition. Messages can be combined at any stage to decide which points are exemplars and, for every other point, which exemplar it belongs to. The "availability" a(i, k) message is sent from candidate exemplar point j to point i and it reflects the accumulated evidence for how appropriate it would be for point i to choose point j as its exemplar. The responsibility" r(i, k) message is sent from data point i to candidate exemplar point j and it reflects the accumulated evidence for how well-suited point j is to serve as the exemplar for point i. At the beginning, the availabilities are initialized to zero: a(i ,j) = 0.Then, the responsibilities are computed using the rule

$$r(i,k) \leftarrow s(i,k) - \max\{ a(i,k') + s(i,k')\} \quad (2)$$

The availabilities are zero in the first iteration, r(i,k) is set to the input similarity between point i and k as its exemplar, minus the largest of the similarities between point i and other candidate exemplars. This update is data-driven and does not take into account how many other points favor each candidate exemplar. In later iterations, when some points are effectively assigned to other exemplars, their availabilities will drop below zero. These negative availabilities will decrease the effective values of some of the input similarities s(i,k′) , removing the corresponding candidate exemplars from competition. For k = i, the responsibility r(k,k) is set to the input preference that point k be chosen as an exemplar, s(k,k), minus the largest of the similarities between point i and all other candidate exemplars. This self-responsibility reflects that point k is an exemplar.

$$a(i,k) \leftarrow \min\{0, r(k,k) + \sum \max\{0, r(i' ,k)\}\} \quad (3)$$

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:

$$a(i,k) \leftarrow \sum \max \{ 0, r(i', k)\} \quad (4)$$

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 eq. (3) 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 :

1. Updating all responsibilities given the availabilities.
2. Updating all availabilities given the responsibilities and
3. Combining availabilities and responsibilities to monitor the exemplar decisions and terminate the algorithm.

2.1.1 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.

2.2 Seeds Affinity Propagation

Seeds Affinity Propagation is based on AP method. The main new features of the new

algorithm are: Tri-Set computation, similarity computation, seeds construction, and messages transmission [11] Similarity measurement are as follows: Co-feature Set (CFS), Unilateral Feature Set (UFS), and Significant Co-feature Set (SCS). The structural information of the text documents is included in the new similarity measurement.

2.2.1 Similarity Measurement

Similarity measurement plays a major role in Affinity Propagation clustering. To give specific and effective similarity measurement for our particular domain, i.e., text document these three feature sets are used . Co-feature Set, the Unilateral Feature Set, and the Significant Co-feature Set.. In this approach, each term in text is still deemed as a feature and each document is still deemed as a vector. All the features and vectors are not computed simultaneously, but one at a time. Let $D = \{d_1, d_2 \ldots d_N\}$ be a set of texts. Suppose, d_i and d_j are two objects in D and can be represented using the following two subsets:

$d_i = \{<f_i^1, n_i^1>, <f_i^2, n_i^2>, \ldots . <f_i^L, n_i^L> \},$
$d_j = \{<f_j^1, n_j^1>, <f_j^2, n_j^2>, \ldots <f_j^M, n_j^M>\},$

where f_i^x and f_j^y ($1 \le x \le L$, $1 \le y \le M$) in the two tuples $<fi^x, n_j^x>$, $<f_j^y, n_j^y>$ represent the x*th* and y*th* feature of *di* and *dj*, respectively. n_i^x and n_j^y are the values of f_i^x and f_j^y. L and M are the counts of the objects features.

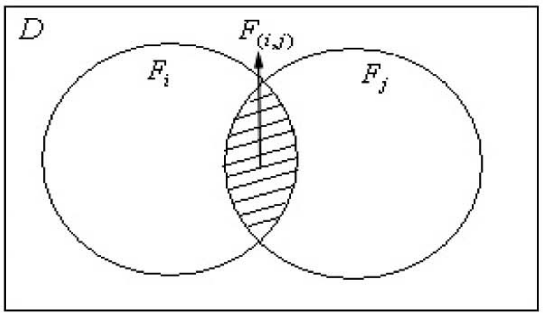
Let *Fi* and *Fj* be the feature sets of the two objects, respectively: $F_i = \{ f_i^1, f_i^2, \ldots f_i^L \}$, $F_j = \{ f_j^1, f_j^2 \ldots . f_j^M \}$. Let the set DF_j composed of the "most significant" features of d_j. Most significant means features that are capable of representing crucial aspects of the document. These most significant features could be key phrases and/or tags associated with each document when available or they could be all the words except stop words in the title of each document. The venn diagram is shown in Fig1.

Definition 1. *Co-feature Set.*
Let *di* and *dj* be two objects in a data set. Suppose that some features of *di*, also belong to *dj*. A new two-tuples subset consisting of these features and their values in *dj* can be constructed.

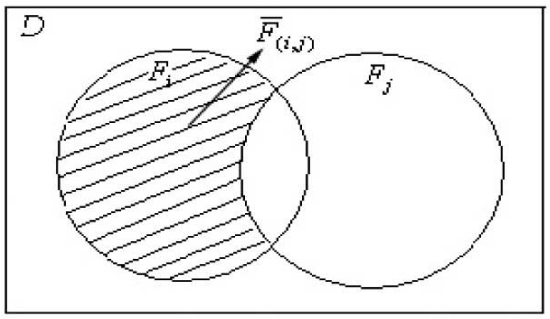
Definition 2. *Unilateral Feature Set.*
Suppose that some features of *di*, do not belong to *dj*. Two-tuples subset consisting of these features and their values in *di can be constructed.*

Definition 3. *Significant Co-feature Set.*
Suppose that some features of *di*, also belong to the most significant features of *dj*. A new two-tuples subset consisting of these features and their values as the most significant features in dj can be constructed. Thus extending the generic definition of the similarity measures based on the Cosine coefficient by introducing the three new sets CFS UFS and SCS namely, Tri-Set similarity.

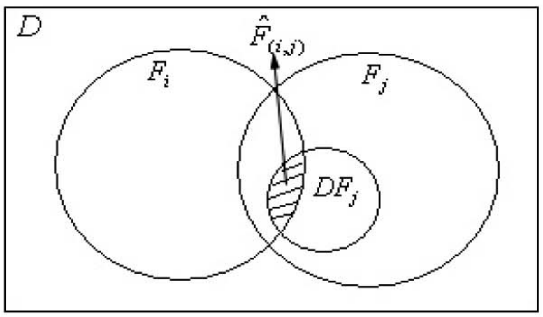
This extended similarity measure can reveal both the difference and the asymmetric nature of similarities between documents. It is quite effective in the application of Affinity Propagation clustering for text documents, image processing and so on, since it is capable of dealing asymmetric problem. The combination of this new similarity with conventional Affinity Propagation is names as Tri-Set Affinity Propagation (AP(Tri-Set)) clustering algorithm.



(a)



(b)



(c)

of d_j. D is the whole data set.

2.2.2 Seed Construction

In semi-supervised clustering, the main goal is to efficiently cluster a large number of unlabeled objects starting from a relatively small number of initial labeled objects. Given a few initial labeled objects, it can be use to construct efficient initial seeds for our Affinity Propagation clustering

algorithm. To avoid a blind search guarantee precision for seeds and imbalance errors, a specific seeds' construction method is presented, namely Mean Features Selection. Let N^O, N^D, N^F and F^C represent, respectively, the object number, the most significant feature number, the feature number, and feature set of cluster C in the labeled set. Suppose F is the feature set and DF is the most significant feature set of seed c .For example, DF of this manuscript could be all the words except stop words in the title, i.e.{Survey, Seeds , Affinity, and Propagation}. Let $f_k \in F_C$; $f_k \in F_C$. Their values in cluster c are nk and nk', the values of being the most significant feature are n_{DK} $(0 \le n_{DK} \le n_k)$ and n_{DK} $(0 \le n_{DK} \le n_K)$. The seeds construction method is prescribed as:

1. iff $n_k \ge \sum \frac{n_k}{N^o}$, $f_k \in F$;
2. iff $n_{DK} \ge \sum \frac{n_{DK}}{N^o}$, $f_{k'} \in DF$.

This method can find out the representative features in labeled objects quickly. Seeds are made up of these features and their values in different clusters. They should be more representative and discriminative than the normal objects. For seeds, their self-similarities are set to $+\infty$ to ensure that the seeds will be chosen as exemplars and help the algorithm to get the exact cluster number. The combination of this semi-supervised strategy with classical similarity measurement and conventional Affinity Propagation is named as Seeds Affinity Propagation with Cosine coefficient (SAP(CC)) clustering algorithm. By introducing both seed construction method and the new similarity measurement into conventional AP, the definition of the complete "Seeds Affinity Propagation algorithm can be generated.

2.2.3 Seeds Affinity Propagation Algorithm

Based on the definitions of SCS, UFS and the described seeds' construction method, the SAP algorithm is developed, with this sequence of steps:

1. Initialization: Let the data set D be an N (N >0) terms superset where each term consists of a sequence of two-tuples:

$$D = \begin{vmatrix} \{<f_1^1, n_1^1>, <f_1^2, n_1^2>, \ldots <f_1^{M1}, n_1^{M1}>\} \ldots \\ \{<f_N^1, n_N^1>, <f_N^2, n_N^2> \ldots <f_N^{MN}, n_N^{MN}>\} \end{vmatrix}$$

where M_x represents the count of the xth object's feature.

2. Seeds construction: Constructing seeds from a few labeled objects according to Mean Features Selection. Adding these new objects into the data set D, and getting a new data set D' which contains N' terms (N≤N');
3. Tri-Set computation: Computing the Co-feature set, Unilateral Feature Set and Significant Co-feature Set between objects i and j.
4. Similarity computation: Computing the similarities among objects in D
5. Self-Similarity computation: Computing the self-similarities for each object in D'.
6. Initialize messages: Initializing the matrixes of messages
   r (i ,j)= s(i,j) - max {s(i,j')}, a(i,j)=0
7. Message matrix computation: Computing the matrixes of messages.
8. Exemplar selection: Adding the two message matrixes and searching the exemplar for each object i which is the maximum of r (i, j) + a(i, j).
9. Updating the messages
10. Iterating steps 6, 7, and 8 until the exemplar selection outcome stays constant for a number of iterations or after a fixed number of iterations end the algorithm.

To summarize, start with the definition of three new relations between objects. Then, assign the three feature sets with different weights and present a new similarity measurement. Finally, a fast initial seeds construction method is defined and detail the steps of the new Seeds Affinity Propagation algorithm in the general case.

2.3.4 Advantages of Seeds Affinity Propagation

1. Reduces the time complexity and improves the accuracy.
2. Avoids Being random initialization and trapped in local minimum.

3.1 Incremental Affinity Propagation

A semi-supervised scheme called incremental affinity propagation clustering is present in the paper.[12] . 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 [13], 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 [14] to compute the similarity between two different points (texts) in the specific I-APC algorithm.

3.1.1 Incremental Affinity Propagation Algorithm

In most cases of real clustering problems, a part of data set, generally a minority of the whole, may have been labeled in prior. To take advantage of the valuable and may be expensive resource ,a semi-supervised scheme based on AP which is called Incremental Affinity Propagation Clustering (I-APC) algorithm is presented.

The similarity between data point *i* and *j* . *s(i,j),* indicates how well the data point *i* is suited to be the exemplars, for data point j. Then if both the points are labeled into the same class, they are more likely to support each other to be an exemplar. Therefore, *s(i,j)* should be set much larger than usual. On contrary, if the two points are labeled differently, *s(i. j*) should be set much smaller.

In the method, incremental learning technique is applied to amplify our pre-knowledge. After each run of AP, the most convinced data point is added to the labeled data set. For non-labeled data point *k*, a score function *score (k,i)* represents how likely the point belongs to cluster *i:*

$$\text{Score}(k,i) = \alpha.\sum(a(k,j)+r(k,j)) + \beta . \sum(a(k,j)+r(k,j)) \quad \textbf{(5)}$$

where *L* is the labeled data set, *I* the data set contained the i_{th} clustering according to the last time results running by AP, and *a* and *β* the coefficients of the so-called convinced and non-convinced items, respectively. Here $a > \beta > O$. Then a convinced function of point *k* can be defined according to the score function:

$$conv(k) = \frac{\min \ \max \text{score}(k,m)}{\text{score}(k,n)} \quad \textbf{(6)}$$

Therefore the most convinced data point is selected by maximizing whose convinced function value within the non-labeled data.

After the labeled data set is updated and the similarity matrix is reset according to the new labeled data. The effect of the labeled data decreases when the labeling time increases. So if points *i* and *j* both are in the labeled data set and at least one is a new labeled, the s *(i,j)* is set to be a function of program time *t:*

$$S(i,j) = -|X_i - X_j|\,(A+e^{-Bt})^{1-2|\text{sign}(Ci-Cj)|} \quad \textbf{(7)}$$

where

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

and *B* are two constants, *t* is the program time, which could be considered as the iteration number of AP running, and C_i and C_j are the labeled numbers for X_i and X_j, respectively.
During the process, preference *s(i,j)* is also adjustable. After a run of AP, when the resulted clustering number is larger than expected, the values of preferences should be reduced as a whole, and vice versa.The rule is as follows:

$$s(i,i) = s(i,i).\ \frac{1}{1+e^{-K'/K}} \quad \textbf{(8)}$$

where K is the expected clustering number and K' the resulted clustering number, respectively.
About the responsibilities and availabilities, their values are kept at the end of last run as the initial values of the next time. It will speed up the convergence of the 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 (according to Eqs, (1) and (7) 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 to Eqs (5-6)and then update *L*. Reset similarities between different points ,according to *Eq 7* and self-similarities ,according to *Eq. 8*
4. Go to step 2.
5. Output results, end.

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

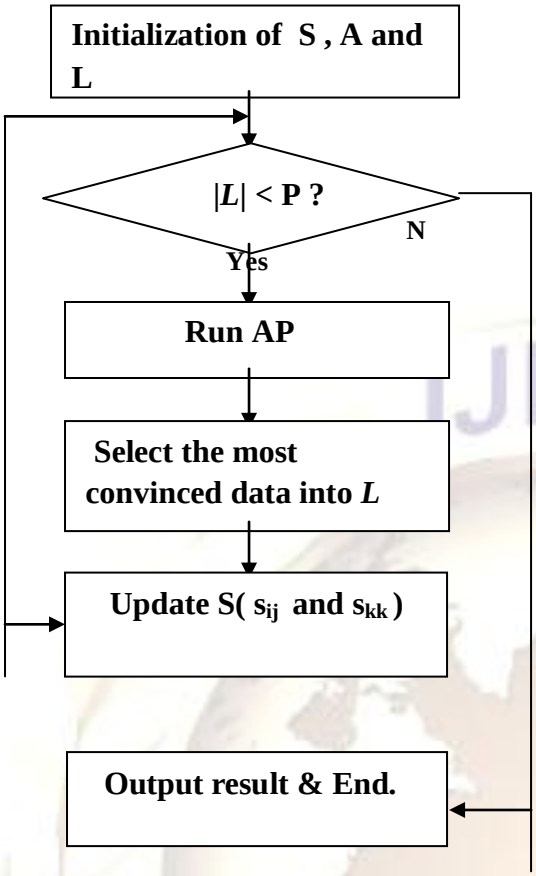


Fig 2. Flowchart of I-APC scheme

3.1.2 Incremental Affinity Propagation For Text Clustering

Text clustering is the process to divide a whole set of texts into several groups according to the similarities among all the texts. It is an important research field of text mining and is often used for benchmark for the new developed clustering methods. For text clustering, each text is considered as a point in the problem space. So the similarity between two different points couldn't be set as negative squared Euclidean distance any longer.

Several similarity measurements are commonly used in information retrieval. The simplest one is Simple matching coefficient method, where the idea is to count the number of shared terms. A most powerful method is Cosine Coefficient. Therefore Cosine coefficient is used to measure the point similarity in this method. During the process, the similarity matrix should be reset according to the new labeled data and s*(i , j)* is a time-dependent function based on the original distance.

3.1.3 Disadvantage of Incremental Affinity Propagation

1. I-APC method costs more CPU time than AP.
2. The performance will be tempered with when the incremental process is run too much. So the selection of the threshold is important.

4.1 Adaptive Affinity Propagation

The affinity propagation clustering algorithm suffers from one limitation that it is hard to know the value of the parameter "preference" which can yield an optimal clustering solution. This limitation can be overcome by a method named, adaptive affinity propagation.[15] The method first finds out the range of "preference", then searches the space of "preference" to find a good value which can optimize the clustering result.

4.1.1 Computing the range of preferences

Affinity propagation tries to maximize the net similarity [16]. Net similarity is a score for explaining the data, and it represents how appropriate the exemplars are. The score sums up all similarities between data points and their exemplar The similarity between exemplar to itself is the preference of the exemplar. Affinity propagation aims at maximizing Net Similarity and tests each data point whether it is an exemplar. Therefore, the method which is using for computing the range of preferences can be developed just as shown in Fig 1.

Algorithm 1 Preference Range Computing

Input: s(i,k) : the similarity between point i and point k ($i \neq k$)
Output: the maximal value and minimal value of preferences; p_{max} , p_{min}
Step1. Initialize s(k, k) to zero:
$s(k, k) = 0$
Step2: Compute the maximal value of Preferences:
$P_{max} = max\{s(i , k)\}$

Step3: Compute the minimal value of Preferences
Step 3.1 Compute the net similarity when the number of clusters is 1:
$dpsim1 = max\{\sum s(i,j)\}$
Step 3.2 Compute the net similarity when the number of clusters is 2:
$dpsim2 = max\{\sum max\{s(i,k), s(j,k)\}$
Step3.3 Compute the minimal value of Preferences:
$P_{min} = dpsim1 - dpsim2$

Fig3: The procedure of computing the range ***of*** *preferences*

The maximum preference (p_{max}) in the range is the value which clusters the N data points

into N clusters, and this is equal to the maximum similarity, since a preference lower than that would make the object better to have the data point associated with that maximum similarity assigned to be a cluster member rather than an exemplar.

The derivation for p_{min} is similar to p_{max}. Suppose that there is a particular preference p' such that the optimal net similarity for one clusters (k=1) and the optimal net similarity for two clusters (k=2) are the same. Optimal net similarity of two clusters can be obtained by searching through all possible pairs of possible exemplars, and the value is $dpsim2+2* p'$. If there is one cluster, the value of optimal net similarity is $dpsim1 + p'$. The minimum preference p_{min} leads to clustering the N data points into one cluster. Since affinity propagation aims at maximizing the net similarity, that is dpsim1 + $p' \geq dpsim2+2*p'$, Then $p' \leq dpsim1-dpsim2$. P_{min} no more than p' , therefore, the minimum value for preferences is $p_{min} = dpsim1-dpsim2$.

4.1.2 Adaptive Affinity Propagation Clustering

After computing the range of preferences, preferences space can be scan to find the optimal clustering result. Different preferences will result different cluster. Cluster validation techniques are used to evaluate which clustering result is optimal for the datasets. Preference step is very important to scan the space adaptively. It is denoted as:

$$p_{step} = \frac{p_{max} - p_{min}}{N * 0.1 * \sqrt{K+50}}$$

In order to sample the whole space, set The base of scanning step *as* $\frac{p_{max} - p_{min}}{N}$

This fixed increasing step cannot meet the different requirement of different cases such as more clusters and less clusters. Because more-clusters case is more sensitive than that of less-cluster case. Therefore the adaptive step method similar to Wang's[17], an adaptive coefficient is more useful ,

$$q = \frac{1}{0.1 \sqrt{K+50}}$$

In this way, the value of *step p* with the count of clusters (K) is set . When K is large, p_{step} will be small and vice versa.

In this paper, global silhouettes index are validity indices. Silhouettes is introduced by Rousseeuw [18] as a general graphical aid for interpretation and validation of cluster analysis, which provides a measure data point classification when it is assigned to a cluster in according to both the tightness of the clusters and the separation between them. Global silhouette index is defined as follows:

$$GS = 1 \sum \frac{S_j}{n_c}$$

Where local silhouette index is:

$$S_j = 1 \sum_{r_j} \frac{b(i) - a(i)}{max\{b(i),a(i)\}}$$

Where r_j the count of the objects in class j, a(i) is the average distance between object i and the objects in the same class j, is the minimum average distance between object i and objects in class closet to class j.

Fig 2 shows the procedure of the adaptive affinity propagation clustering method. The largest global silhouette index indicates the best clustering quality and the optimal number of clusters[19]. A series of *Sil* values corresponding to clustering result with different number of cluster are calculated. The optimal clustering result is found when *Sil* is largest.

Algorithm 3 Adaptive affinity propagation Clustering:

Input: s(i,k)*: the similarity between point i and point k (i≠ k)*

Output: *the clustering result*

Step1: Apply Preferences Range algorithm to computing the range of preferences:
$[p_{min}, p_{max}]$

Step2: Initialize the preferences:
$preference = p_{min} - p_{step}$

Step3: Update the preferences:
step preference = preference + p

Step4: Apply Affinity Propagation algorithm to generating K clusters

Step5: Terminate until Sil is largest.

Fig4; The procedure of adaptive propagation clustering

4.1.3 Adaptive Affinity Propagation Document Clustering

This section discusses the adaptive affinity document clustering, which implements the adaptive affinity propagation algorithm in clustering documents, combined with Vector Space Model (VSM). Vector Space Model is the most common model for representing document among many models of document representation. In VSM, every document is represented as a vector:
$V(d) = (t_1 , w_1(d); t_2, w_2(d); \ldots t_m, w_m (d))$, where t_1 is the word item, $w_1(d)$ is the weight of t_1 in the

document d. The most widely used weighting scheme is Term Frequency with Inverse Document Frequency (TF-IDF).

5.1 Partition Adaptive Affinity Propagation

Affinity propagation exhibits fast execution speed and finds clusters with low error rate when clustering sparsely related data but its values of parameters are fixed. Partition adaptive affinity propagation can automatically eliminate oscillations and adjust the values of parameters when rerunning affinity propagation procedure to yield optimal clustering results, with high execution speed and precision [20]

The premise is that both AP and AAP are a message communication process between data points in a dense matrix. The time spent is in direct ratio to the number of iterations. During each iteration of AP, each element r(i, *k)* of the responsibility matrix must be calculated once and each calculation must be applied to N-1 elements, where N is the size of the input similarity matrix, according to Eq.(2).And each element of the availability matrix can be calculated in the same way. During an iteration of AAP, the convergent of K is detected but the execution speed is still much lower than AP.

Partition adaptive affinity propagation (PAAP). This modified algorithm can eliminate oscillations by using the method of AAP. Our adaptive technique consists of two parts. One is called fine adjustment, another is coarse adjustment. Fine adjustment is used to decrease the values of parameter "preference "slowly, and coarse adjustment is used to rapidly decrease the values of preference correspondingly. The original similarity matrix is decomposed into sub-matrices to gain higher execution speed [21] [22], when executing our method. PAAP can yield optimal clustering solutions on both dense and sparse datasets.

Assuming that *Cmax* is the expected maximal number of clusters, *Cmin* is the expected minimal number of clusters, and *K* (i) is the number of clusters in the iteration, and *maxits* is the maximal number of iterations. λ*step* and *Pstep* are the adaptive factors as in AAP. The PAAP algorithm goes as follows:

Algorithm PAAP algorithm:

1. Execute AP procedure, get the number of clusters:K(i).
2. If $K(i) \leq K(i+1)$, then go to step 4. Else, *count* == 0, then go to step 3.
3. $\lambda \leftarrow \lambda + \lambda_{step}$, then go to step 1. If $A > 0.85$, then $p \leftarrow p + p_{step}$, s(i,i) ←p, Else go to step 1.
4. If $|Cmax - K(i)| > CK$, then $Astep == -20 * |K(i) - Cmin|$ Go to step 6. Else, delay 10 iterations and then go to step 5.
5. If $K(i) \leq K(i+1)$, then *count* == *cout* + 1, *Astep* == *count* *Pstep*. Go to step 6 or Else, go to step 1.
6. *p* == *p* + *Astep*, then *s(i,* i) ← *p.*
7. If i == *maxits* or *K(i)* ~ *Cmin*, the algorithm terminates. Else, go to step 1.

PAAP can find the true or better number of clusters with high execution speed on dense or sparse datasets , meanwhile, it can automatically detect the number oscillations and eliminate them. This verified that both acceleration technique and partition technique are effective. If *Kpart s* and *A step* (acceleration factor) is well chosen, the average number of iteration can be reduced effectively.

5.1.2 Advantages of partition Adaptive affinity propagation.

1. PAAP improved approach based on affinity propagation. It can automatically escape from the oscillation and adjust values of parameters λ and *p*.

III. 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 existing techniques. 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] S. Deelers and S. Auwatanamongkol, *"Enhancing K-Means Algorithm with Initial Cluster Centers Derived from Data Partitioning along the Data Axis with the Highest Variance,"*

International Journal of Electrical and Computer Engineering 2:4 2007

[2] B.J. Frey and D. Dueck, *"Clustering by Passing Messages between Data Points,"* Science, vol. 315, no. 5814, pp. 972-976, Feb. 2007.

[3] B.J. Frey and D. Dueck, *"Non-Metric Affinity Propagation for Un-Supervised Image Categorization,"* Proc. 11th IEEE Int'l Conf. Computer Vision (ICCV '07), pp. 1-8, Oct. 2007.

[4] L. Michele, Sumedha, and W. Martin, *"Clustering by Soft- Constraint Affinity Propagation Applications to Gene-Expression Data,"* Bioinformatics, vol. 23, no. 20, pp. 2708-2715, Sept. 2007.

[5] T.Y. Jiang and A. Tuzhilin, *"Dynamic Micro Targeting: Fitness- Based Approach to Predicting Individual Preferences,"* Proc. Seventh IEEE Int'l Conf. Data Mining (ICDM '07), pp. 173-182, Oct. 2007.

[6] F. Sebastiani, "*Machine Learning in Automated Text Categorization,"* ACM Computing Surveys, vol. 34, pp. 1-47, 2002.

[7] F. Wang and C.S. Zhang, *"Label Propagation through Linear Neighbourhoods,"* IEEE Trans. Knowledge and Data Eng., vol. 20, no. 1, pp. 55-67, Jan. 2008.

[8] Z.H. Zhou and M. Li, *"Semi-Supervised Regression with Co- Training Style Algorithms,"* IEEE Trans. Knowledge and Data Eng., vol. 19, no. 11, pp. 1479-1493, Aug. 2007.

[9] A. Blum and T. Mitchell, *"Combining Labeled and Unlabeled Data with Co-Training,*" Proc. 11th Ann. Conf. Computational Learning Theory, pp. 92-100, 1998.

[10] Z.H. Zhou, D.C. Zhan, and Q. Yang, *"Semi-Supervised Learning with Very Few Labeled Training Examples,"* Proc. 22nd AAAI Conf. Artificial Intelligence, pp. 675-680, 2007

[11] Renchu Guan, Xiaohu Shi, Maurizio Marchese, Chen Yang, and Yanchun Liang "*Text Clustering with Seeds Affinity Propagtaion"* IEEE Transactions on Knowledge and data Engineering , VOL. 23, NO. 4, APRIL 2011

[12] H.F. Ma, X.H. Fan, and J. Chen, *"An Incremental Chinese Text Classification Algorithm Based on Quick Clustering,"* Proc. 2008 Int'l Symp. Information Processing (ISIP '08), pp. 308- 312, May 2008.

[13] Y. Xiao, and J. Yu, *"Semi-Supervised Clustering Based on Affinity Propagation," Journal of Software, Vol. 19, No. 11, November 2008, pp. 2803-2813.*

[14] C. J. van Rijsbergen, *Information Retrieval*, 2nd edition, Butterworth, London, pp. 23-28, 1979.

[15] K.J. Wang, J.Y. Zhang, D. Li, X.N. Zhang, and T. Guo, *"Adaptive Affinity Propagation Clustering,"* Acta Automatica Sinica, vol. 33, no. 12, pp. 1242-1246, Dec. 2007

[16] *FAQ of Affinity Propagation Clustering*: http://www.psi.toronto.edu/affinitypropagation/faq.html

[17] K.J. Wang, J.Y.. Zhang, and D. Li. "*Adaptive Affinity Propagation Clustering.*" Acta Automatic Sinica, 33(12): 1242-1246, 2007

[18] P.J. Rousseeuw, Silhouettes: *"a graphical aid to the interpretation and validation of cluster analysis*", Computational and Applied Mathematics. (20),53-65, 1987

[19] S. Dudoit, J. Fridlyand. *"A prediction-based resampling method for estimating the number of clusters in a dataset".* Genome Biology,3(7): 0036.1-0036.21, 2002

[20] Changyin Sun, Chenghong Wang, Su Song,Yifan Wang *"A Local Approach of Adaptive Affinity Propagation Clustering for Large Scale Data"* Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19, 2009

[21] Guha, S., Rastogi, R., Shim, K., "CURE: *an efficient clustering algorithm for large databases,"* Inf.Syst., 26(1): 35-58, 2001.

[22] Ding-yin Xia, Fei Wu, Xu-qing Zhang, Yue-ting Zhuang, *" Local and global approaches of affinity propagation clustering for large scale data,"* J Zhejiang Univ SCi A, , pp.1373 1381,2008.

.