AN EFFICIENT TEXT CLUSTERING ALGORITHM USING AFFINITY PROPAGATION

Kumar Vasantha*
Asst Professor, Dept of Computer Science and Engineering, Avanthi Institute of Engineering & Technology,Narsipatnam, Andhra Pradesh.
Jagadeesh Majji
Final M.Tech Student, Department of Computer Science & Engineering, Avanthi Institute of Engineering& Technology, Narsipatnam, Andhra Pradesh.

Abstract
The objective is to find among all partitions of the data set, best publishing according to some quality measure. Affinity propagation is a low error, high speed, flexible, and remarkably simple clustering algorithm that may be used in forming teams of participants for business simulations and experiential exercises, and in organizing participants preferences for the parameters of simulations.This paper proposes an efficient Affinity Propagation algorithm that guaranteesthe same clustering result as the original algorithm after convergence. The heart of our approach is (1) to prune unnecessary message exchanges in the iterations and (2) to compute the convergence values of pruned messages after the iterations to determine clusters.

1.Introduction

The problem of clustering has been studied widely in the databaseand statistics literature in the context of a wide variety of data mining tasks. The clustering problem is defined to be that of findinggroups of similar objects in the data.The similarity between the objects is measured with the use of a similarity function. The problemof clustering can be very useful in the text domain, where the objectsto be clusters can be of different granularities such as documents, paragraphs, sentences or terms. Clustering is especially useful for organizingdocuments to improve retrieval and support browsing.

There are so many feature selection and transformation methods.

1.1 Document Frequency-based Selection

The simplestpossible method for feature selection in document clustering is that ofthe use of *document frequency* to filter out irrelevant features. Whilethe use of inverse document frequencies reduces the importance of suchwords, this may not alone be sufficient to reduce the noise effects ofvery frequent words. In other words, words which are too frequent inthe corpus can be removed because they are typically common wordssuch as “a”, “an”, “the”, or “of” which are not discriminative from aclustering perspective. Such words are also referred to as *stop words*.Typically commonly available stop word lists ofabout 300 to 400 words are used for the retrieval process. In addition,words which occur extremely infrequently can also be removed fromthe collection. Thisis because such words do not add anything to thesimilarity computations which are used in most clustering methods.

In some cases, such words may be misspellings or typographical errors indocuments. Noisy text collections which are derived from the web, blogsor social networks are more likely to contain such terms. We note thatsome lines of research define document frequency based selection purelyon the basis of very infrequent terms, because these terms contribute theleast to the similarity calculations. However, it should be emphasizedthat very frequent words should also be removed, especially if they arenot discriminative between clusters. Note that the TF-IDF weightingmethod can also naturally filter out very common words in a “soft” way.Clearly, the standard set of stop words provide a valid set of words toprune. Nevertheless, we would like a way of quantifying the importanceof a term directly to the clustering process, which is essential for moreaggressive pruning. We will discuss a number of such methods below.

1.2. Term Strength

The core idea of this approachis to extend techniques which are used in supervised learning to theunsupervised case. The term strength is essentially used to measurehow informative a word is for identifying two related documents. Forexample, for two related documents x and y, the term strength $s(t)$ ofterm t is defined in terms of the following probability:

$$s(t) = P(t \quad y | t \quad x) \quad (1)$$

Clearly, the main issue is how one might define the document *x* and *y*as related. One possibility is to use manual (or user) feedback to definewhen a pair of documents are related. This is essentially equivalentto utilizing supervision in the feature selection process, and may bepractical in situations in which predefined categories of documents areavailable. On the other hand, it is not practical to manually createrelated pairs in large collections in a comprehensive way. It is thereforedesirable to use an automated and purely unsupervised way to definethe concept of when a pair of documents is related. Apair of documents are defined to be related if their cosine similarity isabove a user-defined threshold. In such cases, the term strength *s*(*t*)can be defined by randomly sampling a number of pairs of such relateddocuments as follows:

s(*t*) =Number of pairs in which *t* occurs in both / Number of pairs in which *t* occurs in the first of the pair (2)

Here, the first document of the pair may simply be picked randomly.In order to prune features, the term strength may be compared to theexpected strength of a term which is randomly distributed in the trainingdocuments with the same frequency. If the term strength of *t* is not atleast two standard deviations greater than that of the random word,then it is removed from the collection.

1.3. Similarity Measures

Before clustering, a similarity/distance measure must be determined. The measure reflects the degree of closeness orseparation of the target objects and should correspond tothe characteristics that are believed to distinguish the clusters embedded in the data. In many cases, these characteristics are dependent on the data or the problem context athand, and there is no measure that is universally best for allkinds of clustering problems.Moreover, choosing an appropriate similarity measure is alsocrucial for cluster analysis, especially for a particular typeof clustering algorithms. Density-based clustering finds clusters as dense areas in the data set, and the density of agiven point is in turn estimated as the closeness of the corresponding data object to its neighboring objects. Recallingthat closeness is quantified as the distance/similarity value,we can see that large number of distance/similarity computations are required for finding dense areas and estimatecluster assignment of new data objects. Therefore, understanding the effectiveness of different measures is of greatimportance in helping to choose the best one.

Euclidean distance is a standard metric for geometrical problems. It is the ordinary distance between two points and canbe easily measured with a ruler in two- or three-dimensionalspace. Euclidean distance is widely used in clustering problems, including clustering text. It satisfies all the above fourconditions and therefore is a true metric. It is also the default distance measure used with the K-means algorithm.Measuring distance between text documents, given two documents da and db represented by their term vectors t_a and $t_b$respectively, the Euclidean distance of the two documentsis defined as

$$D_E(\vec{t_\alpha},\vec{t_b}) = (\sum_{t=1}^{m} |w_{t,\alpha}-w_{t,b}|^2)^{1/2}$$

where the term set is $\vec{T} = \{t1, \ldots, tm\}$. As mentionedpreviously, we use the tf idf value as term weights, that is

$$w_{t,\alpha} = tfidf(d_\alpha,t).$$

Affinity Propagation is derived as anapplication of the max-sum algorithm in a factor graph,i.e., it searches for the minima of an energy function on thebasis of message passing between data points [7]. Theclustering performance depends on the similarity measureand message updating frequency. For its simplicity, generalapplicability, and good performance, AP has already beenused in text clustering. By using AP to preprocess texts, Maet al. developed an incremental method [11] for textclustering. Wang et al. combined AP with a parallelstrategy for e-learning resources clustering [12]. However,they used AP only as an unsupervised algorithm and didnot consider any structural information derived from thespecific documents.

For text mining tasks, the majority of state-of-the-artframeworks employ the vector space model (VSM), whichtreats a document as a bag of words and uses plain languagewords as features [13], [14]. This model can represent the textmining problems easily and directly. However, with theincrease of data set size, the vector space becomes highdimensional,sparse, and the computational complexitygrows exponentially. Moreover, in many practical applications,completely unsupervised learning is lacking relevantinformation. On the other hand, supervised learning needsan initial large number of class label information, whichrequires expensive human labor and time [15], [16]. Therefore,in recent years, semisupervised learning has captured agreat deal of attentions [17], [18], [19], [20], [21]. Semisupervisedlearning is a machine learning paradigm in whichthe model is constructed using both labeled and unlabeleddata for training—typically a small amount of labeled dataand a large amount of unlabeled data [16], [22].

To examine the effectiveness of the proposed method, wehave applied it to the benchmark data set Reuters-21578. In order to analyze the behaviour of the new algorithm (andalso the impact of the two individual proposed contributions),we have performed a detail comparison with fourclustering methods on the same data set, namely,

1. K-Mediods Approach

2.Fast Algorithm of Affinity Propagation Approach.

2. Related Work

2.1. Affinity Propagation

Affinity Propagation is a clustering algorithm that identifiesa set of 'exemplars' that represents the dataset [Freyand Dueck, 2007]. The input of Affinity Propagation isthe pair-wise similarities between each pair of data points,s[i, j](i, j = 1, 2, . . .,N) 1. Any type of similarities is acceptable,e.g. negative Euclidean distance for real valueddata and Jaccard coefficient for non-metric data, thus AffinityPropagation is widely applicable.Given similarity matrix s[i, j], Affinity Propagation attemptsto find the exemplars that maximize the net similarity,i.e. the overall sum of similarities between all exemplarsand their member data points. The process of Affinity Propagationcan be viewed as a message passing process with twokinds of messages exchanged among data points: responsibilityand availability. Responsibility, r[i, j], is a messagefrom data point i to j that reflects the accumulated evidencefor how well-suited data point j is to serve as the exemplarfor data point i. Availability, a[i, j], is a message from datapoint j to i that reflects the accumulated evidence for how appropriateit would be for data point i to choose data point j asits exemplar. All responsibilities and availabilities are set to0 initially, and their values are iteratively updated as followsto compute convergence values:

r[i, j] = (1 − λ)ρ[i, j] + λr[i, j]

a[i, j] = (1 − λ)α[i, j] + λa[i, j](2)

where λ is a damping factor introduced to avoid numerical oscillations,and ρ[i, j] and α[i, j] are, we call, propagating responsibilityand propagating availability, respectively. ρ[i, j]and α[i, j] are computed by the following equations:

$$\rho[i,j] = \begin{cases} s[i,j] - \max_{k \neq j}\{\alpha[i,j] + s[i,k]\} & (i \neq j) \\ s[i,j] - \max_{k \neq j}\{s[i,k]\} & (i = j) \end{cases} \quad (3)$$

$$\alpha[i,j] = \begin{cases} \min\{0, r[j,j] + \sum_{k \neq i,j} \max\{0, r[k,j]\}\} & (i \neq j) \\ \sum_{k \neq i} \max\{0, r[k,j]\} & (i = j) \end{cases} \quad (4)$$

That is, messages between data points are computed from thecorresponding propagating messages. The exemplar of data

Point i is finally defined as:

$$\text{argmax}\{r[i,j] + \alpha[i,j] : j = 1,2\ldots..,N\} \quad (5)$$

As described above, the original algorithm requiresO(N2T) time to update massages, where N and T are thenumber of data points and the number of iterations, respectively.This incurs excessive CPU time, especially when thenumber of data points is large. Therefore, a fast Affinity Propagationalgorithm is demanded as pointed out in [Jia et al.,2008].

In the existing algorithms for the K-means problem, we find that AP performs at least as well as the competingalgorithms in terms of quality. However, due to a memory footprint of O(N2), the algorithmcannot be applied on datasets where the number of data points N is large. Another reason whyAP is not very suited for large N is its O(N2) scaling of the runtime per iteration. The K-meansalgorithm and deterministic annealing (DA) have a runtime that scales with O(NKD).Therefore, when the dimension D and number of clusters K is small, DA and K-means have amuch lower runtime. We observe, that AP's runtime is mostly independent of the dimension Dand the number of clusters K. That means, when K and D is large, e.g K = 50 and D = 100,AP can be much faster than K-means algorithm and DA. Also, the K-means algorithm is notonly slow for large K but has severe problems to find good solutions. Hence, AP works wellin settings where the K-means algorithm has problems. Compared to hierarchical clusteringalgorithms, e.g. Ward's method, AP generally runs much slower. When clusters are well-definedand there is only little noise in the dataset, the performance is comparable. If that is not thecase, AP finds better solutions.

3.Proposed System

3.1. K-Nearest neibourhood algorithm

The theorem presented in the last section shows sufficient conditions under which clustering can beperformed consistently. Now we want to present a generic algorithm which can be used to minimizearbitrary clustering

objective functions. With help of Theorem 1 we can then prove the consistencyof its results for a large variety of clustering objective functions.

We have seen that the key to obtain consistent clustering schemes is to work with an appropriate function class. But of course, given quality functions Q and Qn, the question is how such a function space can be constructed in practice. Essentially, three requirements have to be satisfied:

• The function space Fn has to be "small". Ideally, it should only contain polynomially manyfunctions.

• The function space Fn should be "rich enough". In the limit $n \to \infty$, we would like to be able to approximate any (reasonable) measurable function.

• We need to be able to solve the optimization problem $\operatorname{argmin}_{f \in Fn} Qn(f)$.

This sounds trivial atfirst glance, but in practice is far from easy.One rather straightforward way to achieve all requirements is to use a function space of piecewise constant functions. Given a partitioning of the data space in small cells, we only look atclusterings which are constant on each cell (that is, the clustering never splits a cell). If we makesure that the number of cells is only of the order log(n), then we know that the number of clusterings is at most Klog(n) = nlog(K)

, which is polynomial in n. In the following we will introduce a data-dependent random partition of the space which turns out to be very convenient.

We will construct a function class Fn as follows. Given a finite sample $X1,...,Xn \in Rd$, the numberK of clusters to construct, and a number $m \in N$ with $K \leq m \ll n$, randomly pick a subset of m

"seed points" Xs1,...,Xsm. Assign all other data points to their closest seed points, that is for allj = 1,...,m define the set Zj as the subset of data points whose nearest seed point is Xsj. In otherwords, the sets Z1,...,Zm are the Voronoi cells induced by the seeds Xs1 ,...,Xsm . Then consider all partitions of Xn which are constant on all the sets Z1,...,Zm. More formally, for given seeds we define the set Fn as the set of all functions

$$\mathcal{F}_n := \{f : \mathcal{X} \to \{1,\ldots,K\} \mid \forall\, j=1,\ldots,m: \forall z,z' \in Z_j : f(z) = f(z')\}.$$

Obviously, the function class Fn contains Kmfunctions, which is polynomial in n if the number mof seeds satisfies $m \in O(\log n)$. Given Fn, the most simple polynomial-time optimization algorithmis then to evaluate Qn(f) for all $f \in Fn$ and choose the solution $fn = \operatorname{argmin}_{f \in Fn} Qn(f)$. We call the resulting clustering the nearest neighbor clustering and denote it by NNC(Qn). The entire algorithm is summarized in Figure 1. We have already published results on the empirical performance.

3.2.Nearest Neighbor Clustering NNC(Qn), naive implementation

Parameters: number K of clusters to construct, number $m \in N$ of seed pointsto use (with $K \leq m \ll n$), clustering quality function Qn

Input: data set Xn = {X1,...,Xn}, distances di j = d(Xi,Xj)

• Subsample m seed points from the data points, without replacement.

• Build the Voronoi decomposition Z1,...,Zm of Xn based on the distancesdi j using the seed points as centers

• Define Fn :={f : Xn → {1,...,K} | f constant on all cells Zj

• For all $f \in Fn$ evaluate Qn(f).

Output: $fn := \operatorname{argmin}_{f \in Fn} Qn(f)$

B. Fast Seeds Affinity Propagation

For resolving the computation time issue of the original AffinityPropagation algorithm, Jia et al. recently proposed FSAP[Jia et al., 2008]. One promising idea for improving thespeed of Affinity Propagation is to reduce the number of messagevalues that need to be computed. FSAP aims to reflectthis idea as follows. The first stage of FSAP constructs aK-nearest neighbor graph. If data point i is among the Kdata points that have the largest similarity with data point j,then data point i and j are connected by an edge, otherwisenot. Since FSAP performs message transmissions on the Knearestneighbor graph, too many exemplars (at least N/K)might be generated. Therefore, in order to mergemultiple exemplarsinto one cluster, the second stage adds further edgesbased on the following three criteria:

1. If data point i is the exemplar of data point j, then datapoint i and j are connected by an edge;

2. For two data points i and j, if there exists two data pointsmand n that take data point i and j as their exemplar, respectively,and data point m and n are K-nearest neighbour to each other, and so data point i and j are connectedby an edge; and

3. For two data points i and j, if they are connected bycriterion 2, then all data points that choose data point ias exemplar are connected to data point j, and vise versa.After convergence, the exemplar of data point i is finally determined by Equation (5).They showed that their approach is much faster than theoriginal algorithm described in Section 2. However, FSAPis based on heuristic ideas, i.e., the linked edges are determinedbased on K-nearest

neighbor approximation andheuristic criteria. Therefore, FSAP does not guarantee thesame result as the original Affinity Propagation algorithm.Our algorithm presented in this paperis faster than FSAP while it still theoretically guaranteesthe exactness of the clustering results after convergence.

Algorithm:

Input: pair-wise similarities

Output: exemplars of each data point

1: for each data point pair [i, j] do
2: compute a[i, j], r[i, j], and a[i, j] by Equation (5-7);
3: end for
4: for each data point pair [i, j] do
5: if r[i, j]≥0 or a[i, j]+s[i, j]≥maxk_=j{a[i, k]+s[i, k]}then
6: link data point pair [i, j];
7: end if
8: end for
9: for t = 1 to T do
10: for each linked data point pair [i, j] do
11: update r[i, j] and a[i, j] by Equation (1);
12: end for
13: end for
14: for each unlinked data point pair [i, j] do
15: compute r[i, j] = ρ[i, j] and a[i, j] = α[i, j];
16: end for
17: for each data point i do
18: compute exemplar by Equation (4);
19: end for

4.Conclusion

Our Affinity propagation approach improves clustering process.We used K-nearest neighbourhood method to calculate the distances between the words and then fast affinity propagation used for clustering.It reduces unnecessary message exchangesin the iterations, and reduces the convergencevalues of reduced messages from those of un-pruned messages.Experiments show that our algorithm can achieve efficient clustering without sacrificing clustering accuracy. Affinity propagation is a low error, high speed, flexible, and an easy-to-code clustering algorithm that identifies clusters, exemplars, and outliers.

References

[1] Y.J. Li, C. Luo, and S.M. Chung, "Text Clustering with Feature Selection by Using Statistical Data," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 5, pp. 641-652, May 2008.
[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] 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.
[7] W.H. Wang, H.W. Zhang, F. Wu, and Y.T. Zhuang, "Large Scale of E-Learning Resources Clustering with Parallel Affinity Propagation," Proc. Int'l Conf. Hybrid Learning 2008 (ICHL '08), pp.1-10, Aug. 2008.
[8] 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.
[9] 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.
[10] S. Yu, B. Krishnapuram, R. Rosales, H. Steck, and R.B. Rao, "Bayesian Co-Training," Advances in Neural Information Processing Systems, vol. 20, pp. 1665-1672, MIT Press, 2008.
[11] 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.
[12] O. Chapelle, B. Scho¨lkopf, and A. Zien, Semi-Supervised Learning. MIT Press, 2006.
[13] L.P. Jing, M.K. Ng, and J.Z. Huang, "An Entropy Weighting Kmeans Algorithm for Subspace Clustering of High-Dimensional Sparse Data," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 8, pp. 1026-1041, Aug. 2007.
[14] Hung-Leng Chen, Kun-Ta Chuang, and Ming-Syan Chen, "On Data Labeling for Clustering Categorical Data", *IEEE transactions on knowledge and data ngineering*, vol. 20, no. 11, November 2008
[15] Yongli Liu, Yuanxin Ouyang, Hao Sheng, Zhang Xiong, " An Incremental Algorithm for Clustering Search Results", *IEEE International Conference on Signal Image Technology and Internet Based Systems* in 2008.
[16] Hai-Dong MENG, Yu-Chen SONG, Shu-Ling WANG." An Incremental Clustering Algorithm Based on Subcluster Feature" ,*The 1st International Conference on Information Science and Engineering (ICISE2009)* in 2009.

[17] Renchu Guan, Xiaohu Shi, Maurizio Marchese, Chen Yang and yanchun Liang, "Text Clustering with Seeds Affinity Propogation", IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No.4, April 2011.
[18] Sergio M. Savaresi and Daniel L. Boley, "On the performance of bisecting K-means and PDDP",*Proceedings of the 1* ergio M. Savaresi and Daniel L. Boley, "On the performance of bisecting K-means and PDDP",*Proceedings of the 1st SIAM ICDM*, Chicago, IL, 2001.*st SIAM ICDM*, Chicago, IL, 2001.
[19] http://www.psi.toronto.edu/affinitypropagation/faq.html, 2010. [[20 Z.H. Zhou and M. Li, "Distributional Features for Text Categorization," IEEE Trans. Knowledge and Data Eng., vol. 21, no. 3, pp. 428-442, Mar. 2009.
[20] S. Huang, Z. Chen, Y. Yu, and W.Y. Ma, "Multitype Features Coselection for Web Document Clustering," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 4, pp. 448-458, Apr. 2006.
[21] X.D. Wu et al., "Top 10 Algorithms in Data Mining," Knowledge and Information Systems, vol. 14, no. 1, pp. 1-37, Jan. 2008.
[22] M.J. Brusco and H.F. Kohn, "Comment on 'Clustering by Passing Messages between Data Points,'" Science, vol. 319, no. 5864, p. 726c, Feb. 2008.
[23] J. Wu, F. Ding, and Q.L. Xiang, "An Affinity Propagation Based Method for Vector Quantization," Eprint arXiv 0710.2037, http://rxiv.org/abs/0710.2037v2, Oct. 2007.
[24] 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.
[25] S. Gao, W. Wu, C.H. Lee, and T.S. Chua, "A Maximal Figure-of- Merit (MFoM)-Learning Approach to Robust Classifier Design for Text Categorization," ACM Trans. Information Systems, vol. 24, no. 2, pp. 190-218, Apr. 2006.