Replication on Affinity Propagation: Clustering by Passing Messages Between Data Points

Zhe Zhao

Abstract

In this project, I choose the paper, Clustering by Passing Messages Between Data Points [1], published in Science 2007, as the work to replicate. In [1], a new clustering algorithm named Affinity Propagation is proposed. In my report, three research hypothesizes in this paper that are related to the performance, evaluation and scalability of the algorithm are studied. I implement the algorithm and conduct extensive experiments to test these three research hypothesizes. Based on the results, some of the hypothesizes are accepted and some are rejected. I also try to use Affinity Propagation in the task of clustering tweets as the extension of this project.

I. Introduction

Affinity Propagation (AP) , is a new clustering algorithm proposed by Brendan J. Frey and Delbert Dueck from the University of Toronto, Canada. A new clustering algorithm is studied in their paper[1]. Clustering analysis [2] is a main task of exploitative data mining, and a common technique for statistical data analysis used in many fields. Clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (defined by specific similarity measurement) to each other than to those in other clusters. The title of their paper, Clustering by Passing Messages Between Data Points , from Science 2007, addressed the most important difference of Affinity Propagation comparing to other clustering algorithm such as the well-known centroid-based clustering algorithms K-centers. What this paper does is clustering data by updating types of real-valued messages between data points until a high-quality set of exemplars (center of clusters) and corresponding clusters gradually emerges.

Some examples of clustering analysis are shown in figure 1 and figure 2. Figure 1 shows clustering 2-dimensional data points into 4 clusters, and figure 2 shows using clustering algorithm to find cities and locations in a night scene photo of Europe. Clustering data based on a measure of similarity is a critical step in scientific data analysis and in engineering systems. Clustering algorithms can be used in various of fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

A common approach is called centroid-based clustering. In centroid-based clustering, clusters are represented by a central vector, which can be either a member of the data set (for example, K-Centers) or not (for example K-Means). This paper studied the case in which the centers can only be selected from actual data points. These centers are called “exemplars.” The famous k-centers clustering techniques [3] can be used to find exemplars. When the number of clusters is fixed to k, K-means clustering or K-centers

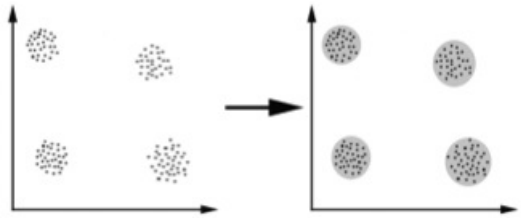

Fig. 1. Example of Clustering Algorithm

Fig. 2. Location Identification by Clustering

clustering uses data to learn a set of centers such that the sum of squared errors between data points and their nearest centers is small. The k-centers clustering technique begins with an initial set of randomly selected exemplars and iteratively refines this set so as to decrease the sum of squared errors. k-centers clustering is quite sensitive to the initial selection of exemplars, so it is usually rerun many times with different initializations in an attempt to find a good solution. However, this works well only when the number of clusters is small and chances are good that at least one random initialization is close to a good solution.

In the replicated paper of this project [1], the authors took a quite different approach and introduced a method that simultaneously considered all data points as potential exemplars. By viewing each data point as a node in a network, they devised a method that recursively transmits real-valued messages along edges of the network until a good set of exemplars and corresponding clusters emerges. Messages were updated on the basis of simple formulas that search for minima of an appropriately chosen energy function. At any point in time, the magnitude of each message reflects the current affinity that one data point has for choosing another data point as its exemplar, so this method is also called affinity propagation by the authors.

In this project, I replicated the Affinity Propagation algorithm to test three hypothesizes made in this paper and evaluate their algorithm in different datasets. The research hypothesizes lying in three aspects to analyze the clustering algorithms are shown as follows.

- The first research hypothesis is made on the performance of the AP algorithm: The proposed algorithm can achieve better performance than centroid based algorithms such as K-centers. The performance includes both effectiveness, i.e., the correctness of clustering and efficiency.
- The second research hypothesis is related to the evaluation method used in this paper: The average squared error for the clustering results is a good metric to evaluate clustering performance.
- The third research hypothesis is made by the authors on the scalability of the AP algorithm: In practical, the running time of this algorithm scales linearly with the number of similarities.

The three hypothesizes are either explicitly made by the authors or implicitly used to support their conclusion. The detail description and formulation of these research hypothesizes can be found in section III in this report.

To replicate this paper and test the hypothesizes, I first implemented their algorithm, tested the code by some random generated toy example and replicating the experiments conducted in the paper. Then to do further test, I extended the code so that it can support sparse similarity matrix as input to deal with data points in a larger scale. Although the codes of the algorithm used in this paper are published on the first author's website, the codes used in this report are written by myself because this is easier for me to tune parameters, and conduct further experiments not covered in this paper to test the research hypothesizes. Then, to test the three hypothesizes in this paper, I designed different types of experiment on two datasets,

i.e., 2-dimensional data points and 900 face images. I also used the algorithm to do tweets clustering, to show the potential ability of this algorithm facing social media data, which usually contains large amount of inaccurate and noisy user generated content.

The rest of this report is organized as follows. Section II introduces the key idea of Affinity Propagation and how it works. Section III discusses the three research hypothesizes in this paper and how I designed the experiments to test these hypothesizes. Section IV covers the topic of using AP to cluster tweets. Experiment results are shown in section V, followed with discussions based on these results in section VI. Section VII is the overview of the work I have done in this project and finally section VIII concludes this report.

II. Affinity Propagation

Affinity Propagation, first proposed in the paper, Clustering by Passing Messages Between Data Points [1], is a new clustering algorithm proposed by professor Brendan Frey and his PhD student Delbert Dueck at that time from University of Toronto, Canada. This paper published in Science on Feb 2007 may change people's way of thinking clustering algorithms in recent future. Before this paper, clustering algorithms such as K-means or K-centers [3] proposed by J. MacQueen 40 years ago are widely used in most clustering problems. Instead of studying the general clustering algorithm, work studying on clustering or similar tasks in specific application, has been proposed [4], [5]. The method proposed in this paper, which is a general clustering algorithm can be used in various of applications, may have huge impact in clustering analysis. This method is compared to K-Centers and shown to be more effective than K-centers in their paper.

Affinity propagation takes as input measures of similarity between pairs of data points and simultaneously considers all data points as potential exemplars. Real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges. Comparing to K-Centers and other centroid based clustering algorithms, AP make use of the information of all data points from neighborhood. The algorithm tries to build a network structure by passing messages between connected data points. Another difference is that AP doesn't need to know the number of clusters. However, in AP, instead of the number of clusters K, an initial preference for each data point need to be assigned, which would be even harder to decide than K without knowing the range of data point values.

Next two subsections first introduce the two types of real-valued messages passed between data points, and then show some experiment results in the paper.

A. Availably & Responsibility

In the paper, there are two kinds of message 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 "responsibility" r(i,k), sent from data point i to candidate exemplar point k, reflects the accumulated evidence for how well-suited point k is to serve as the exemplar for point i, taking into account other potential exemplars for point i. The "availability" a(i,k), sent from candidate exemplar point k to point i, reflects the accumulated evidence for how appropriate it would be for point i to choose point k as its exemplar, taking into account the support from other points that point k should be an exemplar.

In each iteration, the responsibilities are computed using the rule:

$$r(i,l) \leftarrow s(i,k) - \max_{k' s.t. k' \neq k} \{a(i,k') + s(i,k')\} \tag{1}$$

and the availabilities a(i,k) when $i \neq k$ are computed as follow:

$$a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{i' s.t. i' \notin \{i,k\}} \{0, r(i',k)\}\} \tag{2}$$

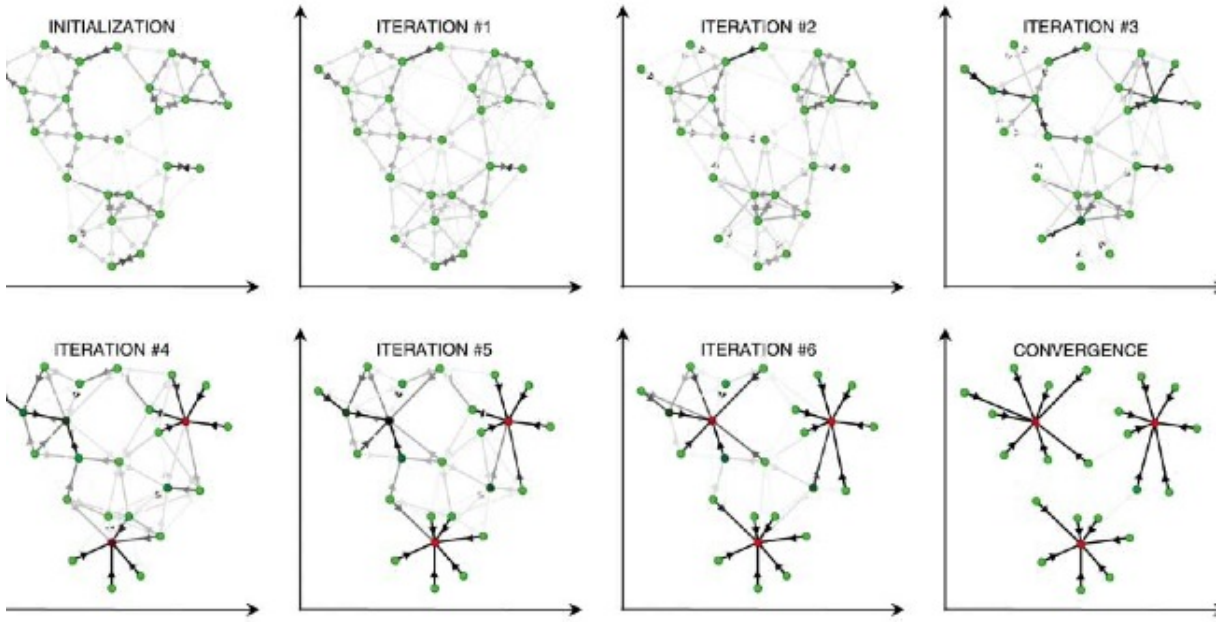

Fig. 3. Data Point Network in each Iteration of Affinity Propagation

When i=k, the "self-availability" a(k,k) is updated differently:

$$a(k,k) \leftarrow \sum_{i's.t.i' \neq k} \max\{0, r(i',k)\} \tag{3}$$

The s(i,k) is the similarity between data points s and k as in the input similarity matrix. And s(k,k) is set to the input preference that point k be chosen as an exemplar.

To begin with, the availabilities are initialized to zero: a(i,k) = 0. In the first iteration, r(i,k) is set to the input similarity between point i and point k as its exemplar, minus the largest of the similarities between point i and other candidate exemplars. Then the algorithm keeps updating responsibilities and availabilities in each iteration until converge. Figure 3 illustrates how clusters gradually emerge during the message-passing procedure.

B. Experiments

In the paper, experiments have been conducted on three data set, i.e., 2-dimensional data points, face detection data set, and gene data set, to test the performance of Affinity Propagation. 2-dimensional data points are used only to show the examples of clustering results. In face detection data set, average squared error is used to evaluate the performance of clustering algorithm. And in gene data set, the true positive and false positive value are sued to evaluate the clustering algorithm.

Figure 4 from the paper shows the example of two clusters generated from Affinity Propagation and K-centers. The two clusters both had the highest squared error in all the clusters. It shows that even with the highest squared error, the precision of the cluster from AP still is much better than the cluster from K-Centers.

Figure 5 from the paper shows the effectiveness of AP comparing to K-centers in the face detection data set. In this experiment, the authors ran AP once on every different input preference to generate different number of clusters, and ran K-centers 10,000 times on every different K. In this figure, the average squared error from AP is the lowest in all clustering results.

Fig. 4. Example of Face Clustering in the paper

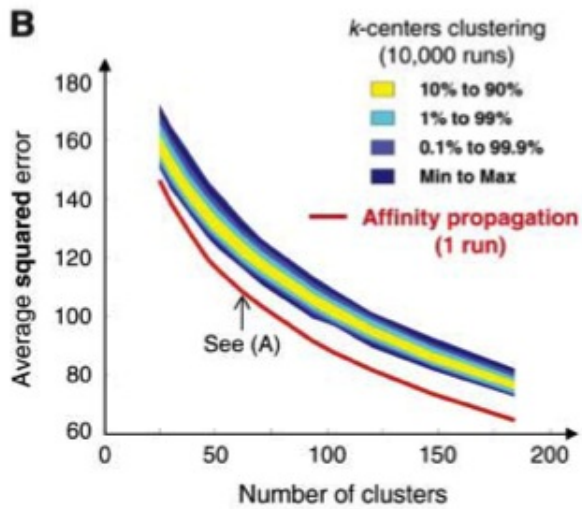

Fig. 5. Average Squared Error Comparison in the paper

Considering all these characteristics of AP and the results shown in their paper, the question is, when facing a clustering problem in future, will people choose to use AP other than K-centers? I think the authors would say yes. In this report, I am also trying to find my answer to this question, by replicating the experiments in their paper and do more extensive experiments to test the research hypothesizes made in this paper.

III. Research Hypothesis & Experiment Design

In this section, I will describe in detail the three research hypothesizes listed in the Introduction, including the content and the meaning of the hypothesizes, how the hypothesizes are shown in the paper, what the authors have done to support or test these hypothesizes, and how I design the experiments to test these hypothesizes. Not all the hypothesizes are explicitly made by the authors, but they can all be inferred from content of the paper. The three hypothesizes lie in three aspects, i.e., effectiveness and efficiency, evaluation metric, and scalability respectively.

A. Hypothesis I: Effectiveness and Efficiency

Research Hypothesis I: The proposed clustering algorithm can achieve better clustering performance in effectiveness and efficiency than other centroid based algorithms. The effectiveness and efficiency can be evaluated as follows.

- The average squared error when the algorithm converge. This evaluates the Effectiveness,

- The number of iterations before the algorithm converge. This evaluates the Efficiency,

For the effectiveness, It is obvious that the authors assumes the performance of AP is better than K-Centers. To test the performance, the author uses the average squared error as one of the evaluation metric. So the authors assumes that AP can achieve lower average squared error than other centroid based algorithms such as K-centers.

For the efficiency, the authors claim that their algorithm would be faster and can cost less than one-hundredth time than other methods. However, I haven't found any experiments testing this. So I measure the efficiency by the number of iterations before converge. In both AP and K-centers, one iteration means to update all data points to choose the clusters they belong to. Converge in AP means the responsibilities and the availability becoming stable, and converge in K-centers means the center of each cluster remaining the same. The reason why I don't use running time directly is because the running time of these two clustering algorithms depends on the number of iterations times the cost in each iteration, for each iteration, the time cost could be different based on different implementations of the clustering algorithm. So it would be clear to use the number of iterations to evaluate efficiency. Meanwhile, the complexity of time cost in each iteration for AP is $O(n^2)$ and for K-centers is O(n), where n is the number of data points. And we also tested the running time for AP and K-centers.

There are various of ways to test effectiveness and efficiency, The evaluation method I used is straightforward for these tasks. In this paper, both average squared error and precision on labeled data are used to test the effectiveness. To test the first research hypothesis, we just use the average squared error and the precision on labeled data for AP and K-centers will also be tested in Research Hypothesis II.

B. Hypothesis II: Evaluation Metric

Research Hypothesis II: average squared error is a good metric to evaluate the correctness of clustering results. The average squared error is calculated by the equation:

$$Err = \sum_{c_i} \sum_{p_j \in c_i} \{(p_j - c_i)^2\}/n \tag{4}$$

where c_i is the ith cluster represented by its center. p_j is the point in the cluster and n is the number of clusters.

It is obvious that the correctness of clustering can be evaluated by the precision and recall on data that already be labeled. Although smaller average squared error indicates higher similarity among nodes in each cluster, but this cannot directly indicate the correctness of the clustering results. The use of average squared error in this paper to test the effectiveness is based on the research hypothesis that averaged squared error is a good evaluation metric for the correctness of the clustering results. Although both average squared error and precision on labeled data are used in this paper, the two metrics are used in different data set and the relation between these two metric are not tested. So this research hypothesis is not tested by the authors directly.

In figure 4, the authors used the cluster having the highest average squared error from AP and K-centers to show that even the average squared error is high, the performance of AP is still much better than K-centers. In this statement, the authors assumed that the clusters having a higher average squared error may have lower precision in detecting faces. While this hypothesis is not tested in this paper.

To test this hypothesis, I just study the correspondence between the precision/recall and the averaged squared error in the clustering results on labeled dataset. I also use another metric that is the average ratio of squared error inside the cluster to squared error outside the cluster.

$$r = \sum_{c_i} \frac{\sum_{p_j \in c_i} \{(p_j - c_i)^2\}}{\sum_{p_j \notin c_i} \{(p_j - c_i)^2}\}/n \tag{5}$$

C. Hypothesis III: Scalability

Research Hypothesis III: In practical, the running time of this algorithm scales linearly with number of similarities.

This hypothesis is stated in the paper without testing. In fact, in each iteration of the AP algorithm, all similarities are scanned at least once. The complexity of each iterations is in $O(n^2)$ where n is the number of data points and n^2 is the number of similarities. And the running time is the time cost for each iteration times the number of iterations. So the authors just assume the number of iterations will not change with growth of number of similarities.

To test this research hypothesis, I have done experiment to find out both the time cost in each iteration and number of iterations with the number of data points increasing. And I will also test the overall running time of AP to deal with data points in different scales.

D. Relationships of the Three Research Hypothesizes

The three hypothesizes covered in this section are the interpretations from the statement and the algorithm in the paper. It seems that the efficiency part in Research Hypothesis I overlaps with the Research Hypothesis III. But in fact, the efficiency part in hypothesis I is trying to compare the efficiency between AP and K-centers, and the hypothesis III is trying to study how the algorithm acts in different scales of data set. Meanwhile, the effectiveness tested in hypothesis I using average squared error may depend on the results from experiments testing hypothesis II. In fact, hypothesis I and II are treated as independent hypothesizes in this report, because even if AP can achieve lower averaged squared error, it may act not as good as K-centers in the precision/recall on labeled data. This is also shown in the experiment results in section V. Some of them are tested by the author in their origin work, some of them are not. I conducted extensive experiments on different data sets to test these hypothesizes in section V.

IV. Extension: Tweets Clustering

This section describes how I use Affinity Propagation to do tweets clustering. Tweet are user posted content sharing his/her status or thinkings in the popular social network Twitter [6]. Twitter is one of the most popular social network websites currently and there are more than 20 millions of tweets posted per day. It is significantly meaningful to summarize all these contents, tag them into different categories and find representative ones in each category, because doing this could make users from twitter or other social media networks feel more conveniet in browsing interesting content in large scale data.

Because the number of tweets posted everyday is in million level and the content is different and lies in various of topics, it is hard to define a complete taxonomy on all the tweets, then it is not possible to label sufficient amount of tweets for specific applications because the data would also be imbalanced. In this situation, unsupervised learning are widely used in data mining in social network. Since the content of each tweet is limited into 140 characters, graphical models such as Latent Dirichlet Allocation used in NLP that requires sufficient content in document cannot be directly used. Therefore, clustering algorithm such as Affinity Propagation may be powerful in this scenario.

In this project, I just simply used a dataset containing 1000 tweets, because it would take a large amount of time to extract features in tweets. And I used bag-of-words model to represent each tweet and use Jarcard distance to measure the similarity between tweets. In preprocessing, I converted all words into lower case and use porter stemming algorithm [7] to reduce inflected words to their stem. Some initial results are also shown in section V.

V. Experiment Results

In this section, The results of experiments designed to test the thee hypothesizes made from the paper are shown. First I will introduce the experiment setup including the implementation and the data set used in the experiments. Then the following three subsections discuss the experiments testing for the three hypothesizes respectively.

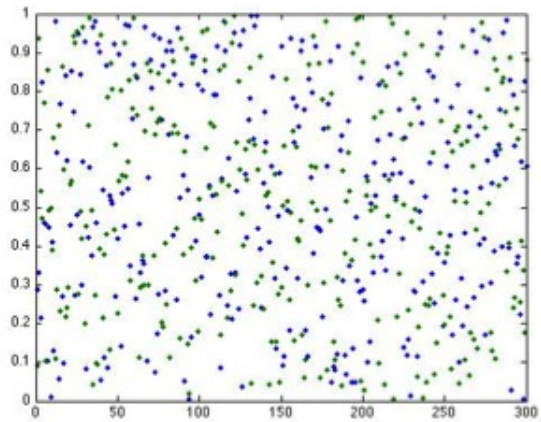

Fig. 6. Example of 2 Dimensional Data Points

A. Experiment Setup

Instead of the codes provided on their website [8], I implemented the algorithm using my own codes, because this is easier for me to record the averaged error and time cost in each iteration in my experiments. I first implemented a simple version that directly reflects the paper, taking $n \times n$ similarity matrix as input, scan over all similarities to update responsibility and availability. The simple version can be ran in scale of 50-300 data points and return the results fast, but either be very slow or taking large size of memory for data set containing more data points. So I implemented a faster version that can take as input the sparse matrix, and have a threshold to filter data point pair with lower similarity. All the codes all written in MATLAB and ran in a Mac Laptop with 2.7 GHz CPU and 4GB memory.

Three data sets have been used in the experiments. The first two are used to test the three hypothesizes and the last one is tweets data set used in the extension. The detail of the data sets is shown as follows.

a) 2-d data points: This data set contains 300 random generated 2 dimensional data points uniformly distributed in $[0, 1] \times [0, 1]$ space. The paper also used random generated 2-d data points to show examples. Figure 6 shows the distribution the data set. Figure 7 and 8 show the clustering results using Affinity Propagation and K-Centers on a 50 and 100 subset of 2-d data points respectively. The figures on the left are the results from Affinity Propagation and the figures on the right are the results from K-centers.

b) Face Data Set: This data set is downloaded from the first author's website. It contains 900 images of faces from 10 people, each person has 90 images. To avoid including the background behind each face, a central window of size 50×50 pixels was extracted by the authors. Finally, the pixels in each 50×50 image were normalized to have mean 0 and variance 0.1. The similarity between two images was set to the negative sum of squared pixel differences. Figure 9 shows a subset of images in this data set.

c) Tweets: I use 1000 tweets crawled from twitter public time line using twitter API [9]. Each tweet is represented by a 25730 dimensional vector, each dimension represents one n-gram extracted from corpus. I use Jarcard distance to measure the similarity between two tweets.

B. Test Research Hypothesis I

To test research hypothesis I, I ran experiments on both the 2-d data set and the face data set. To evaluate effectiveness, I ran Affinity propagation and K-centers on and 2-d data set from size 50 to 300, and ran the two algorithm on face dataset by fixing the number of clusters from 9 to 72. To evaluate efficiency, I record the number of iterations in the two algorithms running in the experiments to test effectiveness.

1) Effectiveness: Figure 10 shows the average squared error of the clustering results from Affinity Propagation, K-centers and K-means. The algorithms are running in different size of the 2-d data points,

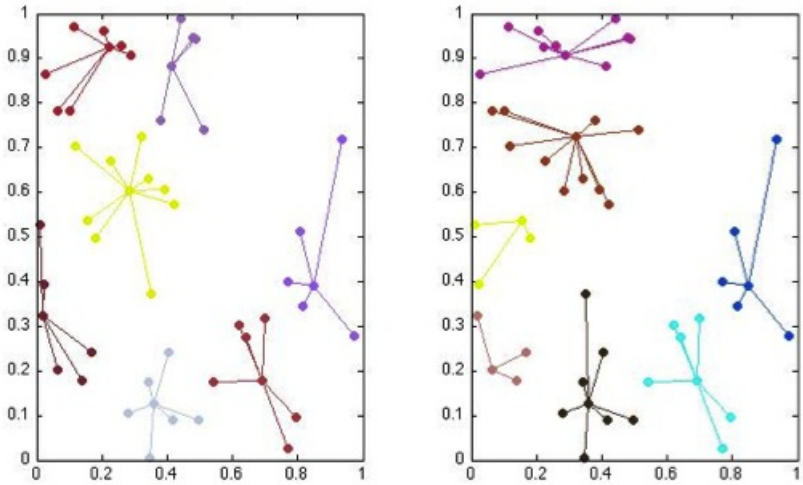

Fig. 7. Running Example of AP and K-Centers in 50 2-d data points

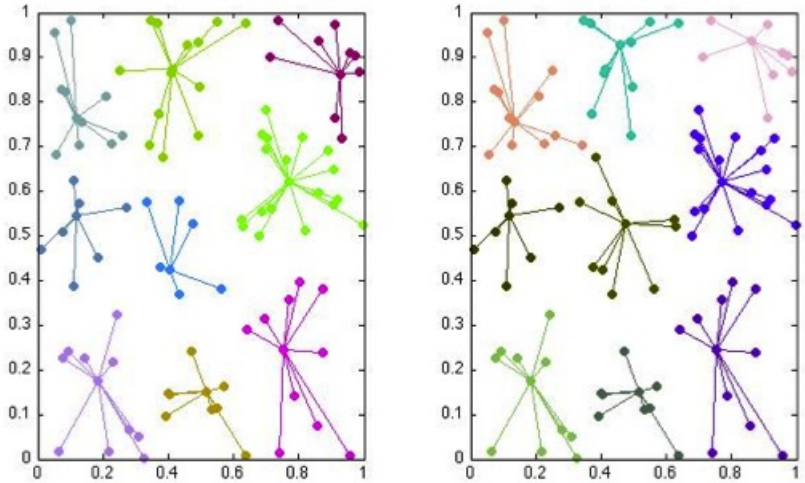

Fig. 8. Running Example of AP and K-Centers in 100 2-d data points

from a subset containing 50 data points to the whole 300 data points. In K-means algorithm, the average squared error is calculated by the data points inside the cluster and the centers, no matter whether the centers found by K-means are in the data points or not. In the results, we can see that the Affinity Propagation has the lower averaged squared error than K-centers in all scales. And even for K-means, where the center need not to be the member of data set, Affinity Propagation still has comparative results.

Figure 11 shows the average squared error of the clustering results from Affinity Propagation and K-centers. The algorithms are running with fixed number of clusters in all the 900 images from face data set. In Affinity Propagation, I try different values of the input parameter preference discussed in section II to get the results with specific number of clusters. The AP algorithm ran only once for each number of clusters shown as the red line and K-centers are ran 100 times shown as the rest purple, yellow, light blue and dark blue areas. This experiments has also been done in the paper, shown in Figure 5.Our results shown in figure 11 is similar, comparing with the results shown in the paper in figure 5. AP had the lowest average squared error comparing to results from K-centers in 100 runs.

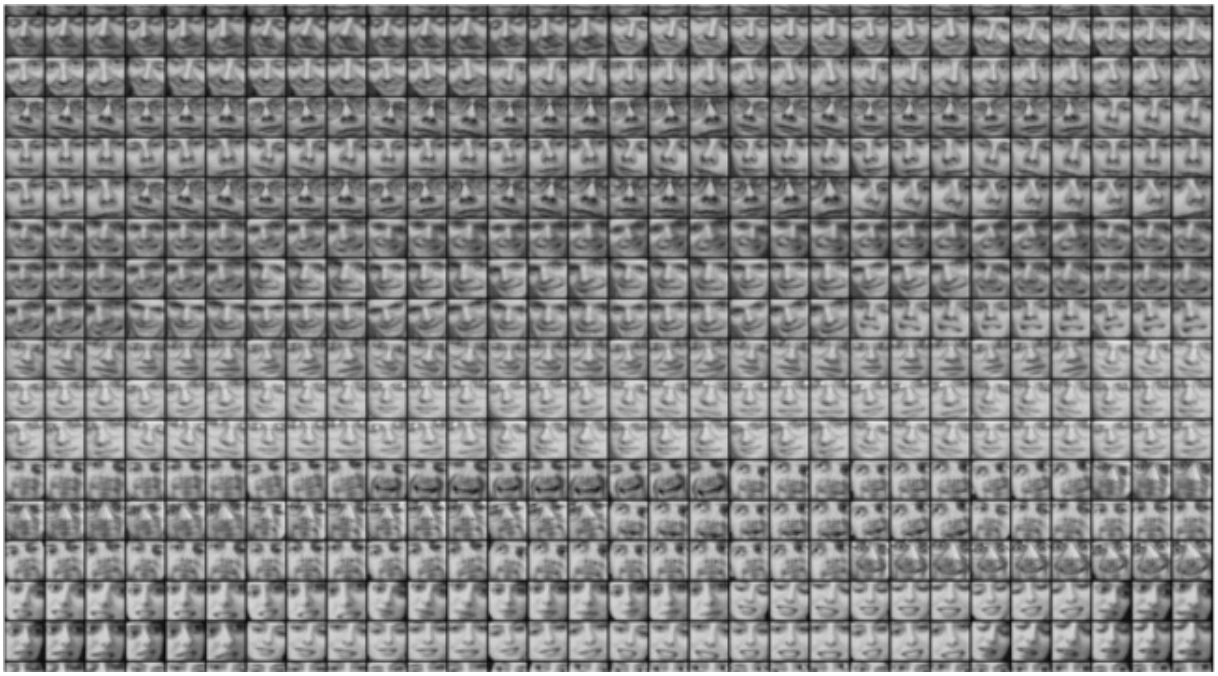

Fig. 9. Collection of the Face Data set

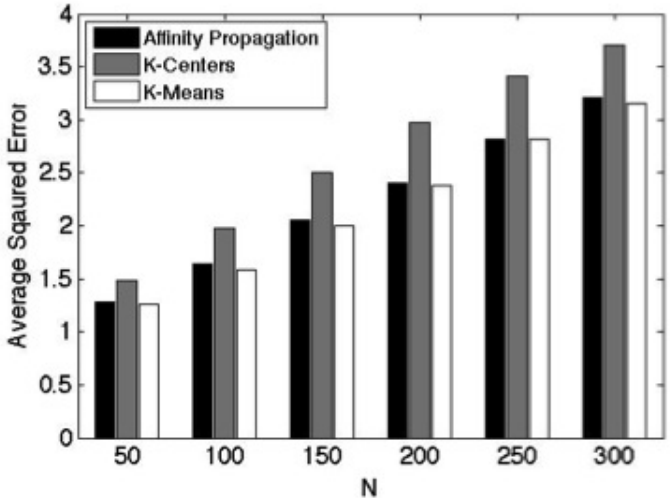

Fig. 10. Average Squared Error on 2-D data set

The experiments testing the effectiveness using average squared error show that Affinity Propagation can achieve better performance than K-centers algorithm measured by average squared error.

2) Efficientcy: Figure 12 shows the number of iterations ran in Affinity Propagation, K-centers and K-means. The figure on the left is the results ran on 2-d data points, and the figure on the right is the results ran on face data set. We can see the the number of iterations ran in Affinity Propagation is much higher than K-centers or K-means in both data set. This means that K-centers can be more efficient than AP. The running time cost for AP on 300 2-d data points is about 2 second, and for K-centers is about 0.02 seconds. Therefore, for single run of AP and K-centers, although AP can achieve lower average squared error, the efficiency of AP is lower than the efficiency of K-centers.

In their paper, the authors claimed that AP can cost less than one hundredth time than other clustering

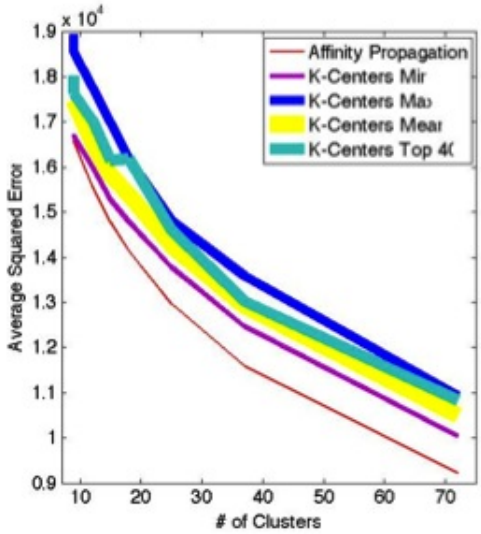

Fig. 11. Average Squared Error on face data set

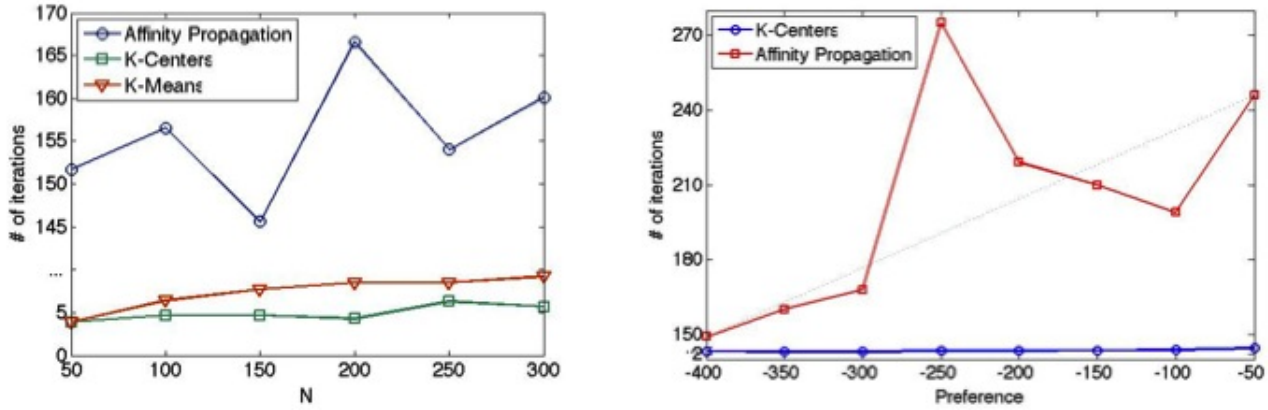

Fig. 12. Average number of Iteration on 2-D data set

algorithms. But the results show in the report is different from this. I think in the paper, the other clustering algorithm, i.e., K-centers may be ran 10,000 times to get one clustering results, as shown in figure 5. This may be why the authors claimed that the time cost for AP is less than other clustering algorithms. In figure 11, we can see the worst K-centers average error is about 20% higher than AP.It may be ok if the K-centers algorithm is a hundred times faster then AP.

Based on the results shown in the experiments tested the efficiency of AP, we can not say that Affinity Propagation is more efficient than K-centers.

C. Test Research Hypothesis II

To test Research Hypothesis II, i.e, average squared error is a good metric to evaluate clustering performance, I used the face data set which contains 10 people's 900 face photos. To evaluate performance of clustering results on labeled data, precision and recall are used. Each cluster's precision and recall are calculated on the people having the most photos labeled in this cluster. For example, if the clustering algorithm group 30 photos in one cluster, 20 of which are from person 1 and 10 of which are from person 2, and both person 1 and person 2 have 40 photos in the whole data set. Then the precision and recall of this cluster is calculated using the matches of person 1, i.e., the precision is 20/30 and the recall is 20/40.

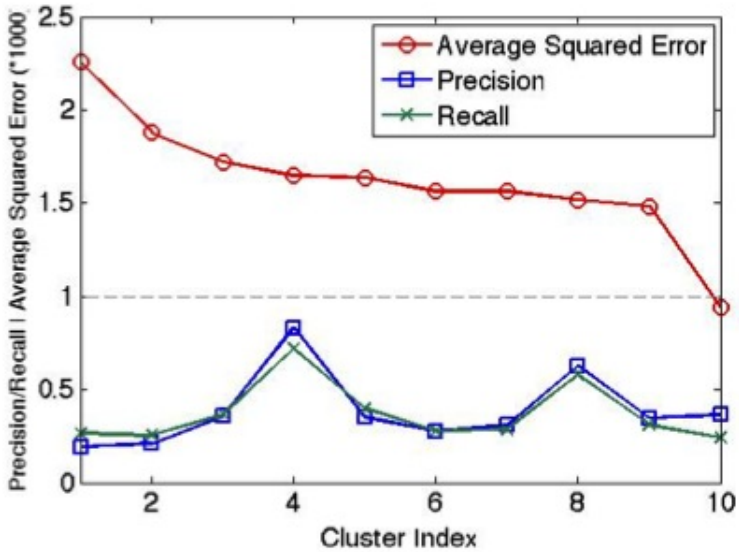

Fig. 13. Precision and Recall Corresponding to Average Squared Error

First, I tested the relationship between the precision/recall and the average squared error in each cluster from clustering results using AP to group the face data set into 10 clusters. The results are shown in figure 13. In this figure, we sorted the clusters by the average squared error in decreasing order, the average squared error is shown by the red line. The blue and green lines show the precision and recall for each clusters. Through this figure we can not see the increasing of precision/recall corresponding to the decrease of average squared error. Therefore it may be not proper for the author to assume that higher average squared error indicates lower precision in the paper when showing the figure 4.

I also tested the relationship between the average precision/recall from all clusters and the average squared error in clustering results. I used the face data set and ran both AP and K-centers with fixed numbers of clusters from 9 to 72 and evaluated the performance by average precision and recall comparing to the figure 11 using average squared error. The results are shown in figure 14. We can see that the average precision is increasing and recall is decreasing when the number of clusters in getting larger. Considering the figure 11, where the average squared error decreasing with the number of clusters increasing. We can infer that the the average precision is increasing corresponding to the decreasing of average squared error.

Although the increase of average precision/recall corresponds to the decrease of average squared error, we can see that in figure 14, the performance of AP is not always better than K-centers. In this figure, we can find out the the performance of K-centers when the number of clusters are small is actually better than Affinity Propagation. This reflects the characteristic of K-centers that this algorithm can do well when the number of clusters are small. And with the increasing of number of clusters, the average precision/recall of AP is higher than K-centers.

We also tested the performance of clustering using the ratio metric introduced in equation (5) on the face data set. The results are shown in figure 15. We can see the results using ratio is similar to results using average squared error shown in figure 11.

D. Test Research Hypothesis III

Since we already have the number of iterations with increase of the scale shown in figure 12, and we can see that the number of iterations has a trend to grow with the data size, then to test Research Hypothesis III, we need the running time for each iteration and the overall running time in one clustering task. The results on the 2-d data point data set are shown in figure 16.

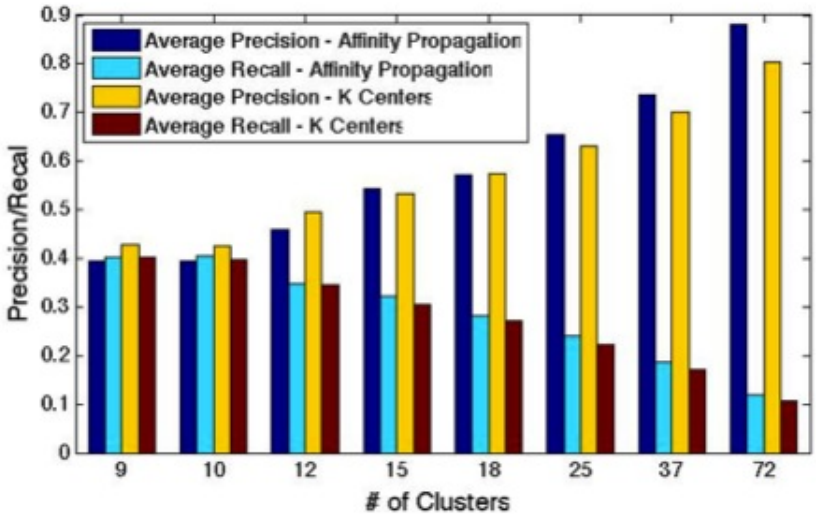

Fig. 14. Average Precision and Recall on Face data set

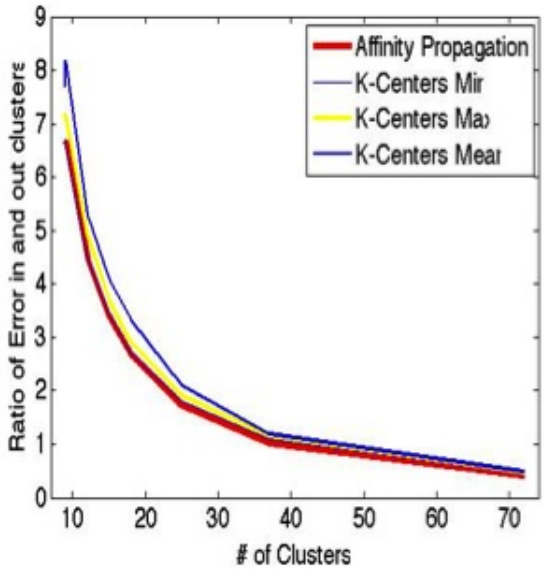

Fig. 15. Clustering Performance Measured by Ratio

The left figure in figure 16 shows the average time cost in each iteration with the increasing of data size from 50 to 300. It can to be fit into polynomial functions based on the results. The right figure in figure 16 shows the overall running of AP with the increasing of data size from 50 to 300. Through the results we can see that the overall running time increases faster than linear and can be fit into exponential approximation.

E. Extension: Tweets Clustering

Here we shows the results of the Affinity Propagation on the task of tweets clustering. Figure 17 shows the average squared error of Affinity Propagation and K-centers algorithm on the tweets data set with different numbers of clusters. For each fixed number of clusters, I ran AP once and K-Centers 100 times. And the results still show that AP has a better performance than K-centers.

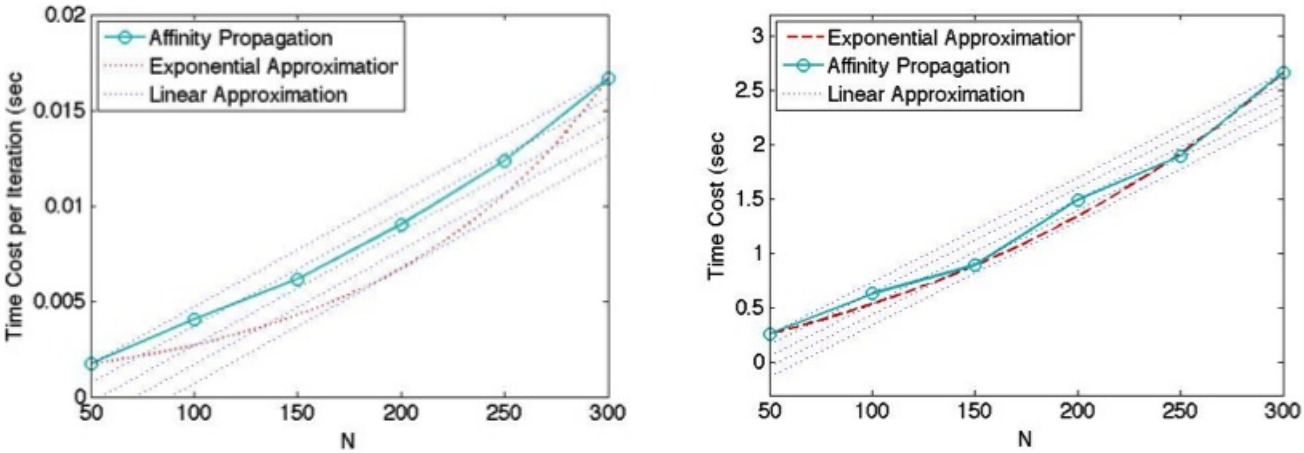

Fig. 16. Time Cost in Each Iteration and Overall Running time in different Scale

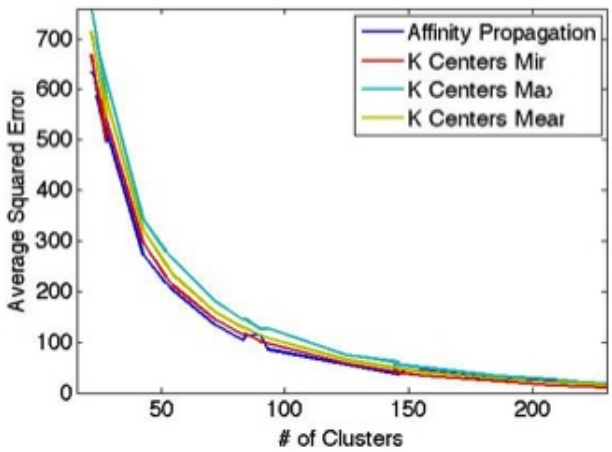

Fig. 17. Averaged Squared Error on Twitter Data Set

Figure 18 shows the representative words from the first 5 clusters in the results of AP. We can see some of the words in same cluster are semantically related, such as “wake up” and “school” in cluster 1, or “camera” and “digital” in cluster 5. But there is still much noise in the results. This may because I only use 1000 tweets that is not sufficient to aggregate enough semantic meanings in each cluster. I think if the scale of the data set is larger, the performance may be better. In current project, I didn’t get enough time to conduct this experiment on a larger data size, and this can be done in my future work.

VI. Discussion on Test Results.

Section V shows all the experiment results in testing the three research hypothesizes made in the paper and an extension doing tweets clustering. In this section, I will discuss whether we will accept or reject the hypothesizes based on the experiment results.

a) Hypothesis I: Effectiveness and Efficiency: Based on the results shown in Section V.B, we found out that the effectiveness of Affinity Propagation measured by average squared error is alway better than K-Centers, as shown in figure 10 and 11, while the number of iterations in Affinity Propagation is much higher than K-Centers in one run, the efficiency of AP may be not as good as the authors claimed in the paper.

cluster 1	cluster 2	cluster 3	cluster 4	cluster 5
wake up	my	work out	ready	digital
that guy	spend by	lfmao	eat	dvd
tell	time	song	to chill	camera
never	day	listen to	good	cheapest
school	chirstmas	know	wait	card
early	hollywood	while he	food	high
say hi	countdown	party	a nap	cause
hi	start	allstar	day	source
sleep	fun	follow	come	business

Fig. 18. Representative Words in Different Clusters

For Research Hypothesis I, our conclusion is that Affinity Propagation can achieve higher performance in effectiveness comparing to other centroid based algorithms, but the efficiency is not better than such algorithms as K-Centers. So we accept the effectiveness part of this hypothesis and reject the hypothesis on efficiency.

b) Hypothesis II: Evaluation Method: Based on the results shown in Section V.C, we have two observations on the relationship between precision/recall on labeled data and average squared error from figure 13 and 14. The first is that the increase or decrease of precision/recall in single cluster does not correspond to the average squared error. The second is that the increase of precision corresponds to the decrease of average squared error in all the clusters of one clustering task. And the results using the ratio metric defined in equation (5) shown in figure 15 also support this hypothesis.

For Research Hypothesis II, our conclusion is that average squared error is good to evaluate overall clustering performance, but the squared error may not be corresponding to precision/recall in each one cluster.

c) Hypothesis III: Scalability: Based on the results shown in Section V.D, we found out that the running time in each iteration is increasing linearly with the number of similarities, i.e., n^2, but the number of iterations does not seem to be same on different data size. So the overall running time will not increase linearly with the number of similarities, the speed should be faster than that. And we need further experiments to test whether the speed is in exponential order or not.

For Research Hypothesis III, our conclusion is that we rejected this research hypothesis, because the time cost does not scale linearly.

VII. Replication Overview

This section contains the goals I achieved in this project, encountered problems in replicating and testing Affinity Propagation, and how I tried to solve them.

A. Achieved Goals

In this project, I replicated the new clustering algorithm, Affinity Propagation, from the paper, Clustering by Passing Messages Between Data Points, published in Science 2007. I found three research hypothesizes made in this paper. Then I replicated the some of the experiments in this paper as well as designed and conducted extensive new experiments to test these hypothesizes. Based on the results, some of the hypothesizes are accepted and some are rejected.

What I have done and the goals I achieved are listed as follows.

- Read and Understand the paper, Clustering by Passing Messages Between Data Points.
- Find three research hypothesizes made in this paper from either the explicit statement or algorithms and experiments.

- Replicate the Affinity Propagation in MATLAB. Finish both simple version and a faster version take sparse matrix as input.
- Design and conduct extensive experiments to test the proposed research hypothesizes.
- The experiment results generated from my implementation is corresponding to the results in the paper.
- Either accept or reject the hypothesizes based on the experiment results.
- I also try to do an extension of tweets clustering using Affinity Propagation.

B. Encountered Problems

The biggest problem I encountered in this project is how to make affinity propagation scale up. Although there is some work after this paper studied how to make this algorithm simpler [10], the origin version is hard to ran on large scale of data. Because it takes as input the similarity matrix, whose size is the square of the number of data points. Although all the data sets used in the experiments contain no more than 1000 items, it still takes too much time to conduct all these experiments. So I implement a faster version of Affinity Propagation, that take sparse matrix as input and eliminate all the similarities less than a threshold. The threshold is set to be 2 times the minimum non-zero value, and since the data points in each cluster usually have high similarity, this threshold doesn't change the overall performance of AP.

Another problem I encountered is on the task of tweets clustering, Since each tweet is represented as a 25730 dimensional vector, and each tweet only have content less than 140 characters, the vector is sparse and most of the 1000 tweets have no intersection between each other. So the performance in tweets clustering is not satisfying. This may be improved if we use a much larger scale data size, but I didn't get time to finish this in time.

VIII. Conclusion

Recall the question asked in section II.B: When facing a clustering problem in future, will people choose to use AP other than K-centers? After testing the three hypothesizes made in this paper, my answer is quite clear: unless the data size is small and we need to find a large number of clusters in this small data set, I will prefer K-centers to the origin version of Affinity Propagation introduced in this paper (there exists improved version in work after this claimed to be more effectiveness and efficiency).

In my project of course 592, I choose Affinity Propagation: Clustering by Passing Messages Between Data Points as the paper to replicate. In this paper, a new clustering algorithm that makes use of the neighborhood information of data points by passing real-valued messages, is proposed. I find three research hypothesizes that are related to the performance, evaluation and scalability of the algorithm in this paper. I replicate the algorithm using MATLAB and conducted extensive experiments to test these hypothesizes. I also use AP in tweets clustering as an extension in this project. Based on the results of the experiments, some of the hypothesizes are accepted and some are rejected.

References

[1] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, 2007. [Online]. Available: www.psi.toronto.edu/affinitypropagation

[2] Wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/Wiki

[3] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of Fifth Berkeley Symp. on Math. Statist. and Prob.*, vol. 1.

[4] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," pp. 282–289, 2001.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[6] Twitter. [Online]. Available: http://www.twitter.com/

[7] S. R. C.J. van Rijsbergen and M. Porter, "New models in probabilistic information retrieval," 1980.

[8] Affinity propagation. [Online]. Available: http://www.psi.toronto.edu/index.php?q=affinity\%20propagation

[9] Twitter api. [Online]. Available: https://dev.twitter.com/

[10] I. E. Givoni and B. J. Frey, "A binary variable model for affinity propagation," *Neural Computation*, vol. 21.