

# A Preference Model on Adaptive Affinity Propagation

Rina Refianti<sup>1</sup>, Achmad Benny Mutiara<sup>2</sup>, Asep Juarna<sup>3</sup>, and Adang Suhendra<sup>4</sup>

<sup>1,2,3</sup>Faculty of Computer Science and Information Technology, Gunadarma University, Indonesia

<sup>4</sup>Department of Informatics, Gunadarma University, Indonesia

---

## Article Info

### Article history:

Received Nov 17, 2017

Revised Feb 5, 2018

Accepted Feb 19, 2018

### Keyword:

Affinity Propagation

Exemplars

Data Points

Similarity Matrix

Preference

---

## ABSTRACT

In recent years, two new data clustering algorithms have been proposed. One of them is Affinity Propagation (AP). AP is a new data clustering technique that use iterative message passing and consider all data points as potential exemplars. Two important inputs of AP are a similarity matrix (SM) of the data and the parameter "preference"  $p$ . Although the original AP algorithm has shown much success in data clustering, it still suffer from one limitation: it is not easy to determine the value of the parameter "preference"  $p$  which can result an optimal clustering solution. To resolve this limitation, we propose a new model of the parameter "preference"  $p$ , i.e. it is modeled based on the similarity distribution. Having the SM and  $p$ , Modified Adaptive AP (MAAP) procedure is running. MAAP procedure means that we omit the adaptive  $p$ -scanning algorithm as in original Adaptive-AP (AAP) procedure. Experimental results on random non-partition and partition data sets show that (i) the proposed algorithm, MAAP-DDP, is slower than original AP for random non-partition dataset, (ii) for random 4-partition dataset and real datasets the proposed algorithm has succeeded to identify clusters according to the number of dataset's true labels with the execution times that are comparable with those original AP. Beside that the MAAP-DDP algorithm demonstrates more feasible and effective than original AAP procedure.

Copyright © 2018 Institute of Advanced Engineering and Science.

All rights reserved.

---

## Corresponding Author:

Achmad Benny Mutiara

Faculty of Computer Science and Information Technology, Gunadarma University

Jl. Margonda Raya No.100, Depok 16424, Indonesia

+62-815-916055

amutiara@staff.gunadarma.ac.id

---

## 1. INTRODUCTION

In big data era, analysis of large amounts of data become as essential area in Computer Science. The methods of data mining, among others clustering methods, classification methods, etc., are needed to extract or mine the knowledge from large amounts of data. To group the data in accordance with their multiple-characteristic based similarities is known as clustering [1].

In recent years, there are two new proposed data clustering algorithms. One of them is Affinity Propagation (AP) that has been proposed by Brendan J. Frey and Delbert Dueck (2007) [2]. Unlike previous clustering method such as  $k$ -means which taking random data points as first potential exemplars, AP considers all the data points as potential cluster centers [3, 4]. AP works by taking an input of similarity between data points and simultaneously considers all the data points as potential cluster centers which called exemplars by iteratively calculating responsibility  $r$  and availability  $a$  based on the similarity until converge. After the points converge, AP found clusters with far less error than  $k$ -means and it takes place in less than one hundredth of the amount of time [3].

AP have several advantage over other clustering methods due to AP consideration of all data points as potential exemplars while most clustering methods find exemplars by recording and tracing the fixed data points and iteratively correcting it [3]. Because of it, most clustering methods does not change the set that much and just keep tracking on the particular sets. Furthermore, AP supports non-symmetrical similarities and it does not depend on initialization that found on other clustering algorithms. Because of these advantages, it has been successfully applied in many disciplines such as image clustering [3] and Chinese calligraphy clustering [5].

The paper is the continue study of our previous works [6, 7]. In [6] we survey and investigate the performance of various AP approaches, i.e. Adaptive AP, Partition AP, Soft Constraint AP, and Fuzzy Statistic AP. And it is found that i) Partition AP (PAP) is the fastest one among four other approaches, ii) Adaptive AP (AAP) can remove the oscillation and more stable than the other, and iii) Fuzzy Statistic AP (FSAP) could result smaller cluster number than the other approach, because its preferences are generated by using fuzzy-statistical methods iteratively. In [7] we investigate a time complexity of various AP approaches, i.e. AAP, PAP, Landmark AP (L-AP), and K-AP. And it is found that the approach that has the most efficient computational cost and the fastest running time is Landmark AP, although its clustering result is very different in clusters number than AP.

Although AP itself has been proven to be faster than  $k$ -means, and it also has shown much success in data clustering, AP still has a limitation, i.e. it is not easy to determine the value of the parameter "preference"  $p$  which can result an optimal clustering solution [8]. The goal of this paper is to resolve this limitation with proposing a new model of the parameter "preference"  $p$ , i.e. it is modeled based on the distribution of similarity. The model will be explain in the subsection 3.1.. Then it will be applied to Adaptive AP (AAP). This is because AAP has a better level of accuracy than other approaches.

In experiment random non-partition dataset, random partition dataset, and real dataset from UCI datasets [9] are used. By partition of dataset  $k$ -means algorithm [10] is applied to generate a four groups of data points. The results of our experiment are shown in section 4..

## 2. THEORETICAL BACKGROUND

### 2.1. Affinity Propagation

#### 2.1.1. Input Preference

Supposed we have a set of data points  $X = \{x_1, x_2, x_3, \dots, x_n\}$ , AP takes as input of similarity matrix (SM) between data points  $s$ , where each similarity  $s(i, j)$  shows how good data point  $x_j$  is fitted to be an exemplar for  $x_i$ . The similarities of any type can be accepted, e.g. for real data, the negative Euclidean distance, and for non-metric data the Jaccard coefficient, so AP can be widely applied in different areas [7].

Instead of requiring that the number of clusters be predetermined, AP takes as input a real number  $s(j, j)$  for each data point  $j$  so that data points with larger values of  $s(j, j)$  are more likely to be chosen as exemplars. These values are referenced to as "preferences". These preferences will affect the number of clusters produced. The preferences values can be the median of the similarities or their minimum.

$$p = \text{median}(s(:)), \text{ or, } p = \text{min}(s(:)) \quad (1)$$

#### 2.1.2. Messages passing

Supposed we have similarity  $s(i, j)$ , ( $i, j = 1, 2, \dots, n$ ), AP attempts to obtain the best exemplars that can make the net similarity maximized, i.e. the roundly sum of similarities between all exemplars and their member data points. Process in AP can be viewed as passing values between data points. There are two values that are passed between data points: responsibility and availability. Responsibility  $r(i, j)$  shows how well-suited point  $j$  is to serve as the exemplar for point  $i$ , taking into account other potential exemplars for point  $i$ . Availability  $a(i, j)$  reflects the accumulated evidence for how appropriate it would be for point  $i$  to choose point  $j$  as its exemplar, taking into account the support from other points that point  $j$  should be an exemplar.

Figure 1 shows us how the availability and responsibility works in AP. Availabilities  $a(i, j)$  are transmitted from candidate exemplars to data points to state the availability of candidate exemplars to data points as cluster point. Responsibilities  $r(i, j)$  are transmitted from data points to candidate exemplars and state how strongly each data point favors the candidate exemplar over other candidate exemplars. All of this message passings are kept done until convergence is met or the iteration reach a certain number.

Initially all  $r(i, j)$  and  $a(i, j)$  are set to 0, and iteratively their values are updated as follows until convergence values achieved:

$$r(i, j) = \begin{cases} s(i, j) - \max_{k \neq j} \{a(i, k) + s(i, k)\} & (i \neq j) \\ s(i, j) - \max_{k \neq j} \{s(i, k)\} & (i = j) \end{cases} \quad (2)$$

$$a(i, j) = \begin{cases} \min\{0, r(j, j)\} + \sum_{k \neq i, j} \max\{0, r(k, j)\} & (i \neq j) \\ \sum_{k \neq i} \max\{0, r(k, j)\} & (i = j) \end{cases} \quad (3)$$

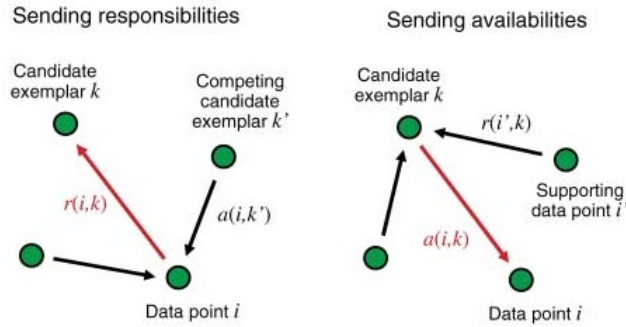


Figure 1. Message Passing in Affinity Propagation [3]

After calculating both responsibility and availability, iteratively their values are updated as follows until convergence values achieved:

$$r(i, j) = (1 - \lambda)r(i, j) + \lambda r_o(i, j) \quad (4)$$

$$a(i, j) = (1 - \lambda)a(i, j) + \lambda a_o(i, j) \quad (5)$$

where  $\lambda$  is a damping factor modeled to reduce numerical oscillations, and  $r_o(i, j)$  and  $a_o(i, j)$  are previous values of responsibility and availability.  $\lambda$  should have value that is greater than or equal to 0.5 and less than 1, i.e.  $0.5 \leq \lambda < 1$ . A high value of  $\lambda$  may make number oscillations avoided, but this is not guaranteed, and a low value of  $\lambda$  will make the AP run slowly [11].

### 2.1.3. Exemplar decision

For a set of data point  $x_i$ , if  $x_j$  can reach the maximal of  $r(i, j) + a(i, j)$ , then it could be deduced that i)  $x_j$  is the most suitable exemplar for  $x_i$ , or that ii)  $x_j$  would be the most exemplar of  $x_i$ . The Exemplar for  $x_i$  is selected as the following formula:

$$c_i \leftarrow \operatorname{argmax}_k \{r(i, j) + a(i, j)\} \quad (6)$$

where  $c_i$  is the exemplar for  $x_i$ .

## 2.2. Adaptive Affinity Propagation

AP has many extensions. One of the extension is Adaptive Affinity Propagation (AAP). AAP is designed to solve AP limitation : we can not know what value of preference can result the best clustering solution, and if oscillations occurs, it cannot be eliminated automatically. To solve the problem, AAP can adapt to the need of the data sets by configuring the value of preferences and damping factor.

As in [11] we assume that  $C(i)$  is the number of clusters in the iteration. We summarize the AAP algorithm as follows:

---

### Algorithm 1 Affinity Propagation Adaptif

---

**Input :** Data Points  $x_i, i = 1, 2, \dots, n$

**Output :** Centers of Clusters  $C(i)$

- 1: Set parameters  $\lambda = 0.5$ ;
  - 2: Calculate  $s(i, j)$ ;
  - 3: Set  $r(i, j) = 0$  and  $a(i, j) = 0$ ;
  - 4: Run AP steps (Eq.2 - Eq.6);
  - 5: Verify whether oscillations occur or not. If oscillations occur, then  $\lambda \leftarrow \lambda + \lambda_{step}$ , else run AP steps continually.
  - 6: If  $C(i) \leq C(i + 1)$ , then  $p \leftarrow p + p_{step}$ , and  $s(i, i) \leftarrow p$ . Go to step 4.
- 

As proposed in [11] when the values of  $C(i)$  is lower than 2, the AAP algorithm stops.

AAP has shown a better quality or at least same quality in making a clustering result as AP and finding optimal solution based on different kind of data sets [11]. AAP has shown to be able to process several type of data

such as gene expression [11], travel route [11], image clustering [11, 12], a mixed numbers and categorical dataset [13], text document [14], and zoogeographical regions [15].

### 3. PROPOSED WORK

#### 3.1. Proposed Preference Model

From a set of data points  $X = \{x_1, x_2, \dots, x_n\}$ , supposed we have randomly two data points  $x_i$  and  $x_j$ , if distance from  $x_i$  to other points is larger than to  $x_j$ , then  $x_i$  has a lower possibility than  $x_j$  to be the dataset center. On the basis of this assumption, preference for each data point can be computed as follows.

For a given data point  $x_i$ , similarities from  $x_i$  to other data points are summed up as:

$$TDS(i) = \sum_{j=1, j \neq i}^n s(x_i, x_j) \quad (7)$$

Then it is normalized as:

$$NTDS(i) = \frac{TDS(i)}{\sum_j^n TDS(j)} \quad (8)$$

Finally, for each data point preference can be defined as follows :

$$p(i) = s(i, i) = N \times NTDS(i) \times Const. \quad (9)$$

where  $Const$  can be real non-zero number or  $\min(s(\cdot))$

Equation (9) of preferences represents and reflects the distribution of data set, and also we hope that it will tend to better results as shown in results section. Then this model is applied to Modified Adaptive Affinity Propagation algorithm (MAAP), as explain in subsection 3.2.. Our model is simple and easy to apply if we compare to a another model proposed by Ping Li et.al (2017) [16].

#### 3.2. Modified Adaptive Affinity Propagation (MAAP)

We modify adaptive AP in the following manner

---

**Algorithm 2** Modified Adaptive Affinity Propagation (MAAP)

---

**Input :** Data Points  $x_i, i = 1, 2, \dots, n$

**Output :** Centers of Clusters  $C(i)$

- 1: Set parameters  $\lambda = 0.5$ ;
  - 2: Calculate  $s(i, j)$ , and set  $p$  as Eq.9
  - 3: Set  $r(i, j) = 0$  and  $a(i, j) = 0$ ;
  - 4: Run AP steps (Eq.2 - Eq.6);
  - 5: Verify whether oscillations occur or not. If oscillations occur, then  $\lambda \leftarrow \lambda + \lambda_{step}$ , else run AP steps continually.
- 

Because we set the proposed preferences  $p$  in algorithm, then we could omit the step 6 in Algorithm 1.

#### 3.3. Cluster Evaluation

Silhouette validation index and Fowlkes-Mallows index [17] are used to evaluate the quality of a clustering process.

For a given dataset  $X = \{x_1, x_2, \dots, x_n\}, x_i \in R^m$ , the Silhouette index of  $x_i$  can be defined as

$$Sil(x_i) = (b(x_i) - a(x_i)) / \max(a(x_i), b(x_i)) \quad (10)$$

where  $a(x_i)$  is defined as a mean distance from other data points in the same cluster  $K_c$  to  $x_i$ ;  $d(x_i, K_{c'})$  represents a mean distance from all data points in cluster  $K_{c'}$  ( $c' \neq c$ ) to  $x_i$ . If  $b(x_i)$  is defined as

$$b(x_i) = \min(d(x_i, K_{c'})), \quad (11)$$

$c' = 1, 2, \dots, C$ , ( $C$  represents the number of cluster) ( $C' \neq C$ ).

And for a given cluster  $X = \{x_1, x_2, \dots, x_n\}$ , the Silhouette Index of the  $X$  dataset can be expressed as follows:

$$Sil(X) = \left( \sum_{i=1}^n Sil(x_i) \right) / n \quad (12)$$

Fowlkes-Mallows Index(FMI) is an external criteria [17], and defined as

$$FMI = \sqrt{\frac{a}{a+b} \times \frac{a}{a+c}} \quad (13)$$

where  $a$  is the number of data with the same label and classified in the same cluster,  $b$  is the number of data with the same label but classified in different clusters, and  $c$  is the number of data with different labels but classified in the same cluster.

#### 4. RESULTS AND DISCUSSION

MAAP algorithm is written and ran with MATLAB R2014b. The test was carried on 4GB RAM Intel(R) Core(TM) i5-2670QM 2.20 GHz machine. We test this MAAP algorithm with:

- two-dimensional random non-partition data point sets of size 100, 500, 1000, 1500 and 2000 respectively to view the scala;
- two-dimensional random partition data point sets of size 100, 300, 500, 800, and 1000 respectively. The random non-partition data points are generated using uniform distribution from 0 to 1. The random partition data points are divided into four group and generated using  $k$ -means algorithm.
- Real datasets are used as shown in table 1. These datasets can be downloaded from *UCI-Repository* [9].

Table 1. UCI Datasets

Datasets	True Cluster	Number of Sampel	Dimension
4k2_far	4	400	2
Ionosphere	2	351	4
Iris	3	150	4
Wine	3	178	13

For the similarity, we use negative Euclidean's distance from the data points: for points  $x_i$  and  $x_j$ ,  $s(i, j) = - \|x_i - x_j\|^2$ .

##### 4.1. Experiments on Random Non-Partition Dataset

The clustering results on random non-partition dataset are presented in table 2, table 3. And Figure 2 shows an example result with number of data  $N = 1000$ . From those tables and Figure 2, although the number of cluster  $NC$  resulting from the MAAP-DDP algorithm are almost the same those from the original AP with  $p = \min(s)$ , MAAP-DDP algorithm is slower than original AP both with  $p = \text{median}(s)$  and with  $p = \min(s)$ . This make sense, because MAAP-DDP algorithm searches adaptively the  $\lambda$ -value in order to eliminate the oscillations.

For various values of  $N$  Silhouette index ( $Sil$ ) for both algorithm are almost the same with the range from 0.3 to 0.45. Interestingly, the  $Sil$  value from MAAP-DDP looks more constant, around 0,325. It means that the clusters resulting from the MAAP-DDP is more stable than those from original AP. The FMI can not be calculated because the random non-partition dataset do not have true labels.

##### 4.2. Experiments on Random Partition Dataset

The clustering results on random 4-partition dataset are presented in table 4, table 5. And Figure 3 shows an example result with number of data  $N = 1000$ . From those tables and Figure 3, the MAAP-DDP has succeeded to identify clusters according to the number of dataset's true labels. The number of dataset's true labels is 4, and the number of clusters ( $NC$ ) resulting from the MAAP-DDP is also 4 for various values of  $N$ .

The speed of the MAAP-DDP is comparable with those of original AP, it means that the execution time of MAAP-DDP is not slower than those original AP. Although the  $Sil$  values of the MAAP-DDP are smaller than

Table 2. AP Clustering Results on Random non-partition Data

$N$	AP ( $p = \text{median}(s)$ )			AP ( $p = \text{min}(s)$ )		
	$NC$	$ET(s)$	$Sil$	$NC$	$ET(s)$	$Sil$
100	10	0.021	0.440	3	0.117	0.420
500	19	0.963	0.374	8	0.531	0.384
1000	27	3.232	0.369	10	2.723	0.371
1500	34	11.019	0.361	12	8.233	0.359
2000	37	16.193	0.358	15	11.022	0.367

$N$  : Number of data     $NC$  : Number of Cluster     $ET$  : Execution Time  
 $Sil$  : Silhouette Index     $FMI$  : Fowlkes-Mallows Index

Table 3. MAAP-DDP Clustering Results on Random non-partition Data

$N$	MAAP-DDP				
	$NC$	$ET(s)$	$Sil$	$Const$	
100	6	0.048	0.325	1.0	
500	8	2.183	0.325	2.0	
1000	10	8.734	0.318	2.0	
1500	13	18.055	0.325	2.0	
2000	15	31.620	0.323	2.0	

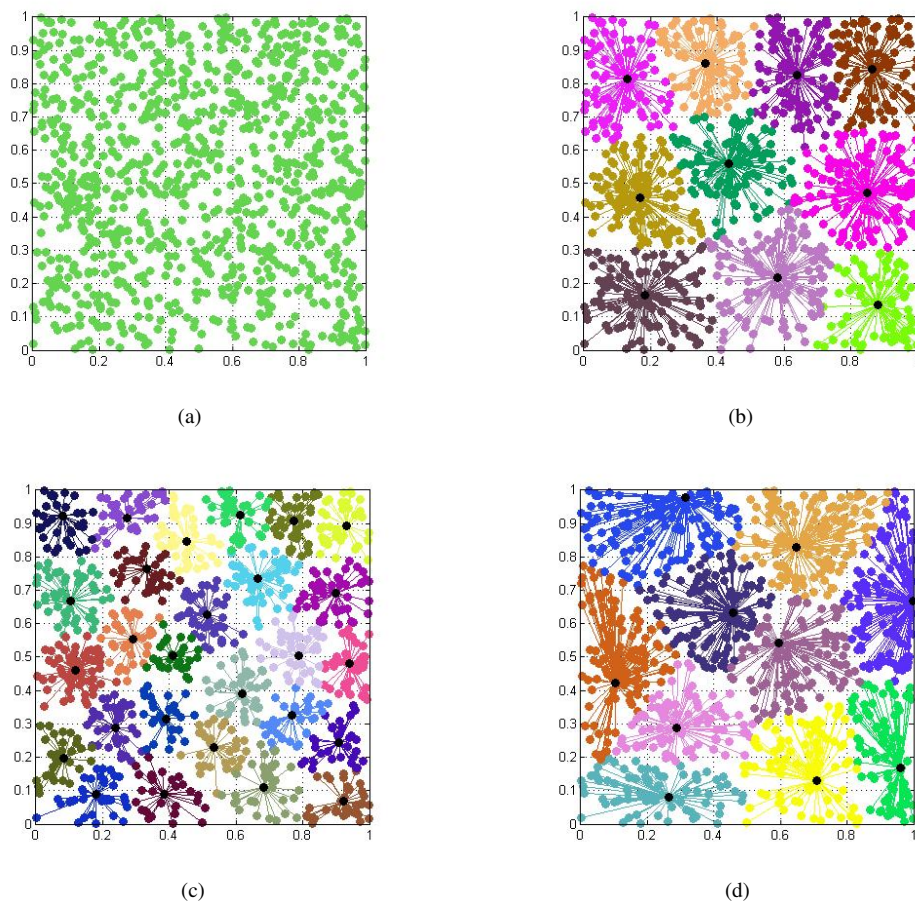


Figure 2. (a) Non-partition random data  $N = 1000$ ; (b) AP with  $p = \text{min}(s)$  for non-partition random data  $N = 1000$ ; (c) AP with  $p = \text{median}(s)$  for non-partition random data  $N = 1000$ ; (d) MAAP with distributed-data based  $p$  for non-partition random data  $N = 1000$ ;

those of AP,  $FMI$  index values of the MAAP-DDP are greater than those of original AP. It means that the MAAP-DDP is better in recovering the true clustering structure. The key parameter of the MAAP-DDP algorithm is  $Const$  in Eq.9. The algorithm is designed to find adaptively  $Const$ -value for obtaining the best clustering solution.

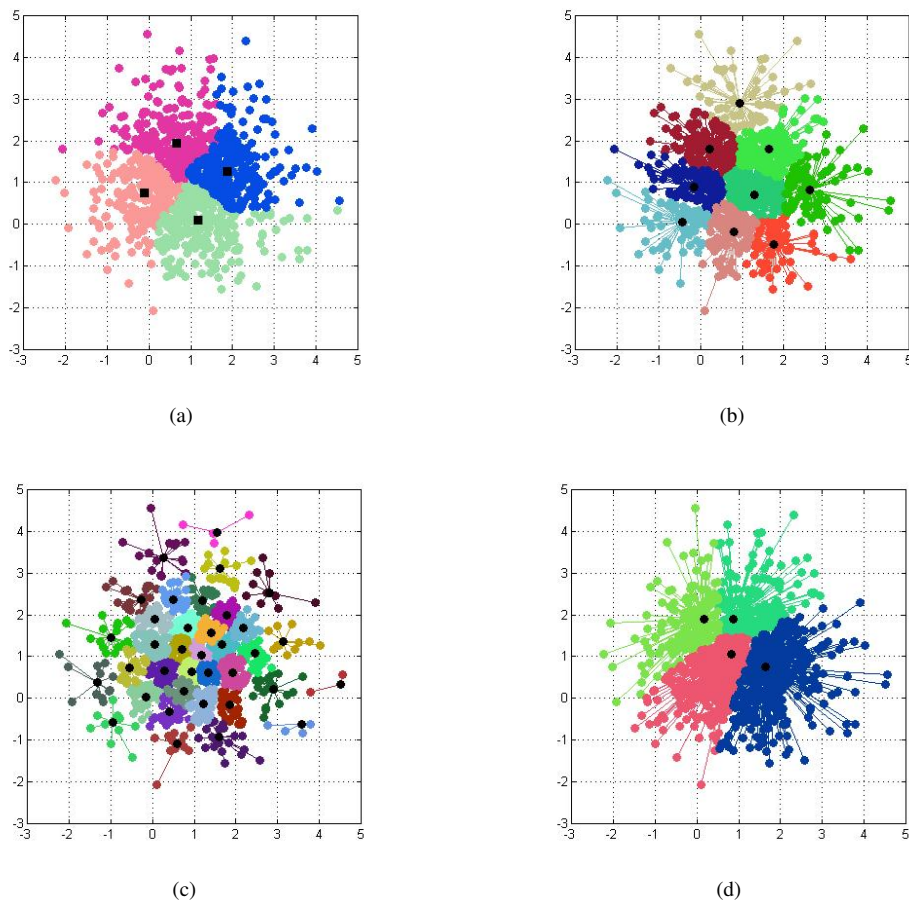


Figure 3. (a) 4-Partition random data  $N = 1000$ ; (b) AP with  $p = \min(s)$  for 4-partition random data  $N = 1000$ ; (c) AP with  $p = \text{median}(s)$  for 4-partition random data  $N = 1000$ ; (d) MAAP with distributed-data based  $p$  for 4-partition random data  $N = 1000$ ;

Table 4. AP Clustering Results on Random 4-partition Data

$N$	$NC$	AP ( $p = \text{median}(s)$ )			AP ( $p = \min(s)$ )			
		$ET(s)$	$Sil$	$FMI$	$NC$	$ET(s)$	$Sil$	$FMI$
100	11	0.094	0.358	0.58	2	0.090	0.368	0.718
300	18	0.477	0.350	0.38	4	0.502	0.338	0.639
500	27	1.502	0.336	0.35	5	1.444	0.292	0.556
800	35	4.252	0.348	0.31	7	4.872	0.317	0.457
1000	36	6.800	0.343	0.31	9	5.642	0.304	0.474

$N$  : Number of data       $NC$  : Number of Cluster       $ET$  : Execution Time  
 $Sil$  : Silhouette Index       $FMI$  : Fowlkes-Mallows Index

Table 5. MAAP-DDP Clustering Results on Random 4-partition Data

$N$	$NC$	MAAP-DDP			
		$ET(s)$	$Sil$	$FM$	$Const$
100	4	0.164	0.300	0.421	0.029
300	4	0.578	0.224	0.545	6.457
500	4	0.898	0.278	0.438	0.410
800	4	3.641	0.228	0.531	27.554
1000	4	6.312	0.262	0.474	30.850

### 4.3. Experiments on Real Datasets

From table 6 and table 7 it shows that MAAP has successfully identified the cluster of all real data, i.e. the number of clusters is equal to the number of true clusters ( $TC$ ), while the original AP did not all succeeded

to identify. The original AP with  $p(k)$  is the minimum value of  $s(i, k)$  succeeded to identify clusters of three real datasets, except for the real datasets of the Ionosphere. The key parameter of the MAAP-DDP algorithm is  $Const$  in Eq.9. The algorithm is designed to find adaptively  $Const$ -value for obtaining the best clustering solution.

The success of MAAP-DDP is supported by the values of its  $Sil$  that are greater than 0,300, and its  $FMI$  that are greater than . The values of its  $Sil$  are as follows: for 4k2\_far 0.455; for Ionosphere 0,513; for Iris 0.522 and for Wine 0.365. The values of its  $FMI$  are as follows: for 4k2\_far 0.681; for Ionosphere 0.719; for Iris 0.679 and for Wine 0.622.

Table 6. AP Clustering Results on Real Datasets

	AP (p = min(s))					AP (p = med(s))			
	$TC$	$NC$	$ET(s)$	$Sil$	$FMI$	$NC$	$ET(s)$	$Sil$	$FMI$
4k2_far	4	4	0,790	0,761	1,000	6	0,808	0,450	0,743
Ionosphere	2	4	0,633	0,386	0,575	28	1,517	0,444	0,413
Iris	3	3	0,129	0,541	0,809	6	0,103	0,469	0,724
Wine	3	3	0,165	0,491	0,830	11	0,174	0,314	0,468
$TC$ : True Cluster			$NC$ : Number of Cluster			$ET$ : Execution Time			
$Sil$ : Silhouette Index			$FMI$ : Fowlkes-Mallows Index						

Table 7. MAAP-DDP Clustering Results on Real Datasets

	MAAP-DDP					
	$TC$	$NC$	$ET(s)$	$Sil$	$FMI$	$Const$
4k2_far	4	4	1,318	0,455	0,681	4,879
Ionosphere	2	2	1,228	0,513	0,719	160,200
Iris	3	3	0,771	0,522	0,679	9,252
Wine	3	3	0,795	0,365	0,622	7,559

## 5. CONCLUSIONS AND FUTURE RESEARCH

From above results it can be concluded: (i) the proposed algorithm, MAAP-DDP, is slower than original AP for random non-partition dataset, (ii) for random 4-partition dataset and real datasets the proposed algorithm has succeeded to identify clusters according to the number of dataset's true labels with the execution times that are comparable with those original AP.

Beside that the MAAP-DDP algorithm demonstrates more feasible and effective than original AAP procedure. As we know that the original AAP algorithm stops when the values of  $C(i)$  is lower than 2, while MAAP-DDP algorithm terminates after the best clusters founded. The key parameter of the MAAP-DDP algorithm is  $Const$  in Eq.9. The algorithm is designed to find adaptively  $Const$ -value for obtaining the best clustering solution.

In the future, for verification of the algorithm we have to test the MAAP-DDP algorithm with the other kinds of dataset, e.g. synthetic dataset, face-image dataset, and so on.

## ACKNOWLEDGMENT

The Authors gratefully acknowledge Gunadarma University for providing research funding and for permission in using the research facilities.

## REFERENCES

- [1] K. R. Nirmal and K. V. V. Satyanarayana, "Issues of k means clustering while migrating to map reduce paradigm with big data: A survey," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 6, pp. 3047–3051, December 2016.
- [2] X. Shi, W. Wang, and C. Zhang, "An empirical comparison of latest data clustering algorithms with state-of-the-art," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 5, no. 2, pp. 410–415, February 2017.
- [3] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, pp. 972–976, 2007.
- [4] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.



- [5] D. Xia, F. Wu, X. Zhang, and Y. Zhuang, "Local and global approaches of affinity propagation clustering for large scale data," *CoRR*, vol. abs/0910.1650, 2009. [Online]. Available: <http://arxiv.org/abs/0910.1650>
- [6] R. Refianti, A. Mutiara, and A. Syamsudduha, "Performance evaluation of affinity propagation approaches on data clustering," *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 7, no. 3, 2016.
- [7] R. Refianti, A. Mutiara, and S. Gunawan, "Time complexity comparison between affinity propagation algorithms," *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 7, pp. 1497–1505, April 2017.
- [8] V. K. Ramana, "Enhanced fuzzy based affinity propagation for multispectral images," *Imperial Journal of Interdisciplinary Research (IJIR)*, vol. 12, no. 12, pp. 1548–1551, 2016.
- [9] A. Asuncion and D. Newman, "UCI machine learning repository," University of California, Irvine, School of Information and Computer Sciences, 2007. [Online]. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [10] J. Kogan, *Introduction to Clustering Large and High-Dimensional Data*. Cambridge University Press, 2007.
- [11] K. Wang, J. Zhang, D. Li, X. Zhang, and T. Guo, "Adaptive affinity propagation clustering," *CoRR*, vol. abs/0805.1096, 2008. [Online]. Available: <http://arxiv.org/abs/0805.1096>
- [12] R. Refianti, A. Mutiara, and A. Syamsudduha, "Performance evaluation of affinity propagation approaches on data clustering," *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 7, no. 3, 2016.
- [13] K. Zhang and X. Gu, "An affinity propagation clustering algorithm for mixed numeric and categorical datasets," *Mathematical Problems in Engineering*, September 2014.
- [14] Y. He, Q. Chen, X. Wang, R. Xu, X. Bai, and X. Meng, "An adaptive affinity propagation document clustering," in *Informatics and Systems (INFOS), 2010 The 7th International Conference on*, March 2010, pp. 1–7.
- [15] M. Rueda, M. A. Rodriguez, and B. A. Hawkins, "Identifying global zoogeographical regions: Lessons from wallace," *Journal of Biogeography*, vol. 40, no. 12, pp. 2215–2225, Dec. 2013.
- [16] P. Li, H. Ji, B. Wang, Z. Huang, and H. Li, "Adjustable preference affinity propagation clustering," *Pattern Recognition Letters*, vol. 85, pp. 72–78, January 2017.
- [17] P. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J. Comp App. Math*, vol. 20, pp. 53–65, 1987.