**IMPLEMENTATION AND TESTING CLUSTERING ALGORITHMS AFFINITY PROPAGATION AND ADAPTIVE AFFINITY PROPAGATION USING STUDENT DATA (GRADE POINT AVERAGE AND HOME DISTANCE)**

**Millati Izzatillah1, Achmad Benny Mutiara2**

Informatics Indraprasta PGRI University1,

Computer and Information Technology Gunadarma University2

mizzatillah@gmail.com 1, amutiara@staff.gunadarma.ac.id 2

**Abstrak**

Clustering is method to classify data easily that has purpose is to look at the correlation between data attributes. Clustering is grouping data points process based on similarity value to determine cluster center. Affinity Propagation (AP) and Adaptive Affinity Propagation (Adaptive AP) are clustering algorithms that produce number of cluster, cluster members and exemplar of each cluster. To find out more effective algorithm when clustering data, there is needs to implements and tests both algorithms. This clustering application was implemented those algorithms by Matlab R2013a 8.10 using Gunadarma University student data based on GPA and home distance as much as 250 records. The result of testing this clustering application was analyzed by comparing both algorithms to find out one of the best algorithm in this case and to find out the correlation between home distance and student GPA in Gunadarma University.

**Kata Kunci :** Clustering, Adaptive Affinity Propagation, Affinity Propagation, GPA, Home Distance, Matlab.

1. **INTRODUCTION**

Clustering is traditionally fundamental problem in data mining. Clustering is a method of grouping data based on the similarity value between data points to determine the cluster center [1].

Affinity propagation (AP) is a clustering algorithm that has a low error rate, high speed, flexible and simple grouping [2]. Affinity propagation works by “message passing” [3], which tested the possibility of all the data points to be cluster center (exemplar). Results of Affinity propagation is the number of clusters, cluster members and exemplar of each cluster.

Along with its development, Affinity propagation does not always produce the ideal cluster. Affinity propagation has two limitations: AP is hard to know what value of parameter “preference” can yield an optimal clustering solution, and oscillations cannot be eliminated automatically if occur [4]. The limitations of AP will produce cluster that have not been optimal. To overcome these limitations, Affinity Propagation developed into Adaptive Affinity Propagation (called Adaptive AP).

Adaptive AP adjustment of the damping factor to eliminate oscillations (called adaptive damping), adaptive escaping oscillations by decreasing p when adaptive damping method fails (called adaptive escape), and adaptive searching the space of p to find out the optimal clustering solution suitable to a data set (called adaptive preference scanning) [4].

Therefore, the author need to tests and implements Affinity Propagation algorithm and Adaptive Affinity Propagation algorithm and compare both the algorithms to find out how effective those algorithms in clustering data. Tests performed used Gunadarma University student data based Grade Point Average (GPA) and home distance implemented by Matlab R2013a 8.10.

Affinity propagation (AP) works on “passing message” (B. J. Frey, Dueck, 2007) [3], AP does not fix the number of clusters beforehand and it seeks to identify each cluster by one of its element, the so-called exemplar, instead of the virtual geometrical center. AP controls the number of identified clusters by parameters called preferences (Zhang, 2008) [4].

Adaptive Affinity propagation (Adaptive AP) is development of Affinity Propagation (AP). AP has two limitations: it is hard to know what value of parameter “preference” can yield an optimal clustering solution and oscillations cannot be eliminated automatically if occur. Adaptive AP adjustment of the damping factor to eliminate oscillations (called adaptive damping), adaptive escaping oscillations by decreasing *i* when adaptive damping method fails (called adaptive escape), and adaptive searching the space of p to find out the optimal clustering solution suitable to a data set (called adaptive preference scanning) (Kaijun Wang, 2007) [4].

1. **RESEARCH METHOD**
2. **Data Collection**

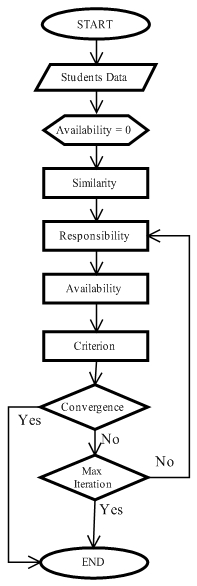
Data collection is the stage to collect data that will be used to test in this research. The data used in this research is the data obtained from the questionnaires to the students at the Gunadarma University as respondents. The amount of data used 250 records. Data is taken randomly without looking at the background of the respondent’s major. From 250 respondents who had filled out questionnaires obtained 38% information system, 23% informatics engineering, 10% accounting, 10% psychology, 6% management, 4% computer system, 2% architecture, 2% electrical engineering, 2% informatics management, 2% english literature, 1% engineering. The data used in this research only two variables, Grade Point Average (GPA) and home distance. Home distance is taken by distance from home address to the campus Gunadarma in Margonda, Depok.

1. **Requirement Analysis**

Requirement analysis is the stage to explain the requirements. Application functional requirements deﬁne the things required by applications that will be built, among others:

1. Ability to cluster data with two algorithms, AP and Adaptive AP.
2. Ability to provide a number of clusters and show its exemplar.
3. Ability to plotting the results of the cluster.
4. Ability to calculate runtime algorithms work AP and Adaptive AP
5. **Affinity Propagation**

Affinity propagation worked through several stages. These stages are described by the program flowchart in figure 1.



Source : Author Documentation

**Figure 1. Program Flowchart Affinity Propagation**

Student data used in this research are Grade Point Average (GPA) and home distance as much as 250 records. The data entered is 250 records, and then the data will be matrix *s*(250x250). Similarity shows the value for the suitability of a data point to another data point. Similarity is denoted by *s(i,k)*, *i* indicates the row and *k* as column, which shows how suitable data point *i* to the data point *k*. Similarity values used in affinity propagation using the negative value of the distance each data point. The distance calculation use Euclidean Distance technique. Author took five records from data to give an example for finding similarity value. Data can be seen in table 1.

**Tabel 1. Five record from student data**

|  |  |  |
| --- | --- | --- |
| No | GPA | Home Distance (KM) |
| 1 | 3.75 | 15 |
| 2 | 3.68 | 25 |
| 3 | 3.55 | 15 |
| 4 | 3.71 | 0.1 |
| 5 | 2.62 | 5 |

Source : Author Documentation

Calculate similarities these data as follows.

After each *s(i, k)* calculated it will get the following matrix.

**Tabel 2. Similarity Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 0 | -10.0002 | -0.2000 | -14.9001 | -10.0636 |
| 2 | -10.0002 | 0 | -10.0008 | -24.9000 | -20.0281 |
| 3 | -0.2000 | -10.0008 | 0 | -14.900 | -10.0432 |
| 4 | -14.9001 | -24.9000 | -14.9009 | 0 | -5.0198 |
| 5 | -10.0636 | -20.0281 | -10.0432 | -5.0198 | 0 |

Source : Author Documentation

Diagonal values above table is 0, the value must be replaced with the value so-called preference. Preference value used is the minimum value or median value of similarity was formed. Minimum preference using the minimum value that is -24.9000 and Median preference using the middle value that is -10.0534. So fill the diagonal with a minimum preference value.

After getting similarity matrix, the next step is count value of responsibility. Counting responsibility matrix is as follows.

After each *r(i,k)* minimum preference calculated it will get the following matrix.

**Tabel 3. Responsibility Matrix Minimum Preference**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | -12.3500 | -4.9001 | 4.9001 | -7.3500 | -4.9318 |
| 2 | 0.0003 | -7.4499 | -0.0003 | -7.4499 | -5.0139 |
| 3 | 4.9004 | -4.9004 | -12.3500 | -7.3504 | -4.9216 |
| 4 | -4.9401 | -9.9401 | -4.9405 | -9.9401 | 4.9401 |
| 5 | -2.5219 | -7.5041 | -2.5117 | 2.5117 | -9.9401 |

Source : Author Documentation

For the first iteration, the initial value of availability that is used to determine responsibility values is zero. Each *a(i,k)* and *a(k,k)* minimum preference calculated it will get the following matrix.

**Tabel 4. Availability Matrix Minimum Preference**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 2.4504 | -3.7249 | -6.1750 | -3.7142 | -2.5000 |
| 2 | -3.7248 | 0 | -3.7249 | -3.7142 | -2.5000 |
| 3 | -6.1749 | -3.7249 | 2.4501 | -3.7142 | -2.5000 |
| 4 | -3.7246 | -3.7249 | -3.7249 | 1.2558 | -4.9701 |
| 5 | -3.7246 | -3.7249 | -3.7249 | -4.9701 | 2.4701 |

Source : Author Documentation

Criterion is the matrix of the sum of the value of responsibility and availability. Diagonal criterion has positive value will be exemplar. Data point that has the nearest value of similarity with exemplar will be in one cluster. If the iteration is done a hundred times for the five records from student data, it will get criterion matrix as follows.

**Tabel 4. Criterion Matrix Minimum Preference**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | -24.6986 | -24.6994 | 14.8394 | -19.6758 | -14.8394 |
| 2 | -24.6986 | -14.8992 | 14.8992 | -19.8750 | -15.0030 |
| 3 | -39.6887 | -39.6901 | 4.9758 | -34.6668 | -29.8091 |
| 4 | -24.6986 | -24.8983 | 4.9758 | -4.9758 | -4.9758 |
| 5 | -24.7191 | -24.8841 | 4.9758 | -4.9758 | -4.9758 |

Source : Author Documentation

Third column on criterion matrix minimum preference has positive value that is 4.9758, then there will be one cluster with third data becomes exemplar. All data points has near value of similarity with data point 3. So, data points 1, 2, 4 and 5 are in one cluster where the exemplar is a data point 3.

After getting the number of clusters of each iteration, the next stage is to determine whether the number of optimal clustering. Optimal number of clusters will happen when it reaches the value converges. Convergent value will be achieved with some condition as follow.

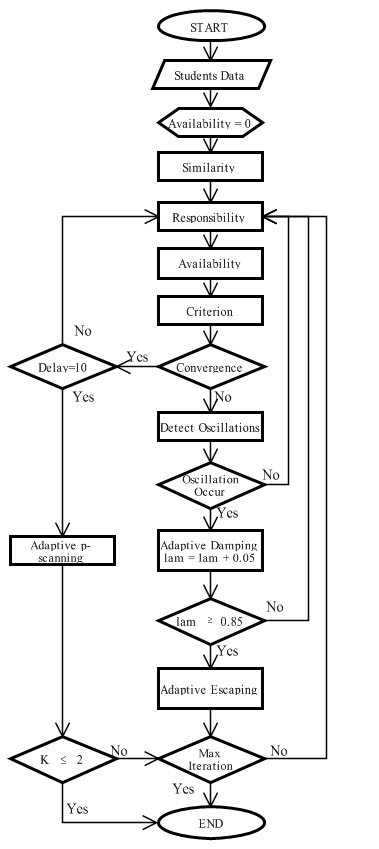
* + Number of clusters generated will always be the same after a minimum iteration. Minimum iteration is 50 times iteration [3].
  + Exemplar (cluster center) unchanging in few times.

If the particular iteration converges stated already in this stage, then the algorithm stops (stop condition) if it does not converge, it will check whether the iteration has reached the maximum iteration (*maxit = 50000* is the large maximal iteration [4]). If not converge, then back to responsibility. If the algorithm has reached the maximum iteration will stop and result in number of clusters although not achieving convergent value.

1. **Adaptive Affinity Propagation**

Adaptive AP is the development of AP algorithm which proposes improvements to achieve more optimal clustering results and overcome the limitations of AP. Therefore, some of the early stages of the Adaptive AP and AP are the same as on the stage (1) input, (2) similarity, (3) responsibility, (4) availability and (5) criterion.

Adaptive Affinity propagation (Adaptive AP) worked through several stages those are described by the program flowchart in figure 2.



Source : Author Documentation

**Figure 2. Program Flowchart Adaptive Affinity Propagation**

After getting the number of clusters of each iteration, the next stage is to determine whether the number of optimal clustering. Optimal number of clusters will happen when it reaches the value converges. Convergent value will be achieved with some condition as follows.

* + Number of clusters generated will always be the same after *v* times (50 iterations), (window size = 40, plus delay = 10).
  + Exemplar (cluster center) unchanging in few times.

In Adaptive AP, after checking convergent, there are several stages before checking the maximum iteration (*maxit = 50000* is the large maximal iteration [4]). If k converge in additional *dy* iteration (*delay = 10*) will perform adaptive p-scanning technique. If not converge in additional dy iteration (*delay ≠ 10*) then will resume responsibility iteration. But if not converge, then it will check the existence of oscillation.

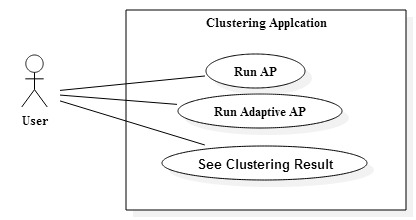
Detect oscillations are too complex to be described. So detect oscillation with defining the non-oscillation features. Conditions of non-oscillations when exemplar of decreasing or unchanging during the iterative process. Exemplar that decreasing or unchanging is going to convergence. Oscillation occurs if the non-oscillation number of features in the monitoring window is less than two third of the window size. So, if exemplar increasing or changing to two-third of the window size *2/3(40)*, then the oscillation occurs. Oscillation occurs and the lam increase and initial lam value is 0.5.

This technique will work when the lambda has reached 0.85 or higher. The large lam Brings little effect suggest that the oscillations are pertinacious under given p, so the alternative is to decrease *p* away from the given *p* to escape from oscillation. Adaptive Escaping is designed to avoid the caused Adaptive damping oscillation fails to depress oscillation. After this technique, check whether it has reached the maximum iteration. If it reached maximum iteration then the algorithm stops, otherwise go to responsibility.

This technique works when *K* converge in additional dy iteration (*delay = 10*) with decrease *p* by step *ps*. Now Adaptive AP gives clustering solutions with different number of clusters. Adaptive *p-scanning* give the cluster validation that used to evaluate the quality of the clustering result. This validity index silhouette used, number of clusters with the highest index value silhouette will be the optimal number of clusters. Adaptive *p-scanning* will stop when the number of clusters less than two (*K ≤ 2*). If the number of clusters are more than two, then it will check the maximum iteration.

1. **Application Design**

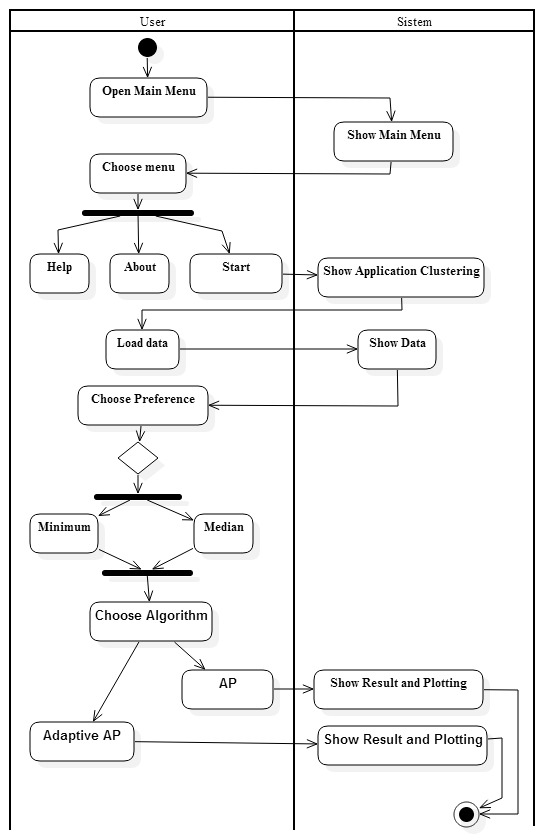
Use Case diagram illustrates the case, the actor and the relations between cases and actors. Use case for clustering applications AP and Adaptive AP can be seen in figure 3.3



Source : Author Documentation

**Figure 3. Use Case Diagram**

Activity diagram describes the activity that occurs when the user running the application. Stages of the process of the activity diagram can be seen in figure 4.



Source : Author Documentation

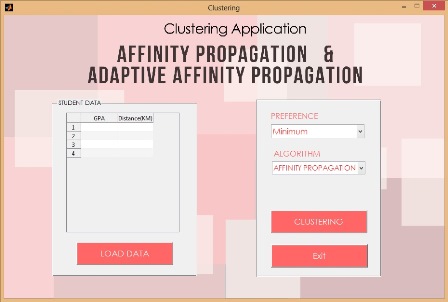
**Figure 3. Activity Diagram**

In the activity diagram on figure 3, user opens main menu and application will display main menu. After main menu appears, there are three activities that can be performed by the user, there are opens help menu, opens about menu and goes start menu to run the application. When a user chooses to run the application, the application will display a menu clustering. After that, user loads data and application will displays data that will be used in clustering. User selects the preferences and the algorithms used before clustering. If user chooses to run the AP algorithm then the application will display the results of clustering and plotting to AP algorithm. However, if user chooses to run the algorithm Adaptive AP then the application will display the results of clustering and plotting for Adaptive algorithms AP.

1. **RESULT AND DISCUSSION**
2. **Implementation**

This clustering application is a desktop-based that implemented in Matlab 8.10 R2013a. Implementation of this application starts from making the main menu page, clustering pages, help pages, about page, output AP page and output Adaptive AP page. Creating an interface in Matlab using Matlab GUIDE and save the file with the extension .fig and file .m will follow automatically. Additions code or function will be written in the .m file.

Here is a view of an application that is designed:

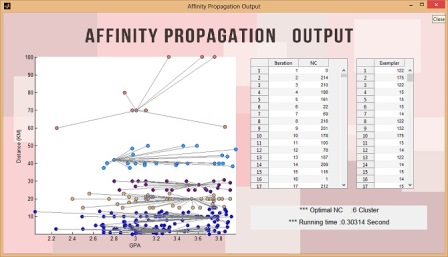


Source : Author Documentation

**Figure 4. Clustering Application**

When load button on click like in figure 4, application will display data into a table for clustering. After data has been loaded, data will be clustered with click on clustering button. Clustering button works after selecting preference (minimum or median) and algorithm (affinity propagation or adaptive affinity propagation) that will be used in clustering.

When clustering button on click in figure 4, application will display clustering results, there are plot, texts and tables like figure 5 below.

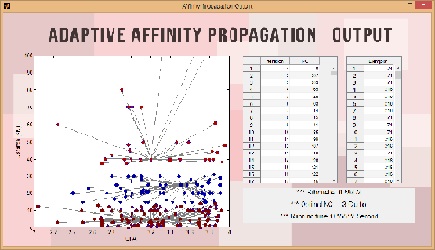


Source : Author Documentation

**Figure 5. Output AP**

In figure 5 output AP, there are Axes to display the results of plotting, table iteration and table number of clusters, table exemplar of each iteration, and text Optimal NC and Runtime in clustering data.

Output Adaptive Affinity Propagation can be seen in figure 6 below.



Source : Author Documentation

**Figure 6. Output AAP**

When clustering button on click with AAP is choosed, application will display like in figure 6 as clustering results, there are plot, texts and tables. In figure output AAP, there are Axes to display the results of plotting, table iteration and table number of clusters, table exemplar of each iteration, and text Optimal NC and Runtime in clustering data.

1. **Testing**

Application has been made in the executable file and ready for use on a computer without installation process Matlab. The amount of data that tested 250 records. The test steps are described as follows.

* The application is run by double clicking on the file Clustering Application. exe. Then, the application will display the main menu page.
* Click the start button to get into the menu clustering. Then the application will display the page clustering. Click load button to display the data.
* In the clustering process, required by selecting the preference value in the pop-up preferences and choose the algorithm used.
* The first test is used the minimum preference clustered by the AP algorithm. Then press the button cluster. Testing performed three times.
* The second test is used the median preference clustered by the AP algorithm. Then press the button cluster. Testing performed three times.
* The third test is used the minimum preference clustered by the Adaptive AP algorithm. Then press the button cluster. Testing performed three times.
* The fourth test is used the median preference clustered by the Adaptive AP algorithm. Then press the button cluster. Testing performed three times.

These algorithms has been testing as much as each of the three times the value of the median and minimum preference for affinity propagation algorithm and the Adaptive AP. The clustering results by using affinity propagation algorithm minimum preference.

1. **CONCLUSION**

Testing and implementation affinity propagation and adaptive affinity propagation algorithms in student data based on GPA and home distance have been successfully implemented with Matlab R2013a 8.10 using 250 records of Gunadarma students. Testing those algorithms implemented in desktop-based clustering application.

This clustering application has been tested as much as each of the three times for AP minimum preference, AP median preference, Adaptive AP minimum preference, Adaptive AP median preference. Testing the AP minimum preference was achieved maximum iteration because it was not convergence until maxit. Testing the Adaptive AP minimum and median preference was not achieved maximum iteration because iteration in every testing stop at convergence. AP produced different NC in every testing. So, AP gives dynamic clustering result and Adaptive AP gives static clustering result because it produced same NC in every testing.

There is not convergence in AP. So, AP produced not optimal clustering result. Adaptive AP managed to overcome the limitations of AP that produces optimal clustering result with achieving convergence without testing that achieve maxit. From the results mentioned above the improvement that given to the AP from Adaptive AP as follow.

* Adaptive AP produced optimal clustering.
* Adaptive AP produced static clustering result.
* Adaptive AP showed convergence is not the final stage to stop the iteration to produce optimal cluster because there is possibility oscillation occurs with both data points that alternate between being exemplar or non-exemplar. Adaptive AP automatically eliminate oscillation.
* Although the AAP has more code lines but in one of the AAP Runtime testing faster than with AP. That is because the AP performs iterations until maxit. The more iteration, the runtime will be longer.
* Optimal number of cluster is generated by AAP from NC that has the highest value of silhouette. Silhouette value is used to evaluate the appropriate NC to generate optimal clustering result.

The results of affinity propagation clustering algorithm and adaptive affinity propagation algorithm in student data based on GPA and home distance is no correlation between the home distance to the GPA that they get. It can be concluded that the student GPA is not influenced by the home distance. Home distance to the campus does not affect study achievement of Gunadarma University students.

In a further development, testing is will be implemented for the larger amount of data and variables used to see the correlation between the data can be increased more than two.

**DAFTAR PUSTAKA**

[1] Xu-qing ZHANG Yue-ting ZHUANG Ding yin XIA, Fei WU. Local and global approaches of affinity propagation clustering for large scale data. *Journal of Zhejiang University SCIENCE A*, 2008.

[2] Precha Thavikulwat. Affinity propagation: A clustering algorithm for computer-assisted business simulations and experiential exercises. *Developments in Business Simulation and Experiential Learning*, 2008.

[3] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *sciencemag*, 2007.

[4] Dan Li-Xinna Zhang Kaijun Wang, Junying Zhang and Tao Guo. Adaptive affinity propagation clustering. *Automatica Sinica*, 2007.