Constraint Rules and Matching Micro-clusters Based Affinity Propagation Clustering Algorithm

Li-min WANG[1], You ZHOU[2], Xu-ming HAN[3], Yi-zhang WANG[4]*, Jing-lin YU[1], Shuai WANG[5]

[1] School of Internet Finance and Information Engineering, Guangdong University of Finance, Guangzhou, 510521, China
wlm_new@163.com, yujinglin@gduf.edu.cn

[2] College of Computer Science and Technology, Jilin University, 130012, Changchun, China
zyou@jlu.edu.cn

[3] College of Information Science Technology, Jinan University, Guangzhou, 510632, China
hanxvming@163.com

[4] College of Information Engineering, Yangzhou University, 225127, Yangzhou, China
wyzhang_new@sina.com (*Corresponding author*)

[5] Business School, Jilin University, 130012, Changchun, China
wang.s@qq.com

Abstract: The performance of original affinity propagation (AP) clustering algorithm is greatly influenced by an important parameter: *preference* (median of similarities between data points), and it may be difficult to identify complex structure data. To address the afore-mentioned issues, this paper proposes two novel methods namely the constraint rules-based affinity propagation (CRAP) and matching micro-clusters hierarchical clustering algorithm (MMHC). The CRAP algorithm can obtain better results by searching the optimal *preference* value by means of the constraint rules-based search algorithm (CRS). The MMHC algorithm initially takes results of AP as micro-clusters, then they are matched in order to achieve the right partitions of complex structure data. Experimental results demonstrate that the improved clustering algorithm performs better than AP.

Keywords: Affinity propagation, Constraint rules, Micro-clusters hierarchical clustering.

1. Introduction

Frey and Dueck proposed a novel clustering algorithm called affinity propagation (AP), which conducts by passing messages (Frey & Dueck, 2007). Unlike k-means, AP begins with a large number of clusters then makes pruning decisions and it does not depend on initial center selection. The above advantages decide that AP is a better tool for data mining and pattern recognition.

At present, this method has been successfully applied to image segmentation (Zhang et al., 2017), image retrieval (Sun et al., 2013), image classification (Li et al., 2017), gene recognition (Sun et al., 2017), text clustering and many other fields (Zhang et al., 2013). Their researches extend AP's application areas.

However, it is difficult to obtain appropriate *preferences* (parameter of AP algorithm) values. Moreover, AP fails to deal with complex structure data because it is based on Euclidian distance similarities.

This paper proposes a parameter search algorithm and employs AP's features of identifying local dense data points to identify complex structure data. The contributions of the proposed method are as follows: the relations of internal evaluation index and clustering results are discussed, constraint rules-based affinity propagation (CRAP) is proposed to obtain optimal number of clusters. This paper discusses how to define distance of inter-class and intra-class in complex structure data, MMHC algorithm is proposed to better identify clusters with complex structure.

The remainder of this paper is as follows: Section 2 presents a brief overview of AP and its related works, while Section 3 draws the proposed constraint rules-based affinity propagation to get more promising results. Section 4 explains the proposed matching micro-clusters hierarchical clustering algorithm, while Section 5 presents experimental results using UCI and synthetic data sets. Section 6 shows the application in Chinese stock quality assessment. Finally, Section 7 concludes this paper.

2. Related Work

In this section, AP and its related works are reviewed. AP takes similarities between different data points as model input. For data points x_i and

x_k , their similarity is computed by $S(i,k) = -\|x_i - x_k\|$ (Wei et al., 2017). Later, the median of similarities is chosen as shared values, which is called *preferences* (*P*) (Wang et al., 2017). Then, there are two kinds of exchanged messages deciding which points are exemplars (cluster centers) and which data point they belong to (Sun et al., 2017; Li et al., 2017). The "Responsibility" reflects how well-suited point *k* is to be chosen as the exemplar for point *i*, the "Availability" reflects how appropriate it is for point *i* to choose *k* as its cent. The two kinds of messages are computed by equations (1) and (2):

$$R(i,k) = S(i,k) - \max_{k's.t.k'\neq k}\left\{A(i,k') + S(i,k')\right\} \qquad (1)$$

$$A(i,k) = \min\left\{0, R(k,k) + \sum_{i's.t.i'\notin\{i,k\}}\max\left\{0, R(i',k)\right\}\right\} \qquad (2)$$

To prevent the algorithm from oscillation when updating the messages, damping factor λ is used by equation (3) and (4):

$$R(i,k)^{(t+1)} = \lambda R(i,k)^t + (1-\lambda)R(i,k)^{(t-1)} \qquad (3)$$

$$A(i,k)^{(t+1)} = \lambda A(i,k)^t + (1-\lambda)A(i,k)^{(t-1)} \qquad (4)$$

The algorithm is terminated when cluster centers do not change for ten iterations (Fan et al., 2019). The procedure of original AP algorithm is shown in Table 1.

Table 1. Procedure of original AP algorithm

Initialization

input *N* data points, compute similarities between different data points: $S(i,k) = -\|x_i - x_k\|$, *P*= median of similarities between data points.

for *i*=1: number of data points

    $S(i,i) = P$

end

for *iter*=1:10000

Update two kinds of messages: responsibility R(i, k) and availability A(i,k) for all the data points.

    In the first iteration, A=zeros(N,N) , R=zeros(N,N) . *A* and *B* are *N**N zero matrix.

    $Eold = A+R$

    AS=A+S

    [*Y,I*]=max(*AS*,[],2): *Y* saves the maximum value of each row of matrix *AS*, *I* save their positions.

---

    for *i*=1: *N*

        AS(*i,I*(*i*))=-*realmax*: *realmax* is the largest positive number in the MATLAB.

    end

    [*Y2,I2*]=max(*AS*,[],2): *Y2* saves the maximum value of each row of matrix *AS*.

    R=*S*-repmat(*Y*, [1,*N*]): repmat() means copying 1*N matrix *Y*.

    for i=1:*N*

        R(*i,I*(i))=*S*(*i,I*(*i*))-*Y2*(*i*)

    end;

    R=(1-λ)*R+ λ *Rold

    Aold=*A*

    Rp=max(*R*,0): choose the larger value between each element in *Rp* and zero.

    for *k*=1:*N*

        Rp(*k,k*)=*R*(*k,k*)

    end;

    A=repmat (sum (*Rp*,1), [*N*,1])-*Rp*: sum(*Rp*,1) means adding each row of *Rp*, than one can copy *N**1 sum(*Rp*,1).

    dA=diag(*A*): diag() means elements on the diagonal of a matrix.

    A=min(*A*,0): choose the smaller value between each element in *A* and zero.

    for *k*=1:*N*

        A(*k,k*)=*dA*(*k*)

    end

    A=(1- λ)*A+ λ *Aold

    $E = A+R$

    if diag(*Eold*) == diag(*E*): diag() means elements on the diagonal of a matrix.

    count=*count*+1

        if *count*==10

        break

        end

    end

end

Obtain final clustering results *idx*

centers=find(diag(*E*)>0): the corresponding diagonal elements greater than 0 are centers.

[tmp *idx*]=max(*E*(:,*I*),[],*2*): each data points is assigned to the center with largest value. The *idx* is final clustering results.

In recent years, scholars have developed many improving methods, all of which focus on three issues: similarity, complex data processing and application. (Gao et al., 2019) identified the problem of vulnerable line in smart grid systems by using an improved AP, which considers the topological parameters and the electrical properties (Gao et al., 2019). (Lu & Carreira-Perpinan, 2008) developed a new constrained spectral clustering algorithm based on AP. (Zhang et al., 2013) redefined manifold similarity so as to identify complex structure data flow. (Zhang et al., 2014) proposed a hierarchical AP. Zhang X. applied AP to deal with streaming data just like (Givoni et al., 2012) applied a combination of AP and hierarchical clustering to simulate HIV strains mutation. (Qasim et al., 2013) combined AP and text concept map semi-automatic generation technique supported by experts. (Hassanabadi et al., 2014) used affinity propagation to process vehicular network.

Beside AP algorithm, there are other important clustering algorithms such as partition-based clustering (Feldman et al., 2020), density-based clustering (Darabkh et al., 2020; Angelova et al., 2020), hierarchical clustering (Dutta et al., 2020), and deep clustering (Fard et al., 2020).

Clustering is an important data mining technology in finance, economics and so on, AP can work well on them. Inspiration is drawn from the related works and then this paper proposes CRAP and MMHC for addressing these issues. For enhancing the performances of AP, CRAP is an optimal parameter method. For identifying the complex structure data, MMHC is different from (Zhang et al., 2013), which redefined manifold similarity, while MMHC constructs a graph and defines new graph similarity to measure intra-class compactness and inter-class separation.

## 3. Constraint Rules based Affinity Propagation (CRAP)

Relations of *Preferences* (*P*) and Number of Clusters (*NC*)

Although there is no exact relation between the parameters P and NC, one can find their fuzzy correlation. As it is shown in Figure 1 (a) ~ (f), a synthetic dataset generated by normal distribution is given (class1~$N(\mu = 2, \sigma = 1)$, class2~$N(\mu = 7, \sigma = 1)$)), NC becomes smaller, while the absolute P value increases gradually. The stable spaces, red thick line area in Figure 1(f), will naturally occur in P-NC plot. High-quality clustering results (the true NC) are between these two stable spaces, but it is hard to know which stable stage is right.
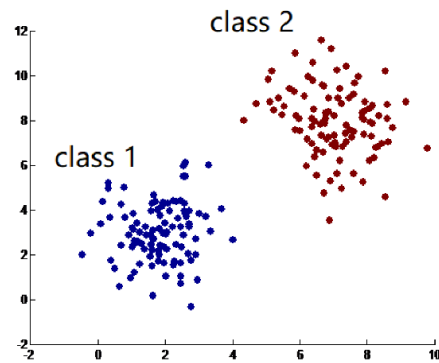
(a) ground-truth of synthetic dataset (class1~$N(\mu = 2, \sigma = 1)$, class2~$N(\mu = 7, \sigma = 1)$)), different colour represents different clusters
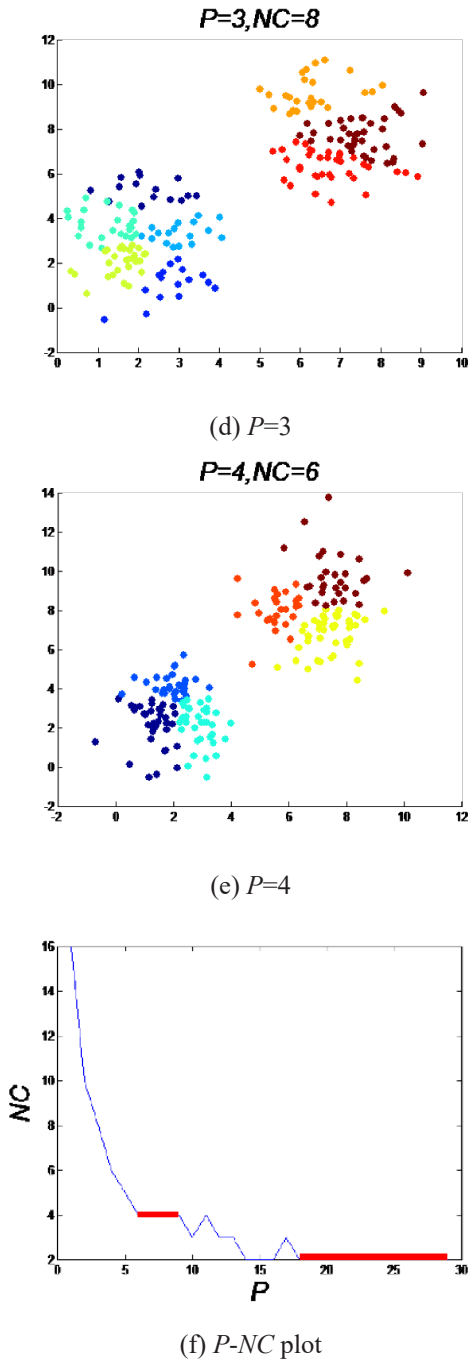

(b) *P*=1

(c) *P*=2

(d) *P*=3

(e) *P*=4

(f) *P-NC* plot
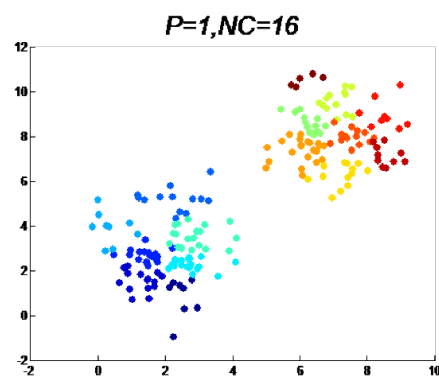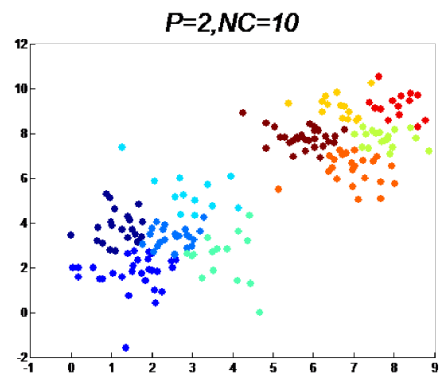
Figure 1. Relation between the parameters *P* and *NC* using synthetic data, which is a two-cluster dataset generated by two different kinds of normal distributions

Relations of Silhouette Index (*Sil*) and Number of Clusters (*NC*)

The popular and important internal evaluation index Silhouette (*Sil*) without labels is introduced to find the fuzzy relationship between number of clusters (*NC*) generated by AP and different *P* values. For data point *t*, $a(t)$ is average dissimilarity between *t* and other data

points in the same cluster, $d(t,C_j)$ is average dissimilarity of *t* from cluster C_j to other clusters $C_i (i \neq j, i = 1,2,...,k)$, $b(t) = \min\{d(t,C_j)\}$. Every sample *t* has its $Sil(t)$ using:

$$Sil(t) = \frac{b(t) - a(t)}{\max\{b(t), a(t)\}} \tag{5}$$

$$Sil = mean\left\{\sum_1^n Sil(t)\right\} \tag{6}$$

Silhouette index is calculated by (6). A higher overall average *Sil* value means the clustering method performs better. According to practical experience, $Sil > 0.5$ means visible boundary between different clusters and AP will oscillate when $Sil < 0.2$.

Sil is used to choose stable stage in Figure 1 (e). Figure 2 shows that *P*=24 is better than *P*=8 because its *Sil* is bigger. In fact, Figure 2(b) is divided into two clusters, which performs better than Figure 2(a) because the true number of clusters for dataset in Figure 2 is two.
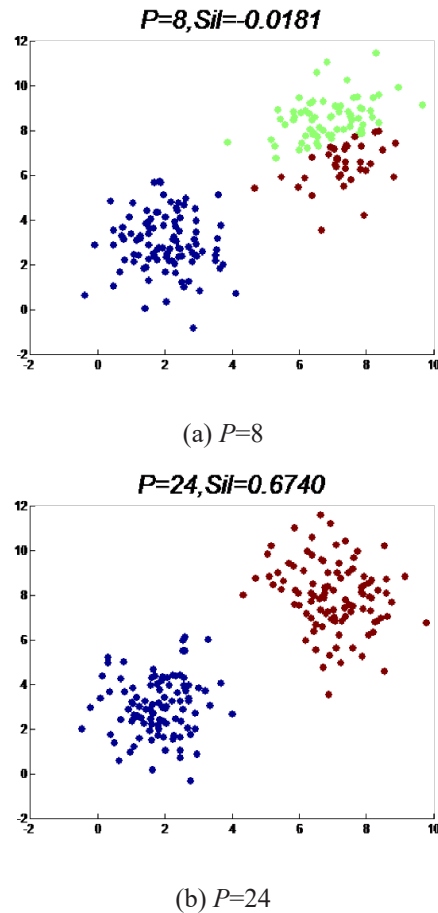

(a) *P*=8

(b) *P*=24

Figure 2. Clustering results using AP with different *P* values

Silhouette may help one find reasonable partition of data points, so that the relationship between *NC* and *Sil* can be further mined out. As it can be seen, Figures 3 (a) and (b) show the effect of input *preference* on *NC* and *Sil* using Wine and Iris, datasets between positive, continuous and steady subspace one can get optimal *NC*, subspace of Wine starts from $P_i = 13$ and Iris is $P_i = 12$. From relations of *Sil* and *NC*, the stable space of *Sil* and *NC* can be found, and better clustering results must be there, where exists true *NC*.
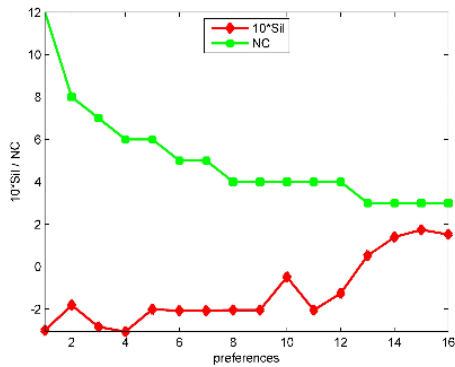
(a) The wine dataset (3-clusters dataset), $\{Sil \, / \, NC \, | \, P \in [13,16]\}$ is the stable space
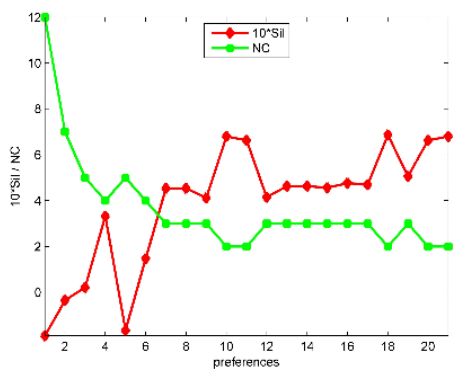
(b) The iris dataset (3-clusters dataset), $\{Sil \, / \, NC \, | \, P \in [12,17]\}$ is the stable space

Figure 3. The effect of input preference (*P*) on *NC* and 10*Sil (10 times Silhouette is set to emphasize its changing trend).

Effect of damping factor on Silhouette index

After finding the stable space between *Sil* and *NC*, one can obtain better *P* area. Next, finding optimal *P* is a key issue of the proposed algorithm.

In AP algorithm, damping factor is set to avoid numerical oscillation. λ is reduced from 0.9 to 0.5 as it is shown in Figure 4, *Sil* changes but high *Sil* is more stable and accurate. If different

P values have the same *Sil*, λ is reduced and the *P* corresponding to higher *Sil* is chosen, this method can be called "λ test". The details of whole constraint rules-based search algorithm are as follows:

Constraint Rules-Based Search Algorithm (CRS)

In the CRAP algorithm, the parameter search starts by a large *P* value, $P_{max} = 100$, then it is divided into ten isometric spaces, the *i*-th isometric space $S_i = [(i-1)P_{max}/10, iP_{max}/10]$, $i \in [1,10]$, $\lambda = 0.5$, and P_j ($j = [1,10]$) is used to represent each *P* value in S_i. For each S_i, if P_j satisfies the rule $|Sil \, | \, P_j - Sil \, | \, P_{j+1}| < 0.02$, $Sil >= 0.2$, the change between $[Sil \, | \, P_j, Sil \, | \, P_{j+1}]$ is very slight. The isometric space S_i is the stable space if this slight change occurs more than twice. After finding the stable space S_i in *P-Sil* plot, one can obtain better *P* area. Next, finding optimal *P* value in this *P* area is a key issue.

Generally speaking, the best *P* value will generate the best *Sil*. In the case of one-to-one mapping (each *Sil* has one corresponding different *P* value), the *P* value is obtained when AP gets the largest *Sil* in the stable space. However, AP may obtain same *Sil* with different *P* values (many-to-one mapping), so damping factor (another parameter in AP algorithm) is used to decide which *P* value is appropriate. When λ is reduced from 0.9 to 0.5 as it is shown in Figure 4, *Sil* also changes. When $\lambda = 0.9$, their results are identical, one adjusts $\lambda_{step} = -0.1$, $S_{il}(p = 25, \lambda = 0.8) > S_{il}(p = 24, \lambda = 0.8)$, $P = 25$ is the best value for Wine dataset. Based on this process, a more stable and accurate *P* value will be chosen. This process is called λ test.
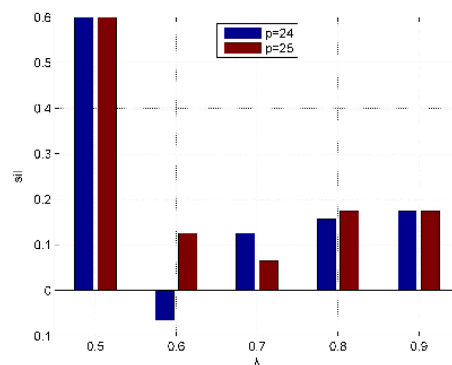

Figure 4. The process of λ test using Wine dataset

In brief, two kinds of strategies under different conditions are proposed to find the optimal P value after obtaining the stable space: (1) if the relationship between P and Sil is one-to-one mapping, the P value is obtained when the corresponding AP gets the largest Sil in the stable space. (2) if the relationship between P and Sil is many-to-one mapping, the λ test is used to find the better P value. The pseudo-codes of constraint rules-based search (CRS) algorithm are listed in Table 2.

Table 2. The pseudo-code of constraint rules-based search (CRS) algorithm

Initialization
$\lambda = 0.5$, $P_{\max} = 100$, $S_i = [(i-1)P_{max}/10, iP_{max}/10]$, $i \in [1,10]$, num=0.
Finding stable space
Running the AP algorithm from S_1 to S_{10}
if $\left\| Sil\|P_j - Sil\|P_{j+1} \right\| < 0.02$, $Sil >= 0.2$
num=num+1
if num >2
the corresponding S_i is a stable space.
end
end
Find the best P value in the stable space S_i
if the relationship between P and Sil is one-to-one mapping
the P value is obtained when the corresponding AP gets the largest Sil in the stable space.
else
the relationship between P and Sil is many-to-one mapping
the λ test is used to find the better P value.
end

4. Matching Micro-clusters Hierarchical Clustering Algorithm (MMHC)

Clustering complex structure data is an important research field (Rodriguez & Laio, 2014; Wang et al., 2020). The AP is unable to identify clusters with complex structure. The aim of this idea is to utilize advantages of AP to solve this problem.

Complex structure data has many topological structures, like manifold or non-manifold structure, uniform density or multiple density, overlapping and so on, MMHC mainly deals with structure of obvious boundaries between different clusters.

MMHC starts by an undirected graph $G = (V, E)$ like that in Figure 5 based on the micro-clusters generated by AP, each cluster C_i has one path linking all the data points, V denotes vertices (data points) and E denotes edge (similarity between two data points).
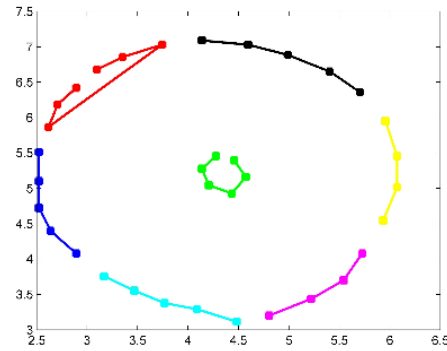

Figure 5. Undirected graph

MMHC has two important variables: inter-class density (ICD) is defined as Equation (7), which describes compactness of a cluster.

$$ICD = \min\{E \in C_i\} \tag{7}$$

Intra-class topological distance ($ICTD$) is defined as Equation (8), which describes the gap between different clusters in the complex structure data:

$$\begin{cases} E_{\max} \in \max\{E \in C_i\} \\ ICTD = \min\{S((V \in E_{\max}), l)\} - ICD \quad l \in \{C_{i+1}\} \end{cases} \tag{8}$$

where the $S((V \in E_{\max}), l)$ means Euclidean distance between data points $V \in E_{\max}$ and $l \in \{C_{i+1}\}$, $S((V \in E_{\max}), l) = \|V - l\|$.

When the $ICTD$ value is negative, the two clusters C_i and C_{i+1} are merged. Then the next micro-cluster becomes new C_i and repeats the above merging process until all the micro-clusters are merged. The pseudo-codes of Micro-clusters Merging-based Clustering (MMHC) algorithm are listed in Table 3.

Table 3. Procedure of Micro-clusters Merging-based Clustering (MMHC) algorithm

---
Generating micro-clusters

P is set to median of similarities between data points, run affinity propagation to generate micro-clusters, $t=1$, $R_1 = \{C_1\}$.

Merging process

For $i=1$: number of micro-clusters

$\quad ICD \leftarrow \min\{E \in C_i\}$

$\quad E_{max} \leftarrow \max\{E \in C_i\}$

$\quad ICTD \leftarrow \min\{S((V \in E_{max}), l)\} - ICD, l \in \{C_{i+1}\}$

$\quad$ **if** $ICTD < 0$

$\quad\quad R_t = \{R_t, C_{i+1}\}$

$\quad\quad$ continue

$\quad$ **else**

$\quad\quad t=t+1, \; R_t = []$

$\quad\quad$ continue

$\quad$ **end**

end

Obtain final clustering results R.

---

5. Experimental Results and Analysis

This section compares the performance of CRAP, MMHC with AP algorithm. To validate the effectiveness of the proposed methods, one uses same datasets, same machine to make experiments. All the experiments use $\lambda = 0.5$ (but λ changes in λ test) and same hardware devices (Pentium G645 2.9 GHz CPU, 4GB memory), CRAP uses $P_{max} = 100$.

Table 4. Features of datasets.

Data Sets	Number of samples	Dimensions	Source
Wine	178	13	UCI
Iris	150	4	UCI
Seeds	210	7	UCI
Harberman	306	3	UCI
Jain	373	2	(Rodriguez & Laio, 2014)
Spiral	312	2	(Rodriguez & Laio, 2014)
2circle	35	2	(Wang et al, 2020)

The clustering results of CRAP and AP are listed in Table 5. For all the datasets, CRAP performs better than AP in term of most evaluation metrics.

Table 5. Clustering results of CRAP and AP

Data sets	CRAP sil	CRAP NC	AP sil	AP NC	Known NC
Wine	0.5996	3	-0.3544	12	3
Iris	0.5057	3	-0.1894	12	3
Seeds	0.4325	3	-0.2594	17	3
Harberman	0.4099	2	-0.2668	31	2

Figure 6 shows clustering results of CRAP and AP using synthetic dataset, Sil(AP) = -0.3282, NC is 7. Sil(CRAP) = 0.3991, NC is 2. Visually, clustering this synthetic dataset into 2 clusters is more reasonable than 7 clusters. Meantime, Sil(CRAP)> Sil(AP). These results demonstrate that the constraint rules and search techniques in CRAP method are effective, resulting in better performance of CRAP in comparison with original AP.
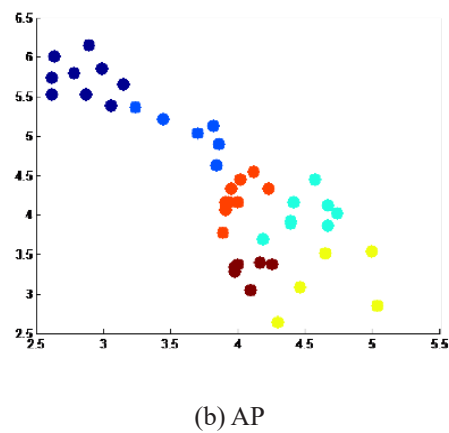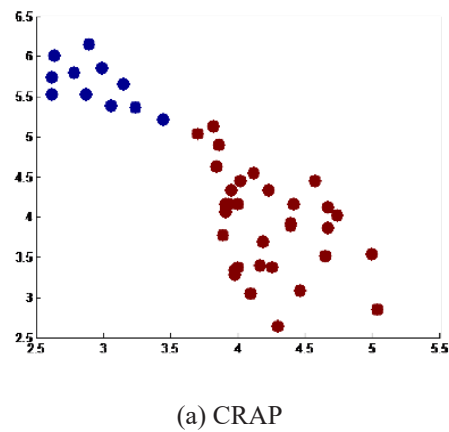

(a) CRAP

(b) AP

Figure 6. Clustering results of CRAP and AP on a synthetic dataset, which has two-clusters in ground-truth

Experiments are also carried out to verify the effectiveness of the MMHC algorithm by contained, occlusive and other complex structure of obvious boundaries. As it is shown in Figure (7), (8) and (9), compared with AP and CRAP, the MMHC algorithm can clearly identify the clusters with complex structure, while AP and CRAP cannot. AP and CRAP are based on Euclidean similarities, which cannot accurately identify complex structure.

The proposed methods have their own strengths and weaknesses, if data has no stable space in *Sil/NC-P* plot, CRAP cannot obtain optimal *NC*. This paper also proposes MMHC algorithm to identify

clusters with complex structure by merging micro-clusters generated by AP.

6. Application in Chinese Stock Quality Assessment

The method proposed in this paper can also be applied to Chinese stock quality assessment. Financial indicators of 98 listed companies are taken as model input, financial indicators include earnings per share (EPS), capital reserve per share (CRPS), undistributed profit per share (UPPS), return on equity (ROE) and net assets per share (NAPS). 98 listed companies are merged into two classes by MMHC algorithm, which include 31 and
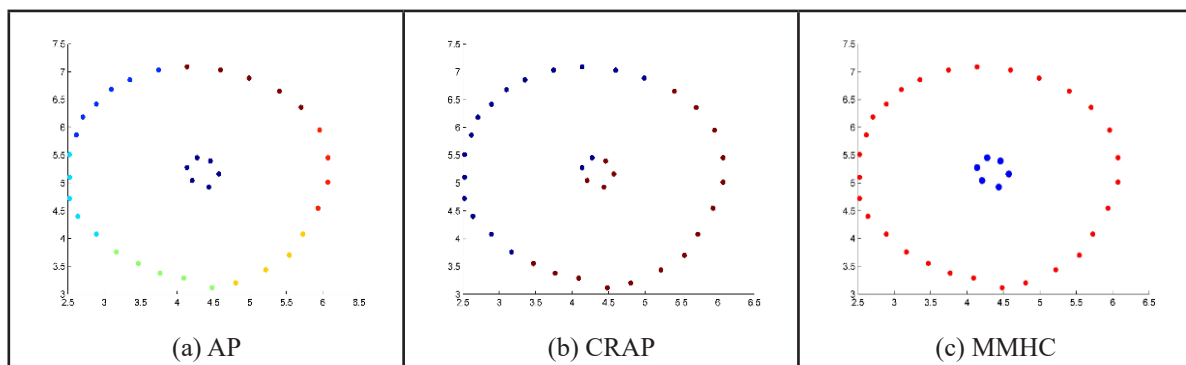

(a) AP   (b) CRAP   (c) MMHC

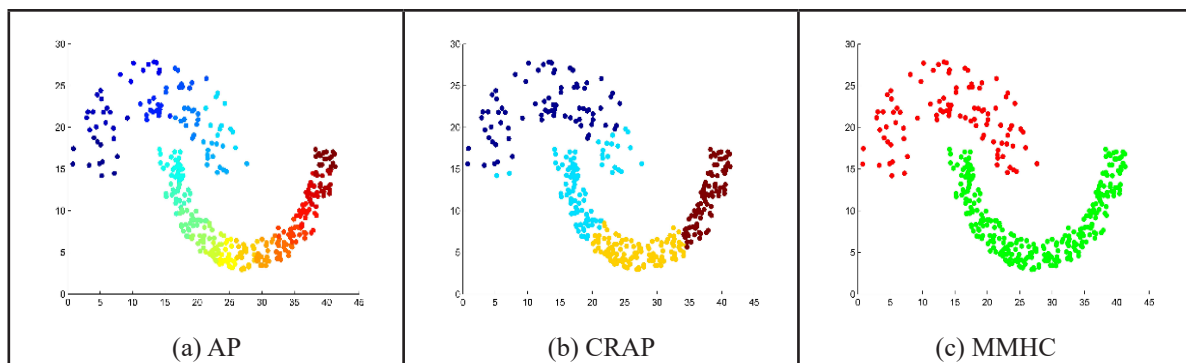**Figure 7.** Clustering results of AP, CRAP and MMHC using 2circle dataset

(a) AP   (b) CRAP   (c) MMHC

Figure 8. Clustering results of AP, CRAP and MMHC using Jain dataset

(a) AP   (b) CRAP   (c) MMHC
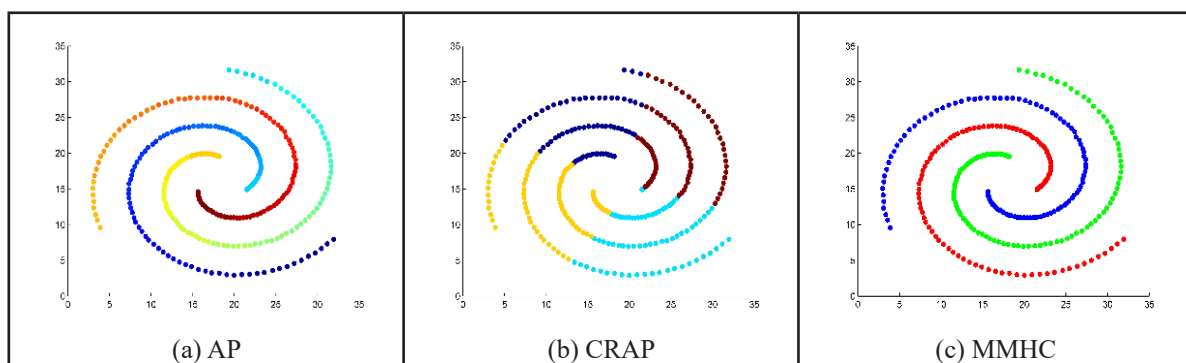
Figure 9. Clustering results of AP, CRAP and MMHC using Spiral dataset

67 listed companies respectively. The statistical data related to the financial indicators of two types of listed companies is shown in the Table 6.

Table 6. Statistical data of financial indicators of two types of listed companies obtained by MMHC

Results	EPS*	CRPS*	UPPS*	ROE*	NAPS*
Class 1 (31)	0.4864	0.816	2.247	12.863	4.45
Class 2(67)	0.0742	0.879	1.084	0.084	3.161

* means mean value

The average earnings per share of the class 1 is 6.55 times that of the second category, and the average return on net assets of the first category is 7.62 times that of the second category. There are obvious differences between these two important financial indicators, which shows that the MMHC algorithm has a significant recognition of the unknown real class data, strong practicability, and can effectively distinguish the status of enterprises in the industry.

Table 7. Statistical data of financial indicators of two types of listed companies obtained by AP

Results	EPS*	CRPS*	UPPS*	ROE*	NAPS*
Class 1 (28)	0.4953	0.851	2.357	12.936	4.86
Class 2(70)	0.0883	0.8623	1.090	0.6024	3.052

* means mean value

98 listed companies have no labels, the popular clustering evaluation indexes are useless. Thus, Mean Square Error (MSE) is used to evaluate the quality of two kinds of methods. The larger the MSE is, the better the method is. A larger MSE means the clustering results are better. Obviously, MSE of AP is 31.43 and MSE of MMHC is 33.298, which reveals that the proposed method is better than original AP.

7. Conclusion

This paper presents new approaches to obtain the optimal *P* and achieve complex structure clustering. A reasonable number of real-world and synthetic datasets are used to verify the proposed CRAP and MMHC algorithms, which perform significantly better than the original AP. The MMHC can combine with many algorithms to realize complex structure clustering. On the other hand, there are also problems with regard to CRAP and MMHC algorithms. Although one can deal with datasets of 2circle, Jain and Spiral, other structures without line between different clusters are unsuited for MMHC, further research is needed.

REFERENCES

Angelova, M., Beliakov, G. & Zhu, Y. (2019). Density-based clustering using approximate natural neighbours, *Applied Soft Computing*, *85*, 1-10. Article ID: 105867.

Darabkh, K. A., Odetallah, S. M., Al-qudah, Z. & Khalifeh, A. (2019). Energy-aware and density-based clustering and relaying protocol (EA-DB-CRP) for gathering data in wireless sensor networks, *Applied Soft Computing*, *80*, 154-166.

Dutta, A. K., Elhoseny, M., Dahiya, V. & Shankar, K. (2020). An efficient hierarchical clustering protocol for multihop Internet of vehicles communication, *Transactions on Emerging Telecommunications Technologies*, *31*(5): e3690. DOI: 10.1002/ett.3690

Fan, Z., Jiang, J., Weng, S., He, Z. & Liu, Z. (2019). Adaptive density distribution inspired affinity propagation clustering, *Neural Computing and Applications*, *31*(1), 435-445.

Fard, M. M., Thonet, T. & Gaussier, E. (2020). Deep k-means: Jointly clustering with k-means and learning representations, *Pattern Recognition Letters*, 1-5. DOI: 10.1016/j.patrec.2020.07.028

Feldman, D., Schmidt, M. & Sohler, C. (2020). Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering, *SIAM Journal on Computing*, *49*(3), 601-657.

Frey, B. J. & Dueck, D. (2007). Clustering by passing messages between data points, *Science*, *315*(5814), 972-976.

Gao, Q., Wang, Y., Cheng, X., Yu, J., Chen, X. & Tao, J. (2019). Identification of Vulnerable Lines in Smart Grid Systems Based on Affinity Propagation Clustering, *IEEE Internet of Things Journal*, *6*(3), 5163-5171.

Givoni, I., Chung, C. & Frey, B. J. (2012). Hierarchical Affinity Propagation, *Psi.toronto.edu*, 1-5.

Hassanabadi, B., Shea, C., Zhang, L., Valaee, S. (2014). Clustering in vehicular ad hoc networks using affinity propagation, *Ad Hoc Networks*, *13*(B), 535-548.

Li, P., Ji, H., Wang, B., Huang, Z. & Li, H. (2017). Adjustable preference affinity propagation clustering, *Pattern Recognition Letters*, *85*, 72-78.

Li, X., Zhang, L., Wang, L. & Wan, X. (2017). Effects of BOW model with affinity propagation and spatial pyramid matching on polarimetric SAR image classification, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *10*(7), 3314-3322.

Lu, Z. & Carreira-Perpinan, M. A. (2008). Constrained spectral clustering through affinity propagation. In *2008 IEEE* Computer *Society Conference on Computer Vision and Pattern Recognition* (*CVPR 2008*), (pp. 1-8).

Qasim, I., Jeong, J.-W., Heu, J.-U. & Lee, D.-H. (2013). Concept map construction from text documents using affinity propagation, *Journal of Information Science*, *39*(6), 719-736.

Rodriguez, A. & Laio, A. (2014). Clustering by fast search and find of density peaks, *Science*, *344*(6191), 1492-1496. DOI: 10.1126/science.1242072

Samaria, F. S. & Harter, A C. (1994). Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision* (pp. 138-142).

Sun, L., Guo, C., Liu, C. & Xiong, H. (2017). Fast affinity propagation clustering based on incomplete similarity matrix, *Knowledge and Information Systems*, *51*(3), 941-963.

Sun, Y., Xie, Z. & Gao, J. (2013). A Novel Kernel Classification Method via Image Novelty Detection, *Acta Optica Sinica*, *33*(10), 1256-1272.

Wang, Y., Wang, D., Pang, W., Miao, C., Tan, A.-H. & Zhou, Y. (2020). A Systematic Density-based Clustering Method Using Anchor Points, *Neurocomputing*, *400*, 352-370.

Wang, Z. J., Zhan, Z.-H., Lin, Y., Yu, W.-J., Yuan, H., Gu, T., Kwong, S. & Zhang, J. (2017). Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems, *IEEE Transactions on Evolutionary Computation*, *22*(6), 894-908.

Wei, Z., Wang, Y., He, S. & Bao, J. (2017). A novel intelligent method for bearing fault diagnosis based on affinity propagation clustering and adaptive feature selection, *Knowledge-Based Systems*, *116*, 1-12.

Zhang, J., Chen, F.-C., Shao-Mei, L. & Liu, L.-X. (2014). Data Stream Clustering Algorithm Based on Density and Affinity Propagation Technique, *Acta Automatica Sinica*, *40*(2), 277-288.

Zhang, X., Furtlehner, C., Germain-Renaud, C & Sebag, M. (2014). Data stream clustering with affinity propagation, *IEEE Transactions on Knowledge and Data Engineering*, *26*(7), 1644-1656.

Zhang, L. M., Jia, Y., Zhou, B. & Zhao, J.-H. (2013). Online bursty events detection based on emoticons, *Chinese Journal of Computers*, *36*(8), 1659-1667.

Zhang, Z., Wang, B.-Q. & Li, X.-T. (2013). Semi-supervised Traffic Identification Based on Affinity Propagation, *Acta Automatica Sinica*, *39*(7), 1100-1109.

Zhang, W., Wu, X., Zhu, W.-P. & Yu, L. (2017). Unsupervised image clustering with SIFT-based soft-matching affinity propagation, *IEEE Signal Processing Letters*, *24*(4), 461-464.