DE GRUYTER OPEN

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **16**, No 6
Special issue with selection of extended papers from 6th International Conference on Logistic, Informatics and Service Science LISS'2016

Sofia • 2016

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2016-0075

Improved Bidirectional CABOSFV Based on Multi-Adjustment Clustering and Simulated Annealing

Minghan Yang[1], *Xuedong Gao*[1], *Ling Li*[2]

[1]*Donlinks School of Economics and Management, University of Science and Technology Beijing (USTB), Beijing, China*
[2]*School of Business, Anhui University, Hefei, China*
Emails: hankmyang@icloud.com

Abstract: *Although Clustering Algorithm Based on Sparse Feature Vector* (*CABOSFV*) *and its related algorithms are efficient for high dimensional sparse data clustering, there exist several imperfections. Such imperfections as subjective parameter designation and order sensibility of clustering process would eventually aggravate the time complexity and quality of the algorithm. This paper proposes a parameter adjustment method of Bidirectional CABOSFV for optimization purpose. By optimizing Parameter Vector* (*PV*) *and Parameter Selection Vector* (*PSV*) *with the objective function of clustering validity, an improved Bidirectional CABOSFV algorithm using simulated annealing is proposed, which circumvents the requirement of initial parameter determination. The experiments on UCI data sets show that the proposed algorithm, which can perform multi-adjustment clustering, has a higher accurateness than single adjustment clustering, along with a decreased time complexity through iterations.*

Keywords: *Data mining, high dimensional sparse data, simulated annealing, clustering validity.*

1. Introduction

Increasing significance has been attached to data mining technologies [1]. With its development, the object data are becoming large-scaled and high dimensional [2]. In these analyses, the clustering algorithms designed for lower dimensional data can no longer meet the requirements, whereas the classic Clustering Algorithm Based On Sparse Feature Vector (CABOSFV) [3] is an efficient algorithm for high dimensional data clustering. Classic CABOSFV uses Sparse Feature Dissimilarity (SFD) to describe the dissimilarity between sets; it uses Sparse Feature Vector

(SFV) to extract features of the set, to reduce data scale, and then to implement clustering by addition of SFV. Classic CABOSFV is insensitive to noise, it is available to cluster both sparse and dense high dimensional data, and has helped solving a series of high dimensional data clustering problems [4-9].

1.1. CABOSFV clustering algorithms

However, there exist several defects of existing CABOSFV related algorithms:

Subjective parameter specifying. SFD threshold *b* is a crucial parameter of CABOSFV clustering. An overestimated *b* increases the risk of objects being assigned to wrong clusters. Conversely, underestimating *b* increases the risk of objects being rejected by the suitable cluster. The only existing method is to designate this parameter subjectively. Song and Xiao [10] proposed a method to determine the cap of *b*; Zhu, Tu, Gao et al. [11] proposed an advanced algorithm based on self-adaptive threshold. Still, the optimal *b* changes with the clustering task and data set, which makes it difficult to be determined objectively in advance. Therefore, a parameter adjustment method of CABOSFV is necessary to perform multiple clustering and optimize the parameter according to the clustering results through iterations.

Complexity of unidirectional CABOSFV clustering through iterations. Classic CABOSFV is an agglomerative clustering algorithm, its process of clustering is unidirectional, that once an object has been assigned to a cluster, it can no longer be reassign to more suitable ones. Restricted by the unidirectionality, each adjustment needs to start over and cannot make use of the previous results, which considerably increases the computational complexity and limits the feasibility of optimization through iterations. Gao, Yang and Li [12] proposed Bidirectional CABOSFV by defining Bidirectional Sparse Feature Vector (B-SFV) and addition-subtraction of B-SFVs, which improved the performance of clustering through multiple adjustments, but gave no method of parameter optimization.

Limitation on clustering quality of single adjustment CABOSFV. The CABOSFV algorithms are sensitive to the clustering order, which is affected by both data input order and clustering pattern. On this issue, Zhu, Gao, Wu and others (see [13-16]) proposed several data pre-processing methods based on object sorting, which can reduce the effects of input order sensibility to some extent. However, none can eliminate the effects of input order, and the effects of clustering pattern have not been addressed. Bidirectional CABOSFV has the ability of performing both decomposing and agglomerative clustering in multiple adjustments; it allows separation and re-aggregation to form the previous results, which can further reduce the influence of the clustering order on the quality of clustering. However, this advantage cannot be presented in single adjustment clustering, in which both decomposing and agglomerative clustering are unidirectional, the deviation affected by clustering order will be accumulated in the clustering process, reduces the quality and stability of clustering. Therefore, to approach the optimal solution of times and parameters of the adjustments is a combinatorial optimization problem.

1.2. Simulated annealing

The Simulated Annealing (SA) approach for optimization problems was proposed by Kirkpatrick, Gelatt and Vecchi [17], and has been widely applied in a variety of optimization problems due to the simple implementation and convergence properties [18], and proved efficient in various fields [19-22].

As pointed out by Peng and Cui [23], Simulated Annealing is known for being a slow method when compared to more recent strategies. However, the solution quality is generally better

Given the high complexity of classic CABOSFV iterations as mentioned above, this paper proposed a method to adjust the parameter of Bidirectional CABOSFV. Based on that, we use simulated annealing and clustering validity indexes to optimize the number and parameters of adjustments, circumvents the requirement of initial parameter determination, thereby improves the efficient of clustering.

All clustering data in this paper is binary, as Wu and Wei [24] have proposed a method to transform categorical variables to binary variables.

2. Bidirectional CABOSFV

2.1. Bidirectional sparse feature vector

Definition 1. Sparse Feature Dissimilarity, SFD. Given n objects, X is a set of the objects; the number of objects contained is $|X|$; a denotes the number of attributes that values 1 for all the objects in X; e denotes the number of attributes that values differently for all the objects in X. Define Sparse Feature Dissimilarity of X as

$$\mathrm{SFD}(X) = \frac{e}{|X| \times a}. \tag{1}$$

Definition 2. Attribute Counting Vector, ACV. Given n objects, each object is described by attributes $A_1, A_2, \ldots, A_m$; X is a set of objects, objects contained are $x_1, x_2, \ldots, x_{|X|}$; $J_{ij}(X)$ denotes the value of attribute A_i for object x_j; $C_1(X)$, $C_2(X)$, ..., $C_m(X)$ denote the times of each attribute valuing 1 for all objects in X, which is given by

$$C_i(X) = \sum_{j=1}^{|X|} J_{ij}(X), \quad i \in \{1, 2, \ldots, m\}. \tag{2}$$

Define ACV of X as vector

$$T(X) = \left(C_1(X), C_2(X), \ldots, C_m(X)\right). \tag{3}$$

Definition 3. Bidirectional Sparse Feature Vector, BSFV. Given n objects, X is a set of the objects, the number of objects contained is $|X|$; $T(X)$ is the ACV of X; S denotes the set of attributes that values 1 for all the objects in X; NS denotes the set of attributes that values differently for all the objects in X; SFD(X) is the Define SFD of X. Define BSFV of X as

$$\mathrm{BSFV}(X) = \left(|X|, T(X), S(X), \mathrm{NS}(X), \mathrm{SFD}(X)\right). \tag{4}$$

2.2. Addition of BSFV

Definition 4. Addition of BSFVs. Given n objects, each object is described by attributes $A_1, A_2,\ldots, A_m$; X and Y are two sets of objects that have no intersection, the SFVs are

$$\mathrm{SFV}(X)=\left(|X|, T(X), S(X), \mathrm{NS}(X), \mathrm{SFD}(X)\right),$$
$$\mathrm{SFV}(Y)=\left(|Y|, T(Y), S(Y), \mathrm{NS}(Y), \mathrm{SFD}(Y)\right).$$

Define addition of BSFVs as

$$\mathrm{SFV}(Y)+\mathrm{SFV}(X)=(N, T, S, \mathrm{NS}, \mathrm{SFD}), \tag{5}$$

where $N=|X|+|Y|$; $T=T(X)+T(Y)$; $S=\{A_i,\ i\in i|C_i=|N|\}$; $\mathrm{NS}=\{A_i,\ i\in i|0<C_i<|N|\}$; $\mathrm{SFD}=|\mathrm{NS}|/(N\times |S|)$.

Theorem 1. BSFV Additivity Theorem. Given n objects, X and Y are two sets of objects that have no intersection, and:

$$\mathrm{SFV}(X)=\left(|X|, T(X), S(X), \mathrm{NS}(X), \mathrm{SFD}(X)\right),$$
$$\mathrm{SFV}(Y)=\left(|Y|, T(Y), S(Y), \mathrm{NS}(Y), \mathrm{SFD}(Y)\right),$$
$$\mathrm{SFV}(X\cup Y)=(|X\cup Y|, T(X\cup Y), S(X\cup Y), \mathrm{NS}(X\cup Y), \mathrm{SFD}(X\cup Y)),$$
$$\mathrm{SFV}(X)+\mathrm{SFV}(Y)=(N, T, S, \mathrm{NS}, \mathrm{SFD}).$$

Then

$$\mathrm{SFV}(X\cup Y)=\mathrm{SFV}(X)+\mathrm{SFV}(Y). \tag{6}$$

Proof:

Since X and $Y$are two sets of objects that have no intersection, numbers of objects are $|Y|$ and $|X|$, then the union set $X\cup Y$ contains $|X|+|Y|$ objects, so

$$N=|X\cup Y|=|X|+|Y|.$$

Let $X=\{x_1, x_2, ..., x_{|X|}\}$, $Y=\{y_1, y_2, ..., y_{|Y|}\}$, each object is described by attributes $A_1, A_2,\ldots, A_m$. $J_{ij}(X)$ denotes the value of attribute A_i for object x_j, $J_{ij}(Y)$ denotes the value of attribute A_i for object y_j. $C_1(X)$, $C_2(X)$, ..., $C_m(X)$ denote the times of each attribute valuing 1 for all objects in X, $C_1(Y)$, $C_2(Y)$, ..., $C_m(Y)$ denote the times of each attribute valuing 1 for all objects in Y. By Definition 2 (ACV):

$$T(X)+T(Y)=(\sum_{j=1}^{|X|}J_{1j}(X)+\sum_{j=1}^{|Y|}J_{1j}(Y), \sum_{j=1}^{|X|}J_{2j}(X)+\sum_{j=1}^{|Y|}J_{2j}(Y),$$
$$..., \sum_{j=1}^{|X|}J_{mj}(X)+\sum_{j=1}^{|Y|}J_{mj}(Y)).$$

Since X and Y have no intersection, then

$$C_i(\mathrm{X}\bigcup \mathrm{Y})=\sum_{j=1}^{|X|}J_{ij}(X)+\sum_{j=1}^{|Y|}J_{ij}(Y),\quad i\in\{1, 2,..., m\}.$$

So $T(X)+T(Y)=(C_1(X\cup Y), C_2(X\cup Y),\ \ldots, C_m(X\cup Y))=T$.

Using Reduction to Absurdity, assume $\exists A_{i^*}\in S$ is subject to $C_{i^*}(N)\neq|N|$. By Definition 3 (BSFV):

$$J_{i*1}(X) = J_{i*2}(X) = ... = J_{i*|N|}(X) = 1.$$

By Definition 2 (ACV)

$$C_{i*}(\mathrm{N}) = \sum_{j=1}^{|N|} J_{i*j}(N) = |N| \times 1 = |N|,$$

which is contradictory to the assumption, so:

$$S = \{A_i, i \in i \,|\, C_i = |N|\};$$

similarly,

$$\mathrm{NS} = \{A_i, i \in i \,|\, 0 < C_i < |N|\}.$$

By Definition 1 (SFD)

$$\mathrm{SFD} = |\mathrm{NS}| / (N \times |S|).$$

Q.E.D.

2.3. Subtraction of BSFV

Definition 5. Subtraction of BSFVs. Given n objects, each object is described by attributes A_1, A_2,…, A_m; X is a set of objects, Y is a proper subset of X, the SFVs are:

$$\mathrm{SFV}(X) = (|X|, T(X), S(X), \mathrm{NS}(X), \mathrm{SFD}(X)),$$
$$\mathrm{SFV}(Y) = (|Y|, T(Y), S(Y), \mathrm{NS}(Y), \mathrm{SFD}(Y)).$$

Define Subtraction of BSFVs as

$$\mathrm{SFV}(Y) - \mathrm{SFV}(X) = (N,\ T,\ S,\ \mathrm{NS},\ \mathrm{SFD}), \tag{7}$$

where $N=|X| - |Y|$; $T=T(X) - T(Y)$; $S=\{A_i,\ i \in i\ |C_i=|N|\}$; $\mathrm{NS}=\{A_i,\ i \in i | 0<C_i<|N|\}$; $\mathrm{SFD}=|\mathrm{NS}|/(N \times |S|)$.

Theorem 2. BSFV Subtractivity Theorem. Given n objects, X is a set of objects, Y is a proper subset of X, and:

$$\mathrm{SFV}(X) = (|X|, T(X), S(X), \mathrm{NS}(X), \mathrm{SFD}(X)),$$
$$\mathrm{SFV}(Y) = (|Y|, T(Y), S(Y), \mathrm{NS}(Y), \mathrm{SFD}(Y)),$$
$$\mathrm{SFV}(X-Y) = (|X-Y|, T(X-Y), S(X-Y), \mathrm{NS}(X-Y), \mathrm{SFD}(X-Y)),$$
$$\mathrm{SFV}(X) - \mathrm{SFV}(Y) = (N, T, S, \mathrm{NS}, \mathrm{SFD}).$$

Then

$$\mathrm{SFV}(X-Y) = \mathrm{SFV}(X) - \mathrm{SFV}(Y). \tag{8}$$

Proof:

Since Y is a proper subset of X, numbers of objects are $|Y|$ and $|X|$, then the difference set $X-Y$ contains $|X|-|Y|$ objects, so $N=|X-Y|=|X|-|Y|$.

Let $X=\{x_1, x_2, ..., x_{|X|}\}$, $Y=\{y_1, y_2, ..., y_{|Y|}\}$, each object is described by attributes A_1, A_2,..., A_m; $J_{ij}(X)$ denotes the value of attribute A_i for object x_j; $J_{ij}(Y)$ denotes the value of attribute A_i for object y_j; $C_1(X)$, $C_2(X)$, ..., $C_m(X)$ denote the times of each attribute valuing 1 for all objects in X; $C_1(Y)$, $C_2(Y)$, ..., $C_m(Y)$ denote the times of each attribute valuing 1 for all objects in Y. By Definition 2 (ACV):

$$T(X)-T(Y)=(\sum_{j=1}^{|X|}J_{1j}(X)-\sum_{j=1}^{|Y|}J_{1j}(Y),$$

$$\sum_{j=1}^{|X|}J_{2j}(X)-\sum_{j=1}^{|Y|}J_{2j}(Y),...,\sum_{j=1}^{|X|}J_{mj}(X)-\sum_{j=1}^{|Y|}J_{mj}(Y)).$$

Since Y is a proper subset of X, then

$$C_i(\mathrm{X}-\mathrm{Y})=\sum_{j=1}^{|X|}J_{ij}(X)-\sum_{j=1}^{|Y|}J_{ij}(Y),\quad i\in\{1,2,...,m\}.$$

So $T(X)-T(Y)=\left(C_1(X-Y),C_2(X-Y),\ \ldots,C_m(X-Y)\right)=T.$

The rest is the same as in the proof of Theorem 1 (BSFV Additivity Theorem). Q.E.D.

2.4. Parameter adjustment of bidirectional CABOSFV

SFD threshold b is the predetermined parameter of CABOSFV clustering. In order to take advantage of the reversibility of Bidirectional CABOSFV, the clustering process with parameter adjustment is proposed, which will further reduce the influence of clustering order on clustering quality during the separation and re-aggregation of objects and clusters.

Definition 6. Adjustment of SFD threshold *b*. Given n result clusters and the parameter SFD threshold b form previous clustering, b' is the new parameter, $b'\neq b$. Taking the n result clusters as initial sets to perform the clustering with the parameter b' is defined as an adjustment of SFD threshold b.

2.5. Steps and example of B-CABOSFV clustering

Classic CABOSFV clustering needs to start over for each adjustment, whereas B-CABOSFV makes use of the previous results, since it is a bidirectional clustering algorithm.

2.5.1. Three-layered structure

The procedures of B-CABOSFV clustering can be described with a three-layered structure (Fig. 1). In the t-th adjustment, $S_{t,1}^{(0)}$, $S_{t,2}^{(0)}$, ..., $S_{t,k}^{(0)}$ are the result sets from previous adjustment with a SFD threshold of $b_{(t\text{-}1)}$ (upper layer). $S_{t,1}^{(1)}$, $S_{t,2}^{(1)}$, ..., $S_{t,k+1}^{(1)}$ are new sets-to-cluster, which are generated by subtraction of BSFV after SFD threshold decreased to b_t (mid layer). $S_{t,1}^{(2)}$, $S_{t,2}^{(2)}$, ..., $S_{t,k}^{(2)}$ are present result sets merged by applying addition of BSFV to new sets-to-cluster (lower layer).

Specifically, when the SFD threshold decreased from b_{t-1} to b_t, check the SFD of each previous result sets successively. If $\mathrm{SFD}(S_{t,1}^{(0)})>b_t$, cull off the last object (X_n) of $S_{t,1}^{(0)}$; if the SFD is still greater than b_t, continue to cull off objects (X_{n-1}, X_{n-2},…) of the set until it drop below b_t. Then regard $S'_{t,1}{}^{(0)}$ and $\{X_n\}$, $\{X_{n-1}\}$, $\{X_{n-2}\}$, … , along with others previous result sets as new sets-to-cluster.

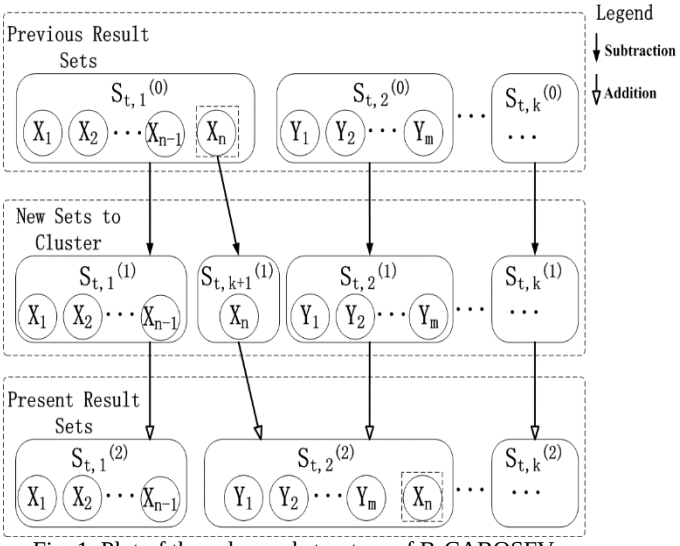


Fig. 1. Plot of three-layered structure of B-CABOSFV

2.5.2. Example

As shown in Table 1, X_1, X_2,…, X_6 are 6 clients, A_1, A_2,…, A_8 are the attributes of clients corresponding the orders of 8 kinds of products, values 1 if ordered and 0 if not. To cluster these clients by order status is a clustering problem of 6 objects of 8 attributes.

Table 1. Client order status

Client	Product ordered	Attribute vector							
		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
X_1	2, 4, 6, 8	0	1	0	1	0	1	0	1
X_2	1, 4, 6, 8	1	0	0	1	0	1	0	1
X_3	1, 2, 4, 6, 8	1	1	0	1	0	1	0	1
X_4	3, 5, 6, 7, 8	0	0	1	0	1	1	1	1
X_5	3, 5, 7, 8	0	0	1	0	1	0	1	1
X_6	1, 2, 4, 8	1	1	0	1	0	0	0	1

To solve this problem, the steps of first and second adjustment of B-CABOSFV clustering are as followed.

Steps of the first adjustment

Step 1. Set the initial SFD threshold b_1=1;

Step 2. Create a set-to-cluster for each client, denote as $S_{1,i}^{(0)}$, $i \in \{1, 2, ..., 6\}$;

Step 3. Calculate the SFDs. Apparently, as the first adjustment, we have

$$\mathrm{SFD}(S_{1,i}^{(1)}) = 0 < b_1 \qquad i \in \{1, 2, ..., 6\},$$

all of which are not greater than b_1, no need to subtract. Regard all the sets as new set-to-cluster, denote as $S_{t,i}^{(1)}$, $i \in \{1, 2, ..., 6\}$, then go to Step 5;

Step 4. Skipped;

Step 5. Merge sets-to-cluster and manage the SFD after merging to be no greater than SDF threshold b_1. The result sets are $S_{1,1}^{(2)}=\{X_1, X_2, X_3, X_4\}$, $S_{1,2}^{(2)}=\{X_5\}$, $S_{1,3}^{(2)}=\{X_6\}$. SFDs of the sets are SFD($S_{1,1}^{(2)}$)=0.75, SFD($S_{1,2}^{(2)}$)=0, SFD($S_{1,3}^{(2)}$)=0;

Step 6. Not satisfied with the results, need another adjustment.

Steps of the second adjustment

Step 1. Reset the SFD threshold to b_2=0.5;

Step 2. Create a set for each previous result sets as $S_{2,1}^{(0)}=\{X_1, X_2, X_3, X_4\}$, $S_{2,2}^{(0)}=\{X_5\}$, $S_{2,3}^{(0)}=\{X_6\}$;

Step 3. Since SFV($S_{2,1}^{(0)}$)=0.75>b_2, we cull off the last client in the set (X_4), denote the remaining part as $S_{2,1}^{(1)}$. Create a new set for X_4, denote as $S_{2,4}^{(1)}$;

Step 4. Calculate the SFD of set $S_{2,1}^{(1)}$:

$$\mathrm{SFD}(S_{2,1}^{(1)}) = \mathrm{SFD}(S_{2,1}^{(0)} - \{X_4\}) = \frac{|\mathrm{NS}|}{N \times |S|} = \frac{2}{3 \times 2} = 0.22 < b_2.$$

At this point, the new sets-to-cluster are $S_{2,1}^{(1)}=\{X_1, X_2, X_3\}$, $S_{2,2}^{(1)}=\{X_5\}$, $S_{2,3}^{(1)}=\{X_6\}$, $S_{2,4}^{(1)}=\{X_4\}$, SFDs are all below b_2;

Step 5. Merge new sets-to-cluster, obtain the result sets of the second adjustment (Table 2);

Step 6. Finish.

Table 2. Clustering result of the example

Clients	All ordered	Partial ordered	SFD
X_1, X_2, X_3, X_6	1, 8	2, 3, 4	0.375
X_4, X_5	5, 6, 7, 8	2	0.125

In this example, B-CABOSFV clustering made use of the results of the previous adjustment, which saved two addition operations.

3. Simulated annealing optimization

The optimization of SFD threshold b combination is crucial to CABOSFV clustering. Different from classic CABOSFV, Bidirectional CABOSFV can make use of the results of previous adjustment, which decreases the time complexity greatly and improves the feasibility of iterative optimization.

Simulated Annealing is derived from the Metropolis algorithm [25]. It has been used to solve large-scale combinatorial problems by Kirkpatrick et al. [17]. The authors created an analogy between combinatorial optimization and the annealing of solids. In this process, an atomic configuration for a solid must be found such that it minimizes internal energy. In optimization cases, a solution to the problem is compared to an atomic configuration, and the internal energy to the objective function.

The main features of the method are Temperature (T) and Temperature Length (TL). In order to achieve the best atomic configuration, the solid temperature must be slowly reduced. In the optimization case, the temperature variable determines the

chances of acceptance of a solution. The Probability of Acceptance (PoA) is a function of temperature and the Objective Function Value (OFV), and is calculated by

$$\text{PoA}(\Delta\text{OFV}, T) = e^{-\text{OFV}/T}. \quad (9)$$

ΔOFV denotes the difference between the OFV of current solution and the new solution. If a new solution is better than the current, it is automatically accepted. If it is worse, it still has a chance of acceptance. When temperature is high, these chances are also high and more uphill moves are accepted. Such strategy leads to local minima avoidance, preventing premature stagnation in non-optimal solutions. The temperature must remain the same for a given number of moves before it is reduced. That given number of moves is represented by the TL. A schedule for temperature reduction must be set. After all the allowed moves are performed in a temperature level, it is reduced according to

$$T_{k+1} = \alpha T_k, \quad (10)$$

where T is the current temperature; k is the current iteration; α is a decreasing rate parameter.

3.1. Objective function

We use two clustering validity indexes as objective function.

3.1.1. Internal clustering validity index CVISFD

CVISFD [26], which is proposed based on DB*, is used to evaluate the clustering results:

$$\text{CVISFD}(n_c) = \frac{1}{n_c}\sum_{i=1}^{n_c} \frac{\max\limits_{j, j\neq i}\left(\frac{1}{n_i}\text{SFD}_i + \frac{1}{n_j}\text{SFD}_j\right)}{\min\limits_{x\in C_i, y\notin C_i}\text{SFD}_{x,y}}, \quad (11)$$

where n_c is the number of result clusters; C_i is the i-th cluster; n_i is the number of objects of C_i; SFD_i is the sparse feature dissimilarity of C_i; $\text{SFD}_{x,y}$ is the sparse feature dissimilarity of object x and y.

A lower value of CVISFD indicates the lower dissimilarity in each clusters, and higher dissimilarity between clusters, and vice versa. Thereby reflects the quality of clustering.

As an internal criterion, CVISFD has no requirement of prior knowledge.

3.1.2. External clustering validity index Averaged Accuracy (AA)

Table 3 shows the four possible cases on the objects.

Table 3. Cases on objects

Desired categories	Result categories	
	Same	Different
Same	*a*	*b*
Different	*c*	*d*

Positive Accuracy (PA):

$$\mathrm{PA} = a/(a+c). \tag{12}$$

Negative Accuracy (NA):

$$\mathrm{NA} = d/(b+d). \tag{13}$$

Averaged Accuracy (AA):

$$\mathrm{AA} = \frac{\mathrm{PA}+\mathrm{NA}}{2} = \frac{a/(a+c)+d/(b+d)}{2}. \tag{14}$$

AA takes both positive and negative accuracy into consideration to evaluate the clustering quality with objectivity and comprehensiveness.

As an external criterion, AA can help to verifying the performance and theoretical limits of the algorithm.

3.2. Data pre-processing: Weighted sorting

Definition 7. Weighted sorting with uncorrelated sequences. Given n objects, object i is described by attributes A_{i1}, A_{i2},…, A_{im}, uncorrelated sequence $M=(M_1, M_2,\ldots, M_m)$, the uncorrelated sequence index of object i is

$$q_i = M_1 A_{i1} + M_2 A_{i2} + \ldots + M_m A_{im}, \tag{15}$$

Sorting the objects by q_i is defined as Weighted Sorting with Uncorrelated Sequences.

Pre-process the input data with this method would decrease the input order sensibility and improve the quality of CABOSFV clustering [16].

3.3. Combinatorial optimization

Definition 8. Parameter Vector, PV. Given n parameters for multiple adjustments, define $\mathrm{PV}(n) = (b_1, b_2, \ldots, b_n)$ as PV.

Strategy 1. Multi-adjustment Clustering. Given $\mathrm{PV}(n) = (b_1, b_2, \ldots, b_n)$ is the input parameter of one iteration, consecutively perform adjustment clustering (Definition 6) with SFD threshold $b_1, b_2, \ldots, b_n$, initial clusters of each adjustment are the result of previous adjustment. Thus n times of adjustments are regarded as one iteration.

Definition 9. Parameter Selection Vector, PSV. Given n parameters for multiple adjustments, n-1 parameter selection indexes $s_1, s_2, \ldots, s_{n-1}$, define $\mathrm{PSV}(n)=(s_1, s_2, \ldots, s_{n-1})$ as Parameter Selection Vector.

Strategy 2. Parameter Selection. Given $\mathrm{PV}(n)=(b_1, b_2, \ldots, b_n)$, $\mathrm{PSV}(n)=(s_1, s_2,\ldots, s_{n-1})$, select parameters by

$$\begin{cases} \text{adopt} & b_i, \quad s_i \geq 0, \\ \text{ignore} & b_i, \quad s_i < 0, \end{cases} \tag{16}$$

to produce the selected parameter vector $\mathrm{PV}'(m)=(b_1, b_2, \ldots, b_m)$, $1 \leq m \leq n$.

Use both PV and PSV as input of SA optimization. By Strategy 1 and Strategy 2, we are able to optimize the number and parameters of adjustments, thereby achieve the optimal clustering result.

3.4. Algorithm steps

The steps of the *p*-th iteration are as followed in Fig. 2.

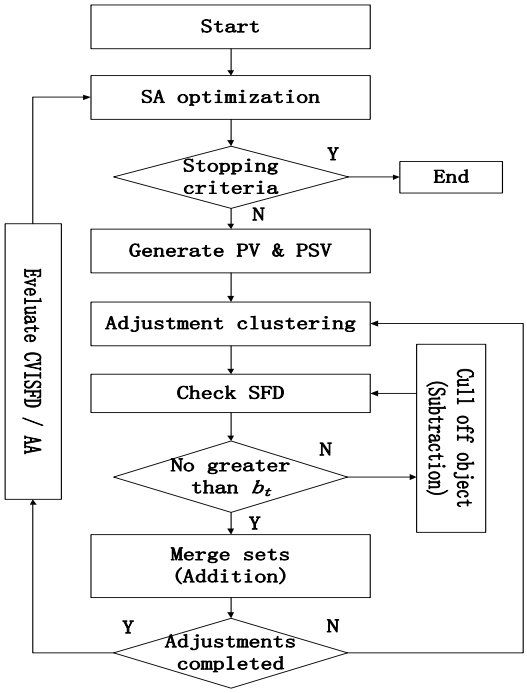


Fig. 2. Plot of steps of B-CABOSFV with SA

Step 1. Generate new PV and PSV by SA, thereby specify the number of adjustment times T, and SFD threshold b_t for each adjustment;

Step 2. Create a set for each of the n objects or sets from the previous result as the initial sets of the t-th adjustment ($1 \le t \le T$), denote as $S_{t,i}^{(0)}$, $i \in \{1, 2, ..., n\}$;

Step 3. Calculate the SFD of each set. Obviously, set contains only one object has a SFD of 0. If the SFD of all sets are no greater than b_t, add 1 to the superscripts of sets, denote as $S_{t,i}^{(1)}$, $i \in \{1, 2, ..., n\}$, regard as new sets-to-cluster and go to Step 5; if SFV($S_{t,i^*}^{(0)}$) is greater than b_t, cull off the last object in the set, denote $S'_{t,i^*}^{(0)}$ as $S_{t,i^*}^{(1)}$. Create a new set-to-cluster for the object culled off, denote as $S_{t,n+1}^{(1)}$, then go to Step 4;

Step 4. By Subtraction of BSFV, calculate

$$\mathrm{SFV}\left(S_{t,i^*}^{(1)}\right) = \mathrm{SFV}\left(S_{t,i}^{(0)} - S_{t,n+1}^{(0)}\right) = \mathrm{SFV}\left(S_{t,i}^{(0)}\right) - \mathrm{SFV}\left(S_{t,n+1}^{(0)}\right),$$

then go back to Step 3;

Step 5. Similar to classic CABOSFV clustering, by addition of BSFV, merge sets-to-cluster and manage the SFD after merging to be no greater than SDF threshold b_t, obtain the clustering result denoted as $S_{t,i}^{(2)}$, $i \in \{1, 2, ..., k\}$. If the adjustment number t reaches T, go to Step 6; else, $t \to t+1$, go back to Step 2;

Step 6. Use cluster validity index to evaluate the result, if it reaches the stopping criteria of SA, terminate the process; else, $p \rightarrow p+1$, go back to Step 1.

3.5. Time complexity

Considering addition and subtraction of BSFVs have the same complexity, the time complexity of one B-CABOSFV iteration is

$$T = O\left(m\sum_{i=1}^{t} k_i q_i \right), \tag{17}$$

where m is the number of the attributes; t is the total number of adjustments; k_i is the number of result clusters after the i-th adjustment. q_i is the number of initial clusters before the i-th adjustment, which is given by

$$q_i = \begin{cases} n, & i = 1, \\ k_{i-1} + p_i, & i > 1, \end{cases} \quad i = 1, 2, ..., t, \tag{18}$$

where n is the number of objects in data set, p_i is the number of objects culled off in the i-th adjustment.

Apparently, in classic CABOSFV clustering, $q_i=n$, $i \in \{1, 2, ..., t\}$. So the ratio of the time complexity of B-CABOSFV to classic CABOSFV is

$$T_{\text{ratio}} = \sum_{i=1}^{t} (k_{i-1} + p_i) k_i \Big/ \sum_{i=1}^{t} n k_i. \tag{19}$$

With the increasing of the times and precision of the adjustments, the total time B-CABOSFV clustering takes is far less than classic CABOSFV clustering.

4. Experiments

4.1. Experimental method

Test on 2 UCI data sets (Table 4) with the objective function of CVISFD and AA. The length of initial PV is 5.

Table 4. Date sets for experiments

Data set	#Instances	#Attributes	#Categories
Zoo	101	16	7
Small soybean	47	35	4

Since each iteration includes multiple adjustments, we use Equivalent Iteration Time (EIT) to compare the time efficiency between bidirectional and unidirectional CABOSFV:

$$\text{EIT}(n) = \sum_{i=1}^{n} (T_i / t_i), \tag{20}$$

where n is the number of iterations; T_i is the time cost of the i-th iteration; t_i is the adjustments times of the i-th iteration. Mean Equivalent Iteration Time (MEIT) is

(21) $$\mathrm{MEIT}(n)=\frac{1}{n}\sum_{i=1}^{n}(T_i/t_i).$$

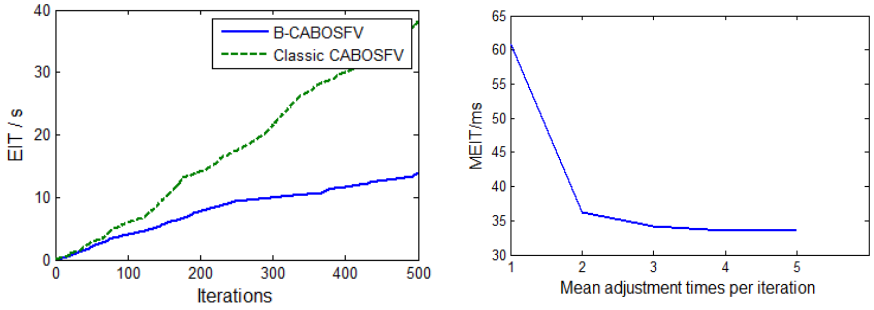
Apparently, to unidirectional CABOSFV, the value of t_i is always and only can be 1.

4.2. Results

Tables 5 and 6 and Figs 3 and 4 show the results with the objective function of AA.

Table 5. Result on data set Zoo (AA)

Input length of PV	Number of optimal adjustment times	AA	Mean adjustment times per iteration	MEIT (ms)
1	1	92.9%	1	60.9
2	2	98.9%	1.94	36.3
3	3	99.1%	2.61	34.2
4	3	99.1%	2.90	33.6
5	3	99.1%	2.95	33.6

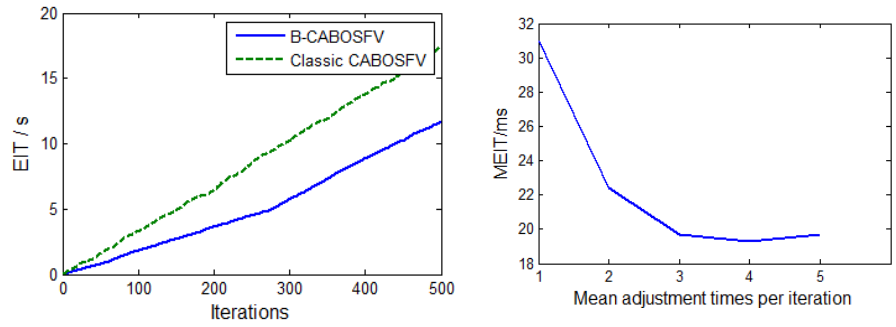


a) Equivalent iteration time b) Mean equivalent iteration time

Fig. 3. Time cost on data set Zoo

Table 6. Result on data set Soybean (AA)

Input length of PV	Number of optimal adjustment times	AA	Mean adjustment times per iteration	MEIT (ms)
1	1	91.3%	1	31.0
2	2	98.5%	1.93	22.4
3	3	100%	2.58	19.7
4	3	100%	2.84	19.3
5	3	100%	2.83	19.7



a) Equivalent iteration time b) Mean equivalent iteration time

Fig. 4. Time cost on data set Soybean

Table 7 shows the result with the objective function of CVISFD.

Table 7. Result with CVISFD

Input length of PV	Zoo			Soybean		
	Number of optimal adjustment times	CVISFD	AA	Number of optimal adjustment times	CVISFD	AA
1	1	0.062	91.9%	1	0.019	78.9%
2	2	0.042	96.9%	2	0.013	78.4%
3	3	0.035	98.5%	3	0.008	75.6%
4	4	0.031	79.7%	3	0.008	75.6%
5	5	0.030	86.1%	3	0.008	75.6%

4.3. Discussion

Discussions of the experimental results are as followed:

In SA iterations, Bidirectional CABOSFV adjustments have an obvious advantage on iterative time than classic CABOSFV, which indicates Bidirectional CABOSFV adjustment's ability of making full use of previous results can reduce considerable number of repeated clustering process. This provides a reference for further extending Bidirectional CABOSFV based on iterative optimization.

MEIT decreases with the increase of adjustment times, but the rate of change decreases gradually. The reason seems to be, that in a probabilistic sense, the length of SA selected parameter vector tends to the median, reducing the influence of higher adjustment number on MEIT. Therefore MEIT can be further reduced as the length of initial PV increases.

According to both internal and external criteria results, the lengths of optimal PVs, the optimal adjustment times, are all greater or equal to 3, validated the improvement on clustering quality of multi-adjustment clustering.

The internal criteria result of dataset Zoo is ideal, but the internal criteria result of Soybean is relatively low, consider the internal clustering validity index is still to be improved. Meanwhile, the external criteria results of both datasets are remarkably good, proved that the theoretical limit of clustering quality has been improved. Also, with proper clustering validity index, the requirement of initial parameter determination can be circumvented by the algorithm.

5. Conclusion

A method of multi-adjustment clustering is proposed on the base of parameter adjustment method and parameter selection method. To approach the optimal solution of multi-adjustment clustering, Simulated Annealing is used with the object function of clustering validity indexes. Both, time complexity analysis and experiments on UCI datasets prove that the proposed algorithm has a fine computational tractability through iterations, the clustering quality is improved, and the requirement of initial parameter determination can be circumvented. In general, the attainable clustering quality is higher by multi-adjustment clustering, which indicates that the sensibility of clustering order has been reduced through separation and re-aggregation of the objects.

In addition, how to design a more reliable internal clustering validity index remains to be studied further.

References

1. N i n g, D., H. L i, H. W a n g. Analysis and Prediction of Logistics Enterprise Competitiveness by Using a Real GA-Based Support Vector Machine. – Journal of System and Management Sciences, Vol. **3**, 2013, No 2, pp. 29-34.
2. J u, C., F. G u o. Distributed Data Mining Model Based on Support Vector Machines. – Systems Engineering-Theory & Practice, Vol. **30**, 2010, No 10, pp. 1855-1863.
3. W u, S., X. G a o. CABOSFV Algorithm for High Dimensional Sparse Data Clustering. – Journal of University Science & Technology Beijing, Vol. **11**, 2004, No 3, pp. 283-288.
4. W u, S., W. Z h a n g, H. H u a n g, Y. Y e. FD-CABOSFV High Dimensional Data Clustering for Interval-Scaled Variables. – China Journal of Information Systems, Vol. **9**, 2011, pp. 77-87.
5. W a n g, D., D. Z h u. Research of Mining Word Category Knowledge Based on CABOSFV. – Computer Science, Vol. **40**, 2013, No 9, pp. 211-215.
6. P a n, J. DS_CABOSFV Clustering Algorithm for High Dimensional Data Stream. – In: 4th International Conference on Awareness Science and Technology (ICAST'12), 2012, pp. 16-19.
7. W u, S., Y. Y e, X. Y u. Clustering for High Dimensional Data Based on Extended Set Dissimilarity. – Application Research of Computers, Vol. **28**, 2011, No 9, pp. 3253-3255.
8. W e i, G., L. Z o u, J. P a n. Improved Text Classification Algorithm for Spam Filtering Based on CABOSFV. – Future Computer & Information Technology, Vol. **86**, 2013, pp. 1131-1139.
9. Z h a n g, Q. Research and Implementation of Clustering Analysis Algorithms Based on I-MINER. – In: 2013 International Conference on Computer Sciences and Applications, ICCSA'13, 2013, pp. 254-257.
10. S o n g, Y., Q. X i a o. The Method of How to Determine Threshold Value of Set-Square-Difference in CABOSFV Algorithm. – Ship Science and Technology, Vol. **28**, 2006, No 1, pp. 99-102.
11. Z h u, Q., G. T u, X. G a o, S. W u, H. C h e n. Enhanced CABOSFV Clustering Algorithm Based on Adaptive Threshold. – In: International Conference on Computer Science and Automation Engineering, ICCSAE'11, 2011, pp. 620-622.
12. G a o, X., M. Y a n g, L. L i. Bidirectional CABOSFV for High Dimensional Sparse Data Clustering. – In: 2016 International Conference on Logistics, Informatics and Service Sciences, LISS'2016, 2016 (in Publishing).
13. Z h u, Q., X. G a o, S. W u, M. C h e n, H. C h e n. High Dimensional Sparse Data Clustering Based on Sorting Idea. – Computer Engineering, Vol. **36**, 2010, No 22, pp. 13-14.
14. W u, S., X. F e n g, Q. W u. Parallel Clustering Algorithm Based on Sparse Index Sort of High Dimensional Data. – Systems Engineering-Theory & Practice, Vol. **S2**, 2011, pp. 13-18.

15. W u, S., J. W a n g, Y. T a n. Improved CABOSFV Clustering Considering Data Sort. – Computer Engineering & Applications, Vol. **47**, 2011, No 34, pp. 127-129.
16. W u, S., Q. W a n g, M. J i a n g, Q. W e i. Clustering Algorithm of Categorical Data in Consideration of Sorting by Weight. – Journal of University of Science & Technology Beijing, Vol. **35**, 2013, No 8, pp. 1093-1098.
17. K i r k p a t r i c k, S., C. D. G e l a t t, M. P. V e c c h i. Optimization by Simulated Annealing. – Science, Vol. **220**, 1983, No 4598, pp. 671-680.
18. R o b i n i, M. C., P. J. R e i s s m a n. From Simulated Annealing to Stochastic Continuation: A New Trend in Combinatorial Optimization. – Glob. Optim., Vol. **56**, 2013, No 1, pp. 185-215.
19. G u o d o n g, Y u et al. Research on the Time Optimization Model Algorithm of Customer Collaborative Product Innovation. – Journal of Industrial Engineering & Management, Vol. **10**, 2014, No 1, pp. 4666-4672.
20. D e l g o s h a e i, A., M. K. M. A r i f f i n, B. T. H. T. B a h a r u d i n. Pre-Emptive Resource-Constrained Multimode Project Scheduling Using Genetic Algorithm: A Dynamic Forward Approach. – Journal of Industrial Engineering & Management, Vol. **9**, 2016, No 3, pp. 732-785.
21. S e y e d k a s h i, S. M. H., et al. Experimental and Numerical Investigation of an Adaptive Simulated Annealing Technique in Optimization of Warm Tube Hydroforming. – Organization Development Journal, Vol. **22**, 2004, pp. 579-583.
22. M a l m b o r g, C. J. A Simulated Annealing Algorithm for Dynamic Document Retrieval. – International Journal of Industrial Engineering, Vol. **10**, 2003, No 2, pp. 115-125.
23. P e n g, F., G. C u i. Efficient Simultaneous Synthesis for Heat Exchanger Network with Simulated Annealing Algorithm. – Appl. Therm. Eng., Vol. **78**, 2015, pp. 136-149.
24. W u, S., G. W e i. High Dimensional Data Clustering Algorithm Based on Sparse Feature Vector for Categorical Attributes. – In: International Conference on Logistics Systems and Intelligent Management, ICLSIM'10, 2010, pp. 973-976.
25. M e t r o p o l i s, N., A. W. R o s e n b l u t h, M. N. R o s e n b l u t h, A. H. T e l l e r, E. T e l l e r. Equation of State Calculations by Fast Computing Machines. – J. Chem. Phys., Vol. **21**, 1953, 1087.
26. W u, S., D. J a n g, Q. W a n g. HABOS Clustering Algorithm for Categorical Data. – Chinese Journal of Engineering, Vol. **38**, 2016, No 7, pp. 1017-1024.