# Weighted Clustering and Evolutionary Analysis of Hybrid Attributes Data Streams

Chen Xinquan
Shangrao Normal University/Department of Mathematics and Computer, Shangrao, China
Email: chenxqscut@126.com

***Abstract*—It presents some definitions of projected cluster and projected cluster group on hybrid attributes after having given some definitions on ordered attributes and sorted attributes to solve clustering analysis problem of infinite hybrid attributes data streams in finite space. In order to improve the clustering quality of hybrid attributes data streams, it presents a two-step projected clustering method, which can often make better clustering effects in two simulated experiments although it is very simple. At last, it gives a dividing and merging framework of infinite hybrid attributes data streams. In order to implement this framework, it presents 8 properties in Section IV, some data structure definitions and 15 algorithms in appendix. The framework is verified and these algorithms are tested by German data set with a better clustering quality than WKMeans sometimes if having set right parameters.**

***Index Terms*—ordered attributes, sorted attributes, hybrid attributes, projected clustering, merging clusters, subtracting of clusters, merging cluster groups, evolutionary analysis of cluster groups**

## I. INTRODUCTION

Clustering is to partition a data set into several disjoint groups so that data points in the same group are near to each other according to some distance metric. Clustering data streams has become an important research direction in data mining since 2000. It is a more challenging problem for high dimensionality, clustering speed, clustering precision, clustering meaning, data sparsity, noise or outliers, and so on. Weighted clustering and evolutionary analysis of hybrid attributes data streams should be an important research direction because of the large number of practical applications.

Joshua Zhexue Huang et al. [1] proposed WKMeans algorithm which was implemented as a component in AlphaMiner [2] after combining k-means clustering algorithm with feature-weighting. Wang X.Z. et al. [3] gave an improved FCM algorithm after a feature-weighting research in FCM clustering algorithm.

For traditional static data sets, there are already many research papers about projected clustering, subspace clustering, and so on. Aggarwal C. et al. [4] proposed a projected data stream clustering method called HPStream. Gabriela Moise et al. [5] presented a projected clustering method, P3C, which can deal with both numerical and categorical data. Aggarwal C. et al. [6] proposed CluStream framework, which contains online micro-cluster maintenance and macro-cluster creation. CluStream can only handle numerical data streams, so it is a valuable research direction to find some framework and methods which can handle both numerical and categorical data streams. In order to solve this difficult problem it gives a dividing and merging framework of infinite hybrid attributes data streams in this paper.

This paper is organized as follows. Section II gives some definitions of hybrid attributes data streams. In Section III, it presents a two-step projected clustering method of hybrid attributes data streams. Section IV presents a dividing and merging framework of hybrid attributes data streams. Section V gives several simulated experiments for the two-step projected clustering method and the dividing and merging framework. Conclusions and future work are presented in Sections VI.

corresponding author: Chen Xinquan.

## II. SOME DEFINITIONS OF HYBRID ATTRIBUTES DATA STREAMS

$< A_1 \times \cdots \times A_{m_1} \times A_{m_1+1} \times \cdots \times A_{m_1+m_2}, T >$ is a domain description of $m(m = m_1+m_2)$ dimensions data stream marked by a time stamp for every data point. In this $m$ dimensions space, which contains $m_1$ ordered attributes and $m_2$ sorted attributes, these is a data stream $SD = \{< X_1, T_1 >, \cdots, < X_N, T_N >, \cdots\}$, where $X_i = (x_{i1}, \cdots, x_{im})$ $(i = 1, \cdots, N, \cdots)$ is the $i$-th data point and $T_i (i = 1, \cdots, N, \cdots)$ is the $i$-th time stamp.

### *A. Some definitions of projected cluster and projected cluster group on ordered attributes*

**Definition 1**(**Projected cluster** on ordered attributes) A **projected cluster** $\mathbf{DSC_1} = \{< X_{i_1}, T_{i_1} >, \cdots, < X_{i_n}, T_{i_n} >\}$ of a $m$ dimensions data stream subset $\mathbf{DS_m} = \{< X_{j_1}, T_{j_1} >, \cdots, < X_{j_N}, T_{j_N} >\}$ on $m_1$ ordered attributes can be defined as a five-member group

$$\mathrm{DS}_{\mathrm{order}}(\mathrm{Cluster}) = < CF2^x, CF1^x, cf2^t, cf1^t, n > \quad (1)$$

This **projected cluster** structure $DS_{order}$(Cluster) is constructed according to the cluster structure of CluStream [6] and [8]. In formula (1),

$CF2^x = (CF2_1^x, \cdots, CF2_{m_1}^x)$ , in which $CF2_k^x = \sum_{j=i_1}^{i_n} (x_{jk})^2 \ (k = 1, \cdots, m_1)$ is the square sum of the *k*-th dimension values of data points in **DSC₁**.

$CF1^x = (CF1_1^x, \cdots, CF1_{m_1}^x)$ , in which $CF1_k^x = \sum_{j=i_1}^{i_n} x_{jk} \ (k = 1, \cdots, m_1)$ is the sum of the *k*-th dimension values of data points in **DSC₁**.

$cf2^t = \sum_{j=i_1}^{i_n} (T_j)^2$ is the square sum of time stamps of data points in **DSC₁**.

$cf1^t = \sum_{j=i_1}^{i_n} T_j$ is the sum of time stamps of data points in **DSC₁**.

*n* is the number of data points in **DSC₁**.

**Definition 2**(**Projected cluster group** on ordered attributes)

A **projected cluster group** of a *m* dimensions data stream subset $\mathbf{DS_m} = \{< X_{j_1}, T_{j_1} >, \cdots, < X_{j_N}, T_{j_N} >\}$ on $m_1$ ordered attributes can be defined as a three-member group [8]

$$DS_{order}(\text{Cluster group}) = (W_1, K_1, p_1) \quad (2)$$

In formula (2),

$W_1 = (w_1, \cdots, w_{m_1})$ is the weighted feature vector of data stream subset **DS_m** on $m_1$ ordered attributes.

$K_1$ is the number of projected clusters in this **projected cluster group**.

$p_1$ is the pointer that points a data structure storing $K_1$ projected clusters of this **projected cluster group**.

*B. Some definitions of projected cluster and projected cluster group on sorted attributes*

**Definition 3**(**Projected cluster** on sorted attributes, **the first definition**)

A **projected cluster** $\mathbf{DSC_2} = \{< X_{i_1}, T_{i_1} >, \cdots, < X_{i_n}, T_{i_n} >\}$ of a *m* dimensions data stream subset $\mathbf{DS_m} = \{< X_{j_1}, T_{j_1} >, \cdots, < X_{j_N}, T_{j_N} >\}$ on $m_2$ sorted attributes can be defined as a five-member group

$$DS_{sort}(\text{Cluster}) = < VF, ND, cf2^t, cf1^t, n > \quad (3)$$

This **projected cluster** structure $DS_{order}$(Cluster) is constructed extendedly according to the cluster structure of CluStream [6]. In formula (3),

$VF = (VF_{m_1+1}, \cdots, VF_{m_1+m_2})$ , in which $VF_k \ (k = m_1 + 1, \cdots, m_1 + m_2)$ is the most frequent value of data points in **DSC₂** in the *k*-th dimension. Especially, $VF_k$ makes null when values of data points in **DSC₂** in the *k*-th dimension are equally distributed in its domain.

$ND = (ND_{m_1+1}, \cdots, ND_{m_1+m_2})$ , in which $ND_k \ (k = m_1 + 1, \cdots, m_1 + m_2)$ is the number of data points in **DSC₂** different from $VF_k$ in the *k*-th dimension.

$cf2^t = \sum_{j=i_1}^{i_n} (T_j)^2$ is the square sum of time stamps of data points in **DSC₂**.

$cf1^t = \sum_{j=i_1}^{i_n} T_j$ is the sum of time stamps of data points in **DSC₂**.

*n* is the number of data points in **DSC₂**.

**Note:** $VF = (VF_{m_1+1}, \cdots, VF_{m_1+m_2})$ of **Definition 3** is similar to center vector, and $ND = (ND_{m_1+1}, \cdots, ND_{m_1+m_2})$ is similar to standard deviation.

**Definition 4**(**Projected cluster group** on sorted attributes)

A **projected cluster group** of a *m* dimensions data stream subset $\mathbf{DS_m} = \{< X_{j_1}, T_{j_1} >, \cdots, < X_{j_N}, T_{j_N} >\}$ on $m_2$ sorted attributes can be defined as a three-member group

$$DS_{sort}(\text{Cluster group}) = (W_2, K_2, p_2) \quad (4)$$

In formula (4),

$W_2 = (w_{m_1+1}, \cdots, w_{m_1+m_2})$ is the weighted feature vector of data stream subset **DS_m** on $m_2$ sorted attributes.

$K_2$ is the number of projected clusters in this **projected cluster group**.

$p_2$ is the pointer that points a data structure storing $K_2$ projected clusters of this **projected cluster group**.

**Definition 5**(**Projected cluster** on sorted attributes, **the second definition**)

A **projected cluster** $\mathbf{DSC_2} = \{< X_{i_1}, T_{i_1} >, \cdots, < X_{i_n}, T_{i_n} >\}$ of a *m* dimensions data stream subset $\mathbf{DS_m} = \{< X_{j_1}, T_{j_1} >, \cdots, < X_{j_N}, T_{j_N} >\}$ on $m_2$ sorted attributes can be defined as a four-member group:

$$DS_{sort}(\text{Cluster}) = < AF, cf2^t, cf1^t, n > \quad (5)$$

This **projected cluster** structure $DS_{order}$(Cluster) is constructed extendedly according to clustering result tree of Wkmeans in AlphaMiner [2] and cluster structure of CluStream [6]. In formula (5),

$AF = (AF_{m_1+1}, \cdots, AF_{m_1+m_2})$ , in which $AF_k \ (k = m_1 + 1, \cdots, m_1 + m_2)$ is dependent on the domain values of the *k*-th dimension. Suppose $|AF_k| = a_k$ is the number of the domain values of the *k*-th dimension, then $AF_k = (< v_{k1}, n_{k1} >, \cdots, < v_{ka_k}, n_{ka_k} >)$ can denote $a_k$ different sort values and its corresponding number of data

points in the *k*-th dimension. In order to handle conveniently, $a_k$ different sort values are arranged orderly.

$cf2^t = \sum_{j=i_1}^{i_n} (T_j)^2$ is the square sum of time stamps of data points in **DSC**$_2$.

$cf1^t = \sum_{j=i_1}^{i_n} T_j$ is the sum of time stamps of data points in **DSC**$_2$.

*n* is the number of data points in **DSC**$_2$.

**Note:** In clustering result tree of WKMeans [2], it records various sort values and its corresponding number of data points of this cluster for ordered attributes, and it also records means and standard deviation for sorted attributes. The cluster structure in this paper can records more detail information. When sorted attributes make too many values, this cluster structure will be too large.

*C. Some definitions of cluster and cluster group on hybrid attributes*

**Definition 6**(**cluster** on hybrid attributes, **the first definition**)

A **cluster DSC** $=\{< X_{i_1}, T_{i_1} >, \cdots, < X_{i_n}, T_{i_n} >\}$ of a *m* dimensions data stream subset **DS$_m$** $=\{< X_{j_1}, T_{j_1} >, \cdots, < X_{j_N}, T_{j_N} >\}$ can be defined as a seven-member group

DS(Cluster) $= < CF2^x, CF1^x, VF, ND, cf2^t, cf1^t, n >$ (6)

In formula (6),

$CF2^x = (CF2_1^x, \cdots, CF2_{m_1}^x)$ , in which $CF2_k^x = \sum_{j=i_1}^{i_n} (x_{jk})^2 \ (k = 1, \cdots, m_1)$ is the square sum of the *k*-th dimension values of data points in **DSC**.

$CF1^x = (CF1_1^x, \cdots, CF1_{m_1}^x)$ , in which $CF1_k^x = \sum_{j=i_1}^{i_n} x_{jk} \ (k = 1, \cdots, m_1)$ is the sum of the *k*-th dimension values of data points in **DSC**.

$VF = (VF_{m_1+1}, \cdots, VF_{m_1+m_2})$ , in which $VF_k \ (k = m_1 + 1, \cdots, m_1 + m_2)$ is the most frequent value of data points in **DSC** in the *k*-th dimension. Especially, $VF_k$ makes null when values of data points in **DSC** in the *k*-th dimension are equally distributed in its domain.

$ND = (ND_{m_1+1}, \cdots, ND_{m_1+m_2})$ , in which $ND_k \ (k = m_1 + 1, \cdots, m_1 + m_2)$ is the number of data points in **DSC** different from $VF_k$ in the *k*-th dimension.

$cf2^t = \sum_{j=i_1}^{i_n} (T_j)^2$ is the square sum of time stamps of data points in **DSC**.

$cf1^t = \sum_{j=i_1}^{i_n} T_j$ is the sum of time stamps of data points in **DSC**.

*n* is the number of data points in **DSC**.

**Definition 7**(**cluster** on hybrid attributes, **the second definition**)

A **cluster DSC** $=\{< X_{i_1}, T_{i_1} >, \cdots, < X_{i_n}, T_{i_n} >\}$ of a *m* dimensions data stream subset **DS$_m$** $=\{< X_{j_1}, T_{j_1} >, \cdots, < X_{j_N}, T_{j_N} >\}$ can be defined as a six-member group:

DS(Cluster) $= < CF2^x, CF1^x, AF, cf2^t, cf1^t, n >$ (7)

In formula (7),

$CF2^x = (CF2_1^x, \cdots, CF2_{m_1}^x)$ , in which $CF2_k^x = \sum_{j=i_1}^{i_n} (x_{jk})^2 \ (k = 1, \cdots, m_1)$ is the square sum of the *k*-th dimension values of data points in **DSC**.

$CF1^x = (CF1_1^x, \cdots, CF1_{m_1}^x)$ , in which $CF1_k^x = \sum_{j=i_1}^{i_n} (x_{jk}) \ (k = 1, \cdots, m_1)$ is the sum of the *k*-th dimension values of data points in **DSC**.

$AF = (AF_{m_1+1}, \cdots, AF_{m_1+m_2})$ , in which $AF_k \ (k = m_1 + 1, \cdots, m_1 + m_2)$ is dependent on the domain values of the *k*-th dimension. Suppose $|AF_k| = a_k$ is the number of the domain values of the *k*-th dimension, then $AF_k = (< v_{k1}, n_{k1} >, \cdots, < v_{ka_k}, n_{ka_k} >)$ can denote $a_k$ different sort values and its corresponding number of data points in the *k*-th dimension. In order to handle conveniently, $a_k$ different sort values are arranged orderly.

$cf2^t = \sum_{j=i_1}^{i_n} (T_j)^2$ is the square sum of time stamps of data points in **DSC**.

$cf1^t = \sum_{j=i_1}^{i_n} T_j$ is the sum of time stamps of data points in **DSC**.

*n* is the number of data points in **DSC**.

**Definition 8**(**cluster group** on hybrid attributes)

A **cluster group** of a *m* dimensions data stream subset **DS$_m$** $=\{< X_{j_1}, T_{j_1} >, \cdots, < X_{j_N}, T_{j_N} >\}$ can be defined as a three-member group:

DS(Cluster group) = (*W*, *K*, *p*) (8)

In formula (8),

$W = (w_1, \cdots, w_{m_1}, w_{m_1+1}, \cdots, w_{m_1+m_2})$ is the weighted feature vector of data stream subset **DS$_m$** on *m* hybrid attributes.

*K* is the number of clusters in this **cluster group**.

*p* is the pointer that points a data structure storing *K* clusters of this **cluster group**.

*D. Constructing method of cluster and cluster group on hybrid attributes*

Because one section of hybrid attributes data stream which can be seen as a static data set and be loaded into *EMS* memory, its cluster group can be obtained by the following **two-step projected clustering method**. And its cluster data structure can be computed according to the cluster's data points. Another constructing method of cluster group can use WKMeans [2] or weighted FCM algorithm [7].

## III. TWO-STEP PROJECTED CLUSTERING METHOD OF HYBRID ATTRIBUTES DATA STREAMS

*A. Two-step projected clustering method*

Suppose $DS_{order}$(Cluster group)= $\{C1_1,\cdots,C1_{K_1}\}$ is a projected clustering result of data stream subset $\mathbf{DS_m}$ on $m_1$ ordered attributes and $DS_{sort}$(Cluster group)= $\{C2_1,\cdots,C2_{K_2}\}$ is a projected clustering result of data stream subset $\mathbf{DS_m}$ on $m_2$ sorted attributes. $C1_i(i=1,\cdots,K_1)$ is a cluster of $DS_{order}$(Cluster group) which is a set containing label information of its data points, and it is the same with $C2_j(j=1,\cdots,K_2)$ for $DS_{sort}$(Cluster group).

After having merged $DS_{order}$(Cluster group) and $DS_{sort}$(Cluster group), we can obtain a clustering result of data stream subset $\mathbf{DS_m}$ on hybrid attributes.

**The two-step projected clustering method:**

```
DS(Cg) = { };  //this cluster group records clustering result on hybrid attributes. Cluster group abbreviates Cg.
for (i = 1; i ≤ K1; i++)
{   for (j = 1; j ≤ K2; j++)
    {
        DS(Cg) = DS(Cg) ∪ {C1i ∩ C2j};
    }
}
```

*B. Improvement of two-step projected clustering method*

The time complexity of the original two-step projected clustering method is $O(K_1\cdot K_2\cdot \max_{i=1,\cdots,K_1}\{|C1_i|\}\cdot \max_{j=1,\cdots,K_2}\{|C2_j|\})$, because the simple algorithm getting intersection of two sets needs $O(\max_{i=1,\cdots,K_1}\{|C1_i|\}\cdot \max_{j=1,\cdots,K_2}\{|C2_j|\})$. The time complexity of getting intersection algorithm can be reduced if ordering the two sets $DS_{order}(Cg)=\{C1_1,\cdots,C1_{K_1}\}$ and $DS_{sort}(Cg)=\{C2_1,\cdots,C2_{K_2}\}$ at first. The best time complexity of the ordering algorithm for $DS_{order}$(Cg) and $DS_{sort}$(Cg) is $O(K_1\cdot \max_{i=1,\cdots,K_1}\{|C1_i|\cdot\log|C1_i|\}+K_2\cdot \max_{j=1,\cdots,K_2}\{|C2_j|\cdot\log|C2_j|\})$.

Suppose the two ordered sets are $CO1_i$ and $CO2_j(i=1,\cdots,K_1;j=1,\cdots,K_2)$, then its time complexity of getting intersection algorithm is $O(|CO1_i|+|CO2_j|)$. It is also equal to $O(|C1_i|+|C2_j|)$ $(i=1,\cdots,K_1;j=1,\cdots,K_2)$. So the time complexity of the improved two-step projected clustering method is

$$O(K_1\cdot \max_{i=1,\cdots,K_1}\{|C1_i|\cdot\log|C1_i|\}+K_2\cdot \max_{j=1,\cdots,K_2}\{|C2_j|\cdot\log|C2_j|\}+K_1\cdot K_2\cdot \max_{\substack{i=1,\cdots,K_1\\ j=1,\cdots,K_2}}\{|C1_i|+|C2_j|\})$$

*C. Getting intersection from two ordered sets*

**Input:** Two ordered sets $S_1=\{X_1,\cdots,X_{n1}\}$ and $S_2=\{Y_1,\cdots,Y_{n2}\}$.

**Output:** S = { };  //S records the intersection of $S_1$ and $S_2$. Initial S is null.

**Procedure:**

```
int i = 1; j = 1;
while (i ≤ n1 && j ≤ n2)
{
    if (Xi == Yj)
    {
        S = S ∪ { Xi };  //append Xi to S
        i++; j++;
    }
    else
        if (Xi < Yj)
            i++;
        else
            j++;
}
if (i ≤ n1)
    for (k = i; k ≤ n1; k++)
    {
        S = S ∪ { Xk };
    }       //append remanent elements in S1 to S
if (j ≤ n2)
    for (k = j; k ≤ n2; k++)
    {
        S = S ∪ { Xk };
    }       //append remanent elements in S2 to S
```

**Note:** This is a simple method getting intersection from two ordered sets. Here it is listed independently only for integrality although it should not be the first method.

## IV. DIVIDING AND MERGING FRAMEWORK OF HYBRID ATTRIBUTES DATA STREAMS

*A. Dividing stratery of hybrid attributes data streams*

The dividing method of hybrid attributes data streams is usually based on the size of *EMS* memory and some field knowledge. For some data streams with special time meaning, we can partition them with a range in per second, per minute, per hour, per day, per week, per month or per year, and so on. One data stream section which can be loaded into the *EMS* memory can be analyzed using weighted clustering method or other analysis methods. At last, we usually select to merge multi data stream sections or do an evolutionary analysis based on application.

*B. Merging stratery of hybrid attributes data streams*

We can use a weighted clustering algorithm on hybrid attributes or **two-step projected clustering method** for each data stream section in order to get its cluster data structure which saves its some necessary information. How to select a merging strategy is often based on application.

*1) Merging and subtracting of projected clusters and projected cluster groups on ordered attributes*

The addition of projected clusters and the merging of projected cluster groups can get some summary information in some range in order to obtain cluster distributed status of data stream on ordered attributes. The subtraction of projected clusters and projected cluster groups can be used to do an evolutionary analysis of data stream on ordered attributes. This is the difference from CluStream [6]. Their implementation methods can be found in [6] and [8].

*2) Merging and subtracting of projected clusters and projected cluster groups on sorted attributes*

*a) Merging and subtracting of projected clusters on sorted attributes*

If we use the first definition(see **Definition 3**) for projected cluster on sorted attributes, then the merging judgement condition of $DS_{sort}$(Cluster1) and $DS_{sort}$(Cluster2) can be designed as

$VF$(Cluster1) = $VF$(Cluster2) (9)

**Property 1** (**the first additive property** of projected cluster on sorted attributes)

Suppose $C_1 = < VF1, ND1, cf2_1^t, cf1_1^t, n_1 >$ and $C_2 = < VF2, ND2, cf2_2^t, cf1_2^t, n_2 >$ are two projected clusters being able to merge on sorted attributes(satisfied $VF1=VF2$). Then their merged cluster can be represented as

$C=C_1+C_2=$

$$< VF, ND1+ND2, cf2_1^t + cf2_2^t, cf1_1^t + cf1_2^t, n_1+n_2 > \quad (10)$$

In formula (10), $VF = VF1 = VF2$.

**Property 2** (**the first subtractive property** of projected cluster on sorted attributes)

Suppose $C_1 = < VF1, ND1, cf2_1^t, cf1_1^t, n_1 >$ and $C_2 = < VF2, ND2, cf2_2^t, cf1_2^t, n_2 >$ are two projected clusters being able to make subtraction on sorted attributes(satisfied $VF1=VF2$). Then the difference of two projected clusters can be represented as

$C = C_2 - C_1 =$

$$< VF, ND2-ND1, cf2_2^t - cf2_1^t, cf1_2^t - cf1_1^t, n_2 - n_1 > \quad (11)$$

In formula (11), $VF = VF1 = VF2$.

The above two properties is obvious according to its definition.

If we use the second definition(see **Definition 5**) for projected cluster on sorted attributes, then the merging judgement condition of $DS_{sort}$(Cluster1) and $DS_{sort}$(Cluster2) can be designed as

$$dis_{sort}(C1,C2) = \sum_{k=m_1+1}^{m_1+m_2} (w_k \cdot dis_k(AF1_k, AF2_k)) \le \delta \quad (12)$$

In formula (12), $\delta$ is a threshold larger than 0, $w_k$ is the merged feature weight of Cluster1 and Cluster2 in the $k$-th dimension, and $dis_k(AF1_k, AF2_k) = \sum_{j=1}^{a_k} |n2_{kj} - n1_{kj}| \ (k = m_1+1, \cdots, m_1+m_2)$.

**Property 3** (**the second additive property** of projected cluster on sorted attributes)

Suppose $C_1 = < AF1, cf2_1^t, cf1_1^t, n_1 >$ and $C_2 = < AF2, cf2_2^t, cf1_2^t, n_2 >$ are two projected clusters being able to merge on sorted attributes(satisfied formula (12)). Then their merged cluster can be represented as

$C=C_1+C_2=$

$$< AF1+AF2, cf2_1^t + cf2_2^t, cf1_1^t + cf1_2^t, n_1+n_2 > \quad (13)$$

In formula (13), Suppose

$AF = AF1 + AF2 = (AF_{m_1+1}, \cdots, AF_{m_1+m_2})$ ,

$AF1 = (AF1_{m_1+1}, \cdots, AF1_{m_1+m_2})$,

$AF2 = (AF2_{m_1+1}, \cdots, AF2_{m_1+m_2})$.

Here, $AF1_k = (< v_{k1}, n1_{k1} >, \cdots, < v_{ka_k}, n1_{ka_k} >)$,

$AF2_k = (< v_{k1}, n2_{k1} >, \cdots, < v_{ka_k}, n2_{ka_k} >)$.

Suppose

$|AF1_k| = |AF2_k| = a_k \ (k = m_1+1, \cdots, m_1+m_2)$ is the number of different sort values in the $k$-th dimension. Because $a_k$ different sort values are arranged orderly, so we can add the number of data points according to corresponding sort value for two projected clusters.

Then

$AF_k = (< v_{k1}, n_{k1} >, < v_{k2}, n_{k2} >, \cdots, < v_{ka_k}, n_{ka_k} >)$

$(k = m_1+1, \cdots, m_1+m_2)$ can be obtained by

$AF_k = (< v_{k1}, n1_{k1} + n2_{k1} >, \cdots, < v_{ka_k}, n1_{ka_k} + n2_{ka_k} >)$

**Property 4** (**the second subtractive property** of projected cluster on sorted attributes)

Suppose $C_1 = < AF1, cf2_1^t, cf1_1^t, n_1 >$ and $C_2 = < AF2, cf2_2^t, cf1_2^t, n_2 >$ are two projected clusters being able to make subtraction on sorted attributes (satisfied formula (12)). Then the difference of two projected clusters can be represented as

$C = C_2 - C_1 =$

$$< AF2-AF1, cf2_2^t - cf2_1^t, cf1_2^t - cf1_1^t, n_2 - n_1 > \quad (14)$$

In formula (14), Suppose

$AF = AF2 - AF1 = (AF_{m_1+1}, \cdots, AF_{m_1+m_2})$ ,

$AF1 = (AF1_{m_1+1}, \cdots, AF1_{m_1+m_2})$,

$AF2 = (AF2_{m_1+1}, \cdots, AF2_{m_1+m_2})$.

Here, $AF1_k = (< v_{k1}, n1_{k1} >, \cdots, < v_{ka_k}, n1_{ka_k} >)$,

$AF2_k = (< v_{k1}, n2_{k1} >, \cdots, < v_{ka_k}, n2_{ka_k} >)$.

Suppose

$|AF1_k| = |AF2_k| = a_k \quad (k = m_1+1, \cdots, m_1+m_2)$ is

the number of different sort values in the *k*-th dimension. Because $a_k$ different sort values are arranged orderly, so we can make subtraction of the number of data points according to corresponding sort value for two projected clusters.

Then $AF_k = (< v_{k1}, n_{k1} >, \cdots, < v_{ka_k}, n_{ka_k} >)$ $(k = m_1 + 1, \cdots, m_1 + m_2)$ can be obtained by

$$AF_k = (< v_{k1}, n2_{k1} - n1_{k1} >, \cdots, < v_{ka_k}, n2_{ka_k} - n1_{ka_k} >)$$

*b) The merging and evolutionary analysis of two projected cluster groups on sorted attributes*

The merging and evolutionary analysis method is similar to [8].

If we use the first definition of projected cluster on sorted attributes, the merging judgement condition is *VF*(Cluster1) = *VF*(Cluster2).

If we use the second definition of projected cluster on sorted attributes, the merging judgement condition can be designed as formula (12).

*3) Merging and subtracting of clusters and cluster groups on hybrid attributes*

*a) Merging and subtracting of clusters on hybrid attributes*

**Merging and subtracting of two clusters $C_1$ and $C_2$ on hybrid attributes satisfied two conditions:**

(1) Projected clusters of $C_1$ and $C_2$ on ordered attributes satisfied the merging judgement condition(see algorithm 2 in [8]).

(2) Projected clusters of $C_1$ and $C_2$ on sorted attributes satisfied its corresponding merging judgement condition.

The merging judgement condition on hybrid attributes is implemented in **Algorithm 2** of appendix.

**Property 5** (**the first additive property** of cluster on hybrid attributes)

Suppose

$C_1 = < CF2^{x1}, CF1^{x1}, VF1, ND1, cf2_1^t, cf1_1^t, n_1 >$ and

$C_2 = < CF2^{x2}, CF1^{x2}, VF2, ND2, cf2_2^t, cf1_2^t, n_2 >$ are two clusters being able to merge on hybrid attributes(both satisfied *VF*1=*VF*2 on sorted attributes and satisfied its merging judgement condition on ordered attributes(see algorithm 2 in [8])). Then their merged cluster can be represented as

$C = C_1 + C_2 =$

$$< CF2^{x1} + CF2^{x2}, CF1^{x1} + CF1^{x2}, VF, ND1 + ND2, cf2_1^t + cf2_2^t, cf1_1^t + cf1_2^t, n_1 + n_2 > \quad (15)$$

In formula (15), *VF* = *VF*1 = *VF*2.

**Property 6**(**the second additive property** of cluster on hybrid attributes)

Suppose $C_1 = < CF2^{x1}, CF1^{x1}, AF1, cf2_1^t, cf1_1^t, n_1 >$ and $C_2 = < CF2^{x2}, CF1^{x2}, AF2, cf2_2^t, cf1_2^t, n_2 >$ are two clusters being able to merge on hybrid attributes(both satisfied formula (12) on sorted attributes and satisfied its merging judgement condition on ordered attributes(see algorithm 2 in [8])). Then their merged cluster can be represented as

$C = C_1 + C_2 =$

$$< CF2^{x1} + CF2^{x2}, CF1^{x1} + CF1^{x2}, AF1 + AF2, cf2_1^t + cf2_2^t, cf1_1^t + cf1_2^t, n_1 + n_2 > \quad (16)$$

In formula (16),

$$AF = AF1 + AF2 = (AF_{m_1+1}, \cdots, AF_{m_1+m_2})$$

$$= (AF1_{m_1+1} + AF2_{m_1+1}, \cdots, AF1_{m_1+m_2} + AF2_{m_1+m_2}).$$

Here,

$$AF_k = (< v_{k1}, n1_{k1} + n2_{k1} >, \cdots, < v_{ka_k}, n1_{ka_k} + n2_{ka_k} >)$$

$$(k = m_1 + 1, \cdots, m_1 + m_2).$$

**Property 7** (**the first subtractive property** of cluster on hybrid attributes)

Suppose

$C_1 = < CF2^{x1}, CF1^{x1}, VF1, ND1, cf2_1^t, cf1_1^t, n_1 >$ and

$C_2 = < CF2^{x2}, CF1^{x2}, VF2, ND2, cf2_2^t, cf1_2^t, n_2 >$ are two clusters being able to make subtraction on hybrid attributes(both satisfied *VF*1=*VF*2 on sorted attributes and satisfied its merging judgement condition on ordered attributes(see algorithm 2 in [8])). Then their difference can be represented as

$C = C_2 - C_1 =$

$$< CF2^{x2} - CF2^{x1}, CF1^{x2} - CF1^{x1}, VF, ND2 - ND1, cf2_2^t - cf2_1^t, cf1_2^t - cf1_1^t, n_2 - n_1 > \quad (17)$$

In formula (17), *VF* = *VF*1 = *VF*2.

**Property 8**(**the second subtractive property** of cluster on hybrid attributes)

Suppose

$C_1 = < CF2^{x1}, CF1^{x1}, AF1, cf2_1^t, cf1_1^t, n_1 >$ and

$C_2 = < CF2^{x2}, CF1^{x2}, AF2, cf2_2^t, cf1_2^t, n_2 >$ are two clusters being able to make subtraction on hybrid attributes(both satisfied formula (12) on sorted attributes and satisfied its merging judgement condition on ordered attributes(see algorithm 2 in [8])). Then their difference can be represented as

$C = C_2 - C_1 =$

$$< CF2^{x2} - CF2^{x1}, CF1^{x2} - CF1^{x1}, AF, cf2_2^t - cf2_1^t, cf1_2^t - cf1_1^t, n_2 - n_1 > \quad (18)$$

In formula (18),

$$AF = AF2 - AF1 = (AF_{m_1+1}, \cdots, AF_{m_1+m_2})$$

$$= (AF2_{m_1+1} - AF1_{m_1+1}, \cdots, AF2_{m_1+m_2} - AF1_{m_1+m_2}).$$

Here,

$$AF_k = (< v_{k1}, n2_{k1} - n1_{k1} >, \cdots, < v_{ka_k}, n2_{ka_k} - n1_{ka_k} >)$$

$$(k = m_1 + 1, \cdots, m_1 + m_2).$$

*b) The merging and evolutionary analysis of two cluster groups on hybrid attributes*

The merging and evolutionary analysis method is similar to [8]. We use **Algorithm 9*** of appendix to make an optimum merging of two cluster groups on hybrid attributes and **Algorithm 15*** of appendix to make an optimum difference of two cluster groups on hybrid attributes. The two algorithms use optimum match cluster-pair strategy which is used and described in [8].

*4) The time complexity of merging framework of hybrid attributes data streams*

According to **Property 5**, **6**, **7**, and **8**, merging framework of hybrid attributes data streams only needs a linear computing(adding or subtracting) between two clusters on hybrid attributes after having searched some optimum match cluster-pairs. So the time complexity of merging framework of hybrid attributes data streams is mainly relied on **Algorithm 9*** and **Algorithm 15*** of appendix. So its time complexity is

$$O(K_1 \cdot K_2 \cdot \min\{K_1, K_2\} \cdot (m_1 + m_2 \cdot \max\{a_{m_1+1}, \cdots, a_{m_1+m_2}\})) \quad (19)$$

In formula (19), $K_1$ and $K_2$ are the cluster numbers of two cluster groups which will be merged or made an evolutionary analysis, $m_1$ is the dimensional number of ordered attributes, $m_2$ is the dimensional number of sorted attributes, and $a_k$ $(k = m_1 + 1, \cdots, m_1 + m_2)$ is the number of different sort values in the *k*-th dimension.

## V. SIMULATED EXPERIMENTS

### *A. Validity experiment for two-step projected clustering method*

*1) Data sets*

*a) German data set [9]*

According to decision tree algorithm J8 in AlphaMiner [2], there are 31 classification rules with 78.4% classification rate when the number of leafage node is set to 8. Then the effective classification attributes are {at1,at2,at3,at4,at5,at7,at10,at13,at17, at20}, and the unused classification attributes are {at6,at8,at9,at11,at12,at14,at15,at16,at18,at19}.

*b) Credit Approval data set [10]*

According to J8 [2], we can obtain a classification rule set. Its effective classification attributes are {at3,at4,at9,at10,at14,at15}, and the unused classification attributes are {at1,at2,at5,at6,at7,at8,at11,at12,at13}.

*2) Experimental method*

The WKMeans [2] is used as the clustering algorithm in this experiment. The number of clusters is set to 2, and weight exponent is set to 2. When using ordered attributes only, WKMeans algorithm can only get one cluster. It gets the same result that the number of clusters is set to 3, or 4. When facing this plight, we use *k*-means in AlphaMiner [2] as the clustering algorithm.

*3) Experimental results*

See TABLE I and TABLE II**.**

It defines the **clustering purity number** as the **number of matched data points between clustering result and its class**.

*4) Analysis and conclusions of experimental results*

It can often get a better clustering quality than WKMeans algorithm first using two-step projected clustering method and then merging the clustering results for these two data sets. Because classification ability between ordered attributes and sorted attributes has large differences for some hybrid attributes data sets. WKMeans algorithm can not get a better clustering quality sometimes because it does not handle ordered attributes and sorted attributes rightly.

TABLE I.
THE COMPARISON EXPERIMENTAL RESULT TABLE FOR GERMAN

| Cluster method | Clustering purity number |
|---|---|
| WKMeans clustering algorithm (using 20 condition attributes of this data set) | 635 |
| WKMeans clustering algorithm (using 10 effective classification attributes {at1, at2, at3, at4, at5, at7, at10, at13, at17, at20}) | 561 |
| First using two-step projected clustering method, then merging the clustering results. (merging the projected clustering result on 7 ordered attributes and the projected clustering result on 13 sorted attributes) | **671** |
| First using two-step projected clustering method, then merging the clustering results. (merging the projected clustering result on 3 effective classification ordered attributes and the projected clustering result on 7 effective classification sorted attributes) | **679** |

TABLE II.
THE COMPARISON RESULT TABLE FOR CREDIT APPROVAL

| Cluster method | Clustering purity number |
|---|---|
| WKMeans clustering algorithm (using 15 condition attributes of this data set) | 526 |
| WKMeans clustering algorithm (using 6 effective classification attributes { at3,at4,at9,at10, at14,at15}) | 547 |
| First using two-step projected clustering method, then merging the clustering results. (merging the projected clustering result on 6 ordered attributes and the projected clustering result on 9 sorted attributes) | 522 |
| First using two-step projected clustering method, then merging the clustering results. (merging the projected clustering result on 3 effective classification ordered attributes and the projected clustering result on 3 effective classification sorted attributes) | **563** |

### *B. Validity experiment of dividing and merging framework*

*1) Data set*

1000 data records of German data set [9] are thrown into two data sets(named German1 and German2, each has 500 records) randomly.

*2) Experimental method*

Do a validity experiment of merging framework like [8]. Some data structures and 15 algorithms for implementing merging framework of hybrid attributes data streams are debugged and have passed in VC++6.00. We can merge clustering results of German1 and German2 after {at1,at2,at3,at4,at5,at7,at10,at13,at17,at20} are used as input attributes in WKMeans [2]. At last do a comparison between the result using merging framework and the result using WKMeans [2].

*3) Experimental results of WKMeans*

**Result 1**: German1 has 500 records. From clustering result table of WKMeans [2], we know clust0 has 171 records and clust1 has 329 records.

361 records with tag class1 contain 112 records in clust0 and **249** records in clust1. 139 records with tag

TABLE III.
THE COMPARISON EXPERIMENTAL RESULT TABLE FOR GERMAN

| Cluster method | Clustering purity number |
|---|---|
| The clustering results using WKMeans is the input of merging framework (merging German1 and German2) | 308 + 291 = **599** |
| WKMeans clustering algorithm for German(ordered) | 572 |

class2 contain **59** records in clust0 and 80 records in clust1.

So the **clustering purity number** is **249**+**59** = **308**.

**Result 2**: German2 has 500 records. From clustering result table of WKMeans [2], we know clust0 has 258 records and clust1 has 242 records.

339 records with tag class1 contain **194** records in clust0 and 145 records in clust1. 161 records with tag class2 contain 64 records in clust0 and **97** records in clust1.

So the **clustering purity number** is **194**+**97** = **291**.

**Result 3**: German(ordered) has 1000 records. From clustering result table of WKMeans [2], we know clust0 has 560 records and clust1 has 440 records.

700 records with tag class1 contain **416** records in clust0 and 284 records in clust1. 300 records with tag class2 contain 144 records in clust0 and **156** records in clust1.

So the **clustering purity number** is **416**+**156** = **572**.

*4) Merged clustering result*

ordered attributes: {at2,at5,at13}, sorted attributes: {at1,at3,at4,at7,at10,at17,at20}.

Using WKMeans [2] can obtain a weighted feature vector participating distance computing between two clusters for German(ordered) data set.

At last, the cluster-pair (cp1–cp4) is merged, and the cluster-pair (cp2–cp3) is also merged.

The condition of merging clusters mainly relies on the distance computing on sorted attributes. So the threshold enactment is important to the merging of clusters between two cluster groups. The merging judgement function of two clusters is **Algorithm 2** in appendix.

When the threshold SortD is below 63, there is no cluster can be merged. When the threshold SortD takes 64, there are 3 clusters after german1 and german2 are merged. Because distance weight on sorted attributes is larger than on ordered attributes, the enactment of threshold is important to the merging of two clusters.

**Note:** After having merged two cluster groups, the **clustering purity number** in merged clustering results is the sum of **Result 1** and **Result 2**. So it is 308 + 291 = **599**. Its clustering result with **599** is larger than **Result 3** with 572.

*5) Conclusions of experimental results*

The merging framework is not only to solve weighted clustering problem of hybrid attributes data streams, but also can get a better clustering quality than WKMeans sometimes if having set right parameters.

## VI. CONCLUSIONS

In order to solve clustering problem of hybrid attributes data streams, it presents some definitions of cluster and cluster group. In order to improve the clustering quality of hybrid attributes data streams, it presents a two-step projected clustering method, which can often make better clustering effects in two simulated experiments. In order to implement the dividing and merging framework of hybrid attributes data streams, it gives some data structures and algorithms in appendix. The framework is verified and these algorithms are tested by German data set with a better clustering quality than WKMeans [2] sometimes if having set right parameters.

It is the next work that applying the framework and its algorithms to other hybrid attributes data streams and doing a comparison with CluStream [6]. Usually, better experimental results are based on appropriate parameters. So there is more research work for adaptive optimization of parameters.

## APPENDIX

The data structure definitions and 15 algorithms described by C language are listed in [11].

## ACKNOWLEDGMENT

This work was supported in part by a grant No GJJ08467 from the JiangXi Provincial Department of Education, China.

## REFERENCES

[1] J. Z. Huang, K. P. Ng, H. Q. Rong, et al, "Automated Variable Weighting in k-Means Type Clustering," *IEEE Trans on pattern analysis and machine intelligence*, 2005, 27(5), pp.657-668.
[2] AlphaMiner. http://bi.hitsz.edu.cn/
[3] X. Z. Wang, Y. D. Wang, and L. J. Wang, "Improving Fuzzy C-Means clustering based on feature-weight learning," *Pattern Recognition Letters*, 2004, 25(10), pp.1123-1132.
[4] C. Aggarwal, J. Han, J. Wang, et al, "A Framework for Projected Clustering of High Dimensional Data Streams," the 30th VLDB, 2004, pp.852-863.
[5] G. Moise, J. Sander, and M. Ester, "Robust projected clustering," *Knowl. Inf. Syst*, 2008, 14, pp.273-298.
[6] C. Aggarwal, J. Han, J. Wang, et al, "A framework for Clustering evolving data streams," the 29th VLDB, 2003, pp.81-92.
[7] X. Q. Chen, "Feature-weighted fuzzy C clustering algorithm,", *Computer Engineering and Design*, China, 2007, 28(22), pp.5329-5333.
[8] X. Q. Chen, "Weighted clustering and evolutionary analysis facing to data streams," http://www.paper.edu.cn, 2008, No: 200803-107.
[9] German dataset, http://www.niaad.liacc.up.pt/old/statlog/datasets/german/german.doc.html.
[10] Credit Approval dataset, http://www.ics.uci.edu/~mlearn/databases/credit-screening/crx.name.
[11] X. Q. Chen, "Weighted clustering and evolutionary analysis of hybrid attributes data streams" http://www.paper.edu.cn, 2008, No: 200805-29.