

# AWST: A Novel Attribute Weight Selection Technique for Data Clustering

**Md Anisur Rahman and Md Zahidul Islam**

School of Computing and Mathematics  
Charles Sturt University  
Panorama Avenue, Bathurst, NSW 2795  
Australia.  
{arahman, zislam}@csu.edu.au

## Abstract

In this paper we propose a novel attribute weight selection technique called AWST that automatically determines attribute weights for a clustering purpose. The main idea of AWST is to assign weight on an attribute based on the ability of the attribute to cluster the records of a dataset. The attributes with higher abilities get higher weights for clustering. We also propose a novel discretization approach in AWST to discretize the domain values of a numerical attribute. The performance of AWST is compared with three other existing attribute weight selection techniques. We compare the performance of AWST with the three existing techniques namely SABC, WKM and EB in terms of Silhouette Coefficient using nine (9) natural datasets that we obtain from the UCI machine learning repository. The experimental results show that AWST outperforms than the existing techniques on all datasets. The computational complexities and the execution times of the techniques are also presented in the paper. Note that, AWST requires less execution time than many of the existing techniques used in this study.

*Keywords:* Clustering, Fuzzy Clustering, Hard Clustering, Cluster Evaluation, Data Mining, Attribute Weight Selection.

## 1 Introduction

Clustering is a process of grouping similar records in a cluster and dissimilar records in different clusters (Rahman, 2014, Tan et al., 2005, Han and Kamber, 2006, Rahman and Islam, 2014, Rahman et al., 2014, Rahman et al., 2015). It extracts hidden patterns, from large datasets, that helps in decision making processes in various fields including medical research, crime detection/prevention, social network analysis and market research (Zhao and Zhang, 2011, Oatley and Ewart, 2003, Adderley et al., 2007, Li et al., 2010, Pirim et al., 2012, Sun et al., 2012, Chan et al., 2012). Therefore, it is important to produce good quality clusters from a dataset.

There are many existing clustering techniques that consider all attributes of a dataset as equally important for the clustering purpose (Redmond and Heneghan, 2007, Chatzis, 2011, Lee and Pedrycz, 2009).

Copyright (C) 2015, Australian Computer Society, Inc. This paper appeared at the Thirteenth Australasian Data Mining Conference, Sydney, Australia. Conferences in Research and Practice in Information Technology, Vol. 168. Md Zahidul Islam, Ling Chen, Kok-Leong Ong, Yanchang Zhao, Richi Nayak, Paul Kennedy, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

However, all attributes of a dataset may not be equally important for clustering (Rahman and Islam, 2011, Ahmad and Dey, 2007, Rahman, 2014, Rahman and Islam, 2012, Hung et al., 2011). Hence, it is important to find the appropriate attribute weights for clustering. There are many existing techniques that automatically identify attribute weights/importance (Ahmad and Dey, 2007, Bai et al., 2011, Hung et al., 2011, Chen et al., 2012, Cordeiro de Amorim and Mirkin, 2012, Huang et al., 2005, Niu et al., 2008, Boongoen et al., 2011, Gançarski et al., 2008, He et al., 2011).

It is assumed in some literatures on clustering (Rahman and Islam, 2012, Rahman, 2014, Rahman and Islam, 2011, Islam and Brankovic, 2011) that the user knows their dataset well and therefore would be able to assign weights to the attributes to meet their clustering purposes. However, this may not always be the case and the user may prefer to use automatic weights for clustering.

Therefore, in this paper we propose a technique called AWST for selecting attribute weights automatically. The main idea of AWST is to assign weight to an attribute based on the ability of the attribute to cluster the records. The attributes with higher abilities get higher weights. By using AWST a user can assign weights automatically on the attributes or can get an idea of the weights of the attributes so that they can assign weights manually for clustering the records of a dataset. We also propose a novel discretization technique in AWST in order to find the attribute weights. Therefore, the contributions of this paper are as follows.

- A novel discretization technique
- A novel attribute weight selection technique

We compare the performance of AWST with three other existing attribute weight selection techniques namely SABC (Ahmad and Dey, 2007), WKM (Hung et al., 2011) and EB in terms of Silhouette Coefficient using nine (9) natural datasets that we obtain from the UCI machine learning repository (Bache and Lichman, 2013). The performance of AWST is better than the three existing attribute weight selection techniques on nine (9) datasets. We also present the computational complexities and execution times of the techniques. Note that, the execution time of AWST is lower than some of the existing techniques used in this paper.

The structure of the paper is as follows: in section 2 we discuss background study; in section 3 we present our proposed technique called AWST; in section 4 an empirical analysis on our discretization approach is presented; in section 5 the experimental results and

discussion are presented; and the conclusion of the paper is presented in section 6.

## 2 Background study

In this study,  $D$  denotes a dataset,  $n$  denotes the number of records of dataset  $D$  i.e.  $D = \{R_1, R_2, \dots, R_n\}$ , and  $m$  denotes the number of attributes of dataset  $D$ , i.e.  $A = \{A_1, A_2, \dots, A_m\}$ . The attributes of a dataset can be numerical and/or categorical (Tan et al., 2005, Han and Kamber, 2006). The numerical and categorical attributes are also known as continuous and nominal attributes, respectively. There is a natural ordering among the domain values of a numerical attribute, whereas there is no natural ordering among the domain values of a categorical attribute. In Table 1, we present an example dataset that has ten records ( $R_1, R_2, \dots, R_{10}$ ) and four attributes (Age, Marital-Status, Qualification, and Occupation), where Marital-Status, Qualification, and Occupation are categorical attributes and Age is a numerical attribute. The domain values of the numerical attribute Age range from 30 to 65. The domain values for the categorical attribute Marital-Status are {Single, Married}. Similarly, the domain values of all the other categorical attributes can be learnt from Table 1.

Record	Age	Marital-Status	Qualification	Occupation
$R_1$	65	Married	PhD	Academic
$R_2$	30	Single	Master	Engineer
$R_3$	45	Married	Master	Engineer
$R_4$	30	Single	Bachelor	Physician
$R_5$	55	Married	PhD	Academic
$R_6$	35	Single	Bachelor	Physician
$R_7$	60	Married	PhD	Academic
$R_8$	45	Single	Bachelor	Physician
$R_9$	35	Single	Master	Engineer
$R_{10}$	42	Married	Master	Engineer

**Table 1: A synthetic dataset**

Many existing clustering techniques consider that all attributes in a dataset have equal weights (significance levels) meaning that all attributes are equally important for clustering (Redmond and Heneghan, 2007, Lee and Pedrycz, 2009, Chatzis, 2011). They do not allow the data miner to assign different weights to different attributes. In these techniques, the data miner can either ignore (i.e. assign a weight equal to 0) or consider (i.e. assign a weight equal to 1) an attribute while clustering the records.

There are of course a number of clustering techniques that automatically (not user defined) assign weights to attributes (Ahmad and Dey, 2007, Bai et al., 2011, Hung et al., 2011, Chen et al., 2012, Cordeiro de Amorim and Mirkin, 2012, Huang et al., 2005, Niu et al., 2008, Boongoen et al., 2011, Fan et al., 2009, Chan et al., 2004, Gańczarski et al., 2008, He et al., 2011, Huang, 1998). Since the weights are calculated automatically the user does not have the opportunity to assign different weights and explore various clustering results. The weight of the

attributes is often calculated using the pair-wise distance of the values belonging to the attribute (with respect to other values belonging to other attributes), where a higher pair-wise distance (on average) indicates a greater ability to separate/cluster the records (Ahmad and Dey, 2007, He et al., 2011). The weight of an attribute can also be calculated from its variation within the clusters. If the total distance between the values of an attribute within a cluster, for all clusters, is low then it shows a low variation of attribute values. In this paper we call the attribute weight selection based clustering proposed by Ahmad and Dey (2007) as SABC. An attribute weight is considered to be inversely proportional to the variation of the attribute (Huang et al., 2005, Cordeiro de Amorim and Mirkin, 2012). The entropy of the values of an attribute is sometimes used for calculating weight where high entropy indicates a low variation and high weight (Hung et al., 2011). The attribute weight (based on entropy,) based K-Means clustering technique is called Weighted K-Means (WKM) (Hung et al., 2011). WKM does not work on a dataset that has both categorical and numerical attributes whereas our proposed attribute weight selection technique called AWST works on a dataset that has both categorical and numerical attributes.

Attribute weights are often calculated separately within each cluster from an initial set of clusters. This approach is generally called Subspace Clustering, which can be an effective clustering method, especially for high dimensional datasets, in order to avoid the curse of dimensionality (Huang et al., 2005, Bai et al., 2011, Chen et al., 2012). Unlike many other techniques, Boongoen et al (Boongoen et al., 2011) proposed a technique which is applicable with various clustering techniques rather than just K-Means (Niu et al., 2008). The technique first finds the k-nearest records of a record and then finds the weight of an attribute with respect to the nearest records. The attribute may have different weights for different records.

Instead of using k-nearest records, many existing techniques rely on an initial set of clusters (or nearest records) for estimating attribute weights through the calculation of the variation of the attribute values within each cluster. If the initial clustering quality is bad then the attribute weight estimation is also likely to be bad. If the initial clustering quality is good then it appears to be arguably unnecessary to find attribute weights and again find the clusters. Additionally, there are often some attributes which are not relevant to the dataset and these can cause noise in the initial clustering and in the weight estimation. These attributes need to be identified and removed before estimating attribute weights.

## 3 AWST: Our Novel Attribute Weight Selection Technique

AWST estimates the significance/weight of an attribute according to the ability of the attribute to cluster the records. That is, the attributes that have a greater ability to cluster the records are given a higher weight. Clustering ability is tested based on the well-known evaluation criterion called the Xie-Beni (XB) Index (Mukhopadhyay and Maulik, 2009, Chou et al., 2004). Note that our technique does not depend on the class attribute of the dataset as we realize that datasets used for clustering generally do not have any class attributes. We

now discuss the basic steps which we use in the AWST algorithm as follows:

Step 1: Divide the dataset into clusters based on the domain values of an attribute. Numerical attributes are discretized automatically using the proposed approach;

Step 2: Calculate the XB of the clusters; and

Step 3: Calculate attribute weights for all attributes based on the XB values.

### Step 1: Divide the dataset into clusters based on the domain values of an attribute

In order to calculate the weight of an attribute, AWST divides the dataset into mutually exclusive horizontal segments based on the values of the attribute where within a segment all records have the same value for the attribute. If the attribute is categorical then all records of a segment will have the same categorical value for the attribute. If the attribute is numerical then we first discretize the attribute and divide the dataset in segments in such a way that all the records within a segment have the same category of the attribute. The dataset is divided into segments for each attribute one by one. If there are  $|A|$  attributes in a dataset it is divided into segments  $|A|$  times, where each time it is divided based on a different attribute.

The values of a categorical attribute are clearly categorized in the dataset. However, finding categories for a numerical attribute may not be so intuitive. If we divide the values into B categories (where B could be the square root of the domain size or any other constant number) with equal ranges then we do not take into account natural properties such as the distribution of the values and instead we discretize them artificially.

Adjacent numerical values are typically similar to each other, but the boundaries of the categories need to be determined carefully considering the distribution of the values so that a category represents a concentration of values that can essentially be thought of as a category by itself. Our initial empirical results also indicate that categorizing the values of a numerical attribute using a predetermined number (B) of equal ranges did not give us a sensible result to indicate the clustering ability of a numerical attribute. The initial empirical results are presented in Section 4.

Therefore, for a numerical attribute  $X = [x_1, x_n]$  (having domain size =  $n$ ), AWST discretize the values of the attribute using a novel approach, which is inspired by intelligent K-Means (IKMeans) (Cordeiro de Amorim and Mirkin, 2012). Note that IKMeans originally dealt with all the attributes of all records aiming to find initial seeds, whereas we deal with the values of a single attribute and we aim to find natural categories for the values of the attribute instead of the seeds of the records.

We find the average of all values and call it the grand average, which is  $\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ . We also find the value  $x_j$  having the maximum distance from the grand average as follows:

$$x_j = x_a: \left| x_a - \frac{\sum_{i=1}^n x_i}{n} \right| > \left| x_b - \frac{\sum_{i=1}^n x_i}{n} \right|; \forall b \neq a \quad (1)$$

All values are then divided into two partitions  $P_1$  and  $P_2$ , where in one partition,  $P_1$ , we have values that are closer to the most distant value from the grand average

and in the other partition,  $P_2$ , we have the remaining values. In equation 2, the value  $x_l$  is closer to  $x_j$  than the grand average  $\bar{x}$  and in equation 3, the value  $x_p$  is closer to the grand average  $\bar{x}$  than  $x_j$ .

$$P_1 = \left\{ x_l: |x_j - x_l| \leq \left| \frac{\sum_{i=1}^n x_i}{n} - x_l \right| \right\} \quad (2)$$

$$P_2 = \left\{ x_p: |x_j - x_p| > \left| \frac{\sum_{i=1}^n x_i}{n} - x_p \right| \right\} \quad (3)$$

We then go to the next iteration and find the average (called partition average,  $P_a$ ) of partition  $P_1$  having all values closer to the most distant value than the grand average.

$$P_a = \frac{\sum_{i=1}^{|P_1|} x_i}{|P_1|}; \forall x_i \in P_1 \quad (4)$$

Two partitions are then again created using the partition average and the grand average, where in one partition we have all records closer to the new partition average  $P_a$  than the grand average, and in the other partition we have the remaining values. We continue the iterations while the difference between two consecutive partition average values is greater than a small default threshold  $\epsilon$ .

We then remove the partition with values closer to the partition average than the grand average. Among the remaining values we next choose a new most distant value from the same grand average. The whole process is then repeated for the remaining values. We continue this until the partition contains more than a user defined number of threshold  $t$ . The partitions are finally used as the categories of the numerical attribute.

We argue that a partition represents a natural concentration of values since in our approach a partition average stabilises when there is a reasonable gap between the values belonging to two partitions.

### Step 2: Calculate XB of the clusters

The dataset is then divided into mutually exclusive segments/clusters where, in a cluster, all records have the same value for the attribute (if it is a categorical attribute) or same category of the attribute (if it is a numerical attribute). The Xie-Beni Index (XB) (Chou et al., 2004, Mukhopadhyay and Maulik, 2009) of the clusters for an attribute is then calculated. Note that while calculating XB, we need to calculate the distance between the records and the seeds. For calculating the distance between records, we use similarity between categorical values (Giggins and Brankovic, 2012) and normalized numerical values. We repeat step 1 and step 2 in order to calculate the XB Index for all attributes (see the algorithm as shown in Figure 1). Note that during the XB calculation, the sequence or order of the attributes of the dataset does not have any impact.

### Step 3: Calculate attribute weights for all attributes based on XB values

We then calculate the attribute weights based on the XB values for the attributes. An attribute having a lower XB has higher cluster ability than attributes having higher XBs. We calculate normalized XB for the  $i^{\text{th}}$  attribute.

$$N(XB_i) = \frac{XB_i}{\sum_{a=1}^{|A|} XB_a} \quad (5)$$

Where  $XB_i$  is the XB value of the  $i^{\text{th}}$  attribute and  $|A|$  is the total number of attributes in the dataset, we then calculate the weight of the  $i^{\text{th}}$  attribute as follows:

$$W_i = 1 - N(XB_i) \quad (6)$$

The user can assign the  $W_i$  values (see Eq. 6) as the weights of the attributes in clustering. Alternatively, they can sort the attributes according to the XB values and assign higher weights as they like on attributes having lower XB values than attributes having higher XB values. We present the algorithm for AWST in Figure 1 above.

```

Algorithm: Attribute Weight Selection Technique (AWST)
Input: A dataset D having A number of attributes, a user defined
number of values t and a user defined threshold ε
Output: Weights of the attributes W
-----
Set XB ← ∅ /*XB stores Xie-Beni (XB) Index of each attribute where XBi
is the XB of ith attribute */
Set W ← ∅ /*W is a set of weights where Wi is the weight of the ith
attribute */
FOR (i=1 to |A|) Do
  IF Ai is numerical attribute DO
    Ai ← Categorize attribute Ai /*Categorize numerical attribute Ai */
    S ← FindCluster (D, Ai) /*divide the dataset into clusters based on
the categories of the attribute */
    XBi ← CalculateXB (D, S) /*calculate the Xie-Beni Index of the
clusters */
    XB ← XB ∪ XBi
  END IF
  ELSE
    S ← FindCluster (D, Ai) /*divide the dataset into clusters based on
the domain values of the attribute s */
    XBi ← CalculateXB (D, S) /*calculate the Xie-Beni (XB) Index
of the clusters */
    XB ← XB ∪ XBi
  END ELSE
END FOR
FOR (i=1 to |A|) Do
  N(XB) ←  $\frac{XB_i}{\sum_{j=1}^{|A|} XB_j}$  /*normalized the Xie-Beni value for each attribute */
  Wi ← 1 - N(XB)
  W ← W ∪ Wi
END FOR
Return W

```

Figure 1: Algorithm for AWST

#### 4 An Empirical Analysis on Discretization

We perform an empirical analysis to evaluate the quality of discretization by using our approach and another approach that discretize using the square root of the domain size of the values of the numerical attributes. We discretize the values of each numerical attribute by using our approach as discussed in Step 1 of Section 3. We next calculate the weight of each attribute by AWST. Based on the attribute weights, we next divide the attributes of the dataset into three equal categories, namely best attributes (BA), medium attributes (MA) and worst attributes (WA). We then assign equal weights (0.4) to the BAs for clustering the records using CRUDAW-F (Rahman, 2014). The clusters produced by CRUDAW-F are

evaluated in terms of silhouette coefficient. The silhouette coefficient values of CRUDAW-F in the PID, CA and CMC datasets that we obtained from the UCI machine learning repository (Bache and Lichman, 2013) are presented in Table 2.

We next discretize the values of the numerical attributes by using the square root of the domain size (SRDS) of the numerical attribute and calculate the weight of each attribute by AWST. We next divide the dataset into three categories in the same way that we did above and apply CRUDAW-F by using 0.4 weights for the BAs. The clusters produced by CRUDAW-F are also evaluated in terms of silhouette coefficient. In the PID, CA and CMC datasets, the silhouette coefficient values of CRUDAW-F are presented in Table 2. From Table 2, by using our discretization approach, the silhouette coefficient value of CRUDAW-F is better than the silhouette coefficient value of CRUDAW-F by using the discretization by the square root of the domain size (SRDS).

Datasets	Silhouette coefficient (higher the better)	
	Discretization using our approach (OA)	Discretization using square root of the domain size of a numerical attribute (SRDS)
Pima Indian Diabetes (PID)	0.2706	0.1910
Credit Approval (CA)	0.5284	0.2372
Contraceptive Method Choice (CMC)	0.6607	0.3612

Table 2: The Silhouette coefficient (SC) of CRUDAW-F with the discretization by our approach and square root over of domain size

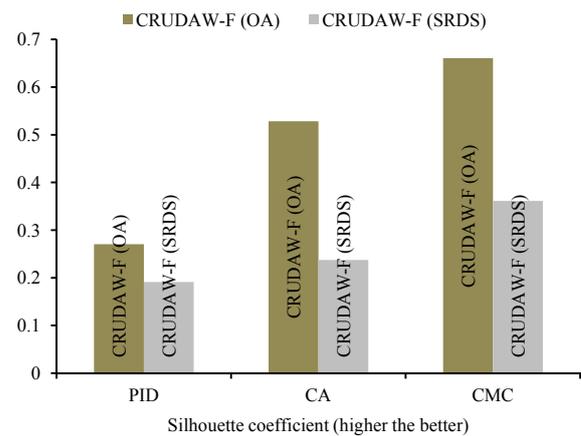


Figure 2: The Silhouette coefficient of CRUDAW-F with the discretization by our approach (OA) and square root over of domain size (SRDS)

For both discretization approaches, we also present the silhouette coefficient values of CRUDAW-F in Figure 2. From Figure 2, we see that our approach for discretization performs better than the discretization by SRDS.

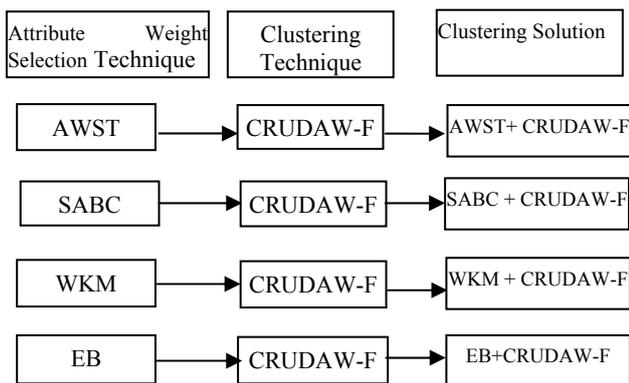
## 5 Experimental Results and Discussion

We compare the performance of our proposed attribute weight selection technique called AWST with three other existing attribute weight selection techniques namely SABC (Ahmad and Dey, 2007), W-K-Means (WKM) (Hung et al., 2011) and the entropy-based (EB) approach (Rahman and Islam, 2012, Rahman, 2014, Rahman et al., 2015).

SABC is a clustering technique that uses its own attribute weight selection method to assign weights on the attributes prior to clustering. In this study we implement the attribute weight selection method of SABC. Once the weights are selected we use the weights in an existing technique called CRUDAW-F (Rahman and Islam, 2012, Rahman, 2014) for clustering the records, as illustrated in Figure 3. Note that CRUDAW-F uses a weight selection approach for first selecting the weights of attributes and then using them for clustering records. CRUDAW-F is a Fuzzy C-Means based clustering technique where it requires weights of attributes for clustering. In our experiment we replace the original weight selection approach of CRUDAW-F by the weight selection approach of SABC. In Figure 3 we refer to this arrangement as “SABC+CRUDAW-F”.

Similarly, WKM is also a clustering technique that uses its own attribute weight selection technique to first compute the weights of the attributes of a dataset. It then uses the weights for clustering the records. In our experimentation, we only implement the attribute weight selection approach of WKM to compute the weights of the attributes. The weights are then fed into CRUDAW-F for clustering the records. In Figure 3 we refer to this arrangement as “WKM+CRUDAW-F”.

EB computes the attribute weights by using the entropy of each attribute of a dataset (Rahman and Islam, 2012, Rahman, 2014). We then feed these weights on the attributes into CRUDAW-F in order to get the clustering result. In Figure 3 this arrangement is called “EB+CRUDAW-F”.



**Figure 3: Interaction between an attribute weight selection technique and CRUDAW-F to produce clustering solution**

Finally, we use our proposed attribute weight selection technique called AWST and then feed the weights into CRUDAW-F to get the final clustering result. This arrangement has been called as “AWST+CRUDAW-F” in Figure 3. Our main goal in the experiments is to use the

same clustering technique (which is CRUDAW-F) for different weight selection methods so that we can compare the performance of the weight selection methods.

### The Datasets

We use nine natural datasets, namely Mushroom (MR), Blood Transfusion (BT), Credit Approval, (CA) Breast Cancer (BC), Pima Indian Diabetes (PID), Liver Disorders (LD), Contraceptive Method Choice (CMC), Chess and Adult. All of these datasets were available at the UCI Machine Learning Repository (Bache and Lichman, 2013). A brief introduction to the datasets is presented in Table 3.

We first remove all records with missing values. After removing these records, the MR, CA, Adult and BC datasets had 5644, 653, 30162 and 277 records respectively. We also remove the class attributes from the datasets before we apply the clustering techniques to them.

Datasets	Records with any missing values	Records without any missing values	No. of categorical attributes	No. of numerical attributes	Class Size
Mushroom (MR)	8124	5644	22	0	2
Blood Transfusion (BT)	748	748	0	4	2
Credit Approval (CA)	690	653	9	6	2
Breast Cancer (BC)	286	277	9	0	2
Pima Indian Diabetes (PID)	768	768	0	8	2
Liver Disorders (LD)	345	345	0	6	2
Contraceptive Method Choice (CMC)	1473	1473	7	2	3
Chess	28056	28056	3	3	18
Adult	32561	30162	8	6	2

**Table 3: A brief introduction to the datasets**

### The Parameters Used in the Experiments

In our proposed discretization approach, we use two user defined parameters: 1) the difference between two consecutive partition averages  $\epsilon$ ; and 2) the number of required values around a grand average  $t$ . In the experiments we consider the value of  $\epsilon = 0.00005$  and  $t = 1$ . The number of iterations for CRUDAW-F is considered as 50 that is mentioned in the original study (Rahman, 2014).

### The Experimental Results

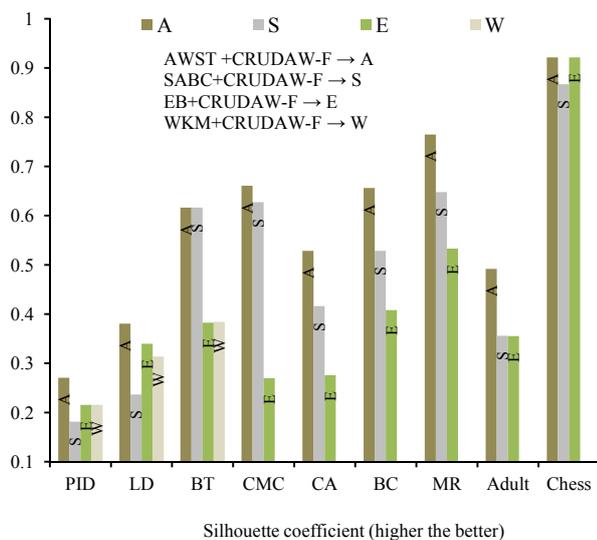
We first calculate the attribute weights using our proposed AWST. Based on the attribute weights obtained by AWST, we next divide the attributes of the datasets into three equal categories, namely best attributes (BA), medium attributes (MA) and worst attributes (WA). We then assign equal weights (0.4) to the BAs for clustering the records using CRUDAW-F (Rahman, 2014). Similarly, we also calculate attribute weights using SABC and assign equal weights (0.4) to the BAs (according to SABC) for clustering the records using CRUDAW-F (Rahman, 2014). We also repeat this process for WKM and EB of finding attribute weights and assigning 0.4 weights to the BA attributes.

So CRUDAW-F clusters the records using the 0.4 weights of the BA attributes four times. The first time the BA attributes were chosen using AWST. Second time the BA attributes were chosen using SABC. Similarly in the third and fourth times the BA attributes were chosen using WKM and EB respectively. Finally, the clustering quality of each of the four CRUDAW-F runs is evaluated using silhouette coefficient. The clustering result producing the best silhouette coefficient indicates the best selection of the attribute weights. Note that, the clustering results produced by CRUDAW-F based on each attribute weight selection techniques are deterministic.

While exploring attribute weights automatically, AWST discretizes the values of numerical attributes using the novel approach explained in Section 3. Although SABC and EB can identify attribute weights automatically, they do not have any techniques for the categorisation of numerical values.

Datasets	AWST + CRUDAW-F	SABC + CRUDAW-F	EB + CRUDAW-F	WKM + CRUDAW-F
PID	<b>0.2706</b>	0.1814	0.2156	0.2156
LD	<b>0.3806</b>	0.2365	0.3398	0.3137
BT	<b>0.6163</b>	<b>0.6163</b>	0.3826	0.384
CMC	<b>0.6607</b>	0.6273	0.2697	NA
CA	<b>0.5284</b>	0.4164	0.2758	NA
BC	<b>0.6562</b>	0.5286	0.4083	NA
MR	<b>0.7649</b>	0.6478	0.5329	NA
Adult	<b>0.4917</b>	0.356	0.3554	NA
Chess	<b>0.9215</b>	0.867	<b>0.9215</b>	NA

**Table 4: The Silhouette Coefficient of CRUDAW-F based on each attribute weight selection techniques on nine datasets**



**Figure 4: The Silhouette Coefficient of CRUDAW-F based on each attribute weight selection technique on nine datasets**

In order to favour SABC and EB, we use our discretization approach for them. Therefore we use the same discretization approach used by AWST for the SABC and EB techniques, basically to favour them and make the experiment a tough evaluation of AWST. The WKM technique was applicable to numerical attributes only. Therefore, we could not evaluate WKM for the datasets having any categorical attributes. The silhouette coefficient (the higher the better) results are presented in Table 4 and Figure 4 for all datasets. From Table 4 and Figure 4 we can see that the performance of AWST is better than the existing techniques in all datasets.

**Complexity and Execution Time of the Techniques**

We now calculate the complexity of AWST. The overall complexity of AWST is  $O(nm^2)$  whereas the overall complexity of WKM is  $O(nm)$  (Hung et al., 2011) and the overall complexity of SABC  $O(nm^2 + m^2S^3)$  (Ahmad and Dey, 2007), where  $n$  is the number of records,  $m$  is the number of attributes, and  $S$  is the average number of distinct categorical values in a dataset.

We also calculate the total execution time required by CRUDAW-F including the weight selection technique (see Table 5). Table 5 shows that for the PID dataset CRUDAW-F required 0.158 seconds when the attribute weights are determined by AWST. Similarly, for the same dataset, CRUDAW-F required 0.626 seconds when the attribute weights were determined by SABC. For other datasets, the execution time of the techniques can be learnt from Table 5. We use a shared computer system with 4x8 core Intel E7-8837 Xeon processors, 256 GB of RAM and 23 TB of disk storage.

Datasets	AWST + CRUDAW-F	SABC + CRUDAW-F	EB + CRUDAW-F	WKM + CRUDAW-F
PID	0.158	0.626	0.431	0.041
BT	0.048	0.385	0.315	0.014
LD	0.046	0.299	0.228	0.009
BC	0.105	0.037	0.005	NA
CA	0.395	0.537	0.343	NA
CMC	0.234	0.394	0.296	NA
MR	2.547	2.474	0.084	NA
Adult	25.918	116.197	73.601	NA
Chess	1.920	30.621	29.333	NA

**Table 5: The execution time (in seconds) of the techniques for all datasets**

**6 Conclusion**

In this paper, we present a novel attribute weight selection technique called AWST. In AWST we discretize the numerical attribute values using our novel approach which was inspired by IKMeans (Cordeiro de Amorim and Mirkin, 2012). AWST calculates the clustering ability of an attribute through the Xie-Beni Index (XB) of the clusters obtained by the categories of the attribute. However, to find the clustering ability of an

attribute, any other internal cluster evaluation criteria such as the Davies-Bouldin Index or Dunn Index could be used (Dunn, 1974, Davies and Bouldin, 1979).

We experimentally compare the performance of AWST with the performances of the SABC, WKM and EB techniques. In the experiments, we select the best attributes (BA) using each technique. We next apply CRUDAW-F (Rahman, 2014) to the datasets by considering the best attributes obtained by each technique separately. Based on each technique, we produce the clusters using CRUDAW-F and calculate the silhouette coefficient of the clusters. The experimental results indicate the superiority of AWST over the existing techniques in all nine datasets.

One of the advantages of AWST is that it requires less execution time when compare with many existing attribute weight selection techniques. The performance of AWST is also shown to be better than many existing attribute weight selection techniques.

## 7 References

- Adderley, R., Townsley, M. & Bond, J. 2007. Use of data mining techniques to model crime scene investigator performance. *Knowledge-Based Systems*, 20, 170-176.
- Ahmad, A. & Dey, L. 2007. A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, 63, 503-527.
- Bache, K. & Lichman, M. 2013. UCI Machine Learning Repository University of California, Irvine, School of Information and Computer Sciences.
- Bai, L., Liang, J., Dang, C. & Cao, F. 2011. A novel attribute weighting algorithm for clustering high-dimensional categorical data. *Pattern Recognition*, 44, 2843-2861.
- Boongoen, T., Changjing, S., Iam-On, N. & Qiang, S. 2011. Extending Data Reliability Measure to a Filter Approach for Soft Subspace Clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41, 1705-1714.
- Chan, E. Y., Ching, W. K., Ng, M. K. & Huang, Z. 2004. An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern Recognition*, 37, 943-952.
- Chan, K. Y., Kwong, C. K. & Hu, B. Q. 2012. Market segmentation and ideal point identification for new product design using fuzzy data compression and fuzzy clustering methods. *Applied Soft Computing*, 12, 1371-1378.
- Chatzis, S. P. 2011. A fuzzy c-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional. *Expert Systems with Applications*, 38, 8684-8689.
- Chen, X., Ye, Y., Xu, X. & Huang, Z. 2012. A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition*, 45, 434-446.
- Chou, C. H., Su, M. C. & Lai, E. 2004. A new cluster validity measure and its application to image compression. *Pattern Analysis and Applications*, 7, 205-220.
- Cordeiro De Amorim, R. & Mirkin, B. 2012. Minkowski metric, feature weighting and anomalous cluster initializing in K-Means clustering. *Pattern Recognition*, 45, 1061-1075.
- Davies, D. L. & Bouldin, D. W. 1979. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Dunn, J. 1974. Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*.
- Fan, J., Han, M. & Wang, J. 2009. Single point iterative weighted fuzzy C-means clustering algorithm for remote sensing image segmentation. *Pattern Recognition*, 42, 2527-2540.
- Gançarski, P., Blansch e, A. & Wania, A. 2008. Comparison between two coevolutionary feature weighting algorithms in clustering. *Pattern Recognition*, 41, 983-994.
- Giggins, H. & Brankovic, L. VICUS - A Noise Addition Technique for Categorical Data. *Proc. Data Mining and Analytics 2012 (AusDM 2012)*, 2012 Sydney, Australia. ACS, CRPIT 134: 139 - 148.
- Han, J. & Kamber, M. 2006. *Data Mining Concepts and Techniques*, San Francisco, Morgan Kaufmann.
- He, Z., Xu, X. & Deng, S. 2011. Attribute value weighting in k-modes clustering. *Expert Systems with Applications*, 38, 15365-15369.
- Huang, Z. 1998. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Mining and Knowledge Discovery*.
- Huang, Z., Ng, M. K., Hongqiang, R. & Zichen, L. 2005. Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 657-668.
- Hung, W.-L., Chang, Y.-C. & Stanley Lee, E. 2011. Weight selection in W-K-means algorithm with an application in color image segmentation. *Computers and Mathematics with Applications*, 62, 668-676.
- Islam, M. Z. & Brankovic, L. 2011. Privacy preserving data mining: A noise addition framework using a novel clustering technique. *Knowledge-Based Systems*, 24, 1214-1223.
- Lee, M. & Pedrycz, W. 2009. The fuzzy C-means algorithm with fuzzy P-mode prototypes for clustering objects having mixed features. *Fuzzy Sets and Systems*, 160, 3590-3600.
- Li, S.-T., Kuo, S.-C. & Tsai, F.-C. 2010. An intelligent decision-support model using FSOM and rule extraction for crime prevention. *Expert Systems with Applications*, 37, 7108-7119.
- Mukhopadhyay, A. & Maulik, U. 2009. Towards improving fuzzy clustering using support vector machine: Application to gene expression data. *Pattern Recognition*, 42, 2744-2763.

- Niu, K., Zhang, S. & Chen, J. 2008. Subspace clustering through attribute clustering. *Frontiers of Electrical and Electronic Engineering in China*, 3, 44-48.
- Oatley, G. C. & Ewart, B. W. 2003. Crimes analysis software: 'pins in maps', clustering and Bayes net prediction. *Expert Systems with Applications*, 25, 569-588.
- Pirim, H., Eksioglu, B., Perkins, A. D. & Yuceer, C. 2012. Clustering of high throughput gene expression data. *Computers & Operations Research*, 39, 3046-3061.
- Rahman, M. A. 2014. *Automatic Selection of High Quality Initial Seeds for Generating High Quality Clusters without Requiring any User Inputs*. PhD thesis in Computer Science, School of Computing and Mathematics, Charles Sturt University, Australia.
- Rahman, M. A. & Islam, M. Z. Seed-Detective: A Novel Clustering Technique Using High Quality Seed for K-Means on Categorical and Numerical Attributes *Proc. Ninth Australasian Data Mining Conference (AusDM 11)*, 2011, Ballarat, Australia, ACS, CRPIT 121: 211-220
- Rahman, M. A. & Islam, M. Z. CRUDAW: A Novel Fuzzy Technique for Clustering Records Following User Defined Attribute Weights. *Proc. Tenth Australasian Data Mining Conference (AusDM 2012)*, 2012, Sydney, Australia, ACS, CRPIT 134: 27 - 42.
- Rahman, M. A. & Islam, M. Z. 2014. A Hybrid Clustering Technique Combining a Novel Genetic Algorithm with K-Means. *Knowledge-Based Systems*, 71, 345-365.
- Rahman, M. A., Islam, M. Z. & Bossomaier, T. DenClust: A Density Based Seed Selection Approach for K-Means. *Proc. 13th International Conference on Artificial Intelligence and Soft Computing (ICAISC 2014)*, 2014, Poland.
- Rahman, M. A., Islam, M. Z. & Bossomaier, T. 2015. ModEx and Seed-Detective: Two Novel Techniques for High Quality Clustering by using Good Initial Seeds in K-Means. *Journal of King Saud University-Computer and Information Sciences*, 27, 113-128.
- Redmond, S. J. & Heneghan, C. 2007. A method for initialising the K-means clustering algorithm using kd-trees. *Pattern Recognition Letters*, 28, 965-973.
- Sun, J., Chen, W., Fang, W., Wun, X. & Xu, W. 2012. Gene expression data analysis with the clustering method based on an improved quantum-behaved Particle Swarm Optimization. *Engineering Applications of Artificial Intelligence*, 25, 376-391.
- Tan, P.-N., Steinbach, M. & Kumar, V. 2005. *Introduction to Data Mining*, Pearson Addison Wesley.
- Zhao, P. & Zhang, C.-Q. 2011. A new clustering method and its application in social networks. *Pattern Recognition Letters*, 32, 2109-2118.