Nat Comput
DOI 10.1007/s11047-015-9492-7

Software cost estimation based on modified *K*-Modes clustering Algorithm

Partha Sarathi Bishnu · Vandana Bhattacherjee

Abstract Unsupervised technique like clustering may be used for software cost estimation in situations where parametric models are difficult to develop. This paper presents a software cost estimation model based on a modified *K*-Modes clustering algorithm. The aims of this paper are: first, the modified *K*-Modes clustering which is an enhancement over the simple *K*-Modes algorithm using a proper dissimilarity measure for mixed data types, is presented and second, the proposed *K*-Modes algorithm is applied for software cost estimation. We have compared our modified *K*-Modes algorithm with existing algorithms on different software cost estimation datasets, and results showed the effectiveness of our proposed algorithm.

Keywords Data mining · Clustering · Software cost estimation · *K*-Modes clustering

1 Introduction

To overcome the drawback of partitional clustering techniques like *K*-Means (Han and Kamber 2007), *K*-Medoids (Han and Kamber 2007) clustering algorithms which are popular clustering algorithms for non categorical datasets, Prof. Huang (1997, 1998) suggested *K*-Modes clustering algorithm to handle categorical data. Software cost estimation (Papatheocharous and Andreou 2009), bioinformatics (Manganaro et al. 2005), and computer networks (Andreopoulos et al. 2005) are some of the application areas for the *K*-Modes clustering algorithm.

K-Modes algorithm is simple and widely used clustering technique for non numeric datasets but there are some issues which directly affect the quality of the clusters: first, the initial selection of the cluster centers (Sun et al. 2002; Bai et al. 2011; Cao et al. 2009; Huang 1997; Wu et al. 2007), second, the similarity or dissimilarity measures between the two non numeric data (Cao et al. 2012; He et al. 2005; Ng et al. 2007), third, proper handling of the boundary data (Huang and Ng 1999; Bishnu and Bhattacherjee 2013) and finally outlier detection (Bishnu and Bhattacherjee 2013). Researchers from the soft computing fields have contributed lots to handle the issues of *K*-Modes clustering (Gan et al. 2007; Aranganayagi and Thangavel 2009; Omar et al. 2012; Gan et al. 2009; Aroba et al. 2008).

In this paper we focus on the problem of software cost estimation. Software cost estimation is an important activity in software development as it is crucial for better project planning, monitoring and control (Mittas and Angelis 2013). Stress is being laid on the importance of improving estimation accuracy and techniques and over the last few decades several models have been proposed for this purpose. In recent years, models based on soft computing and data mining techniques are being built (Papatheocharous and Andreou 2009; Keung 2009; Cuadrado-Gallego et al. 2006).

The major aim of this research paper is to develop a software cost estimation model and the software attributes which serve as input to this model are of mixed nature i.e. categorical as well as numeric. Keeping the mixed nature of software attributes in mind, we establish the main objectives of this paper as follows: first, a combined distance measure is proposed for mixed attribute which is adapted

P. S. Bishnu (✉) · V. Bhattacherjee
Birla Institute of Technology, Ranchi 834001, India
e-mail: psbishnu@gmail.com

V. Bhattacherjee
e-mail: vbhattacharya@bitmesra.ac.in

Published online: 08 February 2015

from the distance measure of Bishnu and Bhattacherjee (2013) and Han and Kamber (2007) to handle the second issue of *K*-Modes clustering algorithm. Second, the modified *K*-Modes algorithm is applied for developing the software cost estimation model.

We have compared our technique with four existing *K*-Modes clustering techniques suggested by Huang (1998) (KM1) for simple *K*-Modes, by He et al. (2005) (KM2) for modified distance calculation for *K*-Modes, by Huang and Ng (1999) (KM3) for fuzzy *K*-Modes, and by Papatheocharous and Andreou (2009) (KM4), for entropy based *K*-Modes, with the help of real datasets from promise datasets (http://promise.site.uottawa.ca/SERepository) to demonstrate the applicability of *K*-Modes clustering algorithm on software cost estimation.

The outline of this paper is as follows: in Sect. 2 we present related work, in Sect. 3 we describe an approriate dissimilarity measure for mixed attribute data and the modified *K*-Modes algorithm. In Sect. 4 we explain the experiments which we carried out. In Sect. 5 we present the results and analysis and this is followed by conclusions in Sect. 6.

2 Related work

In this section we discuss the related work for both software cost estimation and *K*-Modes clustering algorithm. In the field of software cost estimation, Papatheocharous and Andreou (2009) have proposed an estimation approach which attempts to cluster empirical non-homogeneous project data sample using entropy based *K*-Modes clustering algorithm. They identify groups of projects based on similarity in terms of cost attributes descriptors. These descriptive characters are then used to classify a new project in a certain cluster. Subsequently, the projects belonging to the cluster are used to provide an estimate for the new project based on effort prediction intervals of minimum width. Keung (2009) provided an overview of software cost estimation using analogy. The author discusses analogy based systems such as ESTOR, ACE and ANGEL and also the deficiencies of analogy based systems like intolerance of noise, intolerance of irrelevant features, sensitivity to the choice of the algorithms, similarity function etc. The author suggests that using analogy would be ideal in situations where relevant and completed project data consistently measured are available.

Cuadrado-Gallego et al. (2006) and Cuadrado-Gallego and Sicilia (2007) suggested the use of segmented models for software cost estimation. Clustering algorithms are used to determine the classes of projects that a database records at a particular moment in time. After this EM clustering algorithm is used to estimate the software development effort. This work has been validated and found to be appropriate in obtaining segmented parameter estimation models with good adjustment properties. However, crisp clustering has inherent problems and a project under estimation may not be assigned to a segment in a sharp way. To overcome this problem, Aroba et al. (2008) recommended the use of segmented models for software cost estimation where the project space has been divided into fuzzy clusters. The use of fuzzy clustering allows obtaining different mathematical models for each cluster and also allows the items of a project database to contribute to more than one cluster. The validation of this model on the same dataset yields better adjustment figure than its crisp counterparts. Stamelos et al. (2003) proposed an approach for software development cost estimation based on the characterization of the software to be developed in terms of project and environment attributes and comparison with some similar completed projects recovered from historical database. Their best estimation strategy predicted effort with a mean estimation error of 24 % with respect to the actual effort. Lefteris et al. (2001) developed a multi organizational software cost estimation model by analyzing the software cost data collected by the International software benchmarking standards group. The generation of the model is based on categorical regression or regression with optimal scaling. This technique quantifies the qualitative attributes that appear frequently within such data and then uses the obtained scores as independent variables of a regression model. The model is validated by measuring certain indicators of accuracy. Jiang et al. (2012) applied case based reasoning to develop a software cost evaluation model. In this paper the driver factors in software cost evaluation model are found by grey association degree analysis. Then the similarity between the cases in the dataset and the software to be evaluated is taken as the Euclidean distance and the effort is estimated after similarity weighting. For further information on software cost estimation the interested reader may refer the follwing papers Arifoglu (1993), Lin and Tzeng (2010), Dejaeger et al. (2012), Benala et al. (2012) and Zhang et al. (2013).

The *K*-Modes clustering has been proposed by Huang (1997), in which he presented two algorithms which extend the *K*-Means algorithm to categorical domains and domains with mixed numeric and categorical values. The *K*-Modes algorithm uses a simple matching dissimilarity measure to deal with categorical objects, replaces the means of clusters with modes, and uses a frequency-based method to update modes in the clustering process to minimize the clustering cost function. With these extensions the *K*-Modes algorithm enables the clustering of categorical data in a fashion similar to *K*-Means. Further (Huang and Ng 1999) extended this concept to fuzzy *K*-Modes algorithm and presented the effectiveness of the algorithm with experimental results. Next, Ng et al. (2007) suggested a new updating formula of the *K*-Modes clustering algorithm with the new dissimilarity measure and the convergence of the algorithm under the optimization framework. In Papatheocharous and Andreou (2009) the entropy based fuzzy *K*-Modes clustering algorithm has been presented. In this paper authors suggested a new initialization technique for *K*-Modes clustering algorithm. First, the entropy based concept is applied to identify the initial cluster center and then *K*-Modes algorithm is applied with these identified cluster centers. For further information on *K*-Modes clustering algorithms the interested reader may refer the following papers: Sun et al. (2002), Bai et al. (2011), Cao et al. (2009), Wu et al. (2007), He et al. (2005), Gan et al. (2007), Aranganayagi and Thangavel (2009), Omar et al. (2012), Gan et al. (2009) and Aroba et al. (2008).

3 The modified *K*-Modes clustering algorithm for mixed attribute

3.1 *K*-Modes clustering algorithm

First, we describe the symbols used in the paper:

Description of symbols

O	Dataset
o	Data, $o \in O$
n	Number of data
D	Set of attributes
I	Attribute
h	Dimension of the data (categorical part)
d	Dimension of the data (categorical + numerical part)
a, b	Categorical values
u, v	Numerical values
K	Number of clusters
k	Cluster, $1 \leq k \leq K$
dis	Distance between two data (categorical data)
$dist$	Distance between two data (mixed data)
m_k	Mode of the kth cluster
M	Set of modes
C	Clusters
t	To denote iteration
tn	Number of test data
A_k, V_k	Set of categorical and numerical attributes
N_k	Number of data present in the kth cluster
α	Number of categorical attributes for which the values match
β	Number of categorical attributes for which the values do not match
RE	Relative error

Next, we discuss the simple *K*-Modes clustering algorithm for the sake of completeness (Huang 1998; Ng et al. 2007). A categorical dataset (O) is defined by a set (D) of attributes $I_1, I_2, \ldots, I_h$ where h is the dimension of the dataset. Each attribute I_e, $1 \leq e \leq h$ is described by a set of categorical values $a, b \in DOM(I_e)$ which are finite and unordered. For $a, b \in DOM(I_e)$, either $a = b$ or $a \neq b$. A data $o_i \in O$, where $1 \leq i \leq n$ and n is the number of data, can be represented as a conjunction of attribute value pairs $[I_1 = a_1] \wedge [I_2 = a_2] \wedge \cdots \wedge [I_h = a_h]$, where $a_e \in DOM(I_e)$. We use ε to represent the missing value. Moreover, if $o_i, o_j \in O$, and let $o_i = [a_1, a_2, \ldots, a_h]$ and $o_j = [b_1, b_2, \ldots, b_h]$ then we write $o_i = o_j$ if $a_e = b_e$, for $1 \leq e \leq h$ and the relation $o_i = o_j$ does not mean that they are the same data, but rather that the two data have equal values on attributes (I_e), where $1 \leq e \leq h$ (Ng et al. 2007).

Let $o_i, o_j \in O$, be two categorical data represented by $[a_1, a_2, \ldots, a_h]$ and $[b_1, b_2, \ldots, b_h]$, respectively. The distance (simple matching dissimilarity) between o_i and o_j is defined as $dis(o_i, o_j) = \sum_{e=1}^{h} \delta(a_e, b_e)$ where

$$\delta(a_e, b_e) = \begin{cases} 0: & a_e = b_e \\ 1: & a_e \neq b_e \end{cases} \tag{1}$$

In *K*-Modes the objective of clustering categorical dataset into $K(<<n)$ clusters is to search W and M that minimize the cost function $F(W, M)$, where

$$F(W, M) = \sum_{k=1}^{K} \sum_{i=1}^{n} w_{ki}\, dis(m_k, o_i), \tag{2}$$

subject to

$$w_{ki} \in \{0, 1\}, \quad 1 \leq k \leq K \text{ and } 1 \leq i \leq n, \tag{3}$$

$$\sum_{k=1}^{K} w_{ki} = 1, \quad 1 \leq i \leq n \tag{4}$$

and,

$$0 < \sum_{i=1}^{n} w_{ki} < n, \quad 1 \leq k \leq K, \tag{5}$$

where $W = [w_{ki}]$ is a k-by-n $\{0, 1\}$ matrix and $m_k \in M$, $1 \leq k \leq K$, is the kth cluster mode.

Now we give a formal algorithm for *K*-Modes algorithm as follows (Ng et al. 2007):

```
Algorithm 1: The K-Modes algorithm
Input: $O$, $k$
Output: $C$ (clusters)
Step1: randomly choose $K$ data from $O$ as initial modes $M$ and determine $W^1$ such that $F(W, M^1)$ is minimized. Set $t = 1$;
```

Step2: determine M^{t+1} such that $F(W^t, M^{t+1})$ is minimized. If $F(W^t, M^{t+1}) = F(W^t, M^t)$, then stop; otherwise go to step 3;
Step3: determine W^{t+1} such that $F(W^{t+1}, M^{t+1})$ is minimized. If $F(W^{t+1}, M^{t+1}) = F(W^t, M^{t+1})$, then stop; otherwise increment t by one and go to step 2;

To calculate the mode of mixed attribute data we use the following method: Let A_k and V_k be the set of categorical and numeric attributes of kth cluster and N_k be the number of data present in kth cluster. Let m_k be the (mixed) mode of kth cluster, where $m_k = \{m_{k1}, m_{k2}, \ldots, m_{kh}, m_{kh+1}, \ldots, m_{kd}\}$. Where $\{m_{k1}, m_{k2}, \ldots, m_{kh}\}$ is the set of categorical attributes and $\{m_{kh+1}\ , m_{kh+2}, \ldots, m_{kd}\}$ is the set of numerical attributes. To get the m_{kj}, $1 \le j \le h$ i.e. jth categorical attribute value of kth mode, we calculate number of occurrences of each distinct a_j out of the possible n_j values where $n_j = |DOM(I_j)|$ and pick up the maximum occurrence value as the value of m_{kj}. If two or more maximum values are there then choose first one (or any one). It is true for all the categorical attributes and for all the modes (Huang 1998). Next, to get the m_{kj}, $h + 1 \le j \le d$ i.e. the jth numeric attribute value of kth (mixed) mode, we calculate the average of the jth numeric attribute, the average being taken over non missing values. (Note: henceforth in this paper wherever mixed attributes are being referred, the terms mode and mixed mode may be considered to be synonyms.)

3.2 The dissimilarity measure for mixed attribute

A dissimilarity measure (*dist*) was proposed in Bishnu and Bhattacherjee (2013) for the *K*-Modes clustering algorithm for categorical data so as to minimize the cost function F_{BB} (W, M), where $F_{BB}(W, M) = \sum_{k=1}^{K} \sum_{i=1}^{n} w_{ki} dist(m_k, o_i)$ subject to conditions as in (3), (4), and (5). In this paper we have modified the similarity measure to accommodate numeric data in *K*-Modes clustering so that we can create a model for software cost estimation which consist of mixed data types i.e. both categorical and numeric data types.

Definition 1 Let $o_i, o_j \in O$, $1 \le i, j \le n$, $i \ne j$, n is the number of data, be two mixed data represented by $[a_1, a_2, \ldots, a_h, u_{h+1}, \ldots, u_d]$ and $[b_1, b_2, \ldots, b_h, v_{h+1}, \ldots, v_d]$, respectively where a_e, b_e, $1 \le e \le h$ are categorical attributes and u_{nu}, v_{nu}, $h + 1 \le nu \le d$ are the numeric attributes. Let $dist(o_i, o_j)$ be the distance between two objects. The distance is defined as follows:

$$dist(o_i, o_j) = dist_{text}(o_i, o_j) + dist_{numeric}(o_i, o_j) \tag{6}$$

where $dist_{text}(o_i, o_j)$ is the distance between corresponding categorical attributes of the data and $dist_{numeric}(o_i, o_j)$ is the distance between corresponding numeric attributes of the data.

The calculation of $dist_{text}(o_i, o_j)$ is as follows:

$$dist_{text}(o_i, o_j) = 1 - \frac{\alpha_{ij}}{\beta_{ij}}, \tag{7}$$

where

$$\alpha_{ij} = \sum_{e=1}^{n} \chi(a_e, b_e), \tag{8}$$

where,

$$\chi(a_e, b_e) = \begin{cases} 1: & a_e = b_e \\ 0: & a_e \ne b_e \end{cases} \tag{9}$$

and,

$$\beta_{ij} = 2h - \alpha_{ij}, \tag{10}$$

where h is the size of dimension of the categorical part of the data. Here, $\frac{\alpha_{ij}}{\beta_{ij}}$ measures the similarity between two objects as the degree of relative overlap between the two objects considering the categorical attributes. The α_{ij} counts the number of categorical attributes for which the values in o_i, o_j match while the β_{ij} counts those which do not match (Bishnu and Bhattacherjee 2013).

The calculation of $dist_{numeric}\ (o_i, o_j)$ is as follows:

$$dist_{numeric}\ (o_i, o_j) = \frac{\sum_{f=h+1}^{d} |u_{if} - v_{jf}|}{max_f - min_f}, \tag{11}$$

where, max_f is the maximum value of the fth numeric attribute and min_f is the minimum value of the fth numeric attribute (Han and Kamber 2007).

Theorem 1 *Let M be fixed and consider the problem:* $min_w\ F_{BB}\ (W, M)$, *subject to* (3) *to* (5). *The minimizer M is given by*

$$w_{ki} = \begin{cases} 1: & dist(m_k, o_i) \le dist(m_p, o_i), 1 \le k, p \le K, \\ 0: & otherwise. \end{cases}$$

Theorem 2 *Consider only the categorical domain and let* $m_k = [m_{k1}, m_{k2}, \ldots, m_{kh}]$ *be the mode (categorical part) of the kth,* $1 \le k \le K$ *cluster and the domain* ζ_{a_j} *of attributes* a_j *be* $\{a_j^1, a_j^2, \ldots, a_j^{n_j}\}$, $1 \le j \le h$ *and* n_j *is the cardinality of domain of* a_j. *Denote arbitrary object* o_i *by* $[a_{i1}, a_{i2}, \ldots, a_{ih}]$. *Then* $F_{BB}(W, M) = \sum_{k=1}^{K} \sum_{i=1}^{n} w_{ki} dist_{text}$ (m_k, o_i) *is minimized if and only if* $m_{kj} = a_j^r$, *where,* $a_j^r \in \zeta_{a_j}$ *satisfies:* $|\{w_{ki} | o_{ij} = a_j^r, w_{ki} = 1\}| \ge |\{w_{ki} | o_{ij} = a_j^t, w_{ki} = 1\}|$, $1 \le t \le n_j$, $1 \le j \le h$.

Proof For a given W, $F_{BB}(W, M) = \sum_{k=1}^{K} \sum_{i=1}^{n} w_{ki}$ $dist_{text}\ (m_k, o_i)\ = \sum_{k=1}^{K} \psi_k$. Note that all the inner sums ψ_k

of F_{BB} (W, M) are non negative and independent. Then minimizing $F_{BB}(W, M)$ is equivalent to minimizing each inner sum. By definition 1, when $m_{kj} = a_j^t$ we have $\psi_k = \sum_{i=1}^{n} w_{ki} dist_{text}\,(m_k, o_i) = \sum_{i=1}^{n} w_{ki}(1 - \frac{\alpha_{ki}}{\beta_{ki}}) = N_k - \sum_{i=1}^{n} \frac{\alpha_{ki}}{\beta_{ki}})$, where N_k is the number of data in *k*th cluster. Since N_k is nonnegative, ψ_k is minimized if $\frac{\alpha_{ki}}{\beta_{ki}}$ is maximized for $1 \leq i \leq n$. Since $\frac{\alpha_{ki}}{\beta_{ki}}$ is the similarity (or degree of relative overlap) between mode m_k and object o_i, it is maximized when $t = r$, i.e., $m_{kj} = a_r^j$ The result follows.

Theorem 2 holds even in the case of mixed attributes as explained below: consider a mixed mode $m_k = \{m_{k1}, m_{k2}, \ldots, m_{kh}, m_{kh+1}, \ldots, m_{kd}\}$, $1 \leq k \leq K$.

$F_{BB}(W, M) = \sum_{k=1}^{K} \sum_{i=1}^{n} w_{ki}\, dist(m_k, o_i) = \sum_{k=1}^{K} \sum_{i=1}^{n} w_{ki}(dist_{text}\quad(m_k, o_i)\quad + dist_{numeric}(m_k, o_i)) = \sum_{k=1}^{K} (\psi_k + \phi_k) = \sum_{k=1}^{K} (N_k - \sum_{i=1}^{n} \frac{\alpha_{ki}}{\beta_{ki}}) + \sum_{k=1}^{K} \phi_k$. Since N_k is non negative and so is ϕ_k (as defined in [11]), hence F_{BB} is minimized if $\frac{\alpha_{ki}}{\beta_{ki}}$ is maximized. The result follows.

Example 1 Suppose $o_1 = [a, b, c, f, g, 3]$, $m_1 = [a, d, c, f, h, 4]$, and $m_2 = [a, e, c, f, g, 5]$ then $dist_(o_1, m_1) = 1 - \frac{3}{2*5-3} = 0.5714$, $dist_{text}(o_1, m_2) = 1 - \frac{4}{2*5-4} = 0.3333$. Similarly $dist_{numeric}$ $(o_1, m_1) = |3 - 4|/10 = 0.1$ and $dist_{numeric}$ $(o_1, m_2) = |3 - 5|/10 = 0.2$ (assume $max_{f=6th}$ $-min_{f=6th} = 10$).

$dist(o_1, m_1) = dist_{text}(o_1, m_1) + dist_{numeric}\,(o_1, m_1)) = 0.5714 + 0.1 = 0.6714$. Next, $dist(o_1, m_2) = dist_{text}(o_1, m_2) + dist_{numeric}\,(o_1, m_2) = 0.333 + 0.2 = 0.5333$. As the distance from the data o_1 to mode m_2 is lesser, so the data o_1 should be in the second cluster.

Example 2 Consider a synthetic dataset consisting of software cost estimation data. The data are $o_1 = [vl, l, h, n, n, 100, 5.1]$, $o_2 = [h, l, h, n, h, 150, 3.5]$, $o_3 = [h, h, vh, h, h, 160, 3.8]$, $o_4 = [l, l, n, h, n, 120, 5.2]$, and $o_5 = [h, h, h, h, n, 170, 3.9]$, where, the attributes are programmer's capability, language experience, product complexity, required software reliability, use of software tools, lines of codes, and efforts in hours (while creating the model we ignore the last attribute) respectively. The categorical values are *vl*: very low, *l*: low, *h*: high, *n*: normal, *vh*: very high.

Randomly we select two data as initial cluster modes say o_1 and o_5 for constructing two clusters. Then the distance calculations with other data are $dist(o_1, o_2) = dist_{text}(o_1, o_2) + dist_{numeric}(o_1, o_2) = 1 - \frac{3}{2*5-3} + \frac{(|100-150|)}{170-100} = 1.28$. $dist(o_5, o_2) = dist_{text}(o_5, o_2) + dist_{numeric}(o_5, o_2) = 1 - \frac{2}{2*5-2} + \frac{(|170-150|)}{170-100} = 1.03$, $dist(o_1, o_3) = 1.85$, $dist(o_5, o_3) = 0.7142$, $dist(o_1, o_4) = 1.03$, $dist(o_5, o_3) = 1.46$. Since, $dist(o_1, o_2) > dist(o_5, o_2)$, hence, the data o_2 should be with initial cluster center o_5 to form a cluster. Similarly, $dist(o_1, o_3) > dist(o_5, o_3)$, so, data o_3 should be with o_5. Finally, $dist(o_1, o_4) < dist(o_5, 0_4)$, hence o_4 should be with o_1 to form a cluster. The two generated clusters (by initial cluster modes) are $c_1 = (o_1, o_4)$ and $c_2 = (o_2, o_3, o_5)$. Two new modes are $m_1 = [vl, l, h, n, n, 110]$ and $m_2 = [h, h, h, h, h, 160]$. For the next iteration calculate distances $dist(o_1, m_1)$, $dist(o_1, m_2)$, $dist(o_2, m_1)$, $dist(o_2, m_2), \ldots,$ $dist(o_5, m_1)$, $dist(o_5, m_2)$ to reassign the cluster number. After recreating the clusters reidentify the modes and the process will continue till convergence.

3.3 Modified *K*-Modes clustering algorithm

The steps of the modified *K*-Modes clustering algorithm are as follows:

Algorithm 2: Modified *K*-Modes algorithm
Input: *O*, *K*
Output: *C* (clusters)
Step1: Randomly select initial modes from given dataset *O*;
Step2: (re) assign cluster number to each data by calculating the distance (dissimilarity) between the data and the modes;
Step3: (re) identify the (mixed) modes;
Step4: repeat step 2 and 3 till convergence criteria is satisfied;

Explanation of our proposed K-Modes algorithm: A random selection of initial (mixed type) mode is done in the first step. In the next step, our distance measure (refer equation 6) is applied to assign the cluster number to its nearest mode. It should be noted that since the data are mixed types neither simple *K*-Modes (used for categorical data) nor simple *K*-Means (applied for numeric data) can be applied. In step 3 the mode is identified in the follwing manner. The maximum frequency approach is adopted to identify the categorical portion of the mode. For the numeric portion of the data, simple average over non missing values is computed. Finally, both the results are concatenated to get the final mode.

Time complexity: the time complexity of the step 1 is $O(1)$ and steps 2 and 3 is $O(ntKd)$, where n is the number of data, t is the number of iterations, K is the number of clusters, and d is the dimension of the data.

4 Experiments

4.1 Dataset

To show the effectiveness of our proposed algorithm on software cost prediction we use two real datasets CData1 (cocomonasa /software cost estimation) and CData2 (COCOMO NASA 2/Software cost estimation) from website http://promise.site.uottawa.ca/SERepository. The numbers of data are 60 and 93; the dimensions are 17 and 21 respectively. For CData1 the last column (attribute) shows the actual effort in person months and last but one column (attribute) is the lines of source code (numeric). Similarly, for CData2 the last column (attribute) shows the development effort in months and last but one column (attribute) shows the equivalent physical 1,000 lines of source code (numeric). Remaining attributes are categorical in nature. From both the datasets we have removed unique id and year of development which are irrelevant for the model construction. We eliminate attributes or data for any missing values. We use 60 % data for training the model and remaining 40 % data are used for testing the model.

4.2 Performance evaluation parameters

As evaluation parameter for point estimation, the mean magnitude of the relative error (*MMRE*) has been used while for interval estimation *hit ratio* is used. *MMRE* is computed from the absolute value of relative error, or $|RE_i|$, which is the relative value of the difference between the actual and estimated value and is given as: $|RE_i| = |\frac{(estimated_i - actual_i)}{actual_i}|$, $1 \leq i \leq tn$. The *MMRE* is defined as follows: $MMRE = (|RE_1| + |RE_2| + \cdots + |RE_{tn}|)/tn$, where tn is the size of the test dataset (Aroba et al. 2008). To compute the *hit ratio*, we first calculate the range of actual (or development) effort (in months) for each cluster by taking the maximum and minimum values. Now for each test data if the actual development effort falls within the range of the assigned cluster then it is a hit otherwise it is a miss. The *hit ratio* is $\frac{totalHit}{tn}$ (Papatheocharous and Andreou 2009). For a good estimation model the *MMRE* should be low and *hit ratio* should be high.

4.3 Experimental analysis

The steps for software cost estimation using modified *K*-Modes clustering are as follows: the training data are used to create the clusters using modified *K*-Modes clustering algorithm. When a new data (software) is to be estimated, it is assigned to the cluster with the nearest mixed mode. Now, either point estimation or interval estimation of the cost may be done. In point estimation the average of the software cost associated with the data in the target cluster gives the predicted cost of the new data. In interval estimation the range of the software cost associated with the data in the target cluster gives the predicted interval of the cost of the new data (software).

We have compared our technique KM5 with four existing techniques namely; KM1 by Huang (1997) which is the simple and initial *K*-Modes clustering algorithm; KM2 by He et al. (2005) where they suggested a new dissimilarity measure; KM3 (Fuzzy *K*-Mode) by Huang and Ng (1999) where the fuzziness was introduced. In this experiment we set the value of (fuzzy) $\alpha = 1.1$ as suggested by Huang and Ng (1999) and KM4 by Papatheocharous and Andreou (2009), Yao et al. (2000) entropy based fuzzy *K*-Modes clustering algorithm. In entropy based clustering algorithm we use entropy concept to identify the initial cluster centers and then apply *K*-Modes algorithm using these cluster centers. In entropy based clustering we have used two parameters γ percentage (a threshold value to declare a cluster to be a valid one) and β (a threshold value of similarity). The γ values for CData1 are 2, 3, 3, 3, and 3 % and for CData2 are 3, 3, 3, 3, and 3 % for the clusters from 2 to 6 respectively. Similarly the β values for CData1 are 0.71, 0.71, 0.73, 0.74, and 0.90 and for CData2 the β values are 0.70, 0.72, 0.73, 0.80 and 0.85 for the clusters 2–6 respectively. The KM5 is our proposed *K*-Modes algorithm. For software cost estimation we have modified all the four existing *K*-Modes algorithm. Since the existing algorithms can handle only categorical, data we have added the $dist_{numeric}$ concept to each of the existing *K*-Modes algorithms along with our modified *K*-Modes algorithm to handle categorical and numeric data simultaneously. All the algorithms were executed 100 times and the best results were reported. All the algorithms iterate 30 times (we set this value as convergence criteria). The algorithm has been executed to generate two to six clusters (please refer first column of the Tables 1, 2). The values of the evaluation parameters have been reported for the same. Efficiency is expressed in terms of execution time in seconds (s). The experiments were conducted on a PC with an Intel Celeron processor, (1.30 GHz), and 256 MB RAM running the Windows XP operating system. All the *K*-Modes algorithms have been coded in Octave-3.2.4. GNU Octave (http://www.gnu.org/software/octave) is a high level interpreted programming language for numerical computation.

5 Results and analysis

Experiments were conducted to compare the various *K*-Modes algorithms (with mixed attribute) and the evaluation parameters for point estimation and interval estimation of software cost were calculated. The execution time of the algorithms was also noted. These results (*MMRE*, *hit ratio* and *execution time*) for two datasets CData1 and CData2 have been reported in Tables 1 and 2. For the dataset CData1, the *MMRE* is best for 2, 3, and 4 clusters and the *hit ratio* is the best for all the clusters. The *execution time* is the second best for 2, 3, 4 and 5 clusters and it is the best for 6 clusters. For the dataset CData2, the

Table 1 Experimental results for CData1

NOC	Parameter	KM1	KM2	KM3	KM4	KM5
2	*MMRE*	0.7510	0.7964	0.8278	0.8138	0.7503[a]
	Hit ratio	0.9166	0.9166	0.8333	0.8750	0.9166[a]
	Time	0.5301[a]	1.1400	1.9060	3.9876	1.0781[b]
3	*MMRE*	0.8414	0.7565	0.7230	0.8568	0.6100[a]
	Hit ration	0.8333	0.8333	0.8750	0.8333	0.8750[a]
	Time	0.6404[a]	1.6090	3.3906	4.7651	1.3906[b]
4	*MMRE*	0.6556	0.6890	0.5709	0.7671	0.5043[a]
	Hit ratio	0.8750	0.7916	0.8333	0.8750	0.8750[a]
	Time	0.7968[a]	2.1406	5.3125	5.0654	1.5469[b]
5	*MMRE*	0.6534	0.6196	0.6951	0.5918[a]	0.6447
	Hit ratio	0.7750	0.7500	0.7083	0.6250	0.7916[a]
	Time	0.9375	2.5469	7.6250	6.7654	1.9531[b]
6	*MMRE*	0.8993	0.7141	0.4787[a]	0.6271	0.6324
	Hit ratio	0.7368	0.7083	0.7916	0.7543	0.7083[a]
	Time	3.7031	2.9688	10.484	6.8765	2.2656[a]

NOC Number of clusters, [a] for the best result and [b] for the second best

Table 2 Experimental results for CData2

NOC	Parameter	KM1	KM2	KM3	KM4	KM5
2	*MMRE*	1.0844	0.9860	1.0211	0.9237	0.7468[a]
	Hit ratio	0.8921	0.8684	0.8151	0.7631	0.8947[a]
	Time	1.7030	2.4219	3.5090	12.876	1.7123[b]
3	*MMRE*	0.7260	0.9564	0.6930	0.6781[a]	0.9136
	Hit ratio	0.7631	0.7368	0.8151	0.7631	0.8157[a]
	Time	2.2031	3.3750	6.1875	13.002	2.2031[b]
4	*MMRE*	0.7666	0.8024	0.8540	0.6986[a]	0.8889
	Hit ratio	0.7368	0.7631	0.7631	0.7894	0.8684[a]
	Time	2.7188	4.4375	9.7188	14.075	2.7394[b]
5	*MMRE*	0.8394	0.8378	0.6619	0.8199	0.8697
	Hit ratio	0.7894	0.7368	0.5789	0.6578	0.7931[a]
	Time	3.2500	5.3906	14.092	14.876	3.4331[b]
6	*MMRE*	0.8704	0.8748	0.8695	0.9920	0.6397[a]
	Hit ratio	0.6842	0.7368	0.7105	0.6578	0.8157[a]
	Time	3.7031	6.4601	19.367	14.765	3.7560[b]

NOC Number of clusters, [a] for the best result and [b] for the second best

MMRE is the best for 2 and 6 clusters and the *hit ratio* is the best for all the clusters. It should be noted that for CData1 for 3 clusters, there is an approximate 18 and 39 % improvement in *MMRE* as compared to the second best and worst cases. For the same dataset for 4 clusters the said values are 14 and 52 %. Similarly for CData2, for 2 clusters there is an improvement of 24 and 46 %, while for 6 clusters the values are 36 and 57 % as compared to the second best and worst case respectively. For *hit ratio* the improvement is upto 5 % for CData1, while for CData2 this value is upto 10 % in some cases. It may be further noted that the *execution times* in all the cases is either the best or it is the second best being only slightly more than the best *execution time*.

6 Conclusions

The major contribution of this research paper is the proposal of a software cost estimation approach based on modified *K*-Modes algorithm with mixed attribute. To achieve this aim, first a dissimilarity measure proposed for categorical attributes in Bishnu and Bhattacherjee (2013) has been adapted considering the software attributes which are inherently of mixed nature. Next, the computation of modes for mixed attributes and the modified *K*-Modes algorithm are discussed. As experimental validation, the mixed attributes concept has been incorporated with the existing *K*-Modes algorithms and these have been used for point and interval estimation of software cost for two real datasets. The results of Sect. 5 show that the hit ratio values for all the cases are the best for the modified *K*-Modes algorithm. The *MMRE* is the best for five cases out of the total ten in the two datasets. The *execution time* is the second best in almost all the cases. Thus it can be concluded that the proposed algorithm can be efficiently used for building models of software cost estimation. Moreover our proposed modified *K*-Modes clustering may be used for other application areas such as text mining, document clustering, bioinformatics and several others where the mixed data sets are present.

References

Andreopoulos B, An A, Wang X (2005) Clustering the internet topology at multiple layers. WSEAS Trans Inf Sci Appl 2(10):1625–1634

Aranganayagi S, Thangavel K (2009) Improved k-modes for categorical clustering using weighted dissimilarity measure. In: World Academy of Science, Engineering and Technology 3:813–819

Arifoglu A (1993) A methodology for software cost estimation. ACM SIGSOFT Softw Eng Notes 18(2):96–105

Aroba J, Cuadrado-Gallego JJ, Sicilia MA, Ramos I, Garcia-Barriocanal E (2008) Segmented software cost estimation models based on fuzzy clustering. J Syst Softw 81(11):1944–1950

Bai L, Liang J, Dang C (2011) An initialization method to simultaneously find initial cluster centers and the number of clusters for clustering categorical data. J Knowl-Based Syst 24(6):785–795

Benala TR, Dehuri S, Mall R, ChinnaBabu K (2012) Software effort prediction using unsupervised learning (clustering) and functional link artificial neural networks. In: 2012 World congress on information and communication technologies (WICT), pp 115–120

Bishnu PS, Bhattacherjee V (2013) A modified k-modes clustering algorithm. PReMI, Indian Statistical Institute, Kolkata, LNCS 8251:60–66

Cao F, Liang J, Bai L (2009) A new initialization method for categorical data clustering. J Expert Syst Appl 36(7):10223–10228

Cao F, Liang J, Li D, Bai L, Dang C (2012) A dissimilarity measure for the k-modes clustering algorithm. J Knowl-Based Syst 26:120–127

Cuadrado-Gallego JJ, Sicilia MA (2007) An algorithm for the generation of segmented parametric software estimation models and its empirical evaluation. J Comput Inf 26(1):1–15

Cuadrado-Gallego JJ, Sicilia MA, Rodriguez D, Garre M (2006) An empirical study of process-related attributes in segmented software cost-estimation relationships. J Syst Softw 79(3):353–361

Dejaeger K, Verbeke W, Martens D, Baesens B (2012) Data mining techniques for software effort estimation: a comparative study. IEEE Trans Softw Eng 38(2):375–397

Gan G, Yang Z, Wu J (2007) A genetic k-modes algorithm for clustering categorical data. Adv Data Min Appl 3584:195–202

Gan G, Wu J, Yang Z (2009) A genetic fuzzy k-modes algorithm for clustering categorical data. J Expert Syst Appl 36(2):1615–1620

Han J, Kamber M (2007) Data mining concepts and techniques, 2nd edn. Morgan Kaufmann publishers, Burlington

He Z, Deng S, Xu X (2005) Improving k-modes algorithm considering the frequencies of attribute values in mode. In: Computational intelligence and security, LNCS 3801:157–162

Huang Z (1997) Clustering large data sets with mixed numeric and categorical values. In: 1st Pacific Asia knowledge discovery and data mining conference, World Scientific, pp 21–34

Huang Z (1998) Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min Knowl Discov 2(3):283–304

Huang Z, Ng MK (1999) A fuzzy k-modes algorithm for clustering categorical data. IEEE Trans Fuzzy Syst 7(4):446–452

Jiang G, Wang Y, Liu H (2012) Research on software cost evaluation model based on case -based reasoning. In: 2nd world congress on software engineering, pp 338–341

Keung J (2009) Software development cost estimation using analogy: a review. In: ASWEC, pp 327–336

Lefteris A, Ioannis S, Maurizio M (2001) Building a software cost estimation model based on categorical data. In: In Proceedings of 7th international software metrics symposium, pp 4–15

Lin JC, Tzeng HY (2010) Applying particle swarm optimization to estimate software effort by multiple factors software project clustering. In: International computer symposium, pp 1039–1044

Manganaro V, Paratore S, Alessi E, Coffa S, Cavallaro S (2005) Adding semantics to gene expression profiles: new tools for drug discovery. Curr Med Chem 12(10):1149–1160

Mittas N, Angelis L (2013) Ranking and clustering software cost estimation models through a multiple comparisons algorithm. IEEE Trans Softw Eng 39(4):537–551

Ng MK, Li MJ, Huang JZ, He Z (2007) On the impact of dissimilarity measure in k-modes clustering algorithm. IEEE Trans Pattern Anal Mach Intell 29(3):503–507

Omar S, Soliman OS, Saleh DA, Rashwan S (2012) A bio inspired fuzzy k-modes clustering algorithm. LNCS 7665:663–669

Papatheocharous E, Andreou AS (2009) Approaching software cost estimation using entropy-based fuzzy k-modes clustering algorithm. In: AIAI workshop proceedings, pp 231–241

Stamelos I, Angelis L, Morisio M, Sakellaris E, Bleris GL (2003) Estimating the development cost of custom software. J Inf Manag 40(8):729–741

Sun Y, Zhu QM, Chen ZX (2002) An iterative initial-points refinement algorithm for categorical data clustering. Pattern Recognit Lett 23(7):875–884

Wu S, Jiang Q, Huang JZ (2007) A new initialization method for categorical data clustering. LNCS 4426:972–980

Yao J, Dash M, Tan ST, Liu H (2000) Entropy-based fuzzy clustering and fuzzy modeling. J Fuzzy Sets Syst 113(3):381–388

Zhang W, Yang Y, Wang Q (2013) A study on software effort prediction using machine learning techniques. Eval Novel Approach Softw Eng Commun Comput Inf Sci 275:1–15