International Journal of Computational Intelligence Systems
Vol. **13(1)**, 2020, *pp.* 1714–1722
DOI: https://doi.org/10.2991/ijcis.d.201020.001; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

Tree-Based Contrast Subspace Mining for Categorical Data

Florence Sia[1], Rayner Alfred[1,*], Yuto Lim[2]

[1] *Knowledge Technology Research Unit, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Jalan UMS, Kota Kinabalu, Sabah, 88400, Malaysia*
[2] *School of Information Science, Japan Advanced Institute of Science and Technology, Access 1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan*

ARTICLE INFO

Article History
Received 16 Mar 2020
Accepted 27 Sep 2020

Keywords
Mining contrast subspace
Contrast subspace
Categorical data
Feature selection
Data mining

ABSTRACT

Mining contrast subspace has emerged to find subspaces where a particular queried object is most similar to the target class against the non-target class in a two-class data set. It is important to discover those subspaces, which are known as contrast subspaces, in many real-life applications. Tree-Based Contrast Subspace Miner (TB-CSMiner) method has been recently introduced to mine contrast subspaces of queried objects specifically for numerical data set. This method employs tree-based scoring function to estimate the likelihood contrast score of subspaces with respect to the given queried object. However, it limits the use of TB-CSMiner on categorical values that are frequently encountered in real-world data sets. In this paper, the TB-CSMiner method is extended by formulating the tree-based likelihood contrast scoring function for mining contrast subspace in categorical data set. The extended method uses features values of queried object to gather target samples having similar characteristics into the same group and separate non-target samples having different characteristics from this queried object in different group. Given a contrast subspace of the target samples, the queried object should fall in a group having target samples more than the non-target samples. Several experiments have been conducted on eight real world categorical data sets to evaluate the effectiveness of the proposed extended TB-CSMiner method by performing classification tasks in a two-class classification problem with categorical input variables. The obtained results demonstrated that the extended method can improve the performance accuracy of most classification tasks. Thus, the proposed extended tree-based method is also shown to have the ability to discover contrast subspaces of the given queried object in categorical data.

1. INTRODUCTION

The advancement in technology nowadays has enabled the collection and storage of datasets with massive number of features and large volume of data. An analyst may interest to analyze the features that distinguish a particular object from the rest of object in data set. In recent years, mining contrast subspace has been introduced to discover contrast subspaces of queried object. Given a two-class data set, a queried object, and a target class, mining contrast subspace finds subspaces where a queried object is most likely similar to target class while least likely similar to non-target class. Those subspaces are subsets of features that each can comprised of one or more features that represent the data set. These subspaces are called as contrast subspaces. Any object can be queried object which its contrast subspaces are unknown and need to be investigated. Target class is any class label on which the similarity to the queried object is examined in order to find contrast subspaces.

The discovery of contrast subspaces is a crucial task in many application domains including fraud detection, intrusion detection, medical diagnosis, etc. The contrast subspaces can be used to characterize a particular object and enhance the analysis of data. For instance, in banking sector, an analyst may want to know what subspaces do the suspicious transaction is most likely similar to fraudulent cases but dissimilar to normal cases. By knowing the contrast subspaces, further action can be taken in order to avoid the bank account being used by the unauthorized user. In addition to that, an analyst may want to identify the subspaces which cause a suspicious claim most similar to fraudulent cases but different from normal cases in an insurance sector [1,2]. Those contrast subspaces can help the analyst in determining the right action to be performed to tackle the misuse claim.

Tree-Based Contrast Subspace Miner (TB-CSMiner) is a tree-based mining contrast subspace method that finds contrast subspaces of a queried object in a two-class classification problem with continuous input variables [3]. It employs a tree-based likelihood contrast scoring function in estimating the likelihood contrast score of a queried object to the target class against the non-target class regardless of the number of features in the subspace. The tree-based likelihood contrast scoring function uses interval values of features to recursively partition a subspace space so that the queried object and the target samples are grouped together and separated from the non-target samples. It estimates the likelihood contrast score of subspaces with respect to the queried object based on the ratio of probability of target samples (e.g., objects belong to target class) to the probability of non-target samples (e.g., objects

[*] *Corresponding author. Email: ralfred@ums.edu.my*

belong to non-target class). However, the tree-based likelihood contrast scoring function of TB-CSMiner is designed to estimate the likelihood contrast score of a subspace with respect to a queried object in a numerical dataset. Particularly, the partitioning criterion for the process of partitioning the dataset is formed based on the interval values of numerical feature. The partitioning process conducted based on interval values is not applicable for categorical dataset that takes unordered nominal values [4–7]. Thus, TB-CSMiner cannot be applied to handle categorical datasets directly. Mapping nominal values onto numerical or ordinal values may cause loss of information [5,6].

In real-life applications, many research works have been conducted to deal with categorical data in various data mining problem [4,5, 8–14]. This indicates the importance of effectively mining contrast subspace to learn categorical datasets. Other works include summarizing relational datasets based on features selection which emphasized on discretization of numerical data into categorical data [15–17]. This paper addresses this issue by extending the TB-CSMiner method where a tree-based likelihood contrast scoring function is proposed for estimating the likelihood contrast score of subspaces with respect to queried object in a categorical dataset.

In a two-class categorical data set, given a subspace space and a queried object, the proposed tree-based likelihood contrast scoring function divides data objects into two subsets of objects repeatedly by using features values of queried object until the subset comprises of only objects belong to one class or the number of objects in the subset meets a minimum predefined threshold. Then, it measures the ratio of probability of target samples to the probability of non-target samples in the subset containing the queried object. Herein, the contrast subspace of a queried object is defined as the subspace where the queried object falls having higher probability number of target objects than the probability number of non-target objects. Since it is impractical to measure the likelihood contrast score of all possible subspaces, a similar framework of TB-CSMiner method is used to discover contrast subspaces of queried object in categorical dataset. That is, it first measures the likelihood contrast score of individual features or one-dimensional subspaces for the given queried object and selects only certain one-dimensional subspaces having high scores. This is followed by measuring the likelihood contrast score of possible subspaces for the queried object derived from the selected one-dimensional subspaces to find the contrast subspaces of queried object.

In this work, the effectiveness of the proposed extended TB-CSMiner method is validated in terms of performance accuracy of the classification task conducted on the resulting contrast subspaces. For a contrast subspace obtained from using the extended method, a new data set is generated by populating the contrast subspace space with the data sampled from the given data set in such a way that the target class and non-target class are separated well, with reference to a queried object. Subsequently, the newly created data set is used as an input for classification tasks. A contrast subspace shall achieve higher classification accuracy than other subspaces. The experimental results showed that the extended method has the ability to find contrast subspaces for the given queried object in categorical data.

The main contribution of this paper is to provide a tree-based method to find contrast subspaces of the target samples for categorical datasets. For a subspace, the tree-based method focuses on identifying a group of target samples having similar characteristics with the queried object. The tree-based method uses features values of queried object as criterion in partitioning the target and non-target samples. Based on the queried objects, the numbers of target and non-target samples are used to determine the similarity of queried object to the target class and non-target class in a subspace.

The remainder of this paper is organized as follows: Section 2 discusses related works. Section 3 presents the formalization of the tree-based likelihood contrast scoring function that deals with categorical values. The framework of the extended TB-CSMiner method which incorporates the proposed tree-based likelihood contrast scoring function for categorical data set is also presented in Section 3. Section 4 describes the experimental setup and results on eight real world categorical data sets. Finally, the paper is concluded in Section 5 by drawing a conclusion and providing some future works.

2. RELATED WORKS

There are a few methods have been developed to find contrast subspaces of a queried object to the best of our knowledge. Contrast Subspace Miner (CSMiner) is the earliest method introduced to discover contrast subspaces of the given queried object in numerical data set [1]. For a subspace, it employs the probability density of target class and the probability density of non-target class in the region where the queried object is located, to measure the respective likelihood score of the queried object to target class and to non-target class. The ratio of probability density of target class to probability density of non-target class is then used to estimate the likelihood contrast score of a subspace. A contrast subspace is where the queried object is in a region that comprised of higher density of target objects and lower density of non-target objects than average. This is corresponding to the queried object is most likely similar to target class and least likely similar to non-target class respectively. The likelihood contrast scores of contrast subspaces are often high compared to other subspaces. The CSMiner approach considers only non-trivial subspaces which their likelihood scores to the target class are not less than the predetermined minimum likelihood score in searching for contrast subspaces process. Besides that, CSMiner avoids the brute force search by using an upper bound of probability density of target objects to remove trivial subspaces from the search space. The possible subspaces are represented by an enumeration tree and the subspaces set are searched in depth-first search manner. All superspaces of a subspace (i.e., descendents of a subspace) can be removed if the upper bound of probability density of target objects does not meet the minimum likelihood threshold. CSMiner is remaining inefficient for data set with high number of features since the number of subspaces that need to be examined grows exponentially with the number of features in data set.

Contrast Subspace Miner-Bounding Pruning Refining (CSMiner-BPR) has been proposed to accelerate the process of CSMiner in searching contrast subspaces [2]. CSMiner-BPR combines a few new bounding pruning strategies and the pruning rule of CSMiner. CSMiner-BPR creates an upper bound of probability density of target objects and lower bound of probability density of non-target objects based on the ε-neighborhood of queried object. For each subspace, the neighborhood of a queried object is determined by the ε which is the marginal standard deviation of the data. Any objects within the distance ε of a queried object are assigned as the neighbors of the queried object. If the upper bound and lower bound conditions are met, all superspaces of a subspace can be pruned. By taking ε-neighborhood into account, it reduces the computation cost for objects outside from the ε-neighborhood and thus, speeds up the contrast subspaces search process.

However, CSMiner and CSMiner-BPR were developed for mining contrast subspaces in data set comprised of numerical values only. In both of these methods, the pair wise distance computation constituted their likelihood contrast scoring function. They used distance between objects to measure the similarity between objects. Unlike numerical values, categorical values are unordered and thus, it is difficult to define the similarity between categorical values by using distance [3,4]. This causes the existing methods cannot be directly applied to analyze categorical data set.

Recently, TB-CSMiner was introduced that applies a tree-based likelihood contrast scoring function to find contrast subspaces of the queried object in numerical data set [3]. The tree-based likelihood contrast scoring function adopts the divide-and-conquer concept of decision tree in measuring the likelihood contrast score of subspaces with respect to the given queried object. For a subspace, firstly, it selects a partitioning criterion to partition data objects including the queried object into two subsets of objects until the subset that containing the queried object reaches the predetermined minimum number of objects threshold or having objects belong to the same class. A partitioning criterion comprises a feature and its interval value which is the best for discriminating objects including queried object of different class. The likelihood contrast score of subspaces with respect to the queried object is then computed based on the ratio of probability of target objects to probability of non-target objects in the subset that containing the queried object. This also follows the notion applied in formulating the likelihood contrast score presented in the previous works (e.g., CSMiner and CSMiner-BPR) [1,2]. In contrast subspace, the queried object resides in subset that has higher probability of target objects and lower probability of non-target objects than the average value. That is, the contrast subspace should have high tree-based likelihood contrast score. TB-CSMiner accelerates the mining process by selecting a subset of one-dimensional subspaces with high likelihood contrast score from the initial features given in the data set. The contrast subspace of queried object is searched from those non-trivial one-dimensional subspaces. Although the tree-based likelihood contrast scoring function does not use distance in measuring the likelihood between the queried object and the class label, TB-CSMiner method cannot be applied directly for finding contrast subspaces of the queried object in categorical data. This is due to the categorical data contains features that take unordered nominal values. The partitioning criterion of the scoring function which is composed of the interval values of numeric feature cannot be used to partition categorical data in order to measure the likelihood contrast score of subspaces for a queried object.

Local Outlier with Graph Projection (LOGP) characterizes the abnormality of an object against the normal objects for categorical data by using the graph embedding approach [18]. It constructs geometrical structure model from the given data distribution and the model is used to find the subspaces where the object is well discriminated from the normal objects. In the problem of discovering properties, an algorithm called EXPREX was proposed that introduced the notion of exceptional property and defined the concept of exceptionality score, which measures the significance of a property [19]. Exceptional Property Extractor (EXPREX) identifies exceptional features with high exceptionality score for characterizing the abnormality of an object in categorical data. The exceptionality score is based on the randomization test following the Pearson chi-square criterion. A semi-supervised approach has been introduced for detecting outlier in categorical data [5]. It employs distance-based algorithm to characterize normal objects. This characterization model is used to identify outliers. Those features that contribute much in discriminating between normal objects and outliers are exploited in order to extract valuable information about the outlier. A feature-grouping-based outlier detection algorithm, WATCH method is proposed to identify outliers and explanation of the outlier in categorical data [20]. It gathered features into multiple groups based on the correlation among features. Then, it searched for outliers residing in each of the groups. The features group where the outliers are detected is used to explain the deviation of the outliers from other objects. Nevertheless, all of these methods find subspaces for explaining why an object is an outlier with respect to normal objects or inliers in categorical data. This is different from the proposed work in this paper in which the work focuses on finding subspaces where a queried object is most similar to target class than other classes in categorical data.

3. TB-CSMINER FOR CATEGORICAL DATA

In this section, an extended TB-CSMiner method is presented to discover contrast subspaces of the given a queried object in two-class categorical data. This method employs a tree-based likelihood contrast scoring function which is devised to estimate the likelihood contrast score of subspaces for a queried object in categorical data. In the next subsection, the formalization of new tree-based likelihood contrast scoring function utilized in the extended TB-CSMiner is presented. This is followed by a description of phases involved in the framework of the proposed extended TB-CSMiner to efficiently search for contrast subspaces in categorical data.

3.1. Tree-Based Likelihood Contrast Scoring Function

Given a two-class categorical data set, a queried object, and a target class, tree-based likelihood contrast scoring function uses features values of the queried object as partitioning criterion to group objects including queried object that have similar values on the features and also to separate the queried object from the non-target objects that have dissimilar values on the features in a subspace space. In fact, objects that have the same characteristics often belong to the same class and also belong to different class for objects that have dissimilar characteristics [21,22]. The tree-based likelihood contrast scoring function measures the likelihood score of a queried object to the target class and the non-target class based on the probability of target objects and the probability of non-target objects, in the group that contains queried object, respectively. Then, it uses the ratio of probability of target objects to probability of non-target objects to measure the likelihood contrast score of a subspace for the queried object.

The steps of the tree-based likelihood scoring measure process are described in the following. For a subspace space, it starts with choosing a criterion to partition data objects in the root node into a pair of nodes. Each of the nodes contains a subset of objects. We use the value of the selected feature of the given queried object as the partitioning criterion. For example, let consider the selected feature has value *r* for the given queried object, *q*, this value will be used for partitioning a node into a node containing subset of objects that have value *r* and another node which contains the rest of objects. A sequence of partitioning process is performed only on node where the given queried object falls. The partitioning process will be stopped when the number of objects in a node reaches a minimum number of objects threshold, *MinObjs*, or the nodes contains only objects belong to the same class. The node that has met one of the stopping criterions becomes the leaf node.

The likelihood contrast score of a subspace can be estimated based on the ratio of probability of target objects to probability of non-target objects in the leaf node. Sometimes, the leaf node can contain target objects without any non-target objects. In this situation, we use a small constant value $n = 0.001$ to replace the zero denominator of the probability of non-target objects which has been suggested in other relevant related works [23,24]. Let a queried object q and a dataset consists of objects O of two classes, $O = O_+ \cap O_-$ where, O_+ and O_- are subsets of objects of O belong to target class C_+ and non-target class C_- respectively, described by a set of categorical features F, the tree-based likelihood contrast score of a subspace S for a q can be defined as following:

$$TB - LC_S(q) = \frac{P(C_+, X_{leaf})/|O_+|}{n} \tag{1}$$

where $P(C_+, X_{leaf})$ is the amount of target objects in the leaf node, $|O_+|$ is the amount of target objects in O,

$$n = \begin{cases} P(C_-, X_{leaf})/|O_-|, P(C_-, X_{leaf}) > 0 \\ 0.001, P(C_-, X_{leaf}) = 0 \end{cases} \tag{2}$$

In Eq. (2), $P(C_-, X_{leaf})$ is the amount of non-target objects in the leaf node and $|O_-|$ is the number of non-target objects in O. The value of $P(C_+, X_{leaf})$ and $P(C_-, X_{leaf})$ can be affected by the $|O_+|$ and $|O_-|$ respectively. Thus, $|O_+|$ and $|O_-|$ are used to normalize the respective $P(C_+, X_{leaf})$ and $P(C_-, X_{leaf})$. This corresponds to the probability of target objects and the probability of non-target objects in the leaf node. High tree-based likelihood contrast score of a subspace indicates that the queried object is more likely similar to the target class against the non-target class in the subspace. This means the subspace is likely to be the contrast subspace of the queried object.

Figure 1 depicts an illustration of $TB - LC_S(q)$ measure of three dimensional subspaces $\{ s_1, s_2, s_3\}$ and $\{ s_2, s_4, s_6\}$. Suppose each feature in both of the subspaces has nominal values *a*, *b*, and *c*. The red point represents the queried object, the blue points signify objects that belong to target class, and the green points represent objects belong to non-target class. Let the minimum number of threshold *MinObjs*=5. For subspace $\{ s_1, s_2, s_3\}$ in Figure 1, assume that the queried object has value *b*, *a*, and *c* for the respective feature s_1, s_2, s_3, feature s_1 having value *b* based on the queried object is selected first to partition the node that contains all objects in the data set into two groups, left node group and right node group. The left node group and right node group respectively comprise of objects that have feature s_1 having values other than *b* and value *b*. Similar partition process continues using feature s_2 having value *a* on the right node into which the queried object falls. After that, feature s_3 having value *c* is selected to partition the right node. Since the queried object resides on the right node and there are only target objects in the node, the partition has been terminated at the third tier of the tree and the node becomes leaf node. The likelihood contrast score of subspaces $\{ s_1, s_2, s_3\}$ is then computed.

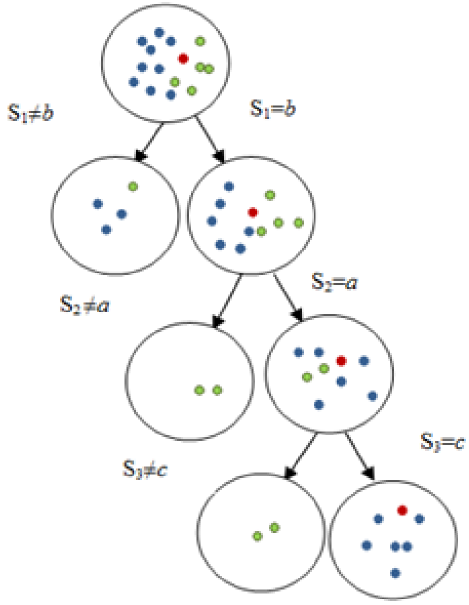

Figure 1 An illustration of $TB - LC_S(q)$ computation subspace $\{s_1, s_2, s_3\}$ for categorical data.

Meanwhile, for subspace $\{ s_2, s_4, s_6\}$ in Figure 2, assume that the queried object has value *b*, *a*, *b* for each feature in the subspace respectively. The partition process begins by selecting feature s_4 having value *a* following the s_4 of the queried object and proceeded on the right node group that contains the queried object using feature s_6 having value *b*. Lastly, the feature s_2 habing value *a* is then used to partition the right node where the queried object falls. The partition process is halted on the right node at the third tier of the tree as the amount of objects in the node is less than *MinObjs*. That node becomes leaf node and subsequently the likelihood contrast score is calculated for subspace $\{ s_2, s_4, s_6\}$.

Based on the likelihood contrast score estimation for subspace $\{ s_1, s_2, s_3\}$ and $\{ s_2, s_4, s_6\}$, the subspace $\{ s_1, s_2, s_3\}$ is identified as the contrast subspace of the given queried object. This is undeniable as there are target objects only in the leaf node which corresponds to the queried object is more likely similar to the target class against non-target class.

3.1.1. The framework of extended TB-CSMiner method

When the number of features in categorical data set is high, the number of subspaces derived can be huge. This causes inefficient to compute the $TB - LC_S(q)$ of each subspace for a queried object. Feature selection is well known with its ability to accelerate the process of mining through reducing the dimensionality of data and at the same time it can improve the mining accuracy by eliminating the trivial features [25–28]. Hence, similar to the TB-CSMiner, the extended method comprises two main phases that are the feature selection and the contrast subspace search. The framework of two phases extended TB-CSMiner method is illustrated in Figure 3.

The pseudo code for phase one is presented in Algorithm 1 as shown in Table 1. In phase one, the tree-based likelihood contrast score of a subspace S for a given queried object q of a one-dimensional subspace in the categorical data set, is computed by using the devised $TB - LC_S(q)$ scoring function specifically for categorical data as shown in Eq. (1). The subspace is then compared with the subspaces in the current list of l one-dimensional subspaces by their likelihood contrast scores. If there is a subspace in the list which its likelihood contrast score is lower than the score of the currently considered subspace, the subspace in the list is removed and replaced with the considered subspace. After finished assessing each of the one-dimensional subspaces given in the data set, the subspaces are sorted in descending order of their $TB - LC_S(q)$ scores. These top l ranked one-dimensional subspaces are selected as a subset of non-trivial subspaces for contrast subspaces search in phase two. The parameter l is a predetermined number of one-dimensional subspaces.

Algorithm 2 provides the contrast subspace search procedure in phase two depicted in Table 2. In phase two, the $TB-LC_S(q)$ score of t random subspaces with respect to the queried object are assessed to find contrast subspaces of the queried object. First, a feature is randomly picked from the subset of selected one-dimensional subspaces. Herein, the value of the feature based on the given queried object is then used as a partitioning criterion to partition data space with respect to the queried object. These processes are performed repeatedly until the stopping criterions as mentioned in the previous section are met. The different features along the partition are taken as a random subspace. After that, the devised $TB - LC_S(q)$ function specifically for categorical data in Eq. (1), is used to compute the $TB - LC_S(q)$ score of the subspace. Then, the $TB - LC_S(q)$ score of the subspace is compared with the score of subspaces in the current list of k subspaces. The subspaces in the list with $TB-LC_S(q)$ score lower than the score of the current subspace being considered are removed from the list and then, the considered subspace is appended to the list. When t random subspaces have been constructed and examined, the k subspaces in the list are ordered by their $TB-LC_S(q)$ scores from the largest score to the smallest score. Finally, the h highly scored subspaces are selected as the contrast subspaces of the queried object.

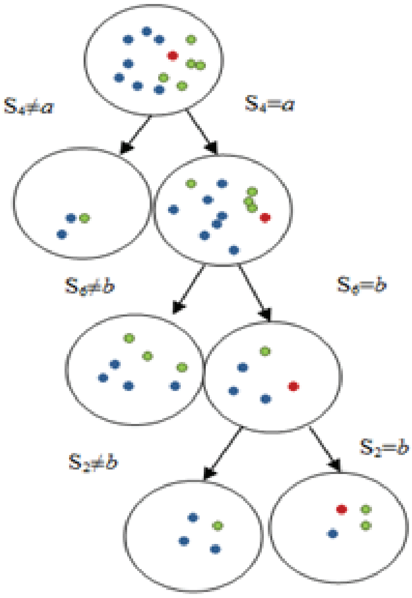

Figure 2 An illustration of $TB - LC_S(q)$ computation subspace $\{s_2, s_4, s_6\}$ for categorical data.

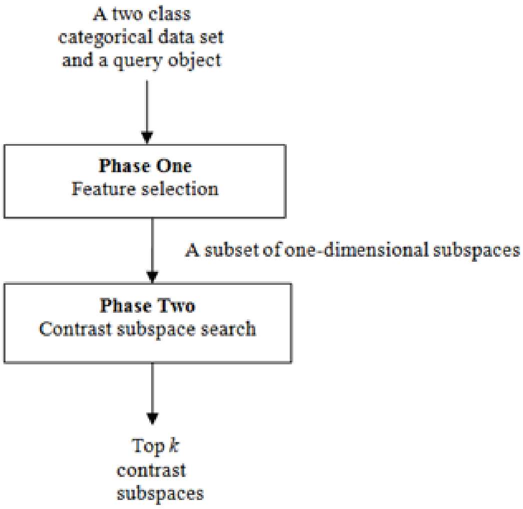

Figure 3 Framework of extended Tree-Based Contrast Subspace Miner (TB-CSMiner) method.

Table 1 Pseudo code for feature selection phase.

Algorithm 1 Feature selection

Input: queried object q, target class C_+, categorical data set X, set of categorical features F, minimum number of objects $MinObjs$, number of one-dimensional subspaces l

Output: Subset of l one-dimensional subspaces with highest tree-based likelihood contrast score

```
Initialize $Fs$ as list of $l$ null subspaces with tree-based likelihood
  contrast score 0
For one-dimensional subspace $s \in F$
    If $\forall o \in X$ belong to same class or $|X| \leqslant$ MinObjs
        Compute $TB - LCs(q)$ using Eq. (1);
        S ← s;
        If ∃ S’ ∈ $Fs$ s.t. TB-LCS(q) > TB-LCS‘(q)
            Insert S into $Fs$ and remove S’ from $Fs$;
        End
    Else
        Select the $s$ value of $q$ to partition X;
        X ← subset of data containing $q$ after partition;
    End
    Sort Ans in descending order of TB-LCs(q);
End for
Sort $Fs$ in descending order of TB-LCs(q);
Return $Fs$;
```

Table 2 Pseudo code for contrast subspace search phase.

Algorithm 2 Contrast subspace search

Input: queried object q, target class C_+, categorical data set X, set of one-dimensional subspace Fs, minimum number of objects $MinObjs$, number of contrast subspaces h, number of random subspaces t
Output: h contrast subspaces with highest tree-based likelihood contrast score

```
1   Initialize Ans as list of $h$ null subspaces with tree-based likelihood
      contrast score 0
2   For $i$:=1:$t$ do
3       S ← ∅;
4       Select a random feature $s$ from $Fs$;
5       Select the $s$ value of $q$ to partition $X$;
6       X ← subset of data containing $q$ after partition;
7       If $\nexists\, s\prime \in$ S s.t. $s\prime = s$
8           Insert $s$ into S;
9       End
10      If $\forall o \in X$ belong to same class or $|X| \leqslant$ MinObjs
11          Compute $TB - LCs(q)$ using Eq. (1);
12          If ∃ S’ ∈ Ans s.t. TB-LCS(q) > TB-LCS‘(q)
13              Insert S into Ans and remove S’ from Ans;
14          End
15      Else
16          Go to Step 4;
17      End
18      Sort Ans in descending order of TB-LCs(q);
19  End for
20  Return Ans;
```

4. EXPERIMENTAL SETUP

4.1. Datasets

In this work, several experiments have been carried out to evaluate the effectiveness of the extended TB-CSMiner method. A total of eight frequently used real world categorical data sets taken from UCI machine learning repository [29] are used in this work to assess the proposed extended TB-CSMiner method based on the classification performance. All objects which have missing features values are eliminated from the data sets. The first data set is the *Mushroom* data containing 8124 hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. All of the samples are described by 22 features of mushroom characteristics. Each sample is classified into either *ediblemushroom* or *poisonousmushroom*. The second data set is the *BreastCancer* data which consists of 286 cases. Each case is described by nine features including tumor size, node-cap, age, etc. These cases are classified into two classes, *no−recurrence−events* class and *recurrence−events* class. The third data set is the *CongressionalVotes* data which has 435 vote records of the U.S. House of Representatives Congressmen described by 16 features correspond to different types of votes identified by the Congressional Quarterly Almanac. Those votes are being classified either *democrat* or *republican*. The fourth data set is the Tic-Tac-Toe data consists of 958 possible board configurations at the end of tic-tac-toe games and described by nine features represent different tic-tac-toe square. Those board configurations are classified into two classes, *positive* (i.e., win) and *negative* classes. The fifth data set is the *Lymphography* data that contains 148 cases of Lymphography provided by the Oncology Institute in Ljubljana. These cases are described by 18 features of lymph characteristics. There are four classes of cases { *normalfind*, *metastases*, *malignlymph*, and *fibrosis*}. The sixth data set is *Chess* data which contains 3196 board descriptions for the end of a chess game and described by 36 features of the board. Those boards are classified into two classes, *win* class and *nowin* class. The seventh data set is *CarEvaluation* data that consists of 1728 records of different concept structures of car. Each record is described by six features which include the buying price, the maintenance price, the number of doors, the persons capacity, the size of luggage boot, and the safety of car. These records are being classified into *unacceptable*, *acceptable*, *good*, and *verygood* class. The last data set is *SPECTHeart* data which has 267 cardiac Single Proton Emission Computed Tomography (SPECT) images of patients. It contains 22 features extracted from the SPECT images. Each patient is classified into either *normal* class or *abnormal* class. The summary of the datasets used in this work is described in Table 3.

4.2. Settings and Procedure

Unfortunately, the ground truth of contrast subspaces are not provided in real-world categorical data sets. Hence, we propose to assess the accuracy of contrast subspaces obtained in terms of how well are the classes separated (i.e., classification accuracy) in the contrast subspace projection with respect to queried object. This is because the contrast subspace can be used to explain the separability of two classes. We use the best parameters setting for TB-CSMiner in finding contrast subspace that has been identified through a series of experiment analysis [30]. That is the minimum number of objects threshold for leaf node is $MinObjs$ = 25%, of total objects in the training data, the number of top ranked one-dimensional subspaces is l=3, and the number of trees is t=400. The extended method is implemented in Matlab 9.2 programming language while the assessment of contrast subspaces is implemented in Java programming language.

The experimental procedure on real-world categorical data sets consists of two stages as follows. During the first stage, for each data set, we take a class as the target class and the rest of the classes as non-target class. For a dataset that has more than two classes, we only take two classes that have the majority of the objects as the target classes. The queried objects are considered as those objects that belong to target class. Then, we apply the extended method on all queried objects to discover their contrast subspaces. In this experiment, we only consider one contrast subspace for each queried object that is the first-ranked contrast subspace. The first-ranked contrast subspace is chosen because it has the highest likelihood contrast score corresponding to the contrast subspace that best characterizes the given queried object. These processes recur but at this time taking non-target class as target class and the remaining classes as non-target class. In the second stage, we generate a new data set of two classes containing objects sampled from the data set for the contrast subspace of each queried object. The first class is labelled as Class A that takes the queried object and the target objects which their distances are less than or equal the k-distance of queried object. We employ a small value k = 30 in this experiment which has been shown sufficient for satisfactory performance in the related existing research works [31]. The reason of sampling data in such a way is to find a group of target objects that have the same characteristics with the queried object. The second class is labelled as Class B that takes objects randomly of the same size as Class A from other class. This is to avoid the imbalance class issue that may affect the performance of the classification task. Finally, the classifier, J48 (decision tree) [32], Naive Bayes (NB) [33], and support vector machine (SVM) [34], in WEKA are used to carry out classification on the new formed data set [35]. The 20-fold cross validation classification accuracy (i.e., percentage of correctly classified objects) on contrast subspaces for all queried objects is averaged. Particularly, for the contrast subspace of each queried object, the classification accuracy is computed based on 20 randomly selected test objects. The average of classification accuracy is used to evaluate the accuracy of the contrast subspaces gained for all queried objects [36,37].

Since there is no method has been developed yet for mining contrast subspace in categorical data, we compare the classification performance on contrast subspace space with the performance on full dimensional space. We generate the full dimensional subspace space following the abovementioned procedure by considering all features given in the data set. Hence, the size of full-dimensional subspace space is the same as the contrast subspace space. High classification accuracy indicates that the contrast subspace is more likely the right contrast subspace for the given queried object. The results of all classifiers on eight real-world categorical data sets for full-dimensional subspace and contrast subspace are reported in Table 4.

5. RESULTS AND DISCUSSION

The results of all classifiers on eight real-world categorical data sets for full-dimensional subspace and contrast subspace are reported in Table 4.

Refering to Table 4, the average classification accuracies of the classification tasks performed by the J48 classifier using the contrast subspace space dataset were higher compared to the one using the full-dimensional space for all data sets except Mushroom data. That is, the J48 classifier achieved 84.56%, 88.23%, 98.58%, 94.99%, 90.78%, 97.69%, 95.04%, and 89.82% for the respective Mushroom, Breast Cancer, Congressional Votes, Tic-Tac-Toe, Lymphography, Chess, Car Evaluation, and Heart data. While for the NB classifier, it shows high average classification accuracy for the contrast subspace compared to the average classification accuracy for the full-dimensional subspace of the Breast Cancer, Congressional Votes, Tic-Tac-Toe, Lymphography, Chess, and Heart datasets. The NB classifier achieved 86.27%, 89.83%, 98.32%, 95.71%, 91.85%, 98.03%, 95.53%, and 91.52% for Mushroom, Breast Cancer, Congressional Votes, Tic-Tac-Toe, Lymphography, Chess, Car Evaluation, and Heart datasets, respectively.

Similarly, the Random Forest (RF) classifier gained high average classification accuracy on contrast subspace compared to the average classification accuracy on full dimensional subspace for Breast Cancer, Congressional Votes, Tic-Tac-Toe, Lymphography, Chess, and Heart data. Particularly, RF achieved 86.74%, 89.85%, 98.84%, 95.37%, 91.69%, 98.22%, 95.45%, and 90.97% for the respective Mushroom, Breast Cancer, Congressional Votes, Tic-Tac-Toe, Lymphography, Chess, Car Evaluation, and Heart data. On the other hand, the average classification accuracy of the classifier SVM on contrast subspace is higher than the average classification accuracy of the classifier SVM on full-dimensional subspace for five data sets which are the Breast Cancer, the Congressional Votes, the Tic-Tac-Toe, the Lymphography, and the Heart data. It achieved 86.14%, 86.88%, 98.45%, 95.25%, 86.31%, 97.65%, 95.37%,

Table 3 Descriptions of eight frequently used real world categorical data sets taken from UCI machine learning repository [29].

	Descriptions			
Data Sets	**Samples**	**Features**	**Classes**	**Names of Classes**
Mushroom	8124	22	2	Edible mushroom, poisonous mushroom
Breast cancer	286	9	2	No-recurrence-events, recurrence-events
Congressional votes	435	16	2	Democrat, republican
Tic-Tac-Toe	958	9	2	Positive, negative
Lymphography	148	18	4	Normal find, metastases, malign lymph, fibrosis
Chess	3196	36	2	Win, no win
Car evaluation	1728	4	6	Unacceptable, acceptable, good, very good
Heart	267	22	2	Normal, abnormal

Table 4 Average classification accuracy (%) for full-dimensional subspace and contrast subspace.

	Full-Dimensional Subspace				**Contrast Subspace**			
Data Set	***J48***	***NB***	***SVM***	***RF***	***J48***	***NB***	***SVM***	***RF***
Mushroom	**98.64**	**99.10**	**99.93**	**99.75**	84.56	86.27	86.14	86.74
Breast cancer	78.04	80.71	80.14	77.17	**88.23**	**89.83**	**86.88**	**89.85**
Congressional votes	82.23	88.20	85.74	88.69	**98.58**	**98.32**	**98.45**	**98.84**
Tic-Tac-Toe	83.16	91.77	92.04	91.05	**94.99**	**95.71**	**95.25**	**95.37**
Lymphography	83.23	88.20	85.74	88.69	**90.78**	**91.85**	**86.31**	**91.69**
Chess	93.94	95.93	**97.69**	97.05	**97.69**	**98.03**	97.65	**98.22**
Car evaluation	94.84	**96.97**	**97.50**	**97.22**	**95.04**	95.53	95.37	95.45
Heart	77.95	83.00	82.50	82.60	**89.82**	**91.52**	**89.64**	**90.97**

and 89.64% for the respective Mushroom, Breast Cancer, Congressional Votes, Tic-Tac-Toe, Lymphography, Chess, Car Evaluation, and Heart data. However, the performance accuracy of the classifiers was lower when using the contrast subspace of the Mushroom and Car Evaluation datasets. This happened because of the high number of objects in Mushroom and Car Evaluation data may require high minimum number of objects in the leaf node to estimate the similarity between the queried object and the target samples. Meanwhile, the classifiers performed very well for the contrast subspace than full dimensional subspace of the Breast Cancer, Congressional Votes, Tic-Tac-Toe, Lymphography, and Heart datasets. This may be due to the fact that those data comprised of only small number of objects. Although there are advanced approaches able to achieve high performance accuracy on some of the data sets, those approaches attempt at either classifying object to a class or searching clusters of objects that are having similar characteristics [38,39]. However, our work interest is to find contrast subspaces for a particular queried object where queried object is most similar to target class against other class.

The classification accuracies obtained using the NB and RF classifiers for the contrast subspace space often provide high classification accuracy compared to other classifiers. This is due to the classifier NB involves probability estimation in classifying data which is similar to the tree-based likelihood contrast scoring function of the tree-based method that estimates the probability of target objects and other objects with respect to the queried object in finding contrast subspaces. Meanwhile, the RF classifier creates multiple random decision trees, thus it reduces the over fitting issues and hence improves the classification accuracy. The SVM classifier produced low classification accuracy for the contrast subspace space compared to the other classifiers on most of the datasets. This is because proper approaches are necessary for SVM to deal with data consisting more than two classes [35].

Overall, the classification tasks being carried on contrast subspace space outperform most of the classification tasks on full-dimensional space that is 24 out of 32 various cases. The classification tasks on full-dimensional subspace outperform the classification tasks on contrast subspace on only eight out of 32 various cases. The superior performance on contrast subspace indicates that the contrast subspaces obtained by using the extended tree-based method is capable to improve the accuracy of the classification tasks. This is due to the fact that the objects of two classes are separated well in contrast subspace space. According to the paired-sample T-test at the significance level of 0.05, the extended tree-based method has achieved significant improvement of classification accuracy against the baseline classification (i.e., classification on full-dimensional space) on most of cases. In summary, the results demonstrate that the extended method exhibits the ability to identify the right contrast subspaces for the given queried objects in categorical data set.

6. CONCLUSION

There are many data contains categorical values in real-life applications. It often requires methods specifically designed for handling those categorical data in order to attain satisfactory performance. In this paper, we have extended the TB-CSMiner for finding contrast subspaces of a given queried object in two-class categorical data set. It uses a tree-based likelihood contrast scoring function that has been designed to handle nominal values of categorical data. The extended method comprises of two phases; Phase (1) finding a subset of relevant one-dimensional subspaces from initial features given in dataset and Phase (2) performing contrast subspace search on the selected subset of one-dimensional subspaces, with respect to a given queried object. We have experimentally evaluated the effectiveness of our extended method on eight real-world categorical data sets. These empirical studies have shown that the extended method was capable to improve the classification accuracy and mine contrast subspaces of a given queried object in categorical data. One of the future works of this paper will be improving the efficiency of mining process and optimizing the contrast subspace search using an evolutionary algorithm.

AUTHORS' CONTRIBUTIONS

This is to inform the editor that all Authors have contributed equally to the publication of this paper.

ACKNOWLEDGMENTS

This work was supported in part by the Universiti Malaysia Sabah internal grant no. SBK0428-2018.

REFERENCES

[1] L. Duan, G. Tang, J. Pei, J. Bailey, G. Dong, A. Campbell, C. Tang, Mining contrast subspaces, in: V.S. Tseng, T.B. Ho, Z.H. Zhou, A.L.P. Chen, H.Y. Kao (Eds.), Advances in Knowledge Discovery and Data Mining, PAKDD 2014, Lecture Notes in Computer Science, vol. 8443, Springer, Cham, Switzerland, 2014.

[2] L. Duan, G. Tang, J. Pei, J. Bailey, G. Dong, V. Nguyen, A. Campbell, C. Tang, Efficient discovery of contrast subspaces for object explanation and characterization, Knowl. Inf. Syst. 47 (2016), 99–129.

[3] F. Sia, R. Alfred, Tree-based mining contrast subspace, Int. J. Adv. Intell. Inf. 5 (2019), 169–178.

[4] J. Li, J. Zhang, X. Qin, Y. Xun, Feature grouping-based parallel outlier mining of categorical data using spark, Inf. Sci. 504 (2019), 1–19.

[5] D. Ienco, R.G. Pensa, R. Meo, A semisupervised approach to the detection and characterization of outliers in categorical data, IEEE Trans. Neural Netw. Learn. Syst. 28 (2017), 1017–1029.

[6] F. Cao, J.Z. Huang, J. Liang, A fuzzy SV-k-modes algorithm for clustering categorical data with set-valued attributes, Appl. Math. Comput. 295 (2017), 1–15.

[7] L. Bai, J. Liang, C. Dang, F. Cao, A cluster centers initialization method for clustering categorical data, Expert Syst. Appl. 39 (2012), 8022–8029.

[8] H. Xu, Y. Wang, L. Cheng, Y. Wang, X. Ma, Exploring a high-quality outlying feature value set for noise-resilient outlier detection in categorical data, in ACM International Conference on Information and Knowledge Management, Torino, Italy, 2018, pp. 17–26.

[9] I.K. Park, G.S. Choi, Rough set approach for clustering categorical data using information-theoretic dependency measure, Inf. Syst. 48 (2015), 289–295.

[10] L. Cheng, Y. Wang, X. Ma, A neural probabilistic outlier detection method for categorical data, Neurocomputing. 365 (2019), 325–335.

[11] D. Ienco, R.G. Pensa, Positive and unlabeled learning in categorical data, Neurocomputing. 196 (2016) 113–124.

[12] M. Azmi, G.C. Runger, B. Abdelaziz, Interpretable regularized class association rules algorithm for classification in a categorical data space, Inf. Sci. 483 (2019), 313–331.

[13] S. Jian, L. Cao, K. Lu, H. Gao, Unsupervised coupled metric similarity for Non-IID categorical data, IEEE Trans. Knowl. Data Eng. 30 (2018), 1810–1823.

[14] S.M. Aboukhamseen, R.A. M'Hallah, Genetic algorithms for cross-calibration of categorical data, J. Mod. Appl. Stat. Methods. 16 (2017), 722–742.

[15] R. Alfred, G. Chiye, J. Obit, M. Hijazi, C. On, H. Lau, A genetic algorithm based clustering ensemble approach to learning relational databases, Adv. Sci. Lett. 21 (2015), 3313–3317.

[16] R. Alfred, G.J. Chiye, Y. Lim, C.K. On, J.H. Obit, A multi-objectives genetic algorithm clustering ensembles based approach to summarize relational data, in SCDS, Int. Conf. Soft Comput. Data Sci. 652 (2016), 113–122.

[17] R. Alfred, Feature transformation: a genetic-based feature construction method for data summarization, Comput. Intell. 26 (2010), 337–357.

[18] X.H. Dang, I. Assent, R.T. Ng, A. Zimek, E. Schubert, Discriminative features for identifying and interpreting outliers, in 2014 IEEE 30th International Conference on Data Engineering, Chicago, IL, USA, 2014, pp. 88–99.

[19] F. Angiulli, F. Fassetti, L. Palopoli, Discovering characterizations of the behavior of anomalous subpopulations, IEEE Trans. Knowl. Data Eng. 25 (2013), 1280–1292.

[20] J. Li, J. Zhang, N. Pang, X. Qin, Weighted outlier detection of high-dimensional categorical data using feature grouping, IEEE Trans. Syst. Man Cybern. Syst. 50 (2018), 4295–4308.

[21] P. Berkhin, A survey of clustering data mining techniques, in: J. Kogan, C. Nicholas, M. Teboulle (Eds.), Grouping Multidimensional Data, Springer, Berlin, Heidelberg, Germany, 2006, pp. 25–71.

[22] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science. 344 (2014), 1492–1496.

[23] Z.H. Zhou, K.J. Chen, H.B. Dai, Enhancing relevance feedback in image retrieval using unlabeled data, ACM Trans. Inf. Syst. 24 (2006), 219–244.

[24] F. Laguzet, A. Romero, M. Gouiffès, L. Lacassagne, D. Etiemble, Color tracking with contextual switching: real-time implementation on CPU, J. Real-Time Image Process. 10 (2015), 403–422.

[25] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Feature Selection for High-Dimensional Data, Springer, 15th floor, Salmon Tower, Manhattan, New York City, USA, 2015.

[26] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (2014), 16–28.

[27] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: a data perspective, ACM Comput. Surv. 50 (2017), 1–94.

[28] J. Tang, H. Liu, Feature selection for social media data, ACM Trans. Knowl. Discov. Data. 8 (2014), 27.

[29] C. Blake, UCI repository of machine learning databases, 1998. http://www. ics. uci. edu/ mlearn/MLRepository. html

[30] F. Sia, Tree-Based Mining Contrast Subspace Method, Universiti Malaysia Sabah, Malaysia, 2020.

[31] B. Micenkov, R.T. Ng, X.H. Dang, I. Assent, Explaining outliers by subspace Separability, in 2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, 2013, pp. 518–527.

[32] J. Ali, R. Khan, N. Ahmad, I. Maqsood, Random forests and decision trees, Int. J. Comput. Sci. Iss. 9 (2012), 272.

[33] I. Rish, An empirical study of the naive bayes classifier, in IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, the venue of the workshop is Seattle, Washington, USA, 2001, pp. 41–46.

[34] S.R. Gunn, Support vector machines for classification and regression, ISIS Tech. Rep. 14 (1998), 5–16.

[35] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explor. Newsl. 11 (2009), 10–18.

[36] K.J. Grimm, G.L. Mazza, P. Davoudzadeh, Model selection in finite mixture models: a k-fold cross-validation approach, Struct. Equ. Model. Multidiscip. J. 24 (2017), 246–256.

[37] S. Arlot, M. Lerasle, Choice of V for V-fold cross-validation in least-squares density estimation, J. Mach. Learn. Res. 17 (2016), 7256–7305.

[38] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, J. Wang, Dendritic neuron model with effective learning algorithms for classification, approximation and prediction, IEEE Trans. Neural Net. Learn. Syst. 30 (2019), 601–614.

[39] J. Jokinen, T. Raty, T. Lintonen, Clustering structure analysis in time-series data with density-based clusterability measure, IEEECAA J. Autom. Sin. 6 (2019), 1332–1343.