IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal
International Research Journal IJNRD Research Through Innovation

MODERN IN CATEGORICAL DATA MERGING

Author: Mr Bani Bhusan Praharaj
Guide: Dr Sushant Ku Das

Chapter 1 Introduction

Clustering is a common and significant technique in various fields of data analysis. The general goal of clustering is to aggregate similar instances (also called 'object- s' interchangeably in this thesis) into a certain number of groups while ensuring dissimilarity between these groups [67] [68].

Clustering analysis can be roughly categorised into two types according to the type of processed data set: numerical data clustering and categorical data clus- tering. Most clustering techniques focus on numerical data clustering because nu- merical features (also called 'attributes' interchangeably in this thesis) have exact values with well-defined distances and are thus more convenient for various types of manipulation, including normalisation, attribute weighting/selection, and distance measurement. The most well-known and commonly used numerical data partition- al clustering algorithm is the k-means algorithm [93], and many variants of this algorithm have been published (see [66] [28] [29] [30] [131] [132] [71]). In contrast, because the possible values of a categorical attribute are categories rather than exact numerical values, the distances between the categories are generally not well-defined [8], which makes the clustering of categorical data more challenging than the cluster- ing of numerical data. The most popular categorical data clustering algorithm in the literature is the k-modes algorithm [63]. Attribute-weighted k-modes [62] and other variants of the k-modes algorithm have also been published, but the distance metric they adopt only assigns binary distances to categories during clustering (i.e., "1" for unequal categories and "0" for identical categories), which cannot reasonably distin- guish the degrees of similarity or dissimilarity between pairs of categories. Although some other categorical data clustering algorithms that adopt more reasonable sim- ilarity metrics or measures have been proposed (e.g., [15] [85] [73]), they do not consider the difference between nominal and ordinal attributes, both of which are very common in categorical data sets. Unlike nominal attributes, ordinal attributes have naturally ordered categories, so treating them in the same way as nominal ones may have an enormous influence on the correctness of the clustering results. Therefore, one focus of this thesis is the design of a categorical data distance metric that is suitable for distance measurement of both nominal and ordinal attributes.

From the perspective of the clustering manner, clustering analysis can be roughly categorised into another two types: partitional clustering and hierarchical clustering [97]. The former type separates objects into a certain number of clusters by maximis- ing the intra-cluster similarity and minimising the objects' inter-cluster similarity, whilst the latter type first assigns each object into an individual cluster and then gradually merges the current most similar cluster pair until the number of clus- ters reaches a pre-set value. In general, the partitional type is more efficient and has been commonly adopted for data analysis tasks. The hierarchical type involves more computation, but it provides nested relationships among data objects and is thus more suitable for finer data analysis [32]. Both partitional and hierarchical clustering clearly have their own merits. However, as far as we know, the existing hierarchical clustering algorithms either have high time complexity (i.e., $O(N^2)$ for the data set with N data objects) or the quality of the constructed hierarchy is unsatisfactory. Moreover, most fast hierarchical clustering approaches are not ap- plicable to categorical data clustering. Therefore, another focus of this thesis is the design of a fast and accurate hierarchical clustering algorithm that is suitable for hierarchical clustering analysis of categorical data.

Section 1.1 - 1.4 provides an in-depth introduction to the research background and motivations. Section 1.5 then reports the main contributions of this thesis. Finally, Section 1.6 provides a brief overview of this thesis.

Ordinal and Nominal Attributes

The attributes that comprise categorical data can be classified into two types: nominal attributes and ordinal attributes [74] [6]. The relationships amongst var- ious data types and attribute types are illustrated in Figure 1.1 One characteristic common to both nominal and ordinal attributes is that their possible val- ues (also called ‘categories’ interchangeably in this thesis) are not exact numerical values and thus are not suitable for processing with arithmetic operations [139]. Therefore, the distances between the categories of both nominal and ordinal at- tributes are generally poorly defined. The most significant difference between nom- inal and ordinal attributes is that the categories of a nominal attribute are un- ordered (e.g., attribute “gender” with categories {male, female}, attribute “shape” with categories {circle, square, star}, etc.), whereas the categories of an ordinal attribute are naturally ordered (e.g., attribute “restaurant rating” with categories {very-poor, poor, marginal, good, very-good}, attribute “paper acceptance/rejection decision” with categories {reject, weak-reject, neutral, weak-accept, accept}, etc). These two types of attributes yield three types of categorical data: those that com- prise only nominal attributes (also called ‘nominal data’ interchangeably in this thesis) and those that comprise only ordinal attributes (also called ‘ordinal data’ interchangeably in this thesis) and those that comprise both nominal and ordinal attributes (also called ‘mixed categorical data’ interchangeably in this thesis). The

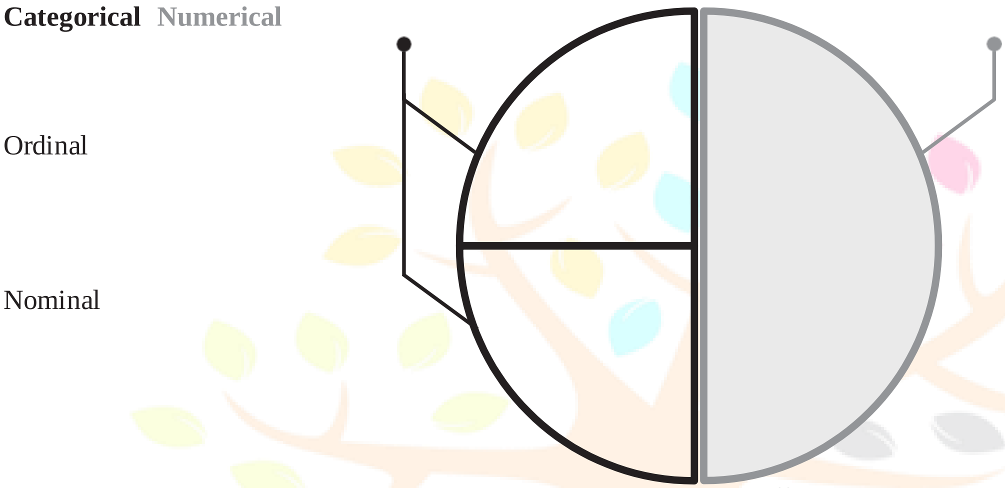


Figure 1.1: Relationships among different data types.

table 1.1: fragment of ta evaluation data set.

# Instances	Attribute 1 (Helpful)	Attribute 2 (Professional)	Attribute (Course)	Attribute (Type)
1	Agree	Agree	Culture	Lecture
2	Disagree	Agree	Oral	Tutorial
3	Marginal	Agree	Culture	Practice
4	Agree	Marginal	Finance	Lab

categorical data in data analysis tasks commonly comprise both nominal and ordi- nal attributes. Many benchmark data sets in the UCI machine learning repository [37], which is the most commonly used database in the field of machine learning and data analysis, contain mixed-categorical data. Table 1.1 demonstrates a fragment of a mixed categorical data set collected from a teaching assistant evaluation system, in which the two ordinal attributes (“Attribute 1” and “Attribute 2”) record the helpfulness and the professional level of each Teaching Assistant (TA), respectively, and the two nominal attributes (“Attribute 3” and “Attribute 4”) record the course served by each TA and the corresponding course type, respectively. Obviously, a desirable clustering algorithm should be able to simultaneously account for the common categorical data characteristics of nominal and ordinal attributes and the differences between the nominal and ordinal attributes for reasonable clustering analysis.

Categorical Data Clustering

Clustering categorical data is a very common data analysis task. Representative cat- egorical data partitional clustering algorithms include the k-modes algorithm [63], the attribute-weighted k-modes algorithm [62] and the attribute-weighted OCIL algorithm [70]. Representative hierarchical clustering approaches include single- linkage, complete-linkage, and average-linkage-based hierarchical clustering [96]. Al- though these hierarchical approaches were not designed for categorical data, they can be modified for categorical data clustering by replacing their adopted Euclidean distance metric with existing categorical data distance/similarity metrics. As far as we know, all existing clustering algorithms that can be applied to categorical data perform clustering analysis under the hypothesis that categorical data comprise only nominal attributes, which is usually unreasonable from a practical view-point.

Table 1.2: An example of TA performance evaluation.

TA Name	Year 2017	Year 2018	Progress Made
Jason	very-bad	good	more
Alex	good	very-good	less

A simple example of TA performance evaluation is shown in Table 1.2 to explain intuitively the problems of treating ordinal attributes as nominal ones. This example demonstrates the performance evaluation results of two TAs named Jason and Alex in 2017 and 2018. The results clearly show that Jason made more progress than Alex. Before the evaluation results are even analysed, we can recognise that the results are ordinal; therefore, we are actually analysing the results on an ordinal scale (i.e., {very-good, good, neutral, bad, very-bad}). On this scale, Jason has progressed by three grades from “very-bad” to “good”, whilst Alex has progressed by just one grade from “good” to “very-good”. However, if we treat the results as nominal values, we can determine only that the evaluation results for both Jason and Alex have changed from 2017 to 2018, but we have no idea how much they have changed or in which direction (i.e., progress or regression).

Because existing partitional and hierarchical categorical data clustering approach- es both treat ordinal attributes as nominal attributes by default during clustering analysis, they will surely twist the natural order information of ordinal attributes and are thus unsuitable for clustering categorical data that comprise ordinal at- tributes. In general, categorical data clustering algorithms rely on the distance measurement of data objects for clustering analysis [72]. As a result, the adopted distance/similarity measures/metrics dominate their clustering performances and are the fundamentals that render them incompetent for the clustering analysis of
categorical data with ordinal attributes.

Categorical Data Metrics

The existing categorical data metrics can be categorised into intra-attribute met- rics and inter-attribute metrics. Intra-attribute metrics measure distances between categories from the same target attribute without considering the relationship be- tween the target attribute and the others, whilst the inter-attribute metrics extract and exploit valuable information from attributes that are correlated to the target attribute for distance measurement. The Hamming distance metric [58] is the most popular and simplest of the existing intra-attribute metrics. It simply assigns the distance “1” to any pair of different categories and assigns the distance “0” to iden- tical categories. Because it cannot distinguish the degrees of dissimilarity for various category pairs and it ignores the relationships between interdependent attributes, the distances it yields are somewhat unreasonable. To overcome the drawbacks of the Hamming distance metric, several inter-attribute metrics have been proposed, including association-based distance metric [80], Ahmad’s distance metric [10], and context-based distance metric [64] [65]. However, these distance metrics treat the information extracted from each correlated attribute equally, which is usually un- reasonable. Moreover, because none of them account for the target attribute’s s- tatistical information, they may fail to measure distances when the attributes are all independent of each other. Therefore, Jia’s distance metric [72] was proposed to simultaneously account for the intra- and inter-attribute information for distance measurement. It also weights the various attributes according to the amount of information they offer.

Nevertheless, the above-mentioned metrics are proposed under the hypothesis that categorical data comprise only nominal attributes. By adopting them for cate- gorical data clustering analysis, the natural order information of ordinal attributes will be twisted and thus result in unsatisfactory clustering performance. We there- fore study the distance measurement problems of ordinal attributes and present an ordinal data distance metric that is suitable for the clustering analysis of ordinal
data in this thesis. In addition, we further propose a unified categorical data dis- tance metric that is expected to 1) inherit the advantages of existing categorical data metrics, 2) have the ability to reasonably exploit the order information of ordinal attributes, 3) measure the distances of nominal and ordinal attributes in a uniform way, and 4) show suitability for the clustering analysis of any type of categorical data, including nominal data, ordinal data, and mixed categorical data.

Fast and Incremental Hierarchical Clustering

Another problem encountered in categorical data clustering is that the time com- plexity of hierarchical clustering approaches is usually very high (i.e., $O(N^2)$ for a data set with N objects). This drawback makes most of the existing hierarchi- cal clustering approaches, including single-linkage-based, average-linkage-based, and complete-linkage-based hierarchical clustering approaches, laborious in the cluster- ing analysis of large-scale or streaming categorical data.

To cope with the problems of large-scale categorical data hierarchical clustering, fast hierarchical clustering algorithms have been proposed to reduce the computation cost of the agglomeration process, which is the most computationally expensive part of hierarchical clustering approaches. A potential-based hierarchical clustering algorithm [91] was proposed to accelerate the agglomeration process with a minimal spanning tree. However, this algorithm saves only a certain computation cost, but its time complexity is still $O(N^2)$. Hashing-based [77], random projection-based [109], and summarisation-based hierarchical clustering approaches [22] [21] [98] [107]
[141] were proposed to improve the time complexity. However, each of these fast hierarchical clustering approaches sacrifices the quality of the constructed hierarchy. Specifically, hashing-based and random projection-based approaches may sacrifice clustering accuracy because their performance is very sensitive to the parameter setting. Summarisation-based approaches construct hierarchies by treating data groups as basic data units; therefore, the detailed hierarchical relationships between specific data objects are lost.

An incremental hierarchical clustering algorithm [121] was proposed to efficient-ly tackle the problem of streaming categorical data clustering analysis. Because this algorithm can dynamically update the constructed hierarchy according to new inputs, it is very efficient for streaming data hierarchical clustering. However, be- cause it does not guarantee a balanced hierarchy, its time complexity is still $O(N^2)$. Moreover, because it approximates a single-linkage-based hierarchical clustering ap- proach, it has bias for certain types of data distribution.

Because none of the existing hierarchical clustering algorithms can achieve both satisfactory efficiency and hierarchy quality in the clustering analysis of categori- cal data, another focus of this thesis is to design a fast and accurate hierarchical clustering approach. It is expected to have lower time complexity than $O(N^2)$; it should not sacrifice the hierarchy quality in comparison with the state-of-the-art fast hierarchical clustering approaches; and it should be efficient for clustering analysis of streaming categorical data.

Main Contributions

This thesis focuses mainly on two significant issues in categorical data clustering analysis: distance measurement of categorical data, especially categorical data that comprise ordinal attributes, and fast and accurate hierarchical clustering of large- scale and streaming categorical data. The main contributions of this thesis can be summarised from four aspects.

An ordinal data distance metric, which can reasonably quantify the distances between ordinal categories from the same attribute according to both the or- der relationship among the categories and the statistical information of the target attribute, was designed from the perspective of information theory. It uses the entropy values of categories to indicate their information amount and simulates human thinking procedures to quantify the distances according to the categories’ entropy values. For this simulation, we quantify the distance between two categories according to the cumulative entropy values of these two categories and all the other categories ordered between them. In this way, the proposed metric can correctly preserve the order relationship between the categories during the distance measurement. It is parameter-free and easy to use, and experimental results have illustrated its effectiveness.

To handle the mixed categorical data clustering problem, we further propose a unified distance metric that defines distances for ordinal and nominal at- tributes in a uniform manner to avoid the information loss caused by combin- ing the distances measured for ordinal and nominal attributes to obtain the distances between data objects. To achieve a more reasonable distance mea- surement, we also provide a unified attribute weighting mechanism to weight the importance of each attribute. Therefore, the concepts of distance and at- tribute weight are unified for both nominal and ordinal attributes. The unified metric remains parameter-free as the proposed ordinal data distance metric. More importantly, it is suitable for any type of categorical data clustering and has the same time complexity as most state-of-the-art categorical data distance/similarity metrics. Experimental evaluations demonstrate that the unified distance metric is competitive in the clustering analysis of nominal da- ta, and it obviously outperforms the existing metrics in the clustering analysis of ordinal data and mixed categorical data.

To achieve fast and accurate hierarchical clustering of categorical data, a Grow- ing Multi-layer Topology Training (GMTT) algorithm is proposed that can train a topology to efficiently represent the structure of a data set. The topol- ogy is self-organised, and its number of layers and nodes depends on the corresponding data set. In the topology, each node represents a group of similar data objects, and the nodes are linked and located in various layers of the topology, which indicates the similarity level of various pairs of object groups. The most important property of the GMTT topology is that the data objects in the same group represented by a node located in a deeper layer of the topol- ogy have more similarity to each other, which is consistent with the expected hierarchy of hierarchical clustering approaches. Thus, GMTT topology can be used to guide the rapid search of the most similar cluster pair during the agglomeration procedure of hierarchical clustering. That is, the most similar pair of clusters can be identified by locally searching the similarity between data objects represented by a same node and the similarity between the object groups indicated by the nodes in the same layer from the bottom to the top of the topology. In this way, the most computationally expensive process, that is, finding and merging the most similar pair of clusters, can be converted from a global search task to a local one. The GMTT-based fast hierarchical cluster- ing approach has less time complexity $O(N^{1.5})$ than most existing hierarchical clustering approaches. Moreover, experimental results have demonstrated the effectiveness and efficiency of a GMTT-based approach.

To cope with the problem of large-scale streaming categorical data hierarchi- cal clustering, an incremental version of the GMTT algorithm (i.e., IGMTT algorithm) is further proposed. The difference between GMTT and IGMTT is that IGMTT uses the former part of streaming inputs to train a coarse topology and that the hierarchy of the present inputs is constructed according to the coarse topology. The topology is then updated dynamically and locally according to each of the following inputs. Meanwhile, the hierarchy of the inputs is also updated dynamically and locally according to the topology. In this manner, restructuring of the whole hierarchy is not required after adopting each new input. Instead, only a small part of the hierarchy is updated when this part is detected to violate the construction rule of the expected hierarchy. The time complexity of the IGMTT-based incremental hierarchical cluster- ing approach remains $O(n^{1.5})$, and experimental results demonstrate that its clustering performance is competitive with each of the hierarchical clustering approaches, including the GMTT-based approach.

Organisation

The remainder of this thesis is organised as follows. Chapter 2 introduces the existing studies related to categorical data clustering, including partitional and hierarchical clustering algorithms, intra- and inter-attribute categorical data distance measures, nominal and ordinal inter-attribute dependence measures, and validity indices for performance evaluation of categorical data clustering. Chapter 3 examines the dis- tance measurement problems of ordinal attributes, and provides a distance metric for ordinal data clustering. In Chapter 4, the ordinal data distance metric is further generalised into a unified metric for ordinal-and-nominal-attribute categorical data clustering. A unified interdependence measure is also presented to weight the contri- butions of various attributes. Chapter 5 then proposes a GMTT topology training algorithm and its incremental version for rapid and incremental hierarchical cluster- ing. By adopting the distance metric proposed in Chapter 4, these two algorithms can be utilized for the hierarchical clustering of any-type of categorical data. Final- ly, Chapter 6 concludes this thesis and discusses some potential directions for future research.

clustering algorithms, intra- and inter-attribute categorical data distance measures, nominal and ordinal inter-attribute dependence measures, and validity indices for performance evaluation of categorical data clustering. Chapter 3 examines the dis- tance measurement problems of ordinal attributes, and provides a distance metric for ordinal data clustering. In Chapter 4, the ordinal data distance metric is further generalised into a unified metric for ordinal-and-nominal-attribute categorical data clustering. A unified interdependence measure is also presented to weight the contri- butions of various attributes. Chapter 5 then proposes a GMTT topology training algorithm and its incremental version for rapid and incremental hierarchical cluster- ing. By adopting the distance metric proposed in Chapter 4, these two algorithms can be utilized for the hierarchical clustering of any-type of categorical data. Final- ly, Chapter 6 concludes this thesis and discusses some potential directions for future research.

Chapter 2

Literature Review of Related Works

In this chapter, existing categorical data clustering algorithms, categorical data metrics, inter-categorical-attribute dependence measures that are related to the pro- posed methods, and validity indices for performance assessment are reviewed. The common definitions and notations are also provided. Other specific notations and definitions about the proposed methods are presented in each chapter accordingly.

Categorical Data Clustering

Existing related partitional clustering algorithms (i.e., k-modes [63], attribute-weighted k-modes [62], and attribute-weighted OCIL [70] algorithms) and hierarchical clus- tering algorithms (i.e., potential-based [91], random projection-based [109], and in- cremental hierarchical clustering algorithms [121]) that are applicable to categorical data clustering are introduced in this section.

Partitional Categorical Data Clustering Algorithms

This part reviews the representative partitional algorithms proposed for categorical data clustering, including k-modes [63], weighted-k-modes [62], and weighted OCIL [70] algorithms. Common notations and general description of partitional clustering are presented as follows. Given a data set **X** with N data objects $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, and the number of clusters k, the goal of partitional clustering is to maximise the value of the objective function Z with variable **Q**:

$$Z(\mathbf{Q}) = \sum_{t=1}^{k}\sum_{i=1}^{N} q_{i,t} \cdot Sim(\mathbf{x}_i, \mathbf{C}_t), \qquad (2.1.1)$$

where $\mathbf{Q} = q_{i,t}$ with $i \in \{1, 2, ..., N\}$ and $t \in \{1, 2, ..., k\}$, is an $N \times k$ matrix. $q_{i,t} = 1$ ($q_{i,t} = 0$) indicates that data object $\mathbf{x}_i$ belongs (does not belong) to cluster $\mathbf{C}_t$. Thus, each row of **Q** satisfy $\sum_{t=1}^{k} q_{i,t} = 1$. $Sim(\mathbf{x}_i, \mathbf{C}_t)$ is a function, which measures the similarity between data object $\mathbf{x}_i$ and cluster $\mathbf{C}_t$.

K-Modes Clustering Algorithm

The most famous and commonly used partitional clustering algorithm is k-means [93]. A lot of variants of it have been presented in the literature, see [63] [15] [28] [85] [62] [29] [30] [73]. Among these variants, k-modes [63] is the most popular one for categorical data clustering. It initializes k modes $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k\}$ from the data set **X**, and partition the whole data set according to the k modes. The goal of k-modes is to minimise the objective function Z with variable **Q** and **U**:

$$Z(\mathbf{Q}, \mathbf{U}) = \sum_{t=1}^{k}\sum_{i=1}^{N} q_{i,t} \cdot Dist(\mathbf{x}_i, \mathbf{u}_t). \qquad (2.1.2)$$

This objective function is minimised by solving the following two problems: 1) fix

$\mathbf{U} = \hat{\mathbf{U}}$, solve the minimisation problem $Z(\mathbf{Q}, \hat{\mathbf{U}})$, and 2) fix $\mathbf{Q} = \hat{\mathbf{Q}}$, solve the minimisation problem $Z(\hat{\mathbf{Q}}, \mathbf{U})$. To solve the first problem, each data object is assigned to a cluster $\mathbf{C}_t$, where t is decided by

$$t = \arg\min_{m} Dist(\mathbf{x}_i, \mathbf{u}_m). \qquad (2.1.3)$$

According to Eq. (2.1.3), we assign $q_{i,t} = 1$ for each object $\mathbf{x}_i$. In this way, the whole data set is partitioned into k clusters, and the optimal $\mathbf{Q}$ is obtained based on $\hat{\mathbf{U}}$.

After the partition, each mode is updated according to the present $\hat{\mathbf{Q}}$ by

$$u_t^r = \arg\max_{o_h^r} \sigma_{o_h^r}(\mathbf{x}_i \in \mathbf{C}_t), \quad (2.1.4)$$

Algorithm 1 K-modes Clustering Algorithm

```
Input: Data set $\mathbf{X}$ and number of clusters $k$.
Output: $k$ clusters described by $\mathbf{Q}$.
/*initialize the values of $\mathbf{Q}$ and $\mathbf{U}$*/
Randomly select $k$ modes $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k\}$ from the data objects of $\mathbf{X}$;
Set $Change = 1$;
while $Change = 1$ do
    Set all values in $\mathbf{Q}$ to 0;
    /*partition the whole data set once*/
    for $i = 1$ to $N$ do
        Find the mode $\mathbf{u}_t$ with shortest distance to $\mathbf{x}_i$ according to Eq. (2.1.3);
        Assign $q_{i,t} = 1$;
    end for
    /*judge if the data set should be partitioned again*/
    if the present $\mathbf{Q}$ equals to the $\mathbf{Q}$ obtained in the last partition epoch then
        $Change = 0$;
    else
        Update $\mathbf{U}$ according to Eq. (2.1.4);
    end if
end while
```

where u_t^r is the r^{th} value of $\mathbf{u}_t$, and $\sigma_{o_h^r}(\mathbf{x}_i \in \mathbf{C}_t)$ counts the number of objects $\mathbf{x}_i$ in $\mathbf{C}_t$ with their r^{th} values equal to o_h^r. In this way, the optimal $\mathbf{U}$ is obtained based on $\hat{\mathbf{Q}}$. These two problems are solved iteratively until convergence. K-modes is summarised as Algorithm 1.

Attribute-Weighted Clustering Algorithms

K-modes treats each attribute equally, which is not always reasonable. To solve this problem, attribute-weighted versions of k-modes [62] [73] [31] are presented in the literature. Here, we review the most representative one proposed in [62]. Its objective function incorporates the weights of attributes by

$$Z(\mathbf{Q}, \mathbf{U}, \mathbf{W}) = \sum_{t=1}^{k} \sum_{i=1}^{N} \sum_{r=1}^{d} q_{i,t} \cdot w_r^{\beta} \cdot Dist(x_i^r, u_t^r), \quad (2.1.5)$$

where $\mathbf{W} = \{w_1, w_2, ..., w_d\}$ are the weights of attributes. Similar to the k-modes, this objective function is minimised by solving the following three problems: 1) fix $\mathbf{U} = \hat{\mathbf{U}}$ and $\mathbf{W} = \hat{\mathbf{W}}$, solve the minimisation problem $Z(\mathbf{Q}, \hat{\mathbf{U}}, \hat{\mathbf{W}})$, 2) fix $\mathbf{Q} = \hat{\mathbf{Q}}$ and $\mathbf{W} = \hat{\mathbf{W}}$, solve the minimisation problem and $Z(\hat{\mathbf{Q}}, \mathbf{U}, \hat{\mathbf{W}})$, and 3) fix $\mathbf{Q} = \hat{\mathbf{Q}}$

$\mathbf{U} = \hat{\mathbf{U}}$, solve the minimisation problem $Z(\hat{\mathbf{Q}}, \hat{\mathbf{U}}, \mathbf{W})$. The former two problems can be solved in the same way as k-modes, and the third problem is solved by

□□ □ 0, *if* $D_r = 0$

$w_r =$ (2.1.6)

□ □ Σ 1 $_1$, $_{d\ s=1}$ ~~*if* D_r~~ 0,

$(\frac{D_r}{D_s})$ 6 1

where

$k \quad N$

$D_r = \Sigma \Sigma \; q_{i,t} \cdot Dist(x^r, u^r).$ (2.1.7) $\quad i \quad t$

$t{=}1 \quad i{=}1$

Subspace Clustering Algorithms

K-modes and attribute-weighted k-modes assume that each attribute has identical contribution in forming different clusters, which is usually not the case in practice. Therefore, subspace partitional clustering algorithms are proposed in the litera- ture, including hard subspace clustering algorithms [5] [27] [3] [4] [126] [87] and soft subspace clustering algorithms [95] [47] [38]. However, all the above-mentioned sub- space clustering algorithms are proposed for numerical data only. Recently, more subspace clustering algorithms have been proposed for categorical data, including hard approaches, see [50] [76] [51] [129] [70], and soft approaches, see [14] [23] [26]. Here, we discuss the attribute-weighted OCIL proposed in [70], which is the most comprehensive one among the state-of-the-art subspace clustering algorithms that are applicable to categorical data. Under the scenario of categorical data clustering, the objective function of WOCIL is

$k \quad N$

$Z(\mathbf{Q}, \mathbf{W}) = \Sigma \Sigma \; q_{i,t} \cdot Sim(\mathbf{x}_i, \mathbf{C}_t),$ (2.1.8) $\quad t{=}1 \quad i{=}1$

where $Sim(\mathbf{x}_i, \mathbf{C}_t)$ is defined as

$Sim(\mathbf{x} , \mathbf{C}) = \frac{1}{} \Sigma \; w$ $\quad d \quad \cdot Sim(x , \mathbf{C}).$ (2.1.9)

r

$i \quad t \quad d \quad r,t \quad i \quad t$

$r{=}1$

$Sim(x^r, \mathbf{C}^r)$ is defined as

$i \quad t$

$Sim(x^r, \mathbf{C}^r) = \frac{\sigma x_r (\mathbf{x}j \in \mathbf{C}t)}{}$, (2.1.10) $\quad \Sigma^{i} \; \overline{qj,t}$

$i \quad t \quad N$

$j{=}1$

where $\sigma_{x^r}(\mathbf{x}_j \in \mathbf{C}_t)$ counts the number of objects $\mathbf{x}_j$ belonging to cluster $\mathbf{C}_t$ with their r^{th} values equal to x^r. $w_{r,t}$ is defined as i

$w \quad = \frac{Fr,t \cdot Mr,t}{}$, $\quad i \quad$ (2.1.11)

$r,t \quad d \; s{=}1 \quad \Sigma \quad Fs,t \quad \cdot Ms,t$

where $F_{r,t}$ and $M_{r,t}$ indicates the inter-cluster difference and intra-cluster similarity, respectively. $F_{r,t}$ is defined as

$'.\Sigma v_r$ $\quad !_2$

$F \quad = \quad , \quad r,t \quad \sqrt{\frac{1}{2}} \quad m{=}1 \; \sigma_{or} \; (\mathbf{x}_i \in \mathbf{C}_t) - \sigma_{or} \; (\mathbf{x}_j \; \in\!/ \; \mathbf{C}_t) \; \Sigma \; \Sigma$, (2.1.12) $\quad N \; i{=}1 \; q_{i,t} \quad N \; i{=}1 \; q_{i,t}$

where v_r is the number of categories of attribute A_r, and o^r_m is the m^{th} category of A_r. $F_{r,t}$ is actually the Hellinger distance derived from the Bhattacharyya coefficient [18] for quantifying the dissimilarity between two probability distributions [101] [17]. $M_{r,t}$ is defined as

$$M_{r,t} = \frac{\sum_{i=1}^{N} (q_{i,t} \cdot Sim(x^r_i, \mathbf{C}^r_t))}{\sum_{i=1}^{N} q_{i,t}}. \qquad (2.1.13)$$

Attribute-weighted OCIL is summarised as Algorithm 2.

Fast Hierarchical Clustering Approaches

This section will give an overview of the representative fast hierarchical clustering approaches, including potential-based [91], random projection-based [109], and incremental [121] hierarchical clustering approaches. Common notations and general description of hierarchical clustering tasks are provided as follows. Given a data set $\mathbf{X}$ with N data objects $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, each data object is assigned into an individual cluster by $\mathbf{C}_1 = \mathbf{x}_1, \mathbf{C}_2 = \mathbf{x}_2, ..., \mathbf{C}_N = \mathbf{x}_N$. The most similar pair of clusters $\mathbf{C}_a$ and $\mathbf{C}_b$ are selected out according to a certain criteria and are merged to form

Algorithm 2 Attribute-weighted OCIL Clustering Algorithm

```
Input: Data set X and number of clusters $k$.
Output: $k$ clusters described by Q.
/*initialize the values of Q and U*/
Randomly select $k$ modes $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k\}$ from the data objects of X;
Set $Change = 1$;
while $Change = 1$ do
    Set all values in Q to 0;
    /*partition the whole data set once*/
    for $i = 1$ to $N$ do
        Find the cluster $c_t$ that is the most similar to $\mathbf{x}_i$ according to Eq. (2.1.9);
        Assign $q_{i,t} = 1$;
    end for
    /*judge if the data set should be partitioned again*/
    if the present Q equals to the Q obtained in the last partition epoch then
        $Change = 0$;
    else
        Update W according to Eq. (2.1.11);
    end if
end while
```

a new cluster $\mathbf{C}_{\{a,b\}}$, which can be expressed as $\mathbf{C}_{\{a,b\}} = \mathbf{C}_a \cup \mathbf{C}_b$. This merging process is repeated until all the clusters are merged to form one cluster $\mathbf{C}_{\{1,2,...,N\}}$, or a pre-set number of clusters k is reached. The merging process is recorded by $\mathbf{H}$, called hierarchy or dendrogram, which is usually visualized with a tree.

Potential-based Hierarchical Clustering

The approach proposed in [91] converts the distance between data objects into potential values to measure the density levels of data objects. Having the potential value of each object, an Edge Weighted Tree (EWT) is constructed, and the hierarchy can be easily read off from it. Specifically, given two objects $\mathbf{x}_i$ and $\mathbf{x}_j$ with distance $Dist(\mathbf{x}_i, \mathbf{x}_j)$, the potential value of $\mathbf{x}_i$ received from $\mathbf{x}_j$ and the potential value of $\mathbf{x}_j$ received from $\mathbf{x}_i$ are the same, which is given by

$$\theta_{\mathbf{x}_i,\mathbf{x}_j} = \theta_{\mathbf{x}_j,\mathbf{x}_i} = \begin{cases} \dfrac{1}{Dist(\mathbf{x}_i,\mathbf{x}_j)} & \text{if } Dist(\mathbf{x}_i, \mathbf{x}_j) \geq \lambda \\ \square\square -\dfrac{1}{\lambda} & \text{if } Dist(\mathbf{x}_i, \mathbf{x}_j) < \lambda, \end{cases} \qquad (2.1.14)$$

where the parameter λ is used to avoid the singularity problem when the value of $Dist(\mathbf{x}_i, \mathbf{x}_j)$ is too small. The total potential value of a data point $\mathbf{x}_i$ is defined as

$$\vartheta_{\mathbf{x}_i} = \sum_{j=1,j\neq i}^{N} \theta_{\mathbf{x}_i,\mathbf{x}_j}, \qquad (2.1.15)$$

which is the sum of $\mathbf{x}_i$'s potential values received from all of the other data objects. Potential value of an object indicates its density level among all the objects. That is, an object with closer neighbors locates in a high-dense region in the data set [140] [84] [90]. According to the potential values of objects, an EWT is constructed by linking objects to its closest neighbor with a higher potential value. Then, the hierarchy of the data set can be read off from the EWT by sequentially merging the linked pair with the closest distance. The potential-based approach is summarised as Algorithm

3. The most computationally expensive procedure of hierarchical clustering (i.e., searching the most similar pair of clusters) is accelerated by PHC, because the very dissimilar pairs of clusters are ignored during the EWT-based searching. However, time complexity of PHC is still $O(n^2)$, which is the same as conventional hierarchical clustering algorithms.

Random Projection-based Hierarchical Clustering

Random projection-based hierarchical clustering approaches [109] improved the time complexity of hierarchical clustering to $O(n(\log n)^2)$. It partitions the entire data set into small-enough subsets, and guarantees that the data objects inside a same subset are very similar to each other. According to these subsets, the most similar pair of clusters can be found locally to reduce the computation cost of hierarchical clustering. To obtain the subsets, data objects are randomly projected onto different lines for splitting [134] [118]. After each projection, the original object set is split into two smaller subsets. The splitting is stopped when the number of objects of

Algorithm 3 Potential-based Hierarchical Clustering Algorithm

```
Input: Data set $\mathbf{X}$ and number of clusters $k$.
Output: Hierarchy $\mathbf{H}$.
/*compute potential values for each data object*/
for $i = 1$ to $N$ do
    Compute potential $\vartheta_{\mathbf{x}_i}$ by Eq. (2.1.15);
end for
/*link all the data objects to form an EWT*/
for $i = 1$ to $N$ do
    Link $\mathbf{x}_i$ and $\mathbf{x}_p$, where $\mathbf{x}_p$ is the nearest one to $\mathbf{x}_i$ among all the objects with higher potential value than $\mathbf{x}_i$;
end for
/*merge data objects according to EWT to form $\mathbf{H}$*/
Assign each data object with an individual cluster;
for $m = 1$ to $N - k$ do
    Find the pair of clusters $\mathbf{C}_a$ and $\mathbf{C}_b$ with the shortest edge in EWT;
    Merge the two clusters to form a new one by $\mathbf{C}_{\{a,b\}} = \mathbf{C}_a \cup \mathbf{C}_b$;
    Remove the edge between $\mathbf{C}_a$ and $\mathbf{C}_b$ from EWT;
end for
Visualize the merging process as a hierarchy $\mathbf{H}$.
```

each subset is smaller than a pre-set parameter *minPts*. After the splitting, each subset contains only a small number of very similar objects. It is guaranteed that each pair of the closest objects that will be found for merging during the hierarchical clustering process are partitioned into the same subset. Finally, all the intra-subset similarity values between object pairs are ranked, and objects are merged according to the similarity ranking. Procedures of the random projection-based framework with single-linkage merging strategy is summarised as Algorithm 4. Average-linkage can also be chosen as the merging strategy for the random projection-based frame- work. However, an improper parameter *minPts* may make random projection-based framework fail to produce a hierarchy with pre-set number of clusters. Thus, a

Algorithm 4 Random Projection-based Single-linkage Algorithm

```
Input: Data set $\mathbf{X}$, number of clusters $k$, and threshold $minPts$.
Output: Hierarchy $\mathbf{H}$.
/*partition $\mathbf{X}$ into small subsets with very similar data objects*/
Perturb the data objects;
while subset with number of objects larger than $minPts$ exists do
    Partition this subset using random projection;
end while
/*form the similarity ranking of object pairs*/
Compute similarity for all the intra-subset object pairs;
Sort all the computed similarities;
/*merge data objects according to the similarity ranking*/
Assign each data object with an individual cluster;
for $m = 1$ to $N - k$ do
    Merge the pair of clusters with their similarity ranked first in the similarity ranking;
    Remove the similarity ranked first in the similarity ranking;
end for
Visualize the merging process as a hierarchy $\mathbf{H}$;
```

parameter-free versions of random projection-based framework has also been pro- posed in [109]. It solves the parameter selection problem by repeatedly performing the random projection-based hierarchical clustering with different *minPts* values, until the desired hierarchy is correctly produced. However, this approach is designed for numerical data only and cannot be applied for fast categorical data hierarchical clustering.

Incremental Hierarchical Clustering

To cope with streaming data, several clustering approaches have been proposed in [81] [44] [45] [19] [40] [121]. Among these works, only the approach presented in [121] focuses on the hierarchical clustering of streaming data. This approach processes each input data object in the following three steps: 1) search the known objects to find the one that is closest to the new input, 2) start from the found object, the present hierarchy is detected in a bottom-up manner to find a proper node to accept the new input, and 3) detect the inhomogeneous region of the hierarchy, and restructure it in a top-down manner. Specifically, for a new input $\mathbf{x}_i$, its nearest neighbor $\mathbf{x}_j$ is found from the leaf nodes of the present hierarchy. Then, the upward searching is performed to $\mathbf{x}_j$'s parent node V_p. If the distance $Dist(\mathbf{x}_i, \mathbf{x}_j)$ between $\mathbf{x}_i$ and $\mathbf{x}_j$ is smaller than the upper limitation and larger than the lower limitation of V_p, V_p is judged to be homogeneous after accepting $\mathbf{x}_i$. In this case, $\mathbf{x}_i$ can be simply inserted under V_p. If $Dist(\mathbf{x}_i, \mathbf{x}_j)$ is smaller than the lower limitation of V_p, it indicates that $\mathbf{x}_i$ and $\mathbf{x}_j$ form a region with higher density among the nodes or objects under V_p. In this case, a new node should be inserted under V_p to be the parent node of $\mathbf{x}_i$ and $\mathbf{x}_j$ in order to maintain the homogeneity of the hierarchy. If $\mathbf{x}_i$ and $\mathbf{x}_j$ form a region with lower density among the nodes or objects under V_p (i.e. $Dist(\mathbf{x}_i, \mathbf{x}_j)$ is larger than the upper limitation of V_p), detection should be performed on V_p's parent node, grandparent node, and so on until reaching a node, which can accept $\mathbf{x}_i$ without influencing its homogeneity. Then, downward inhomogeneous detection and recovery are performed to guarantee that the whole hierarchy is homogeneous after accepting $\mathbf{x}_i$. The incremental hierarchical clustering algorithm is summarised in Algorithm 5. Since the merging strategy of the incremental algorithm approximates the traditional single-linkage, the incremental algorithm is biased towards certain types of data distribution. The incremental algorithm is sensitive to the input ordering of the streaming data, and still has time complexity $O(N^2)$ in the worst case (i.e., the produced hierarchy is extremely imbalanced). Moreover, this approach is designed for numerical data only and cannot be applied for streaming categorical data hierarchical clustering.

Distance Measurement

Five distance metrics (i.e., Hamming distance metric [58], association-based dis- tance metric [80], Ahamd's distance metric [10], context-based distance metric [64]

Algorithm 5 Incremental Hierarchical Clustering Algorithm

```
Input: Streaming data set $\mathbf{X}$ and initialized hierarchy $\mathbf{H}$.
Output: Hierarchy $\mathbf{H}$.
for $i = 1$ to $N$ do
    /*find a proper node from the present hierarchy to accept each new input*/
    Find the nearest neighbor (i.e., $\mathbf{x}_j$ under a node $V_p$) of $\mathbf{x}_i$ from the leaf nodes of the present $\mathbf{H}$;
    while $\mathbf{x}_i$ and $\mathbf{x}_j$ cause a low-dense region do
        Perform upward searching by treating $V_p$ as the nearest neighbor of $\mathbf{x}_i$;
    end while
    /*insert each new input under the proper node*/
    if $\mathbf{x}_i$ and $\mathbf{x}_j$ cause a high-dense region then
        Create a new node under $V_p$ as the parent node of $\mathbf{x}_i$ and $\mathbf{x}_j$;
    else
        Directly insert $\mathbf{x}_i$ under $V_p$;
    end if
    /*adjust the present hierarchy to make it homogeneous*/
    Detect inhomogeneous regions of the present $\mathbf{H}$;
    Recover the inhomogeneous regions of $\mathbf{H}$, and obtain the homogeneous $\mathbf{H}$.
end for
```

[65], and Jia's distance metric [72]) proposed for the distance measurement of cat- egorical data are discussed in this section. Common notations and general de- scription of categorical data distance measurement are provided as follows. Giv- en a data set $\mathbf{X}$ with N data objects $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ represented by d attributes $A_1, A_2, ..., A_d$. Each attribute has a certain number of categories (e.g., A_r has v_r categories $\{o^r_1, o^r_2, ..., o^r_{v_r}\}$). Each data object is described by d values (e.g., $\mathbf{x}_i = \{x^1_i, x^2_i, ..., x^d_i\}$), each value is a possible value (category) of an attribute (i.e., $x^1_i \in \{o^1_1, o^1_2, ..., o^1_{v_1}\}$, $x^2_i \in \{o^2_1, o^2_2, ..., o^2_{v_2}\}$, ..., $x^d_i \in \{o^d_1, o^d_2, ..., o^d_{v_d}\}$). The distance between the rth values of two objects $\mathbf{x}_i$ and $\mathbf{x}_j$ is expressed as $Dist(x^r_i, x^r_j)$, and the overall distance between the two objects is expressed as $Dist(\mathbf{x}_i, \mathbf{x}_j)$.

Hamming Distance Metric

Hamming distance metric [58] is simple and popular for categorical data analysis. It uniformly assigns distance “1” to a pair of different values while assigns distance “0” to a pair of identical values. Specifically, the distance between the r^{th} object values of $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as

$$Dist(x_i^r, x_j^r) = \begin{cases} 1, & if\ x_i^r /= x_j^r \\ 0, & if\ x_i^r = x_j^r. \end{cases} \quad (2.2.16)$$

Accordingly, the distance between two objects $\mathbf{x}_i$ and $\mathbf{x}_j$ is obtained by

$$Dist(\mathbf{x}_i, \mathbf{x}_j) = \sum_{r=1}^{d} Dist(x_i^r, x_j^r). \quad (2.2.17)$$

Because Hamming distance metric only produces two distance values (i.e., “0” and “1”), it is incapable to distinguish the distance between different pairs of categories, and will thus completely ignore the order relationships among ordinal categories. Moreover, it treats each attribute equally, and does not consider the relationships among the attributes, which are unreasonable from the practical view-point [127] [13].

Association-based Distance Metric

To extract valuable information from correlated attributes for more accurate dis- tance measurement, association-based distance metric is proposed in [80]. It adopts the idea that if the probability distributions of the corresponding values from an- other attribute of two categories are dissimilar to each other, distance between the two categories will be larger. Specifically, distance between two object values x_i^r and x_j^r is calculated by

$$Dist(x_i^r, x_j^r) = \sum_{s=1,s \neq r}^{d} Dist(cpd(A_s|A_r = x_i^r), cpd(A_s|A_r = x_j^r)) \quad (2.2.18)$$

i j

where $cpd(A_s|A_r = x^r)$ and $cpd(A_s|A_r = x^r)$ are the conditional probability distributions of the co-occurred values from A_s of x^r and x^r, respectively. $Dist(cpd(A_s|A_r = x^r), cpd(A_s|A_r = x^r))$ is the distance between the two probability distributions. In practice, Kullback-Leibler divergence method [79] [78] is utilized to compute the distance between two probability distributions. Accordingly, $Dist(x^r, x^r)$ can be

i j

written as

v_s □ □ $p(o^s|x^r)$ $p(o^s|x^r)$ □ □

Σ^d Σ

h j

$s=1,s/=r$ $h=1$

$Dist(x^r, x^r) =$ □ $p(o^s|x^r)$ log h i $+p(o^s|x^r)$ h j □ (2.2.19)

log

i j □ h i 2 $p(o^s|x^r)$ h j 2 $p(o^s|x^r)$ □

where $p(o^s|x^r)$ is the conditional probability obtained by $p(o^s|x^r) = \frac{\sigma_{o^s \wedge x^r}(\mathbf{X})}{\sigma_{x^r}(\mathbf{X})}$.

i h i h i

i i

$\sigma_{x^r}(\mathbf{X})$ is an operation fetches the number of objects in **X** with their r^{th} values equal to x^r, and $\sigma_{o^s} \wedge x_r(\mathbf{X})$ fetches the number of objects in **X** with their r^{th} values equal to x^r and s^{th} values equal to o^s .

h

Ahmad's Distance Metric

Later, Ahmad's distance metric [10] is proposed based on the definition of similarity given by [1]. In essence, it adopts the same basic idea as the association-based distance metric. The difference is, Ahmad's distance metric calculates the distance between two categories according to their separating power [116] [33] [9], which makes Ahmad's distance metric more powerful in distinguishing the distances among similar categories. Specifically, the distance between two object values x^r and x^r is calculated by

r r $\frac{1}{}$ Σ i j !

x_r x_r

$Dist(x_i, x_j) =$ $d-1$ $s=1,s$ $\max(p^i(\mu) + p^j(\sim\mu) - 1)$, (2.2.20)

r

i d

where μ is a set of categories comprise A 's categories, called supporting set. $p^{x_r}(\mu)$

s i i j

$(p^{x_r}(\sim\mu))$ is the occurrence probabilities of data objects in **X** with their r^{th} values equal to x^r (x^r) and s^{th} values equal (unequal) to a category in μ. By finding the

x_r x_r

supporting set that makes the value of $p_i(\mu) + p^j(\sim\mu) - 1$ reaches the maximum, the distance between two object values x^r and x^r can be obtained.

i j

Context-based Distance Metric

All the above-mentioned metrics treat each attribute equally, which is not always reasonable. Therefore, context-based distance metric is proposed in [64] [65] to calculate distance between two categories from a target attribute according to the selected relevant attributes, which are called context. In practice, attributes belong- ing to the context of a target attribute A_r is selected according to the symmetrical uncertainty defined in [128]. For two attributes A_r and A_s, the symmetrical uncer- tainty of them is defined as

$SU\ (A\ \ , A\ \ ) = 2 \cdot \frac{IG(Ar|As)}{}$

$_{r} \quad _{s}$ (2.2.21)

$E(A\ ) + E(A\ )$

$r \quad s$

where $E(A_r)$ and $E(A_s)$ are the entropy of attributes A_r and A_s, respectively. $\mathrm{IG}(A_r|A_s)$ is the information gain, which is given by

$IG(A_r|A_s) = E(A_r) - E(A_r|A_s)$ (2.2.22)

where $E(A_r)$ and $E(A_r|A_s)$ are defined as

v_r

Σ

$m \quad m$

$m{=}1$

$E(A_r) = -$

and

$h \quad m \quad h \quad v_s \quad v_r \quad m \quad h$

$E(A_r|A_s) = -\Sigma\, p(o^s\ )\, \Sigma\, p(o^r\ |o^s\ ) \log(p(o^r\ |o^s\ )),$ (2.2.24)

$h{=}1 \quad m{=}1$

respectively. Subsequently, the context of an attribute A_r is selected out by

$\underline{Cont(A\ \ ) = \{A\ \ |s =/}\quad r, SU\ (A\ \ , A\ \ ) \geq \beta \cdot \frac{\Sigma s,s{=}/\ \ r\ \ SU\ (Ar, As)}{d-1}\},$ (2.2.25)

$r \quad s \quad r \quad s$

where β is a parameter in the interval [0, 1]. Then, the distance between two object values x^r and x^r is calculated according to the context by

$i \quad j$

$.' \quad \Sigma \quad v_s$

Σ

$i \quad j \quad i \quad h \quad j \quad h$

$A_s \in Cont(A_r)\ h{=}1$

$Dist(x^r, x^r) = ,\qquad (p(x^r|o^s\ ) - p(x^r|o^s\ ))^2$ (2.2.26)

Jia’s Distance Metric

Association-based, Ahmad’s, and context-based metrics measure distances between categories according to the other attributes. Therefore, when the attributes are independent of each other, they may fail to measure the distances. To solve this problem, Jia’s distance metric is proposed in [72], which measures the distances by simultaneously considering the target attribute and the other attributes that are highly dependent with the target one. Specifically, the dependence degree be- tween two attributes A_r and A_s is measured by calculating their interdependence redundancy, which is defined as

$R(A\ , A\ ) = \frac{I(Ar, As)}{E(A\ , A\ )}$, (2.2.27)

$r \quad s$

$r \quad s$

where $I(A_r, A_s)$ is the mutual information [92] between A_r and A_s, which is defined as

$v_r \quad v_s$

$$I(A_r, A_s) = \sum \sum p(o^r_m, o^s_h) \log \frac{p(o^r\ , o^s\ )}{p(o^r\ )p(o^s\ )} . \qquad (2.2.28)$$

$m{=}1\ h{=}1 \quad m \quad h \quad m \quad h$

Here, $p(o^r\ )$ ($p(o^s\ )$) is the occurrence probability of the objects in **X** with their r^{th}

$m \quad h$

(s^{th}) values equal to o^r (o^s). $p(o^r\ , o^s\ )$ is the occurrence probability of objects in **X**

$m \quad h \quad m \quad h$

with their r^{th} values equal to o^r_m and s^{th} values equal to o^s_h. $E(A_r, A_s)$ is the joint entropy, which is utilized to normalise mutual information [41]. $E(A_r, A_s)$ is defined as

$v_r \quad v_s$

$$E(A_r, A_s) = -\sum\sum p(o^r\ , o^s\ ) \log\ p(o^r\ , o^s\ ). \qquad (2.2.29)$$

$m \quad h \quad m{=}1\ h{=}1 \quad m \quad h$

The values of the interdependence redundancy between each pair of attributes are maintained in a $d \times d$ relationship matrix R. An element R(r, s) in R is obtained by R(r, s) = $R(A_r, A_s)$. For a target attribute A_r, the other attributes that are obviously dependent to it are selected out by

$S_r = \{A_r|\mathrm{R}(r, s) > \beta,\ 1 \leq s \leq d\}$ (2.2.30)

where β is a parameter in the interval [0, 1]. Subsequently, the distance between two object values x^r and x^r can be obtained by

$i \quad j$

$$Dist(x^r, x^r) = \sum_{A_s \in S_r} \mathrm{R}(r, s) Dist((x\ , x\ )(x\ , x\ )), \qquad (2.2.31)$$

$r \quad s$

$i \quad j \quad i \quad i \quad j \quad j$

where

$$Dist((x^r_i, x^s_i)(x^r_j, x^s_j)) = \begin{cases} p((x^r_i, x^s_i) = (x^r_j, x^s_j)) & if\ x^r_i \neq x^r_j \\ \delta(x^s_i, x^s_j) \cdot p((x^r_i, x^s_i) = (x^r_j, x^s_j)) & if\ x^r_i = x^r_j. \end{cases} \qquad (2.2.32)$$

The function $\delta(x^s_i, x^s_j)$ is defined as

$$\delta(x^s_i, x^s_j) = \begin{cases} 1 & if\ x^s_i \neq x^s_j \\ 0 & if\ x^s_i = x^s_j. \end{cases} \qquad (2.2.33)$$

This metric also weights the contributions of different attributes based on the idea that uncommon categories offer more valuable information for the distance measure- ment. As far as we know, Jia's distance metric is the most comprehensive one among the existing metrics that are proposed for the distance measurement of categorical data.

Inter-Attribute Dependence Measurement

Two interdependence measures (i.e., symmetric uncertainty [128] and interdepen- dence redundancy [72]) are designed for nominal attributes and three interdepen- dence measures (i.e., Kendall's rank correlation [75], Spearman's rank correlation [115], and rank mutual information [60]) are designed for ordinal attributes. These two types of measures are discussed in the following because all of them can be applied to measure the interdependence degrees between attributes of categorical data.

Nominal Measures

Symmetric uncertainty [128] and interdependence redundancy [72] are two interde- pendence measures adopted by two of the existing categorical data metrics proposed in [65] and [72], respectively. Since the technical details of these two measures have been reviewed in Section 2.2.4 and 2.2.5, we mainly discuss their differences and common characteristics here.

Both the symmetric uncertainty and interdependence redundancy are symmet- rical, and both of them calculate dependence degrees between categorical attributes from the perspective of information theory. The difference is that, the symmetric uncertainty adopts information gain, and the interdependence redundancy adopts mutual information. Actually, the concepts of information gain and mutual infor- mation are equivalent to each other in the scenario of inter-attribute dependence measurement [34]. These two measures actually differ from each other in how they compensate for the bias of information gain and mutual information toward at-tributes with more categories. Symmetric uncertainty divides the information gain of two attributes by their total entropy, while interdependence redundancy divides the mutual information of two attributes by their joint entropy. However, both of them are inappropriate for ordinal attributes because they cannot take the natural order information of ordinal attributes into account for inter-attribute dependence degree measurement.

Ordinal Measures

Kendall's Rank Correlation

Kendall's rank correlation [75] is presented to measure the association degree be- tween two value lists in terms of the orders of the values. It counts the number of concordant pairs and discordant pairs of observations. The concepts of concordant and discordant are defined as follows.

Definition 1. *Given a data set* **X** *with* N *objects represented by two attributes* A_1 *and* A_2. *If a pair of unequal objects* $\mathbf{x}_i$ *and* $\mathbf{x}_j$ *satisfy* $sign(x_i^1 - x_j^1) = sign(x_i^2 - x_j^2)$, *it is said that* $\mathbf{x}_i$ *and* $\mathbf{x}_j$ *are* ***concordant*** *to each other.*

Definition 2. *Given a data set* **X** *with* N *objects represented by two attributes* A_1 *and* A_2. *If a pair of unequal objects* $\mathbf{x}_i$ *and* $\mathbf{x}_j$ *satisfy* $sign(x^1 - x^1)\, sign(x^2 - x^2)$,

i j i j
it is said that $\mathbf{x}_i$ *and* $\mathbf{x}_j$ *are **discordant** to each other.*

Here, $sign(\cdot)$ fetches the sign of the value inside its brackets. When the number of concordant object pairs (discordant object pairs) is very large, it indicates a higher agreement (disagreement) degree between the two attributes [2]. Specifically, dependence degree between two ordinal attributes A_r and A_s measured according to Kendall's rank correlation can be written as

$$R_{r,s} = \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left(1 \cdot sign(x_j^r - x_i^r) \cdot sign(x_j^s - x_i^s)\right)}{N(N-1)/2}. \qquad (2.3.34)$$

Values of Kendall's rank correlation are in the interval [-1,1], where "-1" indicates perfect disagreement of two value lists, while "1" indicates perfect agreement of two value lists.

Spearman’s Rank Correlation

Spearman’s rank correlation [115] measures the order correlation between two value lists in four steps: 1) sort each value list and assign integer order values to the list,
2) jointly sort the two lists according to the order values of one list, 3) calculate the differences between the order values of the two lists, and 4) measure the correlation between the two lists according to the differences of their order values [130] [54]. Specifically, dependence degree between two ordinal attributes A_r and A_s measured by Kendall’s rank correlation can be written as

$OD(x^r, x^s)^2$

$6 \Sigma_N$

$- {}_{r,s}$

$R \quad = 1 \quad \underline{\quad {}_{i=1} \quad {}_{i} \quad {}_{i} \quad}, \qquad (2.3.35)$

$N\,(N - 1)$

where $OD(x^r, x^s)$ calculates the order difference between x^r and x^s. Suppose x^r

$i \quad i \quad i \quad i \quad i$

ranked 10th among all the r^{th} values of **X**, and x^s ranked 2nd among all the s^{th}
values of **X**, $OD(x^r, x^s) = 10 - 8 = 2$. Values of Spearman’s rank^i correlation are in
the interval [0,1]. A larger value^i of Spearman’s rank correlation indicates that two value lists are more interdependent.

Rank Mutual Information

Rank mutual information [60] is originally proposed for monotonic classification. It exploits the order information of attribute values to measure the order correlation between an ordinal attribute and the decision. It can also be utilized for inter- attribute order correlation measurement. Rank mutual information can be viewed as an ordinal version of mutual information. The original mutual information cannot reflect the dependence between attributes in terms of the orders of their values, because mutual information is calculated by summing up the sub-entropies and sub- conditional entropies of ordinal categories of attributes. Rank mutual information extends mutual information by summing up the sub-entropies and sub-conditional entropies of different dominance rough sets, and is therefore competent for indicating the dependence degree between two attributes in terms of their orders. Specifically, dependence degree between two ordinal attributes A_r and A_s measured by using ascending rank mutual information, and descending rank mutual information can be written as

$$R_{r,s} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\sigma_{\geq x_i^r}(\mathbf{X}) \cdot \sigma_{\geq x_i^s}(\mathbf{X})}{N \cdot \sigma_{\geq x_i^r \wedge \geq x_i^s}(\mathbf{X})}, \qquad (2.3.36)$$

and

$$R_{r,s} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\sigma_{\leq x_i^r}(\mathbf{X}) \cdot \sigma_{\leq x_i^s}(\mathbf{X})}{N \cdot \sigma_{\leq x_i^r \wedge \leq x_i^s}(\mathbf{X})}, \qquad (2.3.37)$$

respectively. The operators $\sigma_{\geq x_i^r}(\mathbf{X})$ and $\sigma_{\leq x_i^r}(\mathbf{X})$ count the number of data objects in the dominance rough sets $\{\mathbf{x}_j \in \mathbf{X} | x_j^r \geq x_i^r\}$ and $\{\mathbf{x}_j \in \mathbf{X} | x_j^r \leq x_i^r\}$, respectively [56] [35].

Validity Indices for Clustering Performance Assessment

Four popular indices for partitional clustering performance assessment (i.e., clus- tering accuracy [120] [59], rand index, adjusted rand index [105] [108] [119] [52], and normalised mutual information [41] [34]) and a popular index for hierarchical clustering performance assessment (i.e., Fowlkes Mallows index [46]) are reviewed in this section.

Clustering Accuracy

Clustering Accuracy (CA) [120] [59] measures the percentage of the data objects that are correctly clustered. Specifically, CA is defined as

$$\mathrm{CA} = \frac{\sum_{t=1}^{k} \sigma_{\{\mathbf{x}_i /= NULL\}}(\mathbf{x}_i \in \mathbf{C}_t^J \cap \mathbf{C}_t)}{N}, \qquad (2.4.38)$$

where $\mathbf{C}_t^J$ is the t^{th} benchmark cluster, and $\mathbf{C}_t$ is the corresponding produced cluster that is mapped to $\mathbf{C}_t^J$. Before the calculation of CA, all the produced clusters are mapped to different benchmark clusters by using the Kuhn-Munkres algorithm [88]. CA has values in the interval [0,1], and a larger value indicates better clustering performance.

Adjusted Rand Index

Adjusted Rand Index (ARI) is a more powerful version of Rand Index (RI) [105] [108] [119] [52]. ARI measures the agreement between the benchmark clusters and the clusters obtained by the assessed clustering algorithm. Specifically, ARI is defined as

$$\text{ARI} = \frac{\text{RI} - Ex(\text{RI})}{Max(\text{RI}) - Ex(\text{RI})}, \qquad (2.4.39)$$

where Ex(RI), and Max(RI) stand for expected value of RI, and maximum value of RI, respectively. RI is defined as

$$\text{RI} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}, \qquad (2.4.40)$$

where TP, FP, FN, and TN stand for true positive, false positive, false negative, and true negative, respectively. ARI has values in the interval [-1,1] while RI has values in the interval [0,1]. If ARI value is less than 0, it indicates that the performance is lower than the expectation, and a larger ARI or RI value indicates better clustering performance.

Normalised Mutual Information

Normalised Mutual Information (NMI) [41] [34] measures the agreement between the benchmark labels and the obtained labels from the perspective of information theory, which is defined as

$$\text{NMI} = \frac{\text{MI}}{\sqrt{\text{E}^J \times \text{E}}}, \qquad (2.4.41)$$

where MI is the mutual information of obtained labels and benchmark labels, which can be written as

$$MI = \sum_{t=1}^{k}\sum_{g=1}^{k} \frac{\sigma_{\{\mathbf{x}_i \neq NULL\}}(\mathbf{x}_i \in \mathbf{C}^J_t \cap \mathbf{C}_g)}{N} \log \frac{N \cdot \sigma_{\{\mathbf{x}_i \neq NULL\}}(\mathbf{x}_i \in \mathbf{C}^J_t \cap \mathbf{C}_g)}{\sigma_{\{\mathbf{x}_i \neq NULL\}}(\mathbf{x}_i \in \mathbf{C}^J_t) \cdot \sigma_{\{\mathbf{x}_i \neq NULL\}}(\mathbf{x}_i \in \mathbf{C}_g)}. \qquad (2.4.42)$$

E^J and E are the entropy values of benchmark labels and obtained labels, respec- tively, which are defined as

$$\text{E}^J = -\sum_{t=1}^{k} \frac{\sigma_{\{\mathbf{x}_i \neq NULL\}}(\mathbf{x}_i \in \mathbf{C}^J_t)}{N} \log \frac{\sigma_{\{\mathbf{x}_i \neq NULL\}}(\mathbf{x}_i \in \mathbf{C}^J_t)}{N} \qquad (2.4.43)$$

and k $E = -\frac{\sum_{g=1} \sigma\{\mathbf{x}_i}{N} \frac{}{N}$ $NULL\}(\mathbf{x}i \in \mathbf{C}g\ )\ \log\ \sigma\{\mathbf{x}_i{=}/$ $NULL\}(\mathbf{x}i \in \mathbf{C}g\ ).$ (2.4.44)

A larger NMI value indicates better clustering performance.

Fowlkes Mallows Index

Fowlkes Mallows Index (FMI) [46] is another commonly used index for evaluating the clustering performance of hierarchical clustering approaches. By using FMI, the constructed hierarchy **H** should be horizontally cut firstly to produce *k* clusters. Then, FMI is computed by

$FM = \ \frac{TP}{TP + FP} \ \frac{TP^{r}}{TP + FN}$, (2.4.45) where TP is the total number of true positives, FP is the total number of false positives, and FN stands for false negative. If the k clusters produced by horizontally cutting **H** and the k benchmark clusters match completely, the value of FMI will take the maximum value 1. Values of FMI are in the interval [0,1], and a larger value indicates better clustering performance.

Summary

In this chapter, existing related works, including partitional clustering algorithm- s, hierarchical clustering algorithms, distance metrics, inter-attribute dependence measures, and validity indices, have been introduced with providing their inter- connections and potential limitations. Partitional clustering algorithms are popular for pre-processing and analysing categorical data sets. Because they cannot provide a hierarchy for indicating the nested similarity relationship for data objects, analysing by using hierarchical clustering algorithms is also an effective way for understanding the unlabeled categorical data sets. Since conventional hierarchical clustering algorithms suffers from high time complexity ($O(N^2)$), fast hierarchical clustering algorithms have been proposed in the literature. However, all these fast algorithms are designed for numerical data only and cannot be directly applied for
categorical data clustering analysis. Therefore, how to efficiently analysing categori- cal data by using hierarchical clustering technique is a significant unsolved problem. In general, a categorical data clustering algorithm adopts a certain categorical data distance metric to perform clustering analysis. In the literature, categorical data distance metrics are proposed for nominal data only, and the clustering algorithms adopting them will result in unsatisfactory clustering results for ordinal and mixed categorical data. Therefore, how to design dedicated distance metric for ordinal data and unified distance metric for mixed categorical data are also important unsolved problems.

Chapter 3

Distance Metric for Ordinal Data Clustering

Introduction

Ordinal data is a major type of categorical data, which is common in the field of data analysis, machine learning, pattern recognition and knowledge discovery [7]. As a major kind of categorical data, ordinal data has similar properties of both nominal and numerical data [12]. On the one hand, the categories of attributes in ordinal data are all qualitative and not suitable for arithmetic calculation. On the other hand, the categories of ordinal data are ordered and comparable. To use the ordinal data, domain experts (also called data designers) usually design the attributes and corresponding ordered categories first. Then, participants (also called data generators) provide their observations/answers (data objects) according to the categories of attributes to form the data set. Since both the designers and generators are human beings, who will be more or less subjective during the data generation, the distances of ordinal data will be different from case to case, which makes the distance measurement of ordinal data a challenging problem.
Since ordinal data is a type of categorical data, the commonly used distance metrics for numerical data, including Euclidean distance [36] and Mahalanobis dis- tance [123], are not applicable to the distance measurement of ordinal data. On the opposite, some distance metrics proposed for categorical data (e.g., Hamming Distance Metric (HDM) [58], Ahmad's Distance Metric (ADM) [10], Association- Based Distance Metric (ABDM) [80], Context-Based Distance Metric (CBDM) [64] [65], and Jia's Distance Metric (JDM) [72]) can be directly applied to measure the distance for ordinal data sets.

Among the existing categorical data distance metrics, HDM is the most conven- tional and popular one. Although HDM is easy to use, it assigns the same distance to all the pairs of different categories, which is unable to distinguish the distance levels between different pairs of ordinal categories or ordinal data objects. Later, ADM and ABDM are proposed adopting a similar basic idea that if the probability distributions of the corresponding values from the other attributes of two target cat- egories are more similar, the two target categories will also be more similar to each other. These two metrics are proved to be more reasonable in the distance measure- ment of categorical data, but they still do not have the ability to extract and exploit the order information of ordinal data for appropriate ordinal data distance measure- ment. Differing from ADM and ABDM, CBDM proposes to select more relevant attributes for the distance measurement of the target attribute, and performs better than ADM and ABDM in the clustering analysis of categorical data. However, all the ADM, ABDM and CBDM metrics have not considered the case that some at- tributes are independent of each other [102], and they have also not considered the case that categorical data comprise ordinal attributes. Recently, JDM is proposed, which considers the case that the attributes are independent of each other. It si- multaneously takes the intra-attribute statistical information and the inter-attribute relationship into account for more robust and reasonable distance measurement. To the best of our knowledge, JDM is the most comprehensive categorical data distance metric in the literature. However, since all the above-mentioned distance metrics are actually proposed under the hypothesis that categorical data comprise only nominal attributes, all of them are incapable for exploiting the order information of ordinal attributes and preserving the natural order relationship between ordinal categories for the distance measurement. Therefore, a distance metric, which can reasonably exploit the underlying order information of ordinal data for clustering analysis is in urgent need.
In this chapter, we therefore propose a distance metric for ordinal data clustering [136]. From the perspective of information theory [69], each category in the data set contains a certain amount of information [94]. By simulating the thinking pro- cedure of human being when trying to change mind from a choice to another for a multiple choice question with ordered choices, distance between ordered categories can be viewed as the thinking cost for changing mind from one category to anoth- er. More

specifically, think about a choice (i.e., category in the scenario of ordinal data distance measurement) containing larger amount of information usually cost more thinking for a human being. Therefore, if the total information amount of the though categories is larger, then the distance between the original choice and the final decision will be larger. Accordingly, the distance between two target categories can be quantified by using the cumulative entropy of the categories ordered between them. To make the distances measured on different attributes with different distance scales comparable, we further provide an entropy-based distance scale normalisation scheme. To prove the effectiveness of the proposed distance metric, and to study the relationship between the exploiting degree of order information and the clustering performance, three experiments are conducted on six real ordinal data sets. The main contributions of this chapter are summarised into three points:

An entropy-based distance metric for ordinal data clustering is proposed. The proposed metric is the first ordinal data distance metric, which considers the underlying order relationship between ordinal categories. It is also parameter- free and easy to use.

An entropy-based distance scale normalisation scheme is designed for making the distances between ordinal categories measured on different attributes com- parable, by which the inter-category distances can be directly combined for forming the distance between data objects.

Relationship between the exploiting degree of order information and clustering performance is studied to prove the effectiveness of the proposed distance metric. This study also provides guidance for the research works that are related to ordinal data in the future.

The rest of this chapter is organised as follows. Preliminaries of ordinal data dis- tance measurement are given in Section 3.2. In Section 3.3, details of the proposed distance metric are presented. In Section 3.4, how to apply the proposed metric for distance measurement in the clustering analysis is discussed, and the correspond- ing time complexity is analysed. Then, we conduct experiments in Section 3.5 to illustrate its effectiveness. Finally, we provide the summarisation of this chapter in Section 3.6.

Preliminaries

Given a data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ with N data objects represented by d ordinal attributes $A_1, A_2, ..., A_d$. Possible values of an attribute A_r are a set of naturally ordered categories $\{o^r, o^r, ..., o^r\}$, where v_r is the number of categories of A_r. In this
chapter, each category is represented in the form of o^{sa}, where, $sa_s(sa \in \{1, 2, ..., d\})$
stands for the sequence number ^{1}of ^{2}an att^{v_r}ribute A_{sa} belongs to, and sc
that o^{sa}_{sc}

($sc \in \{1, 2, ..., v_{sa}\}$) stands for the sequence number of the category o^{sa}_{sc}, which ranked scth among the categories of A_{sa}. For an attribute A_r, its categories satisfy $o^r_1 \prec o^r_2 \prec ... \prec o^r_{v_r}$ where the symbol "≺" means that the categories on its left ranked lower (have smaller order values) than the categories on its right. In this thesis, the sequence numbers of the categories belonging to the same attribute indicate the order values of them. A data object $\mathbf{x}_i = \{o^1_{i_1}, o^2_{i_2}, ..., o^d_{i_d}\}$ is expressed by d categories from different attributes, where the sequence numbers $i_1, i_2, ..., i_d$ indicate that the categories representing $\mathbf{x}_i$ have order values $i_1, i_2, ..., i_d$ among the categories belonging to $A_1, A_2, ..., A_d$, respectively.

For a reasonable ordinal data distance metric, the distances produced by it should consistent with the order relationships between the categories of each attribute. More specifically, the produced distances should satisfy the following two properties:

Table 3.1: Frequently used notations of Chapter 3.

Symbol	Meaning
o^r_s	A category belonging to A_r, and ranked sth among the categories of A_r.
≺	This symbol indicates that the categories on its left ranked lower (have smaller order values) than the categories on its right.
≤	This symbol indicates that the categories on its left ranked not higher than the categories on its right.
$E_{o^r_s}$	Entropy value of a category o^r_s. $E_{o^r_s} = -p_{o^r_s} \log p_{o^r_s}$.
$p_{o^r_s}$	Occurrence probability of the data objects in $\mathbf{X}$ with their rth values equal to o^r_s. $p_{o^r_s} = \frac{\sigma_{o^r_s}(\mathbf{X})}{N}$.
$\sigma_{o^r_s}(\mathbf{X})$	Occurrence frequency of the data objects with their rth values equal to o^r_s in $\mathbf{X}$.
S_{A_r}	Standard information of A_r, see Sectino 3.3.2.
E_{A_r}	Entropy of A_r, see Sectino 3.3.2.

$Dist(\mathbf{x}_i, \mathbf{x}_j) \leq Dist(\mathbf{x}_i, \mathbf{x}_l)$, if the sequence numbers of all the categories representing $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_l$ satisfy $i_r \leq j_r \leq l_r$ or $l_r \leq j_r \leq i_r$ where $i, j, l \in \{1, 2, ..., N\}$, $i_r, j_r, l_r \in \{1, 2, ..., v_r\}$ and $r \in \{1, 2, ..., d\}$;

$Dist(\mathbf{x}_i, \mathbf{x}_j) \leq Dist(\mathbf{x}_m, \mathbf{x}_l)$, if the sequence numbers of all the categories repre- senting $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_l$, and $\mathbf{x}_m$ satisfy $\max(m_r, l_r) \geq \max(i_r, j_r)$ and $\min(m_r, l_r) \leq \min(i_r, j_r)$ where $i, j, l, m \in \{1, 2, ..., N\}$, $i_r, j_r, l_r, m_r \in \{1, 2, ..., v_r\}$ and $r \in \{1, 2, ..., d\}$;

Frequently used notations in this chapter are sorted out in Table 3.1.

The Proposed Metric

Basic Idea

From the perspective of information theory, each category in the data set contain a certain amount of information, which can be measured according to the statistics of the data set. By simulating the thinking procedure of human being when trying to change mind from a choice to another for a multiple choice question with a certain number of ordered choices, it is reasonable to measure the distance between two ordinal categories according to the amount of their contained information. More specifically, if it costs more thinking for a human being to change his/her decision from a choice to another, it usually indicates that the two choices contain a large amount of information that needs to be though for making a decision. Thus, a larger amount of information that needs to be though for changing mind indicates a higher level of difference between the two choices in terms of their contained information. By treating each attribute with ordered categories as a multiple choices question with ordered choices, the distance between ordered categories can be measured according to the "thinking cost for changing mind" between them. For example, the thinking cost can be understood as the amount of thinking that is cost by a reviewer to change his/her decision from "neutral" to "strong accept" for a paper acceptance/rejection decision question with five naturally ordered choices (i.e., "strong accept", "accept", "neutral", "reject", and "strong reject") in a review report. Because the choice "accept" is ordered between "strong accept" and "neutral", it cannot be skipped by the reviewer during the thinking. "accept" choice with a larger information amount will cost more thinking for changing mind, and thus we measure the information amount of categories using Shannon entropy [69] [94], which has been commonly adopted for the information amount measurement in the analysis of categorical data. Therefore, the distance between two ordinal categories can be measured by the cumulative entropy of all the categories that are involved in the "thinking cost for changing mind" between them.

EBDM: Entropy-Based Distance Metric

When trying to select a choice from C and E for a question, all the choices between C and E including themselves (i.e., C, D, and E) should be considered as shown in Figure 3.1. It is obvious that the thinking cost for choosing the choice from two choices is not only related to the two choices themselves, but also related to the

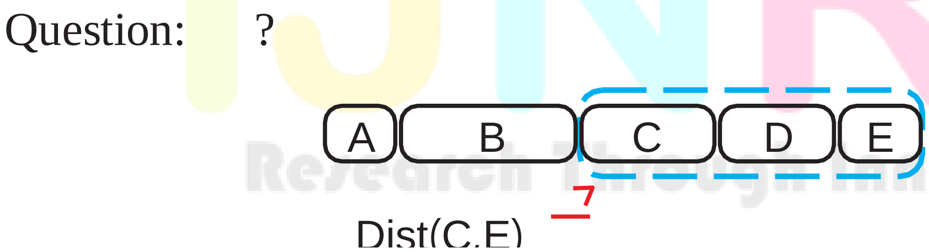


Figure 3.1: An example of a multiple-choice question with ordered choices.

choices ordered between them. That is, if choice D costs more thinking, it will be more difficult for a participant to decide a final answer from C and E. Moreover, choosing a choice from two choices by considering more choices will also cost more thinking. For instance, choosing a choice from C and E will cost more thinking than that from C and D, because one more choice (i.e., E) is involved in the former case. In addition, since all the choices are different from each other, each of them costs different amount of thinking.

Specifically, the distance between two categories o^r and o^r $_{ir}$

from A_r with $j_r > i_r$

can be measured by estimating the cost contributed by all the the $j_r{-}i_r{+}1$ categories (i.e., o^r , o^r , ..., o^r). Accordingly, the distance between the r^{th} values of two

i_r $i_r{+}1$ j_r

objects (i.e., $\mathbf{x}_i$ and $\mathbf{x}_j$) is defined as

$max(i_r,j_r)$) E_{or} , $if\ i_r$ j_r

$Dist(o^r , o^r) =$ $t{=}min(i_r,j_r)$ Σ t (3.3.1)

i_r j_r □□ 0 , $if\ \ i_r = j_r$

where E_{ort} t is the entropy of category o^r, which can be measured by

$E_{or} = -p_{or} \log p_{or}$. (3.3.2)

t t t

p_{or} is the occurrence probability of o^r, which is defined as

t

$\sigma_{or}(\mathbf{X})$

p_{or} = t ———, (3.3.3)

t N

where the operator $\sigma_{or}(\mathbf{X})$ counts the number of data objects in **X** with their r^{th} values equal to o^r.

t t

Scale Normalisation

A significant disadvantage of the distance defined by Eq. (3.3.1) is that the scales of the distances measured on different attributes are not unified. The attributes with larger numbers of categories may produce larger distances. Therefore, we normalise

the scales of the distances measured on different attributes by

□□□ $max(ir,jr)\ t{=}min(ir,jr)$ r , $if\ i_r$ /= j_r Σ $E\ o_t$

S_{Ar}

$Dist(o^r , o^r) =$ (3.3.4)

i_r j_r □□ 0 , $if\ \ i_r = j_r$.

The denominator S_{Ar} is called standard information, which is defined as

1 , (3.3.5)

$S_{Ar} = -\log v$ — r

where v_r is the number of categories of attribute A_r. The standard information is the maximum entropy of an attribute when the occurrence probabilities of the categories belonging to A_r are all the same in **X**. We normalise the distance scales using the standard information instead of the entropy of attributes because the entropy of an attribute actually indicates the total information amount offered by the attribute, which is more suitable to be utilized for attribute weighting, but not scale normalisation [72].

Based on the distances between categories defined by Eq.(3.3.4), the distance between two ordinal data objects $\mathbf{x}_i$ and $\mathbf{x}_j$ can be written as

’.,Σ^d

$r{=}1$

$Dist(\mathbf{x}_i, \mathbf{x}_j) =$ $Dist(o^r_{\ ir}, o^r_{\ jr})^2$. (3.3.6) larger numbers of categories may produce larger distances. Therefore, we normalise the scales of the distances measured on different attributes by

□□□ $\quad {}_{max(ir,jr)\ t=min(ir,jr)}{}^{r}$ $\quad , \ if\ i_r /= j_r$ $\quad \Sigma \quad E\ {}_{o_t}$

$Dist(o^r , o^r) = \quad S_{Ar}$ (3.3.4)

$i_r \quad j_r$ □□ $0 , \quad if\ \ i_r = j_r.$

The denominator S_{Ar} is called standard information, which is defined as

1 , (3.3.5)

$S_{Ar} = - \log\ v \quad {}_{r}$

where v_r is the number of categories of attribute A_r. The standard information is the maximum entropy of an attribute when the occurrence probabilities of the categories belonging to A_r are all the same in **X**. We normalise the distance scales using the standard information instead of the entropy of attributes because the entropy of an attribute actually indicates the total information amount offered by the attribute, which is more suitable to be utilized for attribute weighting, but not scale normalisation [72].

Based on the distances between categories defined by Eq.(3.3.4), the distance between two ordinal data objects $\mathbf{x}_i$ and $\mathbf{x}_j$ can be written as

$' .,\Sigma^{d}$

$$Dist(\mathbf{x}_i, \mathbf{x}_j) = \quad {}_{r=1} \quad Dist(o^r{}_{jr}\, o^r{}_r)^2. \quad (3.3.6)$$

Discussions

In this section, we discuss: 1) the mathematical properties of EBDM, 2) how to use EBDM in the clustering analysis of ordinal data, and 3) the time complexity of the distance measurement using EBDM.

Mathematical Properties

Obviously, the distances between categories defined by Eq.(3.3.4) have the following five properties:

$Dist(o^r , o^r) = Dist(o^r , o^r);$

$i_r \quad j_r \quad j_r \quad i_r$

$0 \le Dist(o^r , o^r) \le 1;$ $_{ir}$ $_{jr}$

$Dist(o^r , o^r) = Dist(o^r ,$ E_{or} $\le o^r$ $\le o^r$ or

$o^r) + Dist(o^r , o^r) -$ $\overline{}_{ir}$, iff o^r

$_{i_r}$ $_{l_r}$ $_{i_r}$ $_{j_r}$ $_{j_r}$ $_{l_r}$ S_{Ar} $_{i_r}$ $_{j_r}$ $_{l_r}$

r r $\le o^r ;$

l_r j_r o $\le o$ $_{ir}$

$_{lr}$ $_{lr}$ $_{jr}$ $_{ir}$

$Dist(o^r , o^r) \le$ $\le o^r$ $\le o^r$, or o^r $\le o^r$ $\le o^r$.

$Dist(o^r , o^r)$, if o^r $_{jr}$ $_{jr}$ $_{ir}$ $_{lr}$ $_{ir}$ $_{jr}$

$Dist(o^r , o^r) \le Dist(o^r$ $, o^r)$, if o^r $\le \forall\{o^r , o^r \} \le o^r$, or o^r $\le \forall\{o^r , o^r \} \le$

$_{i_r}$ $_{j_r}$ $_{m_r}$ $_{l_r}$ $_{m_r}$ $_{i_r}$ $_{j_r}$ $_{l_r}$ $_{l_r}$ $_{i_r}$ $_{j_r}$

$r\ m_r$ o ,

where $i, j, l, m \in \{1, 2, ..., N \}$, $i_r, j_r, l_r, m_r \in \{1, 2, ..., v_r\}$, and $r \in \{1, 2, ..., d\}$.
Moreover, the distances defined by Eq.(3.3.6) have the following five properties:

$Dist(\mathbf{x}_i, \mathbf{x}_j) = Dist(\mathbf{x}_j, \mathbf{x}_i);$

$0 \le Dist(\mathbf{x}_i, \mathbf{x}_j) \le 1;$

$Dist(\mathbf{x}_i, \mathbf{x}_l) \le Dist(\mathbf{x}_i, \mathbf{x}_j) + Dist(\mathbf{x}_j, \mathbf{x}_l)$ if the categories representing $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_l$ satisfy o^r $\le o^r$ $\le o^r$, or o^r $\le o^r$ $\le o^r$;

$_{ir}$ $_{jr}$ $_{lr}$ $_{lr}$ $_{jr}$ $_{ir}$

$Dist(\mathbf{x}_i, \mathbf{x}_j) \le Dist(\mathbf{x}_i, \mathbf{x}_l)$, if the categories representing $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_l$ satisfy

$o^r \le o^r$ $\le o^r$, or o^r $\le o^r$ $\le o^r$;

$_{ir}$ $_{jr}$ $_{lr}$ $_{lr}$ $_{jr}$ $_{ir}$

$Dist(\mathbf{x}_i, \mathbf{x}_j) \le Dist(\mathbf{x}_m, \mathbf{x}_l)$, if the categories representing $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_l$, and $\mathbf{x}_m$ satisfy o^r $\le \forall\{o^r , o^r \} \le o^r$, or o^r $\le \forall\{o^r , o^r \} \le o^r$,

$_{mr}$ $_{ir}$ $_{jr}$ $_{lr}$ $_{lr}$ $_{ir}$ $_{jr}$ $_{mr}$

where $i, j, l, m \in \{1, 2, ..., N \}$, $i_r, j_r, l_r, m_r \in \{1, 2, ..., v_r\}$, and $r \in \{1, 2, ..., d\}$. These properties prove that EBDM is a distance metric.

Distance Measurement

The distance measurement algorithm of EBDM is shown in Algorithm 6. To save computation cost for the clustering procedures, we can record the distances between categories calculated according to Algorithm 6 for each attribute using d distance matrices. In this way, the distances between data objects can be directly read off from the d matrices during clustering analysis.

Algorithm 6 Distance Measurement Using EBDM

```
1: Input: Data objects $\mathbf{x}_i$ and $\mathbf{x}_j$.
2: Output: $D(\mathbf{x}_i, \mathbf{x}_j)$.
3: for $r = 1$ to $d$ do
4:     if $i_r$ /= $j_r$ then
5:         Calculate the distance between the $r^{th}$ values of $\mathbf{x}_i$ and $\mathbf{x}_j$ by $Dist(o^r_{i_r}, o^r_{j_r}) = \sum_{t=min(ir,jr)}^{max(ir,jr)} E_{o^r_t}$;
6:     else
7:         $Dist(o_{rir}, o_{rjr}) = 0$;
8:     end if
9: end for
10: calculate the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ by $Dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{r=1}^{d} Dist(o^r_{ir}, o^r_{jr})^2}$.
```

Time Complexity Analysis

The computation procedures of EBDM-based distance measurement consists of three parts: 1) calculate the v_r occurrence probabilities and entropy values of the cate- gories belonging to each attribute, 2) calculate distance matrices recording the dis- tances between categories of each attribute, and 3) read off the distance between two data objects according to the distance matrices. In clustering analysis, the computation of part 1 and 2 should be executed once, and then the distance matrices produced in part 2 are exploited in part 3 for distance reading off. We analyse the time complexity of these three parts as follows: The time complexity for calculating the occurrence probabilities and entropy values of v_r categories belonging to attribute A_r is $O(N + v_r)$. For the total d attributes, the time complexity is $O(Nd + \sum_{r=1}^{d} v_r)$.

To calculate the distances between categories of an ordinal attribute A_r, we can firstly calculate the distances between the v_r-1 pairs of adjacent categories (i.e., o^r_s and o^r_{s+1} with $s \in \{1, 2, ..., v_r - 1\}$). Then, the distances between the $v_r - 2$ pairs of categories (i.e., o^r_s and o^r_{s+2} with $s \in \{1, 2, ..., v_r - 2\}$) can be obtained by simply adding $Dist(o^r_s, o^r_{s+1}$) and the entropy value of o^r_{s+2} (i.e., $E_{o^r_{s+2}}$). Therefore, the time complexity for producing the distance matrix of

an ordinal attribute A_r is $O(\frac{v_r(v_r-1)}{2})$. For the d attributes in total, the time complexity is $O(\sum_{r=1}^{d} \frac{v_r(v_r-1)}{2})$.

Time complexity for reading off the distance between two data objects according to the produced distance matrices is $O(d)$.

According to the above analysis, we will further discuss if the time complexity of EBDM will influence the time complexity of clustering analysis. Time complexity for calculating the distance matrices using EBDM is $O(Nd + \sum_{r=1}^{d}(vr + \frac{v_r(v_r-1)}{2}))$.

Since the numbers of categories may be different for the attributes, we use $v_{max} = max(v_1, v_2, ..., v_r)$, $r \in \{1, 2, ..., d\}$ instead of v_r in the following analysis. By adopting v_{max}, the time complexity of producing distance matrices using EBDM can be re-written as $O(Nd + v_{max}d + v^2_{max}d)$. Since v_{max} is usually a small constant satisfying $v^2_{max} < N$ for most of the real ordinal data sets, the time complexity can be further modified to $O(Nd)$. According to the produced distance matrices, the time complexity for partitioning the N data objects into k groups is $O(kdNI)$ if the simplest k-modes clustering algorithm is adopted, where I is the number of learning iterations. Obviously, EBDM will not increase the overall time complexity of ordinal data clustering analysis.

Experiments

In this section, comparative experiments are conducted to prove the effectiveness of the proposed EBDM metric.

Experimental Settings

Data Sets

Six ordinal data sets are collected for the experiments in this chapter. Among the data sets, Internship is collected from the students questionnaires of the Education

Table 3.2: Statistics of the six ordinal data sets.

Data Set	♯ Instances	♯ Attributes	♯ Classes
Internship	90	3	2
Photo	66	4	3
Employee	1,000	4	9
Selection	488	4	9
Lecturer	1,000	4	5
Social	1,000	10	4

University of Hong Kong, Photo is collected from the student questionnaires of the College of International Exchange of Shenzhen University, the remainder four (i.e., Employee, Selection, Lecturer, and Social) are collected from the website of Weka [122]. Statistics of the collected data sets are shown in Table 3.2.

Counterparts

Hamming Distance Metric (HDM) [58] is selected as a representative conventional distance metric of categorical data. Ahmad's Distance Metric (ADM) [10], Association- Based Distance Metric (ABDM) [80], Context-Based Distance Metric (CBDM) [65], and Jia's Distance Metric (JDM) [72] are selected as the other four state-of-the- art counterparts in this chapter. All the selected counterparts and the proposed Entropy-Based Distance Metric (EBDM) are embedded into the simplest k-modes clustering algorithm [63] for their performance evaluation.

Validity Indices

We have compared the clustering performance of the above-mentioned six distance metrics using two popular validity indices (i.e. Clustering Accuracy (CA) [59] [120] and Rand Index (RI) [105] [119]), which have been introduced in Chapter 2. To illustrate that the performance of ordinal data clustering analysis depends on if the corresponding distance metric can adequately exploit the order information for the distance measurement, we also propose a new validity index, called Order Consisten-

Table 3.3: Clustering performance in terms of CA.

Data Set	HDM	ADM	ABDM	CBDM	JDM	EBDM
Internship	0.620±0.07	0.571±0.06	0.524±0.01	0.502±0.00	0.573±0.05	**0.734±0.06**
Photo	0.514±0.07	0.503±0.04	0.538±0.08	0.541±0.06	0.486±0.07	**0.614±0.05**
Employee	0.188±0.01	0.202±0.01	0.203±0.01	0.196±0.01	0.186±0.01	**0.206±0.01**
Selection	0.380±0.04	0.396±0.04	0.400±0.05	**0.412±0.03**	0.348±0.04	0.402±0.04
Lecturer	0.330±0.04	0.328±0.02	0.316±0.02	0.313±0.02	0.320±0.04	**0.367±0.02**
Social	0.376±0.03	**0.411±0.02**	0.405±0.02	0.378±0.03	0.362±0.03	0.396±0.04

cy Score (OCS) to evaluate the exploiting degree of order information for distance metrics. More details about OCS are introduced in Section 3.5.4.

Experiments Design

We compare the clustering performance of different metrics to evaluate the effec- tiveness of EBDM. To intuitively observe if the distances calculated by a metric are consistent with the orders of ordinal data, we use the compared metrics to produce distance matrices for each attribute of the six data sets. The distance matrices are compared by converting their normalised values into grey-scale maps. To evaluate the exploiting degree of order information of the selected metrics, we compare their OCS performance on different data sets. To verify that the exploiting of order infor- mation can make a metric performs well in the clustering analysis of ordinal data, we also study the correlation between the averaged OCS and the clustering performance of different metrics. All the results are averaged by 10 runs of the experiments.

Comparative Studies

Clustering performance of the six metrics are evaluated by CA and RI, and the corresponding experimental results are compared in Table 3.3 and 3.4, respective- ly. To evaluate the stability of each distance metric, standard deviation of their performance is also recorded in the two tables. Among the results of each data set, the best and the second best results are highlighted by boldface and underline, respectively.

Table 3.4: Clustering performance in terms of RI.

Data Set	HDM	ADM	ABDM	CBDM	JDM	EBDM
Internship	0.620±0.07	0.571±0.06	0.524±0.01	0.502±0.00	0.573±0.05	**0.734±0.06**
Photo	0.651±0.04	0.688±0.06	0.696±0.06	0.704±0.05	0.672±0.06	**0.721±0.04**
Employee	0.819±0.00	0.822±0.00	0.823±0.00	0.821±0.00	0.819±0.00	**0.824±0.00**
Selection	0.862±0.01	0.866±0.01	0.867±0.01	**0.870±0.01**	0.855±0.01	0.867±0.01
Lecturer	0.732±0.02	0.731±0.01	0.726±0.01	0.725±0.01	0.728±0.02	**0.747±0.01**
Social	0.688±0.01	**0.706±0.01**	0.702±0.01	0.689±0.02	0.681±0.02	0.698±0.02

It can be observed from the results that the proposed EBDM metric outperforms the others on four of the six data sets (i.e., Internship, Photo, Employee, and Lec- turer). For the Selection data set, even EBDM's performance is the second best, the gap between it and the best one is very tiny (i.e., 0.01 and 0.003 in terms of CA and RI, respectively). For the Social data set, the gap between EBDM and the best performing ADM is around 0.01, which is still small. The reason why EBDM has such competitive performance in the clustering analysis of ordinal data is that EBD- M exploits order information for ordinal data distance measurement, but the other five metrics do not. In conclusion, EBDM is obvious competent in comparison with the other distance metrics. In Section 3.5.3 and 3.5.4, we will further analyse the reasons why EBDM cannot outperform some of the counterparts on the Selection and Social data sets according to the experimental results.

Distance Matrices Demonstration

To intuitively observe the distance produced by different metrics, we demonstrate the distance matrices produced by all the compared metrics for Selection data set. All the values of the distance metrics are normalised into the interval [0, 1] and represented by grey-scale blocks as shown in Figure 3.2. In this figure, the distance matrices produced by ADM, ABDM, CBDM, and EBDM are shown in row 1, 2, 3, and 4, respectively. Distance matrices produced by JDM metric is not demonstrated because JDM does not produce the distances between categories. Instead, it directly computes the distances between data objects.

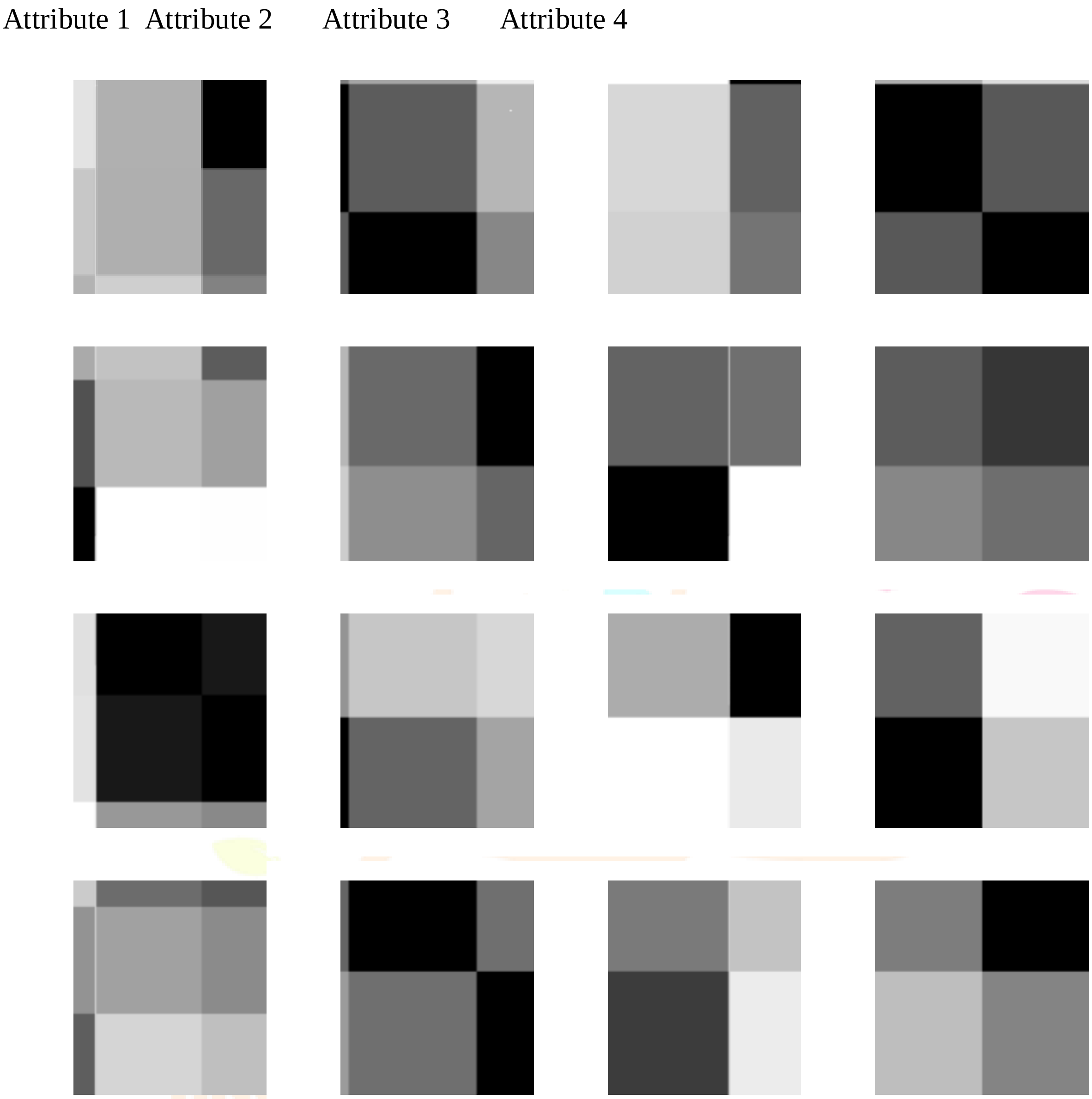


Figure 3.2: Distance matrices of Selection data set.

In this figure, pure black colour indicates distance "0" between two categories while pure white colour indicates distance "1". Therefore, all the blocks on the diagonals from the left top corner to the right bottom corner are pure black. If a distance matrix can reasonably indicate the natural distances of an ordinal attribute, the right top block should be pure white and the other blocks toward the diagonal will be darker. This is because the distance between the two categories with the lowest and highest order values should be the largest and vice versa. Obviously, distance matrices produced by EBDM is closer to the natural distance structure of an ordinal data set. If we consider these distance matrices in combination with the performance demonstrated in Table 3.3 and 3.4, we can find that if the distance matrices produced by a distance metric are closer to the natural distances of an ordinal data set, its clustering performance will be better. This observation will be further studied in Section 3.5.4.

Evaluation of the Order Information Exploiting

To study the relationship between exploiting degree of order information and clus- tering performance for the distance metrics, OCS index is proposed to quantify the consistent level between the distance matrices produced by a distance metric and the order of categories. Specifically, given a distance matrix of attribute A_r, an element of this matrix can be expressed as $\mathbf{M}_r(a, b) = Dist(o^r, o^r)$. The OCS of A_r is ob-

$a \quad b$

tained by searching $\mathbf{M}_r$ and count the sum of the scores earned by each pair of $\mathbf{M}_r$'s elements. If two elements (i.e., $\mathbf{M}_r(a, b)$ and $\mathbf{M}_r(a, c)$) satisfy $\mathbf{M}_r(a, b) < \mathbf{M}_r(a, c)$ with $a < b < c$ or $c < b < a$, this pair of elements will get a score, otherwise, their score will be 0. The score of a pair of elements is defined as

$\underline{v_r-(c-b)-1}\ v_r-2 \quad , \ if\ \mathbf{M}_r(a, b) < \mathbf{M}_r(a, c)$

$Sa,b,c =$ (3.5.7)

□

0 , *otherwise*. Subsequently, OCS □of A_r can be calculated by

$OCS_r \quad = \quad \frac{Sa,b,c}{T}$, (3.5.8) Σ

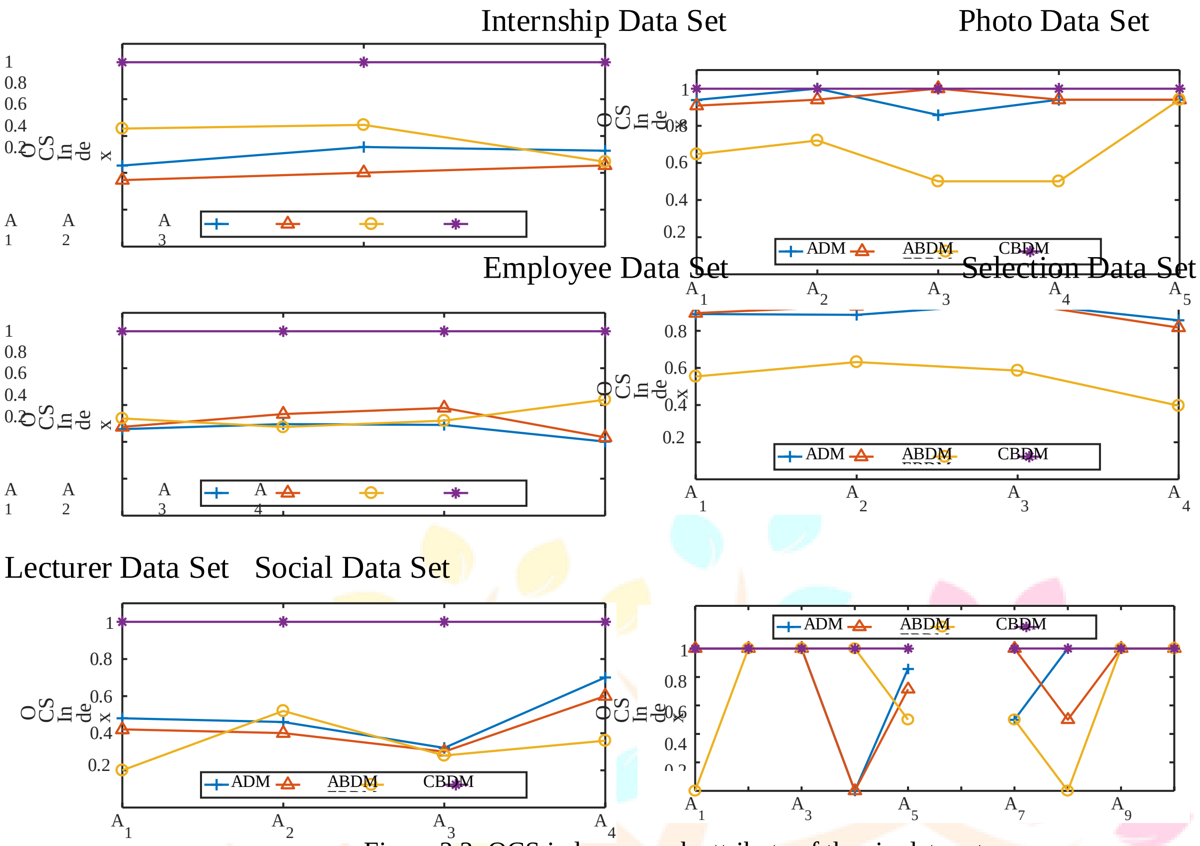
r

where T_r is the maximum score that $\mathbf{M}_r$ can get, which is utilized to normalise the value of OCS_r into the interval [0, 1]. Accordingly, the overall OCS of an ordinal data set is computed by

$\mathrm{OCS} = \frac{1}{d} \Sigma\ \mathrm{OCS}_r$. (3.5.9)

$r{=}1$

OCS of ADM, ABDM, CBDM and EBDM metrics on each attribute of all the six data sets are shown in Figure 3.3. HDM is not included because it is special, and its OCS is 0. It can be seen that attribute 6 of Social data set is not shown. It is because that attribute 6 of Social data set has only 2 categories, which is unavailable for the OCS measurement. It can be seen that EBDM outperforms all the other counterparts in terms of OCS.



Internship Data Set

Photo Data Set

Employee Data Set

Selection Data Set

Lecturer Data Set

Social Data Set

Figure 3.3: OCS index on each attribute of the six data sets.

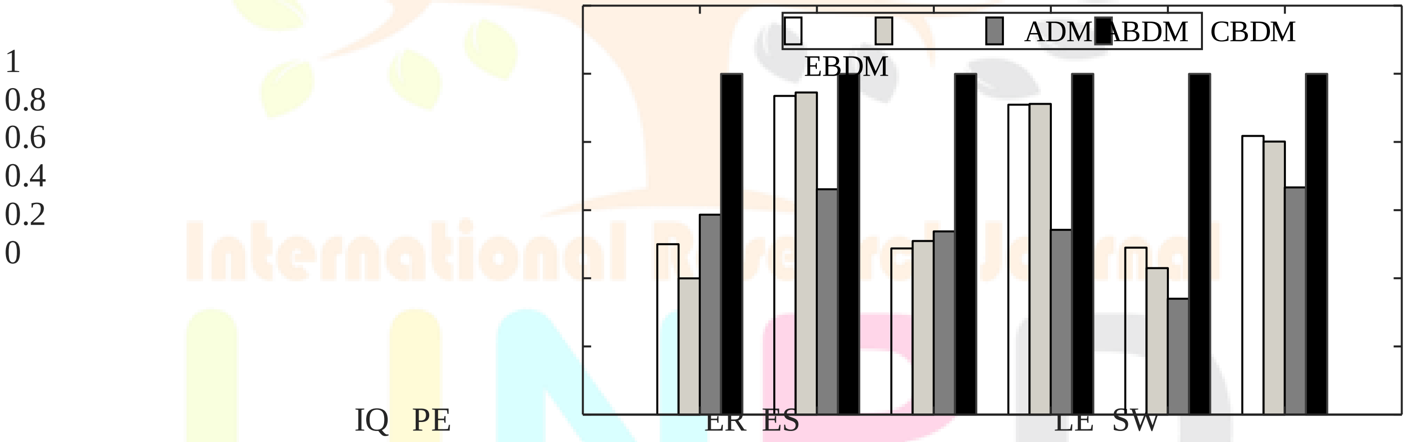


Figure 3.4: Averaged OCS of EBDM on all the six data sets.

To study the relationship between order information exploiting and clustering performance, we first demonstrate the overall OCS of all the compared metrics on all the six data sets in Figure 3.4. In this figure, Internship, Photo, Employee,

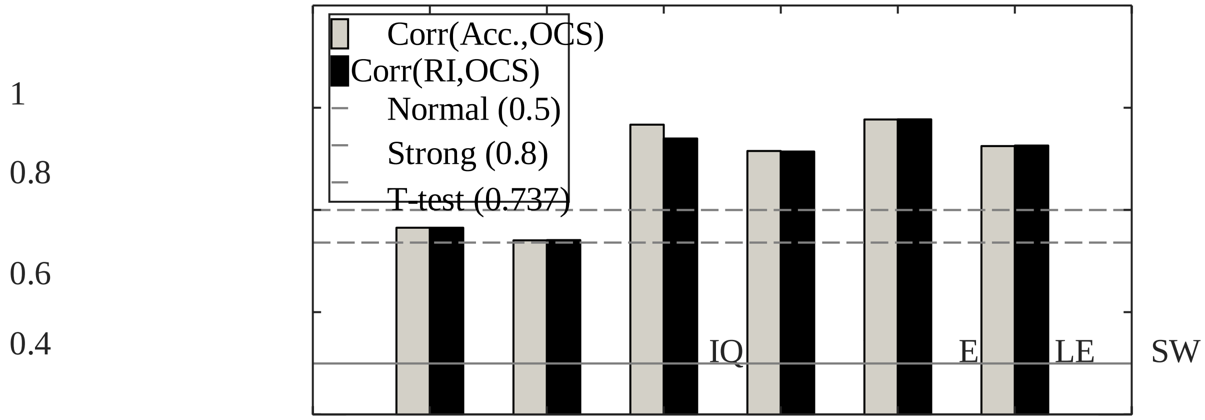


Figure 3.5: OCS-CA correlation and OCS-RI correlation.

Selection, Lecturer, and Social data sets are abbreviated as IQ, PE, ER, ES, LE, and SW, respectively, according to their full data set names. It can be found that the overall OCS performance of EBDM is always better than the others.

Based on the overall OCS results, we evaluation the relationship between order information exploiting and clustering performance. OCS-CA correlation and OCS- RI correlation are demonstrated in Figure 3.5. We use one-sided T-test at 90% confidence level with degree of freedom equals to 2 to prove the significance of the correlation. Moreover, two commonly used critical values of correlation (i.e., “0.5” for normal correlation and “0.8” for strong correlation) are also adopted for reference. It can be observed that the correlations on all the six data sets pass the T-test, which means that the correlations are significant. In addition, four of tests are above the strong correlation level, and all of the tests are above the normal correlation level. To sum up, for a distance metric, its clustering performance on ordinal data sets is obvious in proportion to its exploiting level of order information. Therefore, EBDM outperforms its counterparts because it can better exploit the order information. Since all the ADM, ABDM, CBDM and EBDM have relative high order information exploiting degree on the Selection and Social data sets, their clustering performance shown in Table 3.3 and 3.4 are close to each other.

Summary

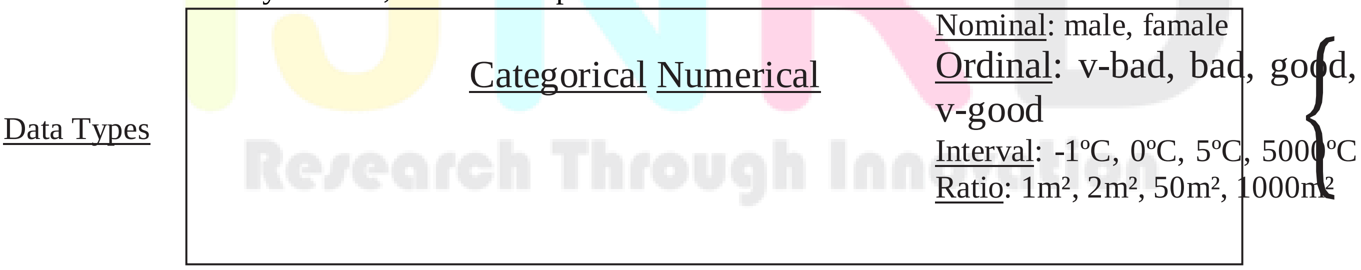
This chapter has presented an entropy-based distance metric for ordinal data clus- tering, which quantifies the distance between ordinal categories with considering the order relationship between them by using cumulative entropy. The proposed metric appropriately exploits the order information and outperforms the existing metrics for ordinal data clustering. In addition, the proposed metric is parameter-free and easy to use. Experimental results have shown the effectiveness of the proposed met- ric in the clustering analysis of ordinal data. In addition, we conduct a study of the relationship between order information exploiting and clustering performance, and prove that the performance of an ordinal data distance metric depends on its exploiting degree of the order information.

Chapter 4

Unified Distance Metric for Categorical Data Clustering

Introduction

In general, there are two common types of data sets (i.e., categorical and numerical data sets) as illustrated in Figure 4.1, in which "v-bad" and "v-good" indicate very- bad and very-good, respectively. Under the categorical class, there are two types of attributes, that are nominal and ordinal attributes. Ordinal attributes inherit some properties of nominal attributes [74][6] but are different from nominal attributes. Like nominal attributes, the categories of ordinal attributes are all qualitative and not suitable for arithmetic operations including mean, division, summation, and so on [8]. Unlike nominal attributes, the categories of an ordinal attribute are naturally ordered, and are comparable in terms of their order values.



Data Types

Categorical Numerical

Nominal: male, famale

Ordinal: v-bad, bad, good, v-good

Interval: -1ºC, 0ºC, 5ºC, 5000ºC

Ratio: 1m², 2m², 50m², 1000m²

Figure 4.1: Relationships among different data types.

In many real categorical data analysis tasks, both nominal and ordinal attributes exist in data sets (e.g., the data obtained through questionnaires, evaluation system, and so on). By taking the fragment of a TA evaluation data set shown in Table 1.1 as an example, if we treat the two ordinal attributes "Helpful" and "Professional" as nominal ones, the preservation of their natural order relationship may not be guaranteed. For example, the distance between "Agree" and "Weak-agree" should be smaller than that between "Agree" and "Disagree". But this order relationship will be ignored if we treat the categories as nominal ones. Therefore, it is more reasonable to treat ordinal attributes differently from nominal ones to take their order information into account for data analysis.

In the literature, several works have been proposed for ordinal data regression [103] [57] [124], ordinal data classification [61] [110] [135] [104], and ordinal data ranking [25] [42] [83]. Nevertheless, all of them focus on ordinal data only. In fact, from the practical perspective, mixed categorical data are common as shown in Table 1.1. Unfortunately, as far as we know, the distance metric for such categorical data has yet to be well explored in the literature. Therefore, this chapter will study the distance measurement problem for such data within the framework of clustering analysis, which is generally a nontrivial task because the heterogeneous information offered by ordinal and nominal attributes should be simultaneously taken into account when assigning the data objects into the proper clusters.

Over the past two decades, a number of clustering algorithms have been proposed for the categorical data, which are essentially applicable to nominal data only. A typical example is k-modes [63] algorithm, which seeks for a partition by iteratively assigning objects into their closest modes. Later, an extended k-modes is proposed in [62], which weight the contribution of different attributes during the data clustering process. After that, another improved version of k-modes has been presented in [73]. Instead of weighting the attributes for a whole data set, this version weights the contributions of different attributes for each cluster. Furthermore, the other examples are the entropy-based clustering algorithms (i.e., [15] and [85]), which provide that the information entropy of a cluster will not increase a lot after adopting an object if this object is similar to the cluster. In addition, paper [70] proposes a clustering algorithm, called attribute-weighted OCIL, which weights the attributes on each cluster by simultaneously considering their contributions in terms of intra- cluster similarity and inter-cluster difference. Although all the above-mentioned algorithms can be applied to mixed categorical data, their clustering results will degrade to a certain degree because the metrics adopted by these algorithms have not taken into account the order information of the ordinal attributes.

In fact, as far as we know, most of the existing metrics proposed for categori- cal data are essentially designed for nominal data only [11]. Among these metrics, the commonly used simple matching distance (also called Hamming distance metric interchangeably [58]) simply assigns distance "1" to unequal categories and assign- s "0" to identical categories without considering the inherent relationships among attributes. Thus, association-based distance metric [80] and Ahmad's distance met- ric [10] have been proposed provided that, for two intra-attribute categories, if the distributions of their corresponding values from the other attributes are similar to each other, the distance between the two categories will be shorter. However, both of these two metrics treat each attribute equally, which is usually unreasonable from the practical view-point. To address this problem, a context-based distance metric [65] has been proposed to measure the distance between two intra-attribute cat- egories according to the selected relevant attributes. Furthermore, categorical data distance metric proposed in [72] not only measures the distance according to the relevant attributes, but also considers the occurrence probability of them. In this way, even all the attributes are independent of each other, this metric still works. N- evertheless, all the above-mentioned metrics are actually proposed for nominal data, which are surely not suitable for exploiting order information of ordinal attributes.

In the literature, some other measures have been presented to measure similarity between two value lists according to the order of the values. For example, Kendall's rank correlation [75] and Spearman's rank correlation [115] measure the correlation degree between two variables according to the matching degree of their order values. However, most of the ordinal attributes in categorical data sets have a small number of possible values, which cannot provide valid order values for the computation of

these two measures. Another measure, called rank mutual information (RMI), has been presented in [60] for monotonic classification. Similar to Kendall's and Spear- man's rank correlation, RMI is designed to measure the monotonic level between attributes. In fact, entropy-based metrics have been successfully used for nominal data clustering (e.g., see [15] [85] [73] [65]). Therefore, along this line, proposing an entropy-based distance metric that can simultaneously exploit valuable information of ordinal and nominal attributes would be a feasible choice for mixed categorical data clustering.
In this chapter, we propose a new categorical data distance metric that can exploit order information of ordinal attribute, and unify the heterogeneous informa- tion offered by ordinal and nominal attributes for mixed categorical data clustering [139]. To exploit the order information of ordinal data, we compute the distance between two ordinal categories according to the entropy values of all categories ordered between them (including themselves). This idea is analogous to making decision between two ordered choices as discussed in Chapter 3. For example, given a multiple-choice question with the ordered choices: {very-good, good, neutral, bad, very-bad}, when we are comparing two choices (i.e., "neutral" and "very-good") to make a final decision for this question, both of these two choices should be consid- ered together with another choice "good" because "good" is an intermediate choice and cannot be skipped. In this chapter, the proposed metric unifies the distance concepts of both ordinal and nominal attributes. Since the choices of a question are unordered in a nominal case, it is not necessary to consider the other choices when we are deciding the final choice from the two nominal choices. According to this, the concept of distance has a uniform definition, which is the "thinking cost" for all the choices that should be considered for making a decision between two choices, no matter the choices are ordered or not. Therefore, information offered by ordinal and nominal attributes can be quantified and combined for indicating the distance between two data objects of a mixed categorical data set. Furthermore, by taking into account the different contributions of attributes in the clustering task, we also present a unified attribute weighting scheme to adjust the contributions of different attributes.

Experimental results on different real and benchmark data sets have shown the effectiveness of the proposed distance metric for mixed categorical data clustering. The main contributions of this chapter are summarised into three-fold:

A unified metric featuring parameter-free, robust, and easy to use is developed, which unifies the distance concepts of both ordinal and nominal attributes. The unified distance metric can be applied for the distance measurement of any type of categorical data, including ordinal data, nominal data, and mixed categorical data.

A unified attribute weighting scheme is also designed to weight the contri- butions of different categorical attributes for the distance measurement. It not only assigns a larger weight to the attributes offering more information for the distance measurement, but also unifies the distance scales of different attributes.

Extensive experiments are conducted to evaluate: 1) the clustering perfor- mance of the proposed metric on ordinal, nominal, and mixed categorical data,
2) the effectiveness of the order information exploiting scheme of the proposed metric, and 3) the effectiveness of the unified attribute weighting scheme of the proposed metric.

The rest of this chapter is organised as follows. In Section 4.2, we formalize the problem of unified categorical data distance measurement, and provide common no- tations. Section 4.3 proposes a unified distance metric for both ordinal and nominal attributes. In Section 4.4, mathematical properties of the unified metric, how to use it for clustering analysis, and its time complexity, are discussed. Section 4.5 presents the experimental results on both real and benchmark data sets. Finally, we summarise this chapter in Section 4.6.

Preliminaries

Given a data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ with N data objects represented by d at- tributes, including d_{ord} ordinal attributes: $A_1, A_2, ..., A_{dord}$ and d_{nom} nominal at- tributes: $A_{dord}+1, A_{dord}+2, ..., A_d$, where $d = d_{ord} + d_{nom}$. In this chapter, it is as- sumed that the former d_{ord} attributes of a categorical data set are ordinal while the latter d_{nom} are nominal. Ordinal data set can be viewed as a special case that $d_{ord} = d$ and $d_{nom} = 0$, while nominal data set is another special case that $d_{ord} = 0$ and $d_{nom} = d$. In this chapter, all the settings about ordinal attributes are the same as that of Chapter 3. The difference between ordinal and nominal categories is that, the ordinal categories from one attribute are naturally ordered, and their sequence numbers are also the order values of them, while the sequence numbers of nominal categories do not indicate their order relationship. A data object $\mathbf{x}_i$ is expressed by $\mathbf{x}_i = \{o^1, o^2, ..., o^{dord}, o^{dord}{}^{+1}, o^{dord}{}^{+2}, ..., o^d\}$, where the former d_{ord}

are the categories of the d_{ord} ordinal attributes, while the latter d_{nom} are the cat- egories of the d_{nom} nominal attributes. For the nominal part of $\mathbf{x}_i$, the sequence numbers i_d +1, i_d +2, ..., i_d indicate that the d_{ord} + 1th, d_{ord} + 2th, ..., dth values

of object $\mathbf{x}_i$ equal to the ith $_{ord}$ $_{ord}$ th $_{dord+1}$ d_{ord}+2 , ..., ith , i categories of the d_{nom} attributes

$A_{dord}+1, A_{dord}+2, ..., A_d$, respectively. $_d$

Frequently used notations in this chapter are sorted out in Table 4.1. Since some of the frequently used notations have been introduced in Chapter 3, we do not repeatedly introduce them again here.

i_1 i_2 i_{dord} i_{dord+1} i_{dord+2} i_d

The Unified Distance Metric

In this section, we first discuss the main limitations of the EBDM metric proposed in Chapter 3. Then, the attribute weighting scheme and the unified distance metric for any type of categorical data clustering is proposed. Finally, we provide the math- ematical properties of the proposed metric, discuss how to use it in the categorical data clustering analysis, and analyse its time complexity.

Table 4.1: Frequently used notations of Chapter 4.

Symbol	Meaning
dord	Number of ordinal attributes in X. Number of nominal attributes in X.
dnom	Weight of A_r, $\omega_A \;= \omega^I \;\; \cdot \omega^S$.
$\omega_{A_r}\,\omega I$	$_r \quad A_r \quad A_r$
A_r	Importance weight of A_r. Scale weight of A_r.
	Reliability of A_r, $R_A = \frac{E_{Ar}}{}$.
	$_r \quad S_{Ar}$
ωS	
A_r	Shannon entropy of A_r, $E_A \;= -\,\Sigma v_r \;\; p_{or} \log p_{or}$.
$R_{A_r} E_{A_r}$	$_r \quad s=1 \quad _s \quad _s$

Limitations of EBDM

In Chapter 3, EBDM metric is proposed for ordinal data clustering analysis. Exten- sive experiments have illustrated its effectiveness in the clustering of ordinal data. However, from the perspective of categorical data clustering analysis, it still has two main limitations, which are discussed as follows:

Treat each attribute equally. EBDM has not considered the importance of different attributes. It treats the valuable information extracted from each attribute equally for the distance measurement, which is usually unreasonable from the practical view-point.

Only applicable to ordinal data. Since EBDM is proposed for ordinal data clustering, it cannot measure distances between categories without knowing their order values. For the nominal categories without order values, EBDM is not applicable to the distance measurement.

To solve the first problem, we present a attribute weighting scheme in Sec- tion 4.3.2. To make the EBDM metric applicable to the clustering analysis of mixed categorical data and nominal data, we generalise it to a nominal case, and com- bine it with the attribute weighting scheme to form a unified distance metric in Section 4.3.3.

Attribute Weighting

From the perspective of information theory, higher entropy of an attribute means that this attribute offers more information [94]. Evidently, a decision made based on larger amount of information will be more convincible. Therefore, we weight the importance of the attributes according to the information amount they offer. Specifically, the weight value for weighting the importance of an attribute A_r is defined as

$\omega^I \quad = \qquad EA_r$ $\qquad \Sigma$ (4.3.1)

$A_r \quad d$ $\qquad E_{A_s}$

$s{=}1$

where E_{A_r} stands for the entropy of A_r, which is defined as

v_r

$E_{A_r} = - \qquad p_{or} \log p_{or}.$ (4.3.2) Σ $_s \quad _s$

$s{=}1$

The attributes with larger number of categories may produce larger distance values, and will contribute more to the distance between two data objects. To avoid this, the weight value for weighting the scale of an attribute A_r is defined as

$\underline{1}$

$\omega S \quad = \dfrac{S_{Ar}}{}$ (4.3.3)

$A_r \quad \Sigma d \qquad \underline{1}$

$s{=}1 \; S_{A_s}$

where the factor S_{A_r} is the standard information, which has been defined by E- q. (3.3.5) in Chapter 3.

To simultaneously weight the attributes using the two above defined weights ω^I and ω^S , the integrated weight of an attribute A_r can be written as $_{Ar}$

$\omega_{A_r} \quad ^{Ar} \qquad A_r \cdot \omega A_r = \omega \qquad .$ (4.3.4)

$I \quad S$

To explain the physical meaning of the integrated weight, we also define another concept called reliability, which is written as

$R_{A_r} \qquad = EA_r.$ (4.3.5)

S

A_r

The reliability indicates the percentage of the maximum information contained by attribute A_r. The higher R_{A_r} is, the more convincible the distances measured ac- cording to attribute A_r will be. Based on Eq. (4.3.5), the weight of A_r can be rewritten as

$\omega \quad = \qquad RA_r \qquad .$ (4.3.6) Σ

$A_r \quad d \qquad R_{A_s}$

$s{=}1$

Since Eq. (4.3.6) is equivalent to the weight defined in Eq. (4.3.4), it is obvious that the defined attribute weight can simultaneously weight the importance and scale of an attribute.

UEBDM: Unified Entropy-Based Distance Metric

In this sub-section, we generalise the EBDM to make it capable for the distance measurement of categorical data with a mixture of nominal and ordinal attributes. For nominal data, we still treat the attributes as questions with multiple choices. The difference is that the choices are unordered. For example, there is a question “what is your favorite course?” in a questionnaire with four choices (i.e., “English”, “Machine Learning”, “Music”, and “Mathematics”). If a participant is trying to choose a choice from “English” and “Mathematics” as shown in Figure 4.2, where A - D stand for “English”, “Machine Learning”, “Music”, and “Mathematics”, re- spectively, he/she will not consider the other two choices (i.e., “Machine Learning” and “Music”) because there is no order relationship among the choices.

Question: ?
Dist(A,D)

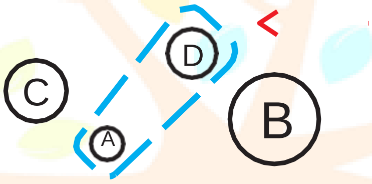


Figure 4.2: An example of multiple-choice question with nominal choices.

Based on the above discussions, the concept of cost can be extended to a unified case. That is, the meaning of the cost is the thinking cost for all the choices that should be considered for choosing a choice from two choices, no matter the choices are ordered or not. Accordingly, the concept of distance induced by the concept of thinking cost can be unified for ordinal and nominal attributes. By combining the unified distance metric and the proposed attribute weighting scheme, the distance between two categories o^r $_{i_r}$ and o^r $_{j_r}$ can be written as

max(i_r,j_r) *s=min*(i_r,j_r) □ $E_{o^r_s}$ □ ω_{A_r} · , *if* i_r j_r, $0 < r \leq d_{ord}$

Σ

$Dist(o^r , o^r) =$ □□ ω . Σ , *if i* j , $d < r \leq d$ (4.3.7)

$_{i_r}$ $_{j_r}$ E $_r$

$_{A_r}$ $_{s=i_r,j_r}$ $_{o_s}$ $_r$ $_r$ $_{ord}$

□ 0 , *if* $i_r = j_r$. □

Based on Eq. (4.3.7), the distance between categorical data objects can be written as

$Dist(\mathbf{x}_i, \mathbf{x}_j) =$ ’.,Σ^d $Dist(o^r , o^r)^2$. (4.3.8)

i_r j_r

r=1

Discussions

This sub-section discusses: 1) mathematical properties of UEBDM, 2) how to apply UEBDM for distance measurement in clustering analysis, and 3) time complexity for clustering analysis using UEBDM.

Mathematical Properties

The generalised distance metric defined in Eq. (4.3.7) can be utilized for calcu- lating the distance between two categories, no matter they are ordinal or nominal. Given *i, j, l, m* ∈ {1, 2, ..., *N* }, i_r, j_r, l_r, m_r ∈ {1, 2, ..., v_r}, and r ∈ {1, 2, ..., d}, the generalised weighted distances have the following properties:

$Dist(o^r , o^r) = Dist(o^r , o^r);$

i_r j_r j_r i_r

$0 \leq Dist(o^r , o^r) \leq 1;$ $_{i_r}$ $_{j_r}$

$Dist(o^r , o^r) = Dist(o^r ,$ E_{o^r} , iff o^r $\leq o^r$ $\leq o^r$ or

$o^r) + Dist(o^r , o^r) - \omega_A$

i_r l_r i_r j_r j_r l_r ^{r} $_{j_r}$ i_r j_r l_r

r r $\leq o^r$;

l_r j_r o $\leq o$ $_{i_r}$

$Dist(o^r , o^r) \leq$ $\leq o^r$ $\leq o^r$, or o^r $\leq o^r$ $\leq o^r$.

$Dist(o^r , o^r)$, if o^r $_{j_r}$ $_{j_r}$ $_{i_r}$ $_{l_r}$ $_{i_r}$ $_{j_r}$ $_{l_r}$ $_{l_r}$ $_{j_r}$ $_{i_r}$

$Dist(o^r , o^r) \leq Dist(o^r , o^r)$, if o^r $\leq \forall\{o^r , o^r \} \leq o^r$, or o^r $\leq \forall\{o^r , o^r \} \leq$

i_r j_r m_r l_r m_r i_r j_r l_r l_r i_r j_r

r m_r o ,

Calculate distance matrics and weight values for each attribute

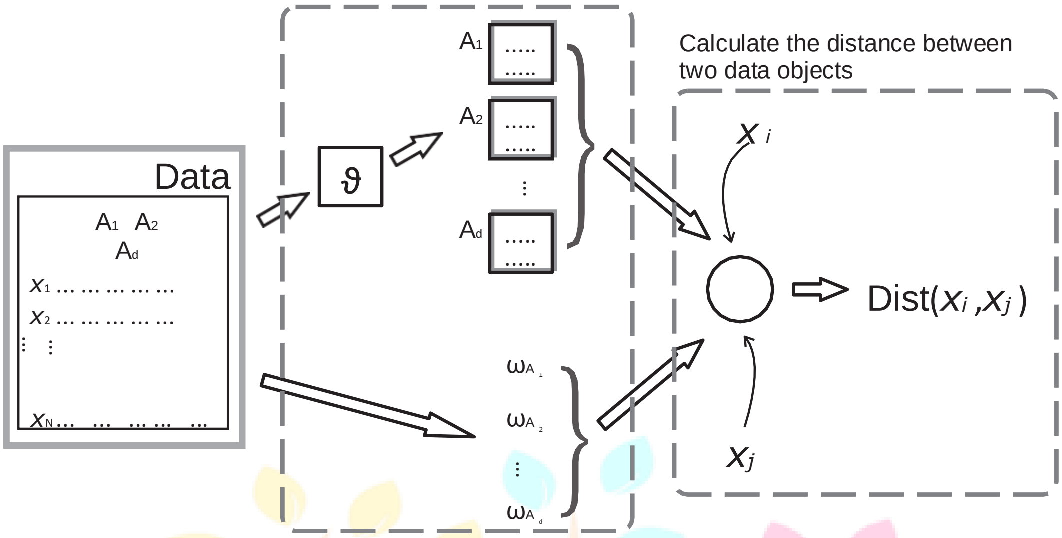


Figure 4.3: Work flow of UEBDM.

Accordingly, the object level distance defined by Eq. (4.3.8) has the follow- ing properties when $i, j, l, m \in \{1, 2, ..., N\}$, $i_r, j_r, l_r, m_r \in \{1, 2, ..., v_r\}$, and $r \in \{1, 2, ..., d\}$:

$Dist(\mathbf{x}_i, \mathbf{x}_j) = Dist(\mathbf{x}_j, \mathbf{x}_i)$;

$0 \le Dist(\mathbf{x}_i, \mathbf{x}_j) \le 1$;

$Dist(\mathbf{x}_i, \mathbf{x}_l) \le Dist(\mathbf{x}_i, \mathbf{x}_j) + Dist(\mathbf{x}_j, \mathbf{x}_l)$ if the categories representing $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_l$ satisfy $o^r_{i_r} \le o^r_{j_r} \le o^r_{l_r}$, or $o^r_{l_r} \le o^r_{j_r} \le o^r_{i_r}$;

$Dist(\mathbf{x}_i, \mathbf{x}_j) \le Dist(\mathbf{x}_i, \mathbf{x}_l)$, if the categories representing $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_l$ satisfy $o^r_{i_r} \le o^r_{j_r} \le o^r_{l_r}$, or $o^r_{l_r} \le o^r_{j_r} \le o^r_{i_r}$;

$Dist(\mathbf{x}_i, \mathbf{x}_j) \le Dist(\mathbf{x}_m, \mathbf{x}_l)$, if the categories representing $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_l$, and $\mathbf{x}_m$ satisfy $o^r_{m_r} \le \forall\{o^r_{i_r}, o^r_{j_r}\} \le o^r_{l_r}$, or $o^r_{l_r} \le \forall\{o^r_{i_r}, o^r_{j_r}\} \le o^r_{m_r}$,

Distance Measurement

The work-flow of distance measurement by using UEBDM is shown in Figure 4.3, where ϑ in the figure indicates the category level distance. The corresponding distance measurement algorithm is shown in Algorithm 7. To save computation cost in

Algorithm 7 Distance Measurement Using UEBDM

```
Input: Data set $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$.
Output: $Dist(\mathbf{x}_i, \mathbf{x}_j)$ for $i, j \in \{1, 2, ..., n\}$.
for $r = 1$ to $d$ do
    $R_{A_r} = \frac{E_{A_r}}{S_{A_r}}$;
end for
for $r = 1$ to $d$ do
    $\omega_{A_r} = \frac{R_{A_r}}{\sum_{s=1}^{d} R_{A_s}}$;
end for
for $r = 1$ to $d_{ord}$ do
    if $i_r$ /= $j_r$ then
        $Dist(o^r_{i_r}, o^r_{j_r}) = \omega_{A_r} \cdot \sum_{s=min(i_r,j_r)}^{max(i_r,j_r)} E_{o^r_s}$;
    else
        $Dist(o^r_{i_r}, o^r_{j_r}) = 0$;
    end if
end for
for $r = d_{ord} + 1$ to $d$ do
    if $i_r$ =/ $j_r$ then
        $Dist(o^r_{i_r}, o^r_{j_r}) = \omega_{A_r} \cdot \sum_{s=i_r,j_r} E_{o^r_s}$;
    else
        $Dist(o^r_{i_r}, o^r_{j_r}) = 0$;
    end if
end for
$Dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{r=1}^{d} Dist(o^r_{i_r}, o^r_{j_r})^2}$.
```

clustering analysis, distance matrices containing the distances between each pair of categories of each attribute can be calculated according to Algorithm 7 in advance. Then, distances between data objects can be easily read off from these matrices.

Time Complexity Analysis

The computation procedures of UEBDM-based distance measurement is composed of four parts: 1) calculate the v_r occurrence probabilities and entropy values of the categories from each attribute A_r, 2) calculate a $v_r \times v_r$ distance matrix for each A_r according to the v_r occurrence probabilities and entropy values, 3) calculate a weight value for each A_r according to the v_r entropy values, and 4) read off the distance between two data objects according to the prepared distance matrices and attribute wights. In clustering analysis, the computation in part 1-3 should be executed once, and then the distance matrices and attribute weights produced by part 2 and 3 are exploited in part 4 for distance reading off. We analyse the time complexity of each of the four parts as follows:

Time complexity for calculating the occurrence probabilities and entropy val- ues of v_r categories belonging to an attribute A_r is $O(N + v_r)$. For d attributes, the time complexity is $O(Nd + \sum_{r=1}^{d} v_r)$.

To calculate the distance between each pair of the v_r categories of an ordinal attribute A_r according to Eq. (4.3.7), the time complexity is the same as an- alyzed in Section 3.4.3 of Chapter 3, which is $O(\frac{v_r(v_r-1)}{2})$. If A_r is a nominal attribute, distances of $\frac{v_r(v_r-1)}{2}$ pairs of its categories can be directly calculated using Eq. (4.3.7) by adding up the two entropy values of each pair of the categories. This procedure has the same time complexity as calculating dis- tances for an ordinal attribute. Therefore, for d attributes in total, the time complexity is $O(\sum_{r=1}^{d} \frac{v_r(v_r-1)}{2})$.

To calculate the weight value of A_r, we should firstly sum up the v_r entropy values of A_r's categories and divide it by the standard information of A_r ac- cording to Eq. (3.3.5) and (4.3.5) to obtain the reliability R_{A_r} of A_r. Then the weight value of A_r can be obtained by dividing the reliability of A_r by the summation of the reliabilities of all the d attributes according to Eq. (4.3.6). Therefore, the time complexity for calculating the weight values of all the d attributes is $O(\sum_{r=1}^{d} v_r)$.

Time complexity for reading off the distance between two data objects accord- ing to the distance matrices and the weight values is $O(d)$.
According to the above analysis, we will further discuss if the time complexity of UEBDM will influence the time complexity of clustering analysis. Time complexity for calculating the distance matrices and weight values of all the d attributes is
$O(Nd + \sum_{r=1}^{d}(2v_r + \frac{v_r(v_r-1)}{2}))$. Since the value of v_r is different for each attribute,

we use $v_{max} = max(v_1, v_2, ..., v_d)$ instead of v_r in the following analysis. Based on v_{max}, the time complexity can be re-written as $O(Nd + v_{max}d + v_{max}^2 d)$. Since v_{max} is usually a small constant satisfying $v_{max}^2 < N$ in most of the real categorical data

sets, time complexity for calculating the distance matrices using UEBDM can be modified to $O(Nd)$. If the distance matrices of all the attributes are given, the time complexity for partitioning the data objects into k clusters using the simplest k-modes is $O(kdNI)$, where I is the number of iterations. Therefore, UEBDM still does not increase the overall time complexity of clustering analysis as EBDM.

Experiments

We embed the proposed UEBDM metric and its counterparts into different repre- sentative clustering algorithms. Their performance on different real and benchmark data sets is evaluated using several popular validity indices. Various experiments are conducted to illustrate the efficacy of UEBDM in categorical data clustering analysis.

Experimental Settings

Data Sets

To comprehensively evaluate the performance of the proposed distance metric, the selected data sets should include ordinal, nominal, and mixed categorial data set- s with different sizes and numbers of attributes. Twelve data sets, including five real data sets (i.e., Internship, Photo, Assistant, Fruit, Pillow) and seven bench- mark data sets (i.e., Employee, Lecturer, Hayes, Nursery, Solar, Voting, Tictac), are

Table 4.2: Statistics of the twelve data sets

Data set	Data type	# Ins.	# Att.(O)	# Att.(N)	# Class
Internship	Ordinal	90	3	0	3
Photo	Ordinal	66	4	0	3
Employee	Ordinal	1,000	4	0	9
Lecturer	Ordinal	1,000	4	0	5
Assistant	Categorical	72	2	2	3
Fruit	Categorical	100	3	2	5
Hayes	Categorical	132	2	2	3
Nursery	Categorical	12,960	6	2	4
Pillow	Nominal	100	0	4	5
Solar	Nominal	323	0	9	6
Voting	Nominal	435	0	16	2
Tictac	Nominal	958	0	9	2

collected for the experiments. Among the twelve data sets, four of them (i.e., Intern- ship, Photo, Employee, and Lecturer) are ordinal data sets. Another four of them (i.e., Assistant, Fruit, Hayes, and Nursery) are mixed categorical data sets. The Remainder four (i.e., Pillow, Solar, Voting, and Tictac) are nominal data sets. Em- ployee and Lecturer are collected from Weka website [122]. Hayes, Nursery, Solar, Voting, and Tictac are collected from the UCI Machine Learning Repository [37]. Internship is collected from the students' questionnaires of the Education University of Hong Kong. Photo and Assistant are collected from the student questionnaires of the College of International Exchange of Shenzhen University. Fruit and Pillow are collected from the business survey of an advertising company. Statistics of the twelve data sets are shown in Table 4.2. "Att.(O)" and "Att.(N)" indicate ordinal and nominal attributes, respectively.

Counterparts

To compare the metrics, we embed them into different distance-based clustering algorithms, and then evaluate their clustering performance. The metrics, clustering algorithms, and validity indices are described as follows.

The commonly used Hamming Distance Metric (HDM) [58] is selected as a base- line. In addition, Ahmad's Distance Metric (ADM) [10], Association-Based Distance Metric (ABDM) [80], Context-Based Distance Metric [65] and Jia's Distance Metric (JDM) [72] are selected as state-of-the-art counterparts in the experiments.

K-Modes (KM) clustering algorithm [63], which is the most commonly used one for categorical data clustering, is selected as a baseline. The attribute weighting ver- sion of KM (i.e., WKM [62]), which can automatically weight the contributions of attributes for clustering, is also selected. In general, subspace clustering algorithm can achieve better performance than the conventional ones. Therefore, a representative subspace clustering algorithm (i.e., Entropy Weighting k-means (EW) [73]) and a state-of-the-art subspace clustering algorithm (i.e., Weighted OCIL (WOC) [70]) are also selected. Besides the above four distance-based clustering algorithms, a representative evaluation-based clustering algorithm (i.e., Entropy-Based Clustering (EBC) algorithm [85]) is also chosen.

Validity Indices

For each data set, the performance of different metrics embedded in different clus- tering algorithms is averaged on 10 runs. The clustering performance is evaluated using three powerful and popular validity indices (i.e., Clustering Accuracy (CA)

[120] [59], Adjusted Rand Index (ARI) [105] [108] [119] [52], and Normalised Mutual Information (NMI) [41] [34]).

Clustering Performance on Ordinal and Mixed Cate- gorical Data

To prove the superiority of the proposed UEBDM in clustering categorical data set- s with ordinal attributes, we embed UEBDM and all its counterparts (i.e., HDM, ADM, ABDM, CBDM, and JDM) into the four selected clustering algorithms (i.e., KM, WKM, EW, and WOC) and compare the clustering performance of them and the EBC clustering algorithm on the four ordinal data sets (i.e., Internship, Photo, Employee, and Lecturer) and the four mixed categorical data sets (i.e., Assistant, Fruit, Hayes, and Nursery). In WKM, EW, and WOC clustering algorithms, dis- tance between intra-attribute categories should be calculated to update the weights of attributes. Because JDM directly calculates distance between objects, and can- not calculate the distance between intra-attribute categories, JDM is not embedded into them for experiments. WOC with its original object-cluster similarity measure is also compared in this experiment. In addition, since EBC is not a distance-based algorithm, we directly compare it with the other algorithms without embedding metrics into it.

Clustering performance in terms of CA, ARI, and NMI on the four ordinal data sets are compared in Table 4.3 - 4.5. Hereinafter, experimental results highlighted by boldface and underline indicate the best and the second best results, respectively.

It can be observed that the performance of UEBDM is the best on almost all the ordinal data sets no matter which clustering algorithm is utilized. Only its CA performance on Employee data set and NMI performance on Internship data set by using WKM is not the best. But it is still the second best and the

gap between it and the best result is very tiny (i.e., 0.005 in terms of CA on Employee and 0.003 in terms of NMI on Internship). Among all the compared metrics, only UEBDM has the mechanism to specially exploit order information of ordinal attributes. This is the reason why UEBDM is superior to the other existing metrics for ordinal data clustering.
To further evaluate the performance of UEBDM on categorical data sets composed of both ordinal and nominal attributes, clustering performance in terms of CA, ARI, and NMI on the four mixed categorical data sets are compared in Table 4.6 - 4.8.

Table 4.3: Averaged CA on four ordinal data sets.

Alg.	Metric	Internship	Photo	Employee	Lecturer
KM	UEBDM	**0.582±0.06**	**0.614±0.05**	**0.208±0.01**	**0.370±0.03**
	HDM	0.562±0.06	0.514±0.07	0.190±0.01	0.344±0.03
	ADM	0.533±0.01	0.503±0.04	**0.208±0.01**	0.313±0.03
	ABDM	0.528±0.01	0.538±0.08	0.200±0.01	0.316±0.02
	CBDM	0.507±0.01	0.541±0.06	0.198±0.01	0.308±0.03
	JDM	0.558±0.02	0.486±0.07	0.189±0.01	0.331±0.04
WKM	UEBDM	**0.571±0.04**	**0.535±0.09**	0.195±0.01	**0.364±0.03**
	HDM	0.559±0.05	0.486±0.09	0.192±0.01	0.339±0.04
	ADM	0.503±0.01	0.506±0.08	**0.200±0.01**	0.340±0.06
	ABDM	0.517±0.03	0.512±0.07	**0.200±0.01**	0.332±0.06
	CBDM	0.502±0.01	0.465±0.07	**0.200±0.01**	0.326±0.01
EW	UEBDM	**0.608±0.07**	**0.609±0.06**	**0.207±0.01**	**0.377±0.04**
	HDM	0.558±0.03	0.532±0.06	0.193±0.01	0.344±0.04
	ADM	0.529±0.02	0.530±0.06	0.205±0.01	0.317±0.03
	ABDM	0.562±0.04	0.529±0.05	0.205±0.01	0.325±0.02
	CBDM	0.516±0.01	0.544±0.07	0.202±0.01	0.310±0.02
WOC	UEBDM	**0.640±0.12**	**0.586±0.08**	**0.203±0.01**	**0.362±0.03**
	HDM	0.563±0.05	0.542±0.10	0.187±0.01	0.332±0.06
	ADM	0.508±0.02	0.498±0.08	0.201±0.01	0.325±0.02
	ABDM	0.500±0.00	0.515±0.08	0.198±0.01	0.337±0.01
	CBDM	0.500±0.00	0.568±0.04	0.197±0.01	0.318±0.02
	-	0.553±0.06	0.521±0.09	0.196±0.01	0.331±0.03
EBC	-	0.566±0.06	0.512±0.08	0.196±0.01	0.348±0.03

Table 4.4: Averaged ARI on four ordinal data sets.

Alg.	Metric	Internship	Photo	Employee	Lecturer
KM	UEBDM	**0.024±0.04**	**0.245±0.08**	**0.026±0.00**	**0.068±0.03**
	HDM	0.004±0.05	0.096±0.07	0.009±0.01	0.045±0.01
	ADM	-0.005±0.00	0.117±0.06	0.018±0.00	0.036±0.01
	ABDM	-0.007±0.00	0.158±0.09	0.011±0.01	0.035±0.01
	CBDM	-0.014±0.00	0.117±0.06	0.014±0.00	0.033±0.02
	JDM	0.004±0.01	0.071±0.06	0.013±0.00	0.042±0.02
WKM	UEBDM	**0.011±0.02**	**0.108±0.09**	**0.018±0.01**	**0.055±0.03**
	HDM	0.008±0.03	0.072±0.09	0.013±0.00	0.042±0.03
	ADM	-0.012±0.00	0.086±0.08	0.013±0.01	0.040±0.03
	ABDM	-0.010±0.00	0.100±0.10	0.010±0.01	0.037±0.05
	CBDM	-0.014±0.00	0.050±0.07	0.017±0.00	0.027±0.00
EW	UEBDM	**0.049±0.08**	**0.246±0.08**	**0.026±0.00**	**0.073±0.03**
	HDM	0.003±0.02	0.121±0.06	0.010±0.01	0.046±0.02
	ADM	-0.006±0.01	0.141±0.06	0.016±0.00	0.038±0.02
	ABDM	0.007±0.02	0.141±0.05	0.011±0.01	0.042±0.01
	CBDM	-0.012±0.00	0.115±0.06	0.015±0.00	0.035±0.01
WOC	UEBDM	**0.109±0.12**	**0.171±0.09**	**0.024±0.01**	**0.060±0.01**
	HDM	0.008±0.03	0.131±0.09	0.012±0.00	0.046±0.04
	ADM	-0.010±0.00	0.082±0.08	0.016±0.01	0.033±0.01
	ABDM	-0.011±0.00	0.113±0.07	0.012±0.00	0.037±0.00
	CBDM	-0.017±0.00	0.124±0.04	0.015±0.01	0.032±0.01
	-	0.004±0.04	0.090±0.09	0.014±0.00	0.038±0.02
EBC	-	0.013±0.03	0.121±0.10	0.011±0.00	0.041±0.01

According to the results, it can be found that UEBDM is still competitive for mixed categorical data clustering, because most of the best and the second best results are achieved by UEBDM-based clustering algorithms. However, superiority

Table 4.5: Averaged NMI on four ordinal data sets.

Alg.	Metric	Internship	Photo	Employee	Lecturer
KM	UEBDM	**0.018±0.02**	**0.281±0.09**	**0.083±0.01**	**0.092±0.04**
	HDM	0.015±0.02	0.126±0.06	0.048±0.01	0.070±0.02
	ADM	0.005±0.00	0.170±0.05	0.062±0.01	0.055±0.01
	ABDM	0.004±0.00	0.222±0.08	0.055±0.01	0.056±0.02
	CBDM	0.006±0.00	0.157±0.04	0.052±0.01	0.059±0.02
	JDM	0.014±0.01	0.099±0.06	0.056±0.01	0.074±0.03
WKM	UEBDM	0.015±0.02	**0.162±0.11**	**0.065±0.02**	**0.085±0.04**
	HDM	**0.018±0.01**	0.128±0.12	0.055±0.01	0.071±0.04
	ADM	0.003±0.00	0.146±0.09	0.051±0.01	0.065±0.04
	ABDM	0.001±0.00	0.157±0.10	0.050±0.01	0.068±0.06
	CBDM	0.004±0.00	0.113±0.07	0.061±0.01	0.052±0.01
EW	UEBDM	**0.039±0.06**	**0.288±0.08**	**0.083±0.01**	**0.100±0.04**
	HDM	0.018±0.02	0.142±0.05	0.051±0.01	0.073±0.03
	ADM	0.005±0.00	0.194±0.06	0.058±0.01	0.056±0.02
	ABDM	0.012±0.01	0.213±0.06	0.057±0.01	0.065±0.02
	CBDM	0.008±0.00	0.158±0.05	0.057±0.01	0.061±0.01
WOC	UEBDM	**0.068±0.08**	**0.223±0.10**	**0.077±0.01**	**0.088±0.02**
	HDM	0.016±0.01	0.196±0.11	0.052±0.01	0.072±0.05
	ADM	0.002±0.00	0.142±0.09	0.056±0.01	0.054±0.01
	ABDM	0.001±0.00	0.192±0.07	0.060±0.00	0.059±0.01
	CBDM	0.007±0.00	0.201±0.06	0.054±0.01	0.057±0.02
	-	0.020±0.02	0.140±0.10	0.054±0.01	0.061±0.03
EBC	-	0.015±0.02	0.160±0.11	0.049±0.01	0.058±0.02

of UEBDM in clustering mixed categorical data is not as significant as its superiority in clustering ordinal data. This is because that, all the other compared metrics are actually designed for nominal attributes. A data set with more nominal attributes

Table 4.6: Averaged CA on four mixed categorical data sets.

Alg.	Metric	Assistant	Fruit	Hayes	Nursery
	UEBDM	**0.603±0.07**	0.540±0.05	**0.417±0.05**	**0.384±0.02**

KM	HDM	0.538±0.07	0.456±0.04	0.386±0.03	0.360±0.04
	ADM	0.556±0.10	0.516±0.03	0.381±0.03	-
	ABDM	0.579±0.08	**0.548±0.06**	0.397±0.03	-
	CBDM	0.586±0.06	0.527±0.05	0.389±0.05	-
	JDM	0.526±0.06	0.451±0.06	0.380±0.02	0.329±0.03
WKM	UEBDM	0.539±0.11	0.504±0.03	**0.497±0.06**	**0.429±0.11**
	HDM	0.525±0.10	0.449±0.03	0.439±0.05	0.387±0.05
	ADM	0.560±0.12	**0.509±0.02**	0.440±0.03	-
	ABDM	**0.596±0.15**	0.499±0.01	0.359±0.04	-
	CBDM	0.499±0.07	0.494±0.03	0.405±0.08	-
EW	UEBDM	**0.604±0.08**	0.546±0.05	0.402±0.06	**0.372±0.04**
	HDM	0.567±0.08	0.460±0.03	0.392±0.04	0.333±0.00
	ADM	0.565±0.09	0.516±0.03	0.379±0.03	-
	ABDM	0.568±0.07	**0.561±0.05**	**0.415±0.03**	-
	CBDM	0.581±0.06	0.525±0.05	0.386±0.05	-
WOC	UEBDM	**0.628±0.10**	0.521±0.05	0.403±0.07	**0.365±0.03**
	HDM	0.508±0.10	0.496±0.05	0.379±0.05	0.360±0.04
	ADM	0.539±0.10	0.513±0.02	0.381±0.05	-
	ABDM	0.531±0.08	**0.537±0.05**	**0.413±0.05**	-
	CBDM	0.553±0.04	0.505±0.04	0.396±0.08	-
	-	0.565±0.10	0.484±0.06	0.402±0.07	0.355±0.05
EBC	-	0.522±0.07	0.447±0.04	0.360±0.04	0.360±0.04

will therefore shorten the performance gap between a nominal data distance metric and UEBDM. In addition, the CA, ARI, and NMI performance of several metrics on Nursery data set are not reported because these metrics are incapable for the

Table 4.7: Averaged ARI on four mixed categorical data sets.

Alg.	Metric	Assistant	Fruit	Hayes	Nursery

KM	UEBDM	**0.210±0.07**	0.293±0.05	**0.012±0.04**	**0.068±0.02**
	HDM	0.111±0.07	0.195±0.07	-0.002±0.02	0.044±0.03
	ADM	0.161±0.08	0.283±0.04	-0.003±0.01	-
	ABDM	0.196±0.09	**0.319±0.06**	0.001±0.01	-
	CBDM	0.168±0.06	0.283±0.04	0.002±0.02	-
	JDM	0.102±0.05	0.176±0.07	-0.004±0.01	0.030±0.02
WKM	UEBDM	0.124±0.11	0.240±0.02	**0.061±0.03**	**0.124±0.17**
	HDM	0.107±0.09	0.207±0.04	0.021±0.02	0.043±0.06
	ADM	0.145±0.14	0.253±0.02	0.025±0.01	-
	ABDM	**0.215±0.20**	**0.266±0.01**	-0.009±0.01	-
	CBDM	0.061±0.06	0.261±0.03	0.011±0.04	-
EW	UEBDM	**0.213±0.07**	0.297±0.05	0.007±0.04	**0.055±0.03**
	HDM	0.145±0.09	0.208±0.06	0.001±0.02	0.000±0.00
	ADM	0.172±0.06	0.283±0.04	-0.004±0.01	-
	ABDM	0.186±0.07	**0.329±0.06**	**0.011±0.01**	-
	CBDM	0.165±0.06	0.278±0.04	0.000±0.02	-
WOC	UEBDM	**0.228±0.12**	0.241±0.05	**0.015±0.04**	**0.056±0.03**
	HDM	0.091±0.10	0.180±0.06	-0.001±0.02	0.044±0.03
	ADM	0.121±0.10	0.256±0.03	0.001±0.02	-
	ABDM	0.141±0.10	**0.282±0.03**	0.011±0.02	-
	CBDM	0.094±0.04	0.240±0.03	0.013±0.04	-
	-	0.142±0.09	0.205±0.06	0.012±0.04	0.023±0.02
EBC	-	0.094±0.07	0.160±0.05	0.044±0.03	0.044±0.03

distance measurement of a data set with independent attributes, like Nursery. Since UEBDM exploits more order information offered by the ordinal attributes of Nursery data set, UEBDM-based clustering algorithms perform better than the others.

Table 4.8: Averaged NMI on four mixed categorical data sets.

Alg.	Metric	Assistant	Fruit	Hayes	Nursery
	UEBDM	**0.246±0.06**	0.460±0.04	**0.027±0.05**	**0.081±0.02**
	HDM	0.136±0.07	0.358±0.08	0.019±0.03	0.047±0.02

KM	ADM	0.209±0.08	0.446±0.04	0.011±0.01	-
	ABDM	0.239±0.10	**0.492±0.05**	0.013±0.01	-
	CBDM	0.190±0.05	0.447±0.04	0.017±0.03	-
	JDM	0.122±0.07	0.322±0.09	0.012±0.01	0.036±0.03
WKM	UEBDM	0.163±0.13	0.428±0.01	**0.065±0.03**	**0.156±0.23**
	HDM	0.147±0.12	0.383±0.03	0.032±0.03	0.052±0.08
	ADM	0.194±0.13	0.438±0.03	0.030±0.01	-
	ABDM	**0.256±0.18**	**0.458±0.02**	0.005±0.01	-
	CBDM	0.106±0.07	0.455±0.04	0.023±0.03	-
EW	UEBDM	**0.249±0.06**	0.464±0.04	**0.024±0.05**	**0.063±0.03**
	HDM	0.166±0.09	0.371±0.08	0.021±0.04	0.000±0.00
	ADM	0.218±0.07	0.446±0.04	0.010±0.01	-
	ABDM	0.231±0.07	**0.501±0.04**	0.022±0.01	-
	CBDM	0.185±0.06	0.441±0.05	0.013±0.02	-
WOC	UEBDM	**0.267±0.11**	0.419±0.04	0.026±0.03	**0.073±0.05**
	HDM	0.131±0.12	0.346±0.08	0.011±0.02	0.047±0.02
	ADM	0.171±0.11	0.436±0.02	0.012±0.02	-
	ABDM	0.197±0.09	**0.455±0.03**	0.019±0.01	-
	CBDM	0.159±0.04	0.421±0.03	0.024±0.04	-
	-	0.201±0.12	0.379±0.08	**0.030±0.05**	0.045±0.04
EBC	-	0.122±0.07	0.293±0.07	0.047±0.02	0.047±0.02

It has been pointed out by [20] and [39] that distance metric is data sensitive, and cannot always outperform the others on different data sets. Therefore, although the clustering performance of UEBDM is not always the best on the above mentioned eight data sets, the above experimental results are still sufficient to prove the effectiveness and robustness of UEBDM in clustering analysis.

According to the comparison of the clustering performance of different clustering algorithms, EBC performs slightly better than the traditional KM in general. The other three state-of-the-art algorithms (i.e., WKM, EW, and WOC) are obviously more powerful because the best clustering results of each distance metric on different data sets are usually produced by one of them. In the following experiments, all the metrics are embedded into WOC, which is the most comprehensive one among the state-of-the-art clustering algorithms.

Clustering Performance on Nominal Data

To illustrate that UEBDM is also competent in clustering nominal data, we compare the clustering performance of it with the other counterparts on the four nominal data sets (i.e., Pillow, Solar, Voting, and Tictac). Corresponding clustering performance is shown in Figure 4.4. The clustering performance on Pillow, Solar, Voting, and Tictac data sets are demonstrated in row 1, row 2, row 3, and row 4 of Figure 4.4, respectively. In this experiment, original object-cluster similarity measure of WOC (denoted by MWOC) is also compared for completeness.

According to the results, we can find that even all the four data sets are nominal data, and all the other compared metrics are originally designed for nominal data, UEBDM is still competitive. More specifically, the performance of UEBDM is always ranked in the top 3, and it even outperforms the other counterparts on Solar and Tictac data sets.

Evaluation of UEBDM and UEBDMnom

The core idea of the proposed UEBDM is to exploit the order information of ordinal attributes for distance measurement. To verify the reasonableness of its order infor- mation exploiting mechanism, we compare the clustering performance of UEBDM with UEBDM^{nom}, which is the nominal version of UEBDM. UEBDM treats ordinal and nominal attributes differently according to Eq. (4.3.7), while UEBDM^{nom} treats

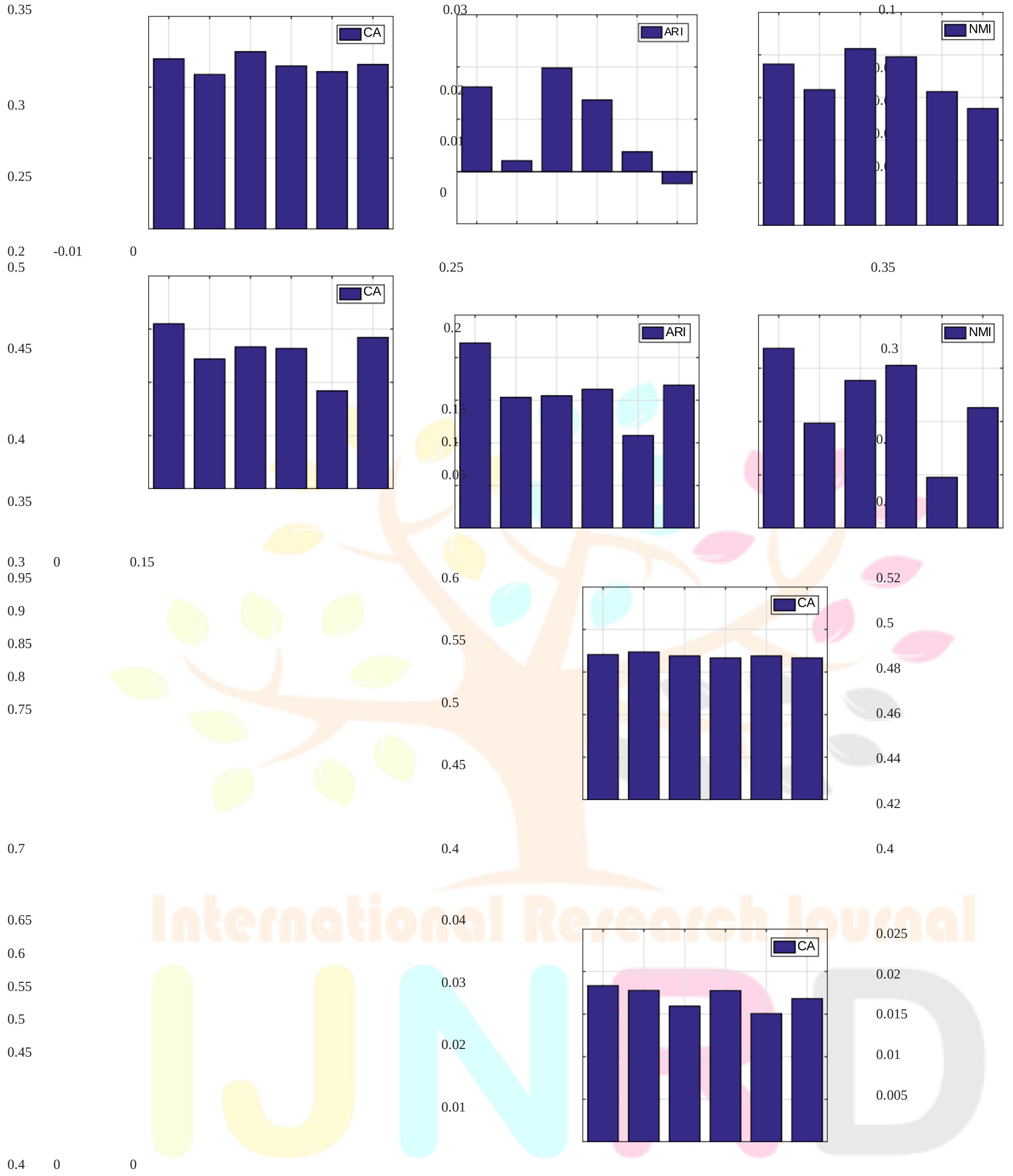


Figure 4.4: Clustering performance on four nominal data sets.

all types of attributes as nominal ones. If the performance of UEBDM outperform- s UEBDM^{nom}, effectiveness of the order information exploiting mechanism can be proved. Since clustering performance of UEBDM and UEBDM^{nom} on nominal data sets are identical, experimental results on the four nominal data sets are omitted in this experiment. Clustering performance of UEBDM and UEBDM^{nom} on the eight data sets with ordinal attributes are compared in Figure 4.5 - 4.7.

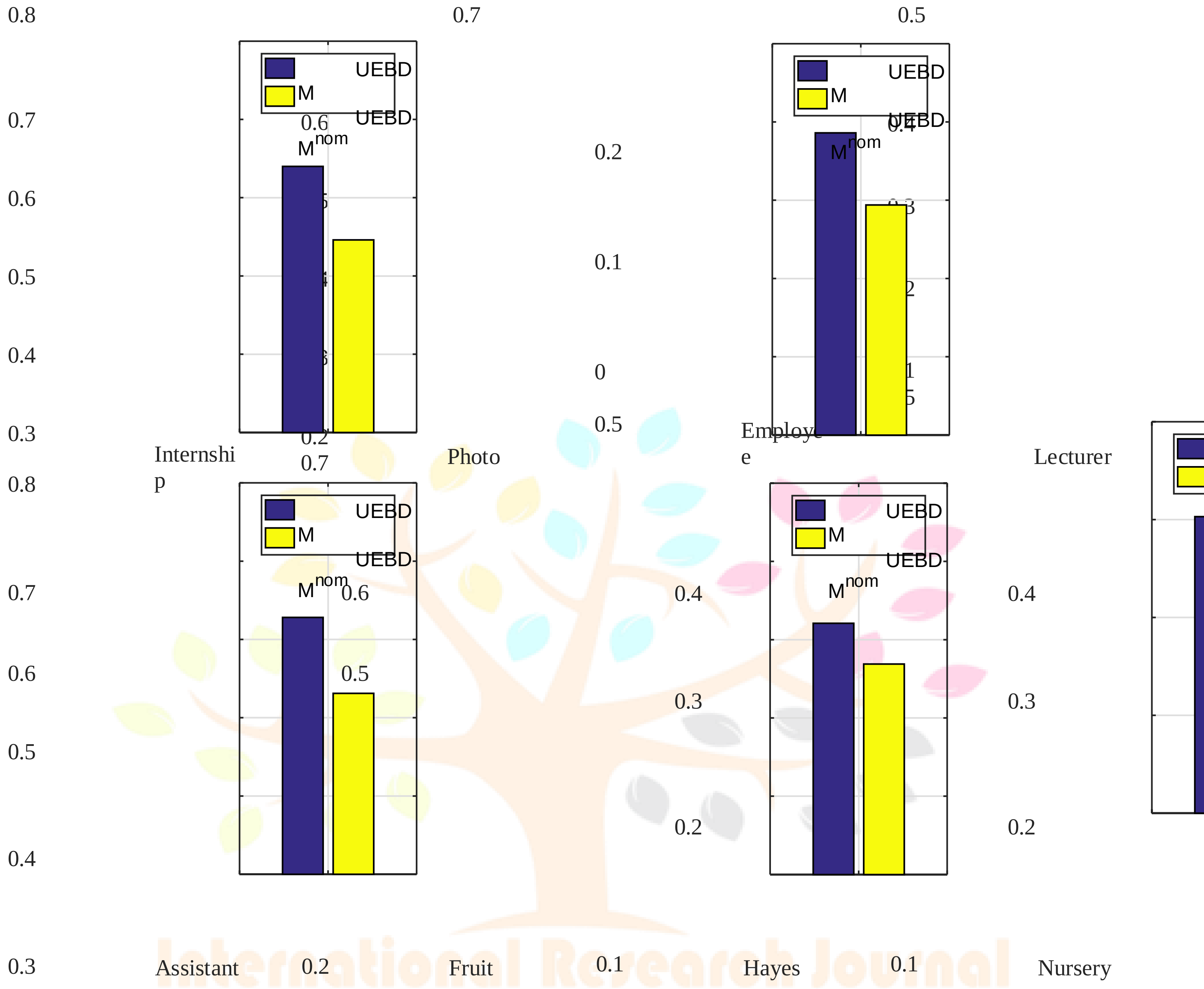


Figure 4.5: Averaged CA of UEBDM and UEBDM^{nom} on four ordinal (row 1) and four mixed categorical (row 2) data sets .

It can be observed from the histograms that UEBDM outperforms UEBDM^{nom} on all the eight data sets. This indicates that UEBDM is effective in exploiting the order information of ordinal attributes for more accurate clustering analysis. Since UEBDM^{nom} treats all the attributes as nominal ones, order information is completely ignored by it. Results of this experiment also once again proved the reasonableness of our core idea (i.e., ordinal attributes should be treated differently to exploit more valuable information for clustering analysis).

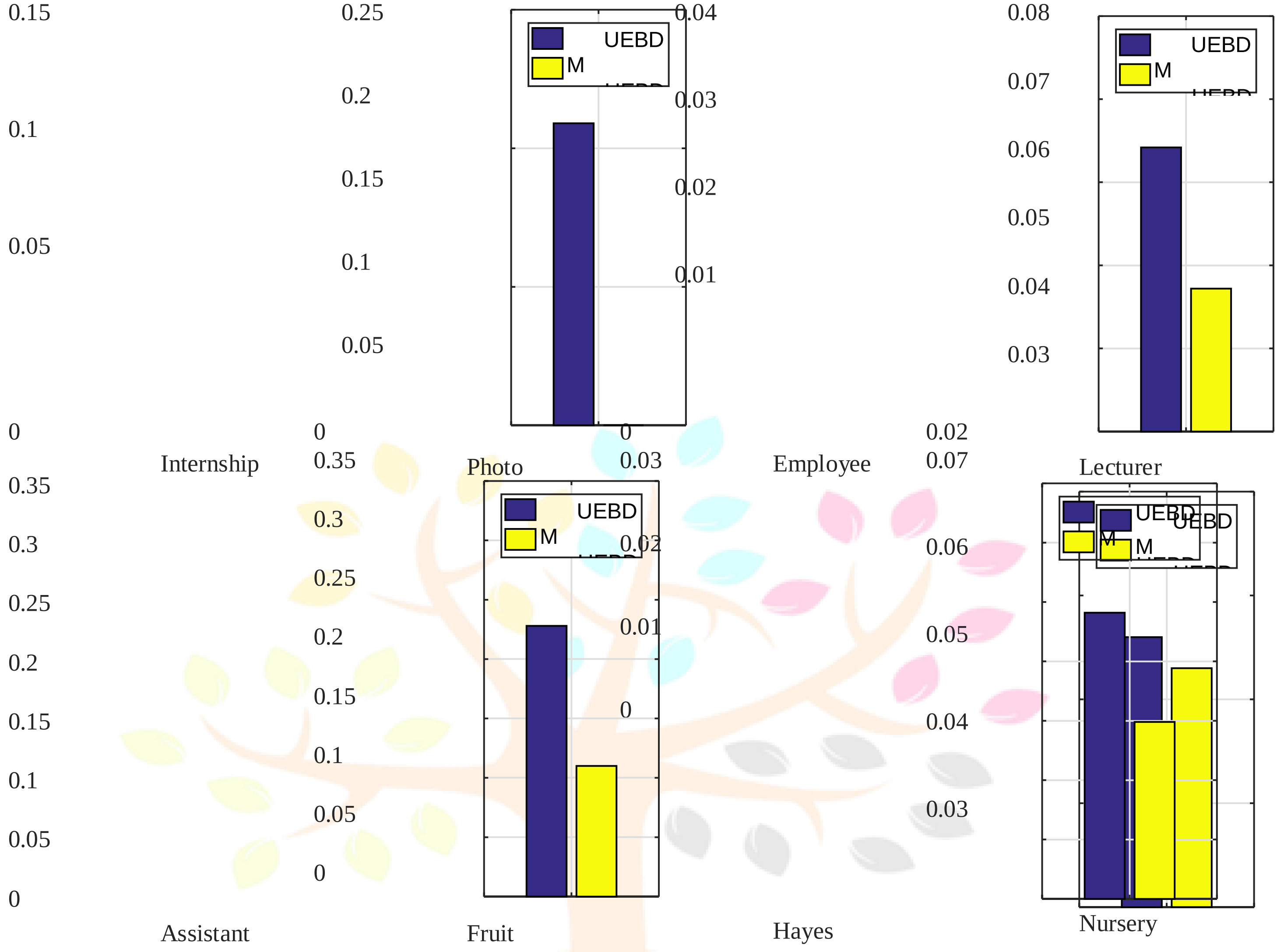


Figure 4.6: Averaged ARI of UEBDM and UEBDM^{nom} on four ordinal (row 1) and four mixed categorical (row 2) data sets.

Weighting Scheme Evaluation

To illustrate the effectiveness of the attribute weighting scheme in UEBDM, we com- pare the clustering performance of UEBDM and its no-weighting version (denoted by UEBDM^{o}) on all the twelve data sets. Their performance is compared in Table
4.9 - 4.11. The best results are highlighted using boldface.

It can be observed that the clustering performance of UEBDM with attribute weighting outperforms the version without attribute weighting on most of the data sets, which indicates that the attribute weighting scheme can effectively weight the contributions of different attributes during the distance measurement.

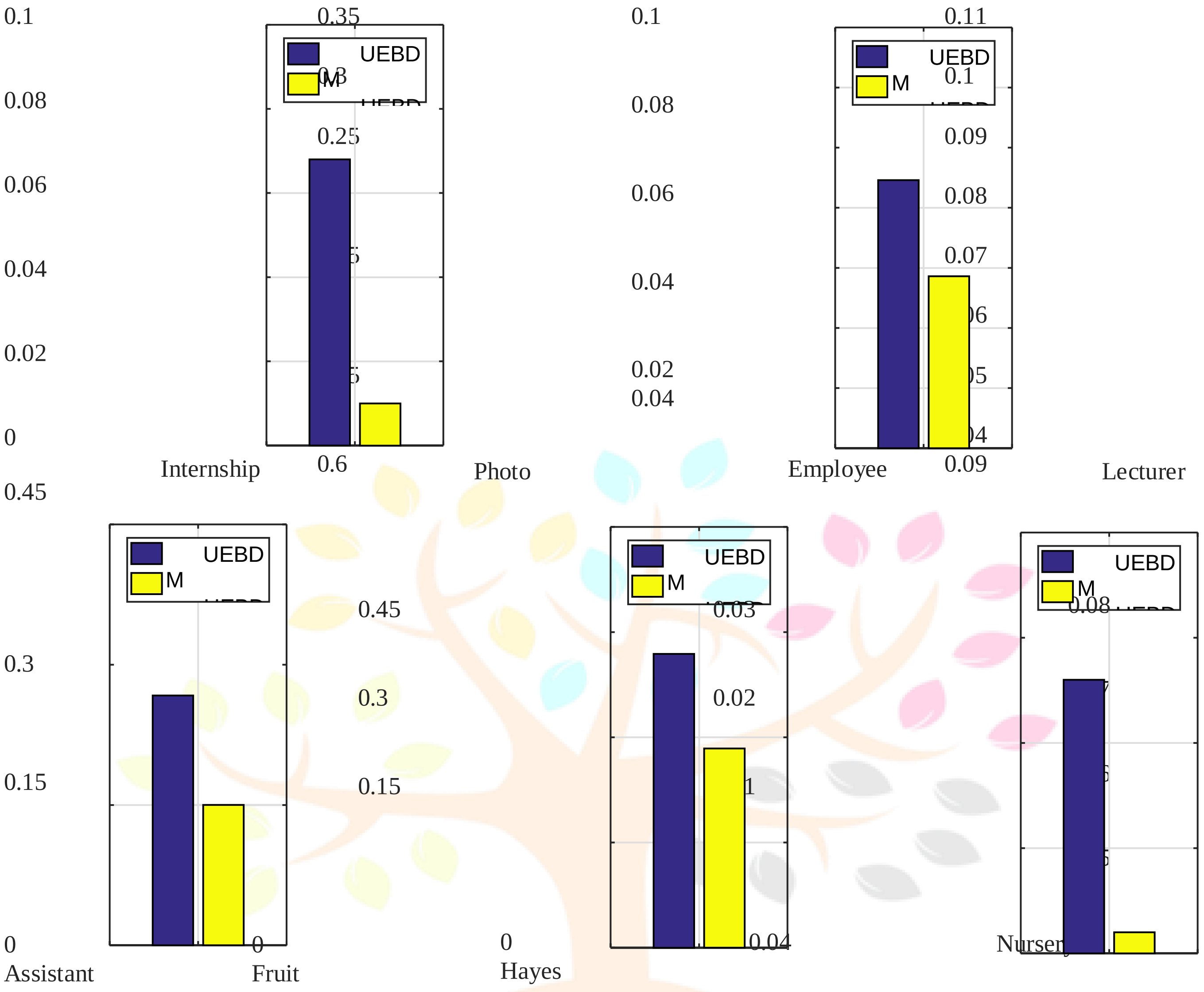


Figure 4.7: Averaged NMI of UEBDM and UEBDM^{nom} on four ordinal (row 1) and four mixed categorical (row 2) data sets.

Table 4.9: Performance of UEBDM and UEBDM^o on four ordinal data sets.

Index	Metric	Internship	Photo	Employee	Lecturer
CA	UEBDM	**0.640±0.12**	**0.586±0.08**	**0.203±0.01**	**0.362±0.03**
	UEBDM^o	0.618±0.10	0.573±0.08	**0.203±0.01**	0.352±0.02
ARI	UEBDM	**0.109±0.12**	**0.171±0.09**	**0.024±0.01**	**0.060±0.01**
	UEBDM^o	0.073±0.09	0.166±0.09	**0.024±0.01**	0.057±0.01
NMI	UEBDM	**0.068±0.08**	**0.223±0.10**	**0.077±0.01**	0.088±0.02
	UEBDM^o	0.057±0.06	0.215±0.10	**0.077±0.01**	**0.091±0.02**

Table 4.10: Performance of UEBDM and UEBDM^o on four mixed categorical data sets.

Index	Metric	Assistant	Fruit	Hayes	Nursery
CA	UEBDM	**0.628±0.10**	**0.521±0.05**	**0.403±0.07**	**0.365±0.03**
	UEBDM*o*	0.622±0.09	0.512±0.04	0.402±0.06	0.360±0.03
ARI	UEBDM	**0.228±0.12**	**0.241±0.05**	**0.015±0.04**	0.056±0.03
	UEBDM*o*	0.220±0.10	0.236±0.04	0.011±0.03	**0.057±0.03**
NMI	UEBDM	0.267±0.11	**0.419±0.04**	**0.026±0.03**	**0.073±0.05**
	UEBDM*o*	**0.270±0.10**	0.416±0.04	0.023±0.03	0.070±0.04

Table 4.11: Performance of UEBDM and UEBDMo on four nominal data sets.

Index	Metric	Pillow	Solar	Voting	Tictac
CA	UEBDM	**0.320±0.02**	**0.455±0.07**	**0.871±0.00**	**0.584±0.03**
	UEBDM*o*	0.316±0.02	0.425±0.03	**0.871±0.00**	0.577±0.03
ARI	UEBDM	**0.016±0.02**	**0.217±0.10**	0.548±0.00	**0.029±0.02**
	UEBDM*o*	0.013±0.03	0.186±0.06	**0.549±0.00**	0.026±0.02
NMI	UEBDM	**0.076±0.02**	**0.319±0.10**	**0.483±0.00**	**0.020±0.01**
	UEBDM*o*	0.074±0.02	0.289±0.06	**0.483±0.00**	0.018±0.01

Distance Matrices Demonstration

To intuitively observe if the distances produced by different metrics are consisten- t with the natural distance structure of the data sets, we compare the distance matrices produced by different metrics in this experiment. All the distance values are normalised into the interval [0,1], and the distance matrices are converted into grey-scale maps accordingly. Lighter pixels indicate larger distance and vice versa. Therefore, for the distance matrix of an ordinal attribute, pixels on the diagonal from left-top corner to the right-bottom corner should be pure black, while the pix- els locate towards the right-top and left-bottom corners should be lighter. Distance matrices of Assistant data set are demonstrated in Figure 4.8. “(O)” and “(N)” indicate ordinal and nominal attributes of Assistant data set, respectively. Distance matrices produced by UEBDM, HDM, ADM, ABDM, and CBDM are demonstrated in row 1, row 2, row 3, row 4, and row 5 of Figure 4.8, respectively. JDM metric is not compared in this experiment because it cannot directly compute the distance between intra-attribute categories.

Att. 1 (O) Att. 2 (O) Att. 3 (N) Att. 4 (N)

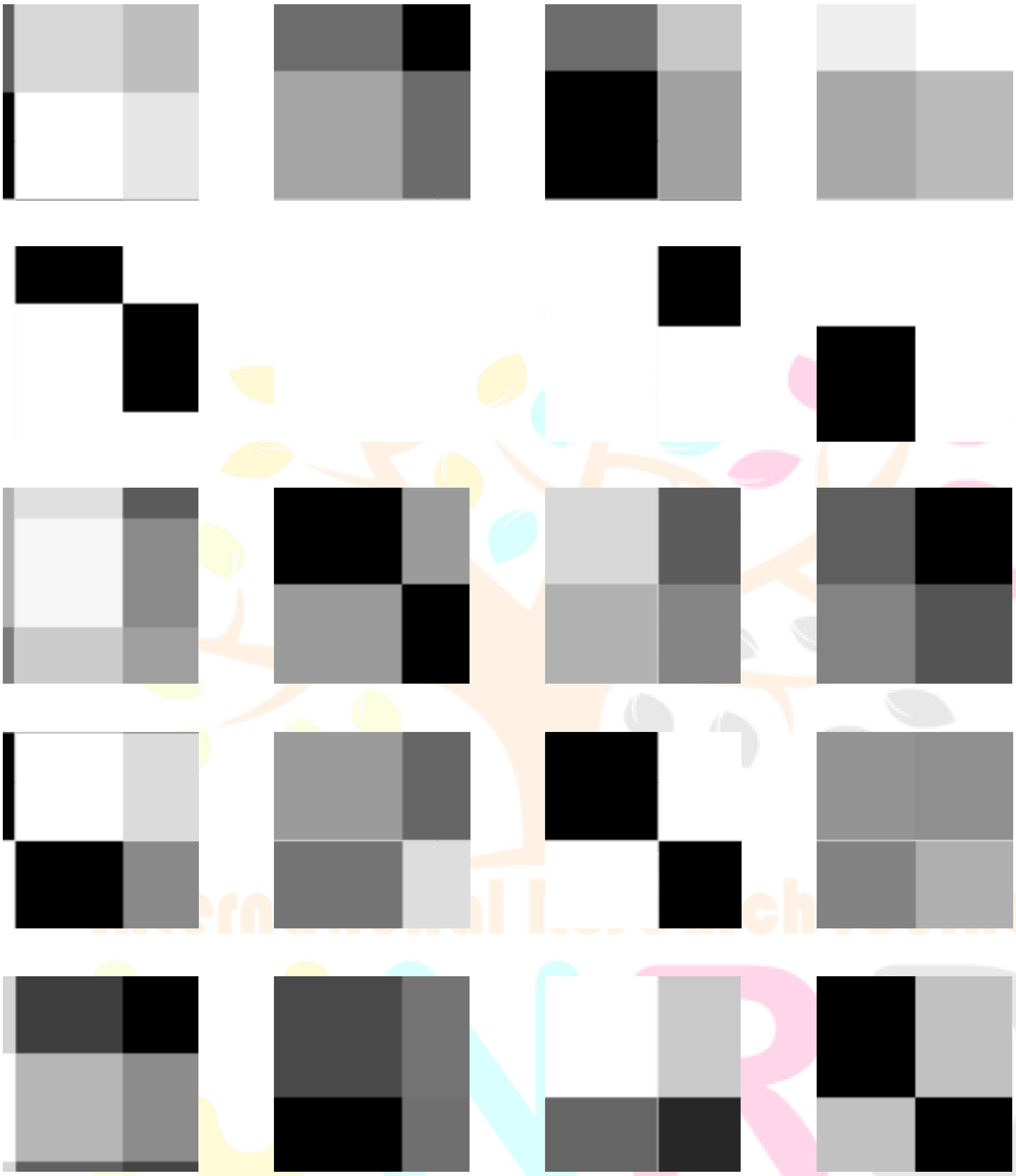


Figure 4.8: Distance matrices produced for Assistant data set.

It can be observed that only the distance matrices produced by UEBDM are completely consistent with the order structure of the two ordinal attributes. Since HDM assigns distances “0” and “1” to all the pairs of identical and different categories, it is incapable to indicate the distance structures of ordinal attributes. The distance matrices produced by ADM, ABDM, and CBDM can roughly indicate the distance structures of the two ordinal attributes, but a certain amount of distances produced by them are still disordered (i.e., the pixels are not gradually lighter towards the right-top and left-bottom corners from the diagonal). This is also the reason why their performance is superior to HDM, but is inferior to UEBDM as shown in the experimental results in Section 4.5.2. For the other two nominal attributes, it is reasonable that their distance matrices produced by all the compared metrics are unordered.

4.6 Summary

In this chapter, we have proposed a distance metric for categorical data clustering, called UEBDM, from the perspective of information entropy. In contrast with the existing categorical data metrics, the proposed one treats ordinal attributes and nominal attributes differently, but unifies the concept of the distance and impor- tance of them, which avoids information loss during the distance measurement. For ordinal attributes, order information is taken into account for the distance measure- ment while for nominal attributes, statistical information is exploited. Since the distance concepts of ordinal and nominal attributes are unified, it is not necessary to separately compute the distances on ordinal and nominal attributes, and then weight and combine them to produce the final distances. Moreover, the proposed metric is easy to use and non-parametric, which can be easily applied for the clus- tering analysis of different types of categorical data. Experiments have shown that the proposed UEBDM metric outperforms its counterparts on different real and benchmark categorical data sets.

Chapter 5

Fast Hierarchical Clustering of Categorical Data

Introduction

Clustering methods can be classified into two types [67] [68] [100]: partitional and hierarchical [96] [111]. Partitional clustering separates a set of data objects into a certain number of clusters to minimise the intra-cluster distance and maximise the inter-cluster distance, while hierarchical clustering views each data object as an individual cluster and builds a nested hierarchy by gradually merging the current most-similar pair of them. Compared to partitional clustering, hierarchical cluster- ing offers more information regarding the distribution of the data set. Often, the hierarchy is visualized using dendrograms, which can be 'cut' at any level to produce the desired number of clusters. Due to the rich information it offers, hierarchical clustering has been extensively applied to different fields (e.g., data analysis, knowl- edge discovery, pattern recognition, image processing, bio-informatics, and so on [53] [82] [24] [86] [43]).
In general, a traditional hierarchical clustering framework can be summarised as follows:
Step 1. Each single data object is assigned to an individual cluster.

Step 2. The most-similar pair of clusters is found according to a certain linkage strategy and distance/similarity metric.

Step 3. The most-similar pair of clusters is merged to form a new cluster.

Step 4. Step 2 and Step 3 are repeated until only one cluster exists or a particular stop condition is satisfied.
In the above, the commonly used linkage strategies are: single-linkage(SL), average- linkage(AL), and complete-linkage(CL), which compute the maximum, average, and minimum similarity between the data objects of two clusters, respectively [111] [100]. The Traditional hierarchical clustering frameworks with SL, AL and CL linkages are abbreviated as TSL, TAL and TCL hereinafter. Although these three traditional frameworks are parameterless and simple to use, they have three major problems:
Their performance is sensitive to different data distribution types. TSL "has a tendency to produce clusters that are straggly or elongated" [67]; TCL and TAL tend to produce compact and spherical-shaped clusters, respectively.
All three only consider the local distance between pairs of data objects but ignore the global data structure during clustering [68].
Their time complexity is $O(N^2)$, which limits their applications, particularly for large-scale data and streaming data.
To tackle the above three problems, various types of hierarchical clustering ap- proaches have been proposed in the literature. To solve the first two problems, potential-based hierarchical clustering approaches based on potential theory [113] have been proposed (e.g., see [91] and [89]), where the potential field is utilized to measure the similarity between data objects. Because this type of approach merges the data objects by considering both their global distribution (i.e., potential fields of data objects) and local relationship (i.e., exact distance between neighbors), they show robustness when processing data sets with

different data distribution types and overlapped clusters. Nevertheless, their time complexity is still $O(N^2)$. To cope with the third problem, locality-sensitive hashing-based hierarchical clustering [77] has been proposed with a time complexity of $O(Nb)$ to speed up the closest-pair search procedure of TSL, where b is the bucket-size. However, the setting of param- eters for this approach is non-trivial, and its clustering accuracy is generally lower than that of TSL. Furthermore, hierarchical clustering based on Random Projec- tion (RP) [109] with time complexity of $O(N(\log N)^2)$ has also been proposed. It accelerates TSL and TAL by iteratively projecting data objects into different lines for splitting. In this manner, the data set is partitioned into small subsets, and the similarity can be measured locally to reduce computation cost. However, RP-based approaches will inherit the drawbacks of TSL and TAL as discussed before due to approximation. To simultaneously tackle the above-mentioned three problems, summarisation-based hierarchical clustering frameworks have also been proposed in the literature. Data bubble-based hierarchical clustering and its variants [22] [21]
[107] [141] [98] [138] [137] have been proposed to summarise the data objects by randomly initializing a set of seed points to incorporate nearby data objects into groups (data bubbles). The hierarchical clustering is only performed on the bubbles to avoid the similarity measurement for a large number of original data objects. However, the performance of data bubble and most of its variants is sensitive to the compression rate and the initialization of seed points. Another common short- coming of the summarisation-based approaches is that the hierarchical relationship between data objects is lost due to summarisation. In addition, none of the above- mentioned approaches are fundamentally designed for streaming data. Specifically, the entire clustering process should be executed to update the hierarchy structure for each new input, which may sharply increase the computation cost. To solve this problem, the Incremental Hierarchical Clustering (IHC) approach [121] has been proposed. It saves a large amount of computation cost by dynamically and locally restructuring the inhomogeneous regions of the present hierarchy structure. There- fore, this approach performs hierarchical clustering with a time complexity as low as $O(N \log N)$ when the hierarchy structure is completely balanced. However, the constructed hierarchy is not guaranteed to be balanced, which makes its worst-case time complexity still $O(N^2)$. Furthermore, because IHC is an approximation of TSL, it will also have bias for certain data distribution types.

In this chapter, we concentrate on: 1) addressing with the three above-mentioned problems of traditional hierarchical clustering frameworks, 2) proposing a new hier- archical clustering framework for fast categorical data hierarchical clustering, and 3) proposing an incremental hierarchical clustering framework for streaming categorical data hierarchical clustering. We first propose a Growing Multi-layer Topology Train- ing (GMTT) algorithm to dynamically partition the data and learn the relationship between the partitioned clusters in terms of their similarity levels. In the literature, topology training has been widely utilized for partitional clustering [112] [49] [48]
[133] [16] [125] [106]. However, to the best of our knowledge, it has yet to be utilized for hierarchical clustering. We make the topology grow by creating new layers with new object groups based on the existing object groups if the existing ones cannot represent the corresponding data objects well. The growth is continued until each leaf node can appropriately represent its child object group. As a result, the GMT- T algorithm assigns more layers and object groups to finely describe the crowded region of data sets (i.e., a group of many similar data objects). With the topology, the merging steps of hierarchical clustering are performed under its guidance. More- over, similarity between data objects is only measured within their groups, which can significantly reduce the computation cost. In addition, an incremental version of the GMTT framework, denoted as the IGMTT framework, is also presented to cope with streaming data. In the IGMTT framework, each new input can easily find its nearest neighbor by searching the topology from top to bottom. After that, both the topology and hierarchy are locally updated to recover the influence caused by the input. Both the GMTT and the IGMTT frameworks have competent perfor- mance in terms of clustering quality and time complexity. Their effectiveness and efficiency have been empirically investigated. The main contributions of this chapter are three-fold:

The GMTT algorithm is proposed for data partition and representation. The topology constructed accordingly can appropriately represent the data objects. The training is automatic without prior knowledge of the data set (e.g., number of clusters, proper number of object groups, etc).

A fast hierarchical clustering framework has been proposed based on GMTT. According to the topology trained through GMTT, distance measurement is locally performed to reduce computation cost. Merging is also guided by the topology to make the constructed hierarchy able to distinguish different clusters.

An incremental version of the GMTT framework (i.e., the IGMTT framework) is provided for streaming data hierarchical clustering. Similar to the GMTT framework, it is also fast and accurate.

The rest of this chapter is organised as follows. Section 5.2 introduces the com- mon notations and basic concepts in this chapter. In Section 5.3, details regarding the proposed GMTT framework and IGMTT framework are presented. In Section 5.4, time complexity analysis of GMTT and IGMTT frameworks are provided. Then, Section 5.5 presents the experimental results for various benchmark and synthetic data sets. Lastly, we summarise this chapter in Section 5.6.

Preliminaries

In this chapter, the basic settings and notations about data object, clusters, at- tribute, category, and distance are all the same as Chapter 2 - 4. Thus, we only introduce the basic notations about hierarchy and topology, which are the two most important concepts in this chapter.

Hierarchy

Given a data set **X** with N data objects. The hierarchical clustering results of **X** is a hierarchy **H**, which is a tree structure linking all the objects using branches, and representing data objects that are similar in different resolutions by nodes. In a hierarchy **H**, an element **H**(l, p, h) is a node in the l^{th} layer with the sequence number of its parent node p, and its own sequence number h. The sequence numbers of all the nodes in a hierarchy are unique, and the root node is **H**(1, 0, 1). Each non-leaf node in **H** has a certain number of child nodes. If the number of child nodes is fixed
for the non-leaf nodes, we call the number of child nodes branching factor, which is denoted by B. Each leaf node is a specific data objects, and the total number of leaf nodes in **H** equals to N .

Topology

Topology **T** of **X** has similar structure as the hierarchy **H** of **X**. The difference between **T** and **H** lies in the way of constructing them and the meaning of their nodes. **H** is constructed by hierarchical clustering algorithms, which gradually merge the most similar pair of clusters of **X**. That is, linking the most similar clusters (each cluster indicates a single object at the beginning of the merging) using branches and represent them using a node. **T** is a structure indicating the similarity levels of object groups of **X**, and does not care about the detailed similarity between data objects. Therefore, the leaf nodes of **T** are the parent nodes of object groups.

The Proposed Method

This section will propose a topology training algorithm that can gradually and automatically make a topology grow to better represent the similarity relationship of object groups. Subsequently, a framework based on it is presented to achieve fast and accurate categorical data hierarchical clustering. Furthermore, an incremental version of the framework is also presented for streaming categorical data hierarchical clustering.

GMTT: Growing Multi-layer Topology Training

The GMTT algorithm is presented, which partitions and represents the data set using a topology. At the beginning, there is only one object group (i.e., the whole data set **X**). Obviously, a topology with only one node indicating this group cannot represent the objects in the data set well, especially for complex real-world data sets. To better represent the data objects, a number of new object groups are initialized and trained, and are indicated by the child nodes of the original one in the topology.

For each of the new nodes, growing training is performed repeatedly until all of the existing data groups represent their subsets well. It is expected that more groups are assigned to the regions of the data set that are hard to represent well. There are many criteria for defining a region that is hard to represent well (e.g., inhomogeneous data objects, crowded data objects, border region of benchmark clusters, overlapped data objects, etc). From the perspective of hierarchical clustering, the merging of data objects happens in the region with crowded data objects at the beginning and gradually moves to the regions that with not that crowded data objects. Obviously, the merging of the regions with crowded data objects dominates the processing time. According to this information, we choose to better represent the crowded region via the GMTT algorithm. Consequently, the structure of the trained topology is similar to the desired hierarchy, and thus the topology can offer better guidance to accelerate the hierarchical clustering procedures.

Specifically, given a data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ with N data objects, the topology **T** is trained by randomly inputting data objects from **X** to create new nodes. Each node in **T** is expressed in the form of $v_{l,p,h}$, where l indicates the layer of the node in **T**, p is the sequence number of its parent node, and h is its own sequence number. For simplicity, $v_{l,p,h}$ can be denoted as v_h if the information of its layer and parent node is not considered in some cases. The nodes in the topology does not have physical meaning. They are just used to indicate the layers and relationship of the object groups. The corresponding object group of a node $v_{l,p,h}$ is expressed as $\mathbf{X}_h$, which contains N_h data objects belonging to **X**. During the training, we need to decide if the topology should grow or not. In other words, we should decide if a node $v_{l,p,h}$ can represent its corresponding subset $\mathbf{X}_h$ well and when to make the topology grow by creating B child nodes for $v_{l,p,h}$ in Layer $l + 1$. Here, B is a constant referred to as the branching factor, and it controls the number of child nodes created for each node. The nodes that cannot represent their subset well are defined as coarse nodes. The definitions of full-coarse node and semi-coarse node are given as follows.

Definition 1. *Let $v_{l,p,h}$ be a leaf node with an N_h-object corresponding group. Given* *the branching factor B and the upper limitation U_L, the node $v_{l,p,h}$ is a full-coarse node if and only if $N_h > U_L \cdot (B-1)$.*

Definition 2. *Let $v_{l,p,h}$ be a leaf node with an N_h-object corresponding group. Given the branching factor B and the upper limitation U_L, the node $v_{l,p,h}$ is a semi-coarse node if and only if $U_L < N_h \leq U_L \cdot (B-1)$.*

In the above two definitions, U_L controls the upper bound of the size N_h of $v_{l,p,h}$'s corresponding object group. For a full-coarse node, B new child nodes should be created by training B new sub-groups for $\mathbf{X}_h$. For a semi-coarse node, B_s new child nodes should be created in the same manner, where B_s is the branching factor of a semi-coarse node. During the training, the value of B_s will dynamically change according to the size of the semi-coarse node's corresponding object group. The value of B_s is decided by

$$B_s = \left[\frac{N_h}{U_L} \right|. \qquad (5.3.1)$$

Supposing that $v_{l,p,h}$ is a full-coarse node, B child nodes (i.e., $v_{l+1,h,t+1}$, $v_{l+1,h,t+2}$, ..., $v_{l+1,h,t+B}$) should be created, where t is the total number of nodes before the creation of B new child nodes. After the creation, the value of t is updated by $t^{(new)} = t^{(old)} + B$, and then, the corresponding object groups are formed by training the objects in $\mathbf{X}_h$. Before the training, all the objects in $\mathbf{X}_h$ are randomly assigned to one of the B groups. Then, the training is performed by repeatedly assigning each data object in $\mathbf{X}_h$ to its closest group until convergence of the training. For a data object $\mathbf{x}_{h,i}$, its closest group is found by

$$w = \underset{j}{\operatorname{argmin}}\, Dist(\mathbf{x}_{h,i}, v_{l+1,h,j}), \qquad (5.3.2)$$

with $t - B + 1 \leq j \leq t$. $Dist(\mathbf{x}_{h,i}, v_{l+1,h,j})$ measures the distance between object $\mathbf{x}_{h,i}$ and the group indicated by node $v_{l+1,h,j}$. Distance $Dist(\mathbf{x}_{h,i}, v_{l+1,h,j})$ is defined by

$$Dist(\mathbf{x}_{h,i}, v_{l+1,h,j}) = \sum_{r=1}^{d} \sum_{s=1}^{v_r} Dist(o^r_s, o^r) \cdot \mathbf{u}^r_{l+1,h,j}(s), \qquad (5.3.3)$$

where $\mathbf{u}^r_{l+1,h,j}$ records the occurrence probability of each category belonging to A_r in the object group indicated by node $v_{l+1,h,j}$, and $\mathbf{u}^r_{l+1,h,j}(s)$ is the occurrence probability of o^r_s in the object group indicated by $v_{l+1,h,j}$. We adopt this object-group

Algorithm 8 Topology Training

```
Input: Data set $\mathbf{X}_h$ and branching factor $B$ ($B_s$).
Output: $B$ ($B_s$) new child nodes and object groups.
Create $B$ ($B_s$) new nodes as the child nodes of the detect coarse-node;
Randomly assigning the data objects of the group indicated by the detected coarse-node into the $B$ ($B_s$) new groups;
while $Convergence = false$ do
    Randomly select a data object $\mathbf{x}_{h,i}$ from $\mathbf{X}_h$;
    Find its closest group among the $B$ ($B_s$) new groups according to Eq. (5.3.2);
end while
```

distance instead of the object-mode distance adopted by k-modes [63] because the statistics of the group more finely describe the intra-group objects than the selected representative mode of the group. Therefore, training the new groups using the object-group distance can make each group containing more similar objects, and thus the constructed topology can better represent the data set. The new object groups are iteratively trained through Eq. (5.3.2) and (5.3.3) until convergence. The training procedure can be summarised as Algorithm 8.

After B new child nodes are created and the corresponding B new groups are trained for $\mathbf{X}_h$, $\mathbf{X}$ is more precisely represented by the topology $\mathbf{T}$ because each new group indicated by the corresponding node contains a smaller number of more similar objects. Here, we also define the concept of fine node to judge when to stop the growth of the topology.

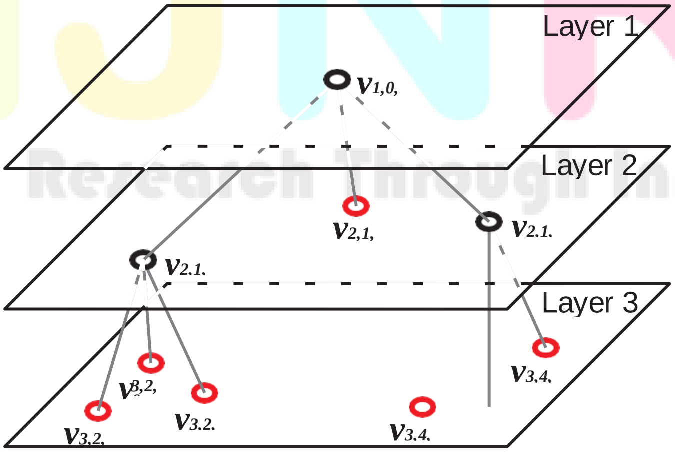
Definition 3. *Let $v_{l,p,h}$ be a child node with an N_h-object corresponding group. Given the upper limitation U_L, the node $v_{l,p,h}$ is a fine node if and only if $N_h \leq U_L$.*

When all the leaf nodes in the topology are judged as fine nodes, it means that all the groups indicated by the leaf nodes are fine enough for describing the data set, and the growth of the topology is thus stopped. The entire GMTT algorithm is summarised as Algorithm 9.

An example of the topology trained through GMTT algorithm is illustrated in Figure 5.1, where a 3-layer topology is trained for a 20-object data set with $B = 3$

Algorithm 9 GMTT Algorithm

```
Input: Data set $\mathbf{X}$, upper limitation $U_L$, and branching factor $B$.
Output: Topology $\mathbf{T}$.
Initialize a node $v_{1,0,1}$ indicating the whole $\mathbf{X}$;
while full-coarse or semi-coarse nodes exist do
    Find a coarse node $v_{l,p,h}$ in $\mathbf{T}$;
    Generate $B$ ($B_s$) new nodes through Algorithm 8;
end while
```



Node Information		
Node	Subset	Size
$v_{1,0,1}$	X_1	20

$\mathbf{v}_{2,1,2}$	X_2	9
$\mathbf{v}_{2,1,3}$	X_3	4
$\mathbf{v}_{2,1,4}$	X_4	7
$\mathbf{v}_{3,2,5}$	X_5	3
$\mathbf{v}_{3,2,6}$	X_6	2
$\mathbf{v}_{3,2,7}$	X_7	4
$\mathbf{v}_{3,4,8}$	X_8	4
$\mathbf{v}_{3,4,9}$	X_9	3

Figure 5.1: Topology trained for a 20-objects data set.

and U_L = 4. In the topology shown in this figure, Layer 1 contains only one top node $v_{1,0,1}$ with the corresponding object group $\mathbf{X}_1$, where $\mathbf{X}_1 = \mathbf{X}$. Because $N_1 > U_L$, B child nodes are created and B new groups are trained in the next layer using data objects from $\mathbf{X}_1$. One of the branches stops its growth with fine node $v_{2,1,3}$ in Layer 2 because $N_3 \leq U_L$. Finally, the topology stops its growth in Layer 3 because all of the leaf nodes are judged as fine nodes in Layer 3, which means that the entire data set can be represented well by the present topology. It can be seen from the figure that the union of all the groups indicated by the leaf nodes is the entire data set (i.e., $\mathbf{X} = \mathbf{X}_3 \cup \mathbf{X}_5 \cup \mathbf{X}_6 \cup \mathbf{X}_7 \cup \mathbf{X}_8 \cup \mathbf{X}_9$).

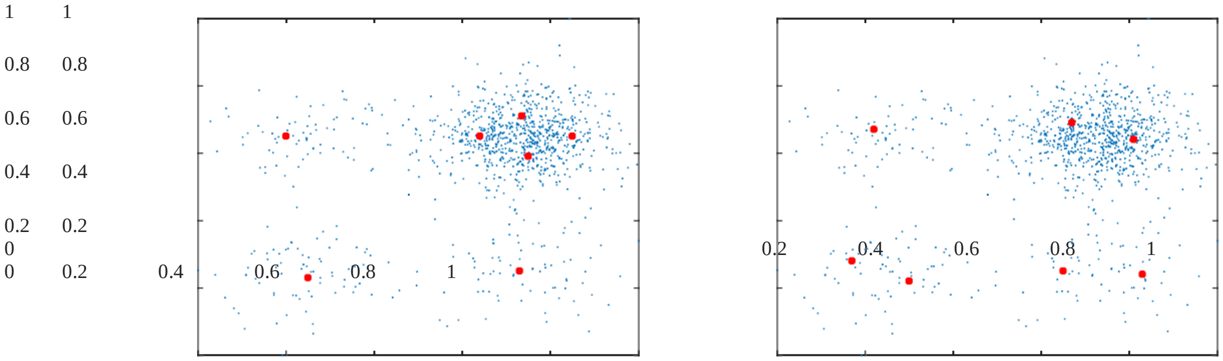
Here, we also discuss the two reasons about why we design the GMTT algorithm to gradually train a multi-layer topology instead of directly initializing enough nodes in one layer and training their indicated object groups:

For a topology trained through GMTT, the number of corresponding data objects of each leaf node will be smaller than U_L due to the growing multi- layer topology training. This guarantees that crowded regions of the data set are represented by more nodes, and sparse regions are represented by less nodes. If we directly initialize a sufficiently large number of object groups, groups will be trapped locally and cannot represent the data set well. In Figure 5.2, we use a numerical 2-d synthetic data set to illustrate the difference between the nodes trained through GMTT, and the nodes trained in just one- layer. Although we focus on categorical data clustering, this example can still intuitively demonstrate the difference between the two topology training strategies. Obviously, the nodes of the topology trained through GMTT can better fit the distribution of the data objects. In the one-layer case, it is possible that a very sparse region with a small number of objects is forced to be partitioned and represented by several nodes, because too many nodes are randomly initialized for indicating the object groups in a sparse region, and the groups are locally trapped due to the single time initialization. Therefore, GMTT is more proper for data set representation.

The structure of the topology trained through GMTT is consistent with the expected hierarchy of hierarchical clustering that the nodes in deeper layers indicate a more crowded region of data objects, and vice versa. Moreover, links in the topology indicate the affiliation between the nodes and their child nodes, which is similar to the links in the expected hierarchy. These properties make the topology suitable for guiding and accelerating the hierarchical clustering. In contrast, if all the object groups are directly initialized and trained in just one layer, their corresponding nodes in the topology cannot offer the desired affiliation information, and thus cannot be utilized for the accelerating of hierarchical clustering.

Although the trained topology has similar structure as the desired hierarchy, they still have two significant differences:

The leaf nodes in the topology indicate groups of objects, while the leaf nodes



Results of GMTT training Results of one-layer training

Figure 5.2: Comparison of GMTT and one-layer training. of a hierarchy are specific data objects.
The links in the topology connecting two nodes only indicate the affiliation
between their indicated object groups, while the links of a hierarchy indicate the nested similarity relationship among the object clusters.

Therefore, how to efficiently and effectively obtain the desired hierarchy through further processing the topology will be discussed in Section 5.3.2.

Fast Hierarchical Clustering Based on GMTT

From the perspective of hierarchical clustering, the constructed hierarchy should satisfy two properties: homogeneity and monotonicity [96]. Suppose we cut a den- drogram horizontally to produce a certain number of clusters, homogeneity is the property that the similarity between intra-cluster objects is higher than that of the inter-cluster objects. Monotonicity is the property that the clusters produced by cutting the hierarchy in a layer close to the bottom are more homogeneous than the clusters produced by cutting the hierarchy in a layer close to the top. In the topolo- gy obtained through GMTT, because the object group of each node is a local part of the object group indicated by their parent node, the topology roughly satisfies the property of homogeneity. The monotonicity is also satisfied among the nodes that are lineal consanguinity of each other, where the concept of lineal consanguinity is defined in Definition 4.

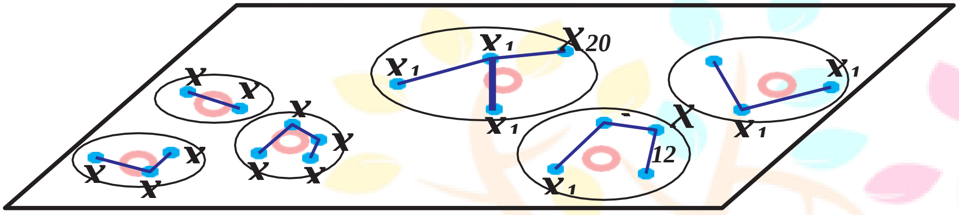


Figure 5.3: Data objects in the subsets are linked to form sub-MSTs.

Definition 4. *Let v_h be a node in* **T**. *If another node v_m in* **T** *can be found by searching* **T** *in a certain direction (bottom-up or top-down) from v_h, then v_h and v_m are said to be lineal consanguinity of each other.* For instance, $v_{3,2,5}$ and $v_{1,0,1}$ shown in Figure 5.1 are lineal consanguinity of each other, but $v_{3,2,5}$ and $v_{2,1,3}$ are not. Even node $v_{2,1,3}$ is in layer 2, its homogeneity is not guaranteed to be lower than that of node $v_{3,2,5}$ in layer 3 because the object group indicated by $v_{3,2,5}$ is not a local part of the object group indicated by $v_{2,1,3}$.

To merge all of the data objects according to the topology, data objects belonging to the groups indicated by leaf nodes, and the groups themselves, should be merged according to a certain linkage strategy. The merging procedures should also comply with the lineal consanguinity relationship between topology nodes to exploit the homogeneity and monotonicity of the topology. We introduce how to efficiently merge the data objects according to the topology to form a complete hierarchy in the following.

For an object group $\mathbf{X}_h$ indicated by a leaf node v_h, a certain linkage strategy (e.g., SL, AL, and CL) and a certain distance/similarity metric (e.g., HDM, ADM, ABDM, CBDM, and UEBDM) should be utilized to link these objects to form a sub-Minimal Spanning Tree (MST) for $\mathbf{X}_h$. In Figure 5.3, we take the same data set and topology shown in Figure 5.1 as an example to show the formed sub-MSTs. After the sub-MSTs are formed, they should be linked to form a complete MST. Therefore, nodes in the same layer sharing the same parent node are also linked to form sub-MSTs according to a certain linkage strategy and metric. It is commonly recognized that hierarchical clustering result can be expressed in the form of an MST instead of a hierarchy because they contain the same information and can be converted to each other easily [55] ~~[67] [96]. Therefore, the hierarchical clustering task of our GMTT hierarchical clustering~~ framework is to form an MST for **X**. When sub-MSTs are formed for: 1) all of the object groups of leaf

Algorithm 10 GMTT Hierarchical Clustering Framework

```
Input: Data set $\mathbf{X}$, upper limitation $U_L$ and branching factor $B$.
Output: Minimal Spanning Tree MST.
Train a topology $\mathbf{T}$ through Algorithm 9;
Form sub-MSTs for the objects in the group indicated by each leaf node;
Form sub-MSTs for the child nodes of each parent node and obtain the complete MST.
```

nodes, and 2) all of the nodes sharing the same parent node, a complete MST linking all of the data objects has been formed. The entire GMTT hierarchical clustering framework is summarised as Algorithm 10. Here, we also introduce how to transform the complete MST into a hierarchy in three stages:

Stage 1. Only the data objects belonging to the groups indicated by the leaf n- odes are considered for merging. Specifically, for a leaf node v_h, all the pairs of data objects in $\mathbf{X}_h$ that are linked by the corresponding sub-MST are stacked together according to the ascending order of the lengths of their linking edges. The stacked pairs form a Local Merging Queue (LMQ) $\mathbf{q}_h$, and a corresponding Linking Distance Queue (LDQ) $\mathbf{d}_h$. After the LMQs $\{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_{u_l}\}$ and the corresponding LDQs $\{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_{u_l}\}$ are formed for all the u_l leaf nodes, a candidate set $\mathbf{C} = \{\mathbf{q}_1(1), \mathbf{q}_2(1), ..., \mathbf{q}_{u_l}(1)\}$ containing the pairs with shortest linking edges in each LMQ, and the corresponding dis- tance set $\mathbf{D} = \{\mathbf{d}_1(1), \mathbf{d}_2(1), ..., \mathbf{d}_{u_l}(1)\}$ is formed. Then, the most-similar pair $\mathbf{q}_g(1)$ that should be merged firstly is found through

$$g = \operatorname*{argmin}_{1\leq i\leq u_l} \mathbf{D}(i). \qquad (5.3.4)$$

After the merging, $\mathbf{q}_g(1)$ is removed from both set $\mathbf{C}$ and $\mathbf{q}_g$, and $\mathbf{d}_g(1)$ is

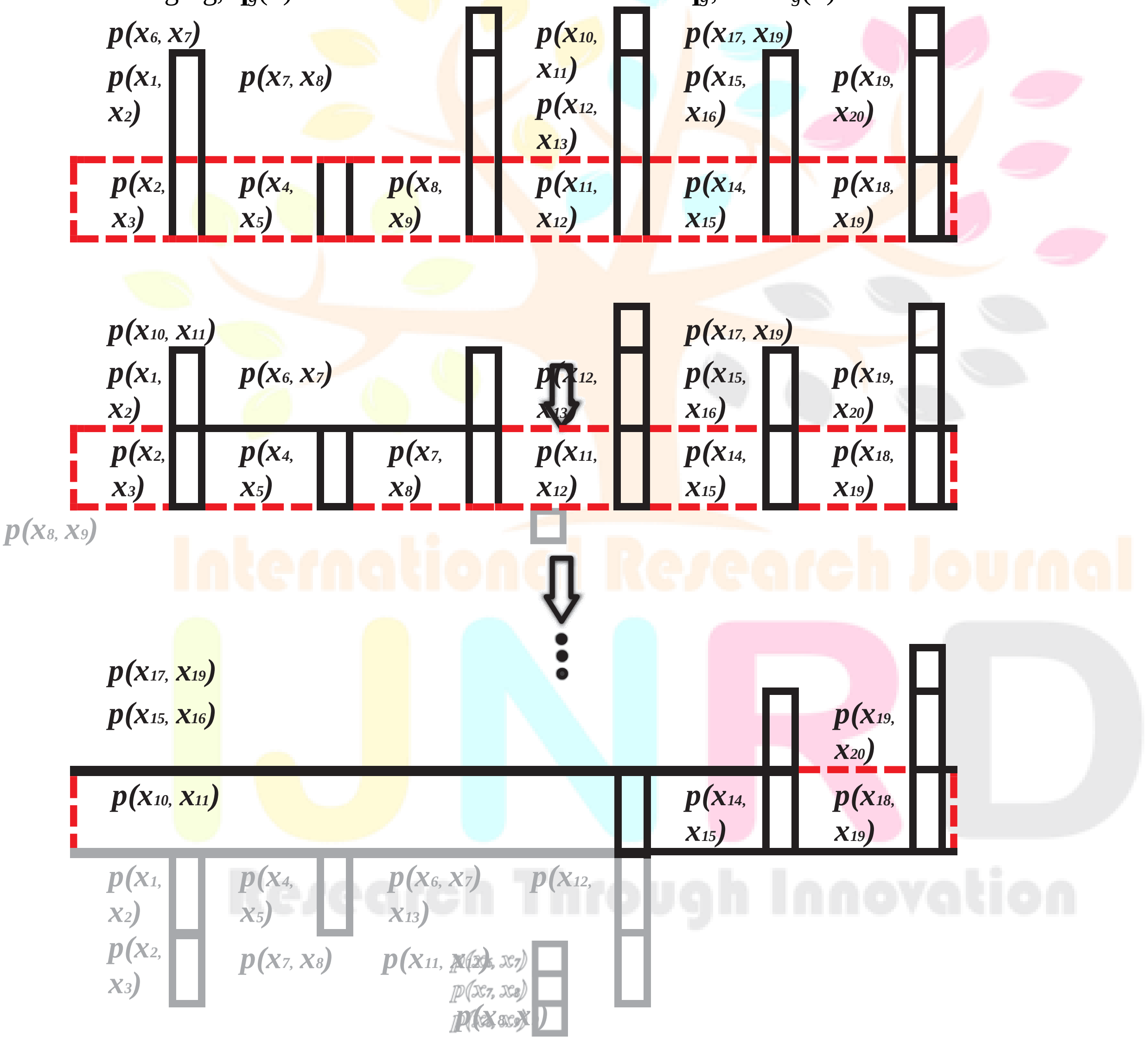


Figure 5.4: Merging procedure demonstration of Stage 1.

removed from both set $\mathbf{D}$ and $\mathbf{d}_g$. Subsequently, the current most-similar pair in $\mathbf{q}_g$ is popped up into $\mathbf{C}$, and the corresponding linking distance is popped up into $\mathbf{D}$. The above operations are iteratively performed until an All-Leaf- Parent (ALP) node in $\mathbf{T}$ becomes an All-Candidate-Parent (ACP) node. For a non-leaf node, if all its child nodes are leaf nodes, it is an ALP. When all the data objects belonging to the groups of ALP's child nodes are merged together within their groups, the ALP becomes an ACP. Figure 5.4 illustrates the merging procedure of Stage 1 using the 20-object data set demonstrated in Figure 5.3 as an example. It can be seen that six LMQs {$\mathbf{q}_1$, $\mathbf{q}_2$, ..., $\mathbf{q}_6$} are formed according to the corresponding sub-MSTs in Figure 5.3. At the beginning, $\mathbf{q}_3(1) = p(\mathbf{x}_8, \mathbf{x}_9)$ is the most similar pair among the candidates in **C** (the candidate set is shown by red the dashed frame in Figure 5.4). Therefore, $\mathbf{x}_8$ and $\mathbf{x}_9$ are merged firstly. Figure 5.5 shows the corresponding hierarchy of the example in Figure 5.3. After merging the data objects as shown in Figure 5.5, an ALP $v_{2,1,2}$ becomes an ACP. Then, both the object groups indicated by the child nodes of ACP, and data objects belong to the groups indicated by all the existing leaf nodes should be considered for merging in Stage 2.

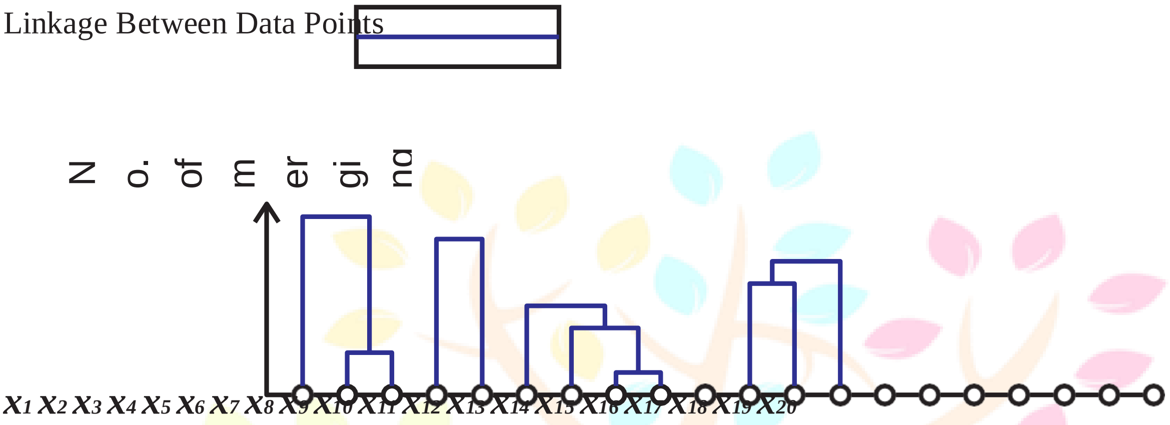

Linkage Between Data Points
N o. of m er gi ng
x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15 x16 x17 x18 x19 x20

Figure 5.5: Hierarchy demonstration for Stage 1.

Stage 2. Because the topology only guarantees the monotonicity of the object groups that are lineal consanguinity of each other, lengths of the linking edges between the groups indicated by ACP's leaf nodes are not guaranteed to be larger than the edges linking unmerged data objects belonging to the groups of all the existing leaf nodes. Therefore, pairs of object groups indicated by ACPs' leaf nodes are viewed as merging candidates and should be considered together with the data object candidates for merging in Stage 2. Suppose that v_h is the only ACP at the beginning of Stage 2, pairs of the groups indicated by its leaf nodes should also be stacked into the candidate set **C** for merging. Because data objects belonging to the groups indicated by ALP's child nodes continue to be merged, more ALPs will become ACPs in Stage 2. Because the groups indicated by ACPs' child nodes also continue to be merged in Stage 2, when all the groups indicated by the child nodes of an ACP are merged together, this ACP becomes a leaf node. In Stage 2, the merging of leaf nodes and data objects is performed repeatedly until all of the ALPs becomes ACPs. Figure 5.6 demonstrates the topology at the end of Stage 2. The corresponding hierarchy is presented in Figure 5.7.

Stage 3. After Stage 2, the candidate set **C** only contains pairs of object groups indicated by nodes. These groups are finally merged according to the lengths of the linking edges of the corresponding sub-MSTs until all of the object groups in the candidate set are merged. The final hierarchy of the 20-object example formed after Stage 3 is shown in Figure 5.8.

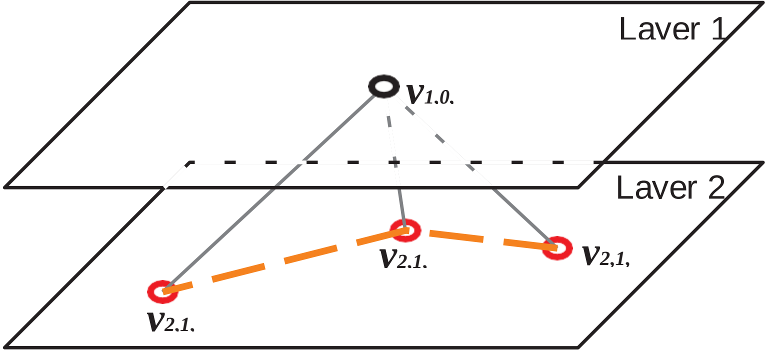


Figure 5.6: Topology at the end of Stage 2.

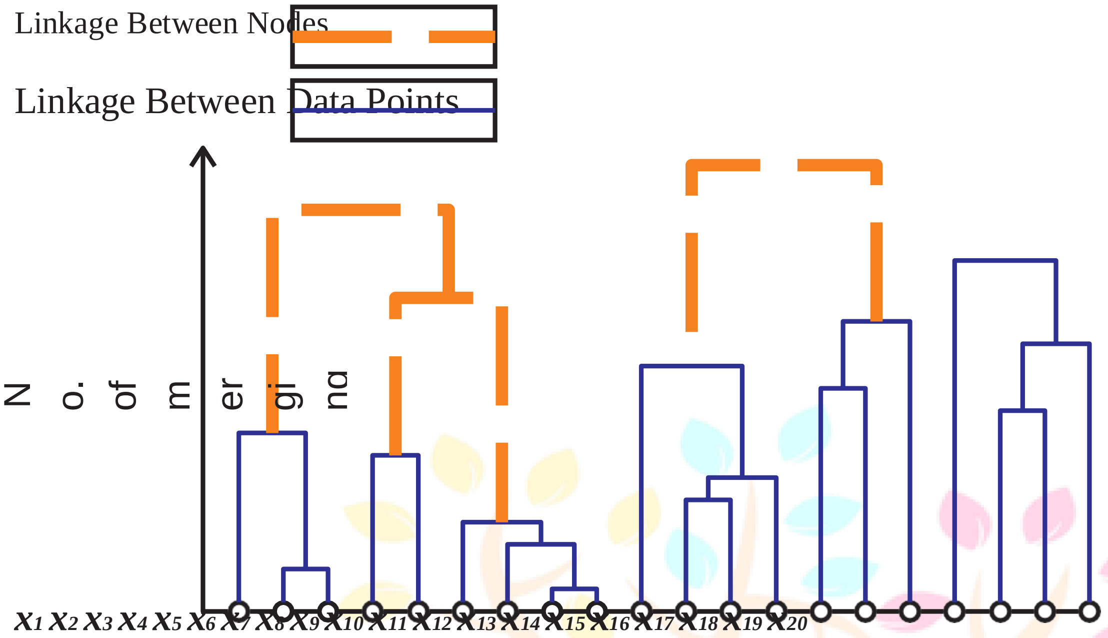

Linkage Between Nodes
Linkage Between Data Points
N o. of m er gi na
X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20

Figure 5.7: Hierarchy at the end of Stage 2.

The transformation algorithm is summarised as Algorithm 11.

In the GMTT framework, traditional linkages (i.e., SL, AL, and CL) can be adopted for hierarchical clustering. Here, we offer the detailed discussions about how to combine them with the GMTT framework: 1) SL can be directly used to form sub-MSTs for hierarchical clustering according to Algorithm 11, 2) AL merges data objects according to the average distance between the members of clusters and cannot produce sub-MSTs as shown in Figure 5.3. Thus, AL should be adopted to directly produce the candidate set **C** without forming LMQs. Whenever a pair of objects (data objects or object groups) is selected from **C** for merging, AL will

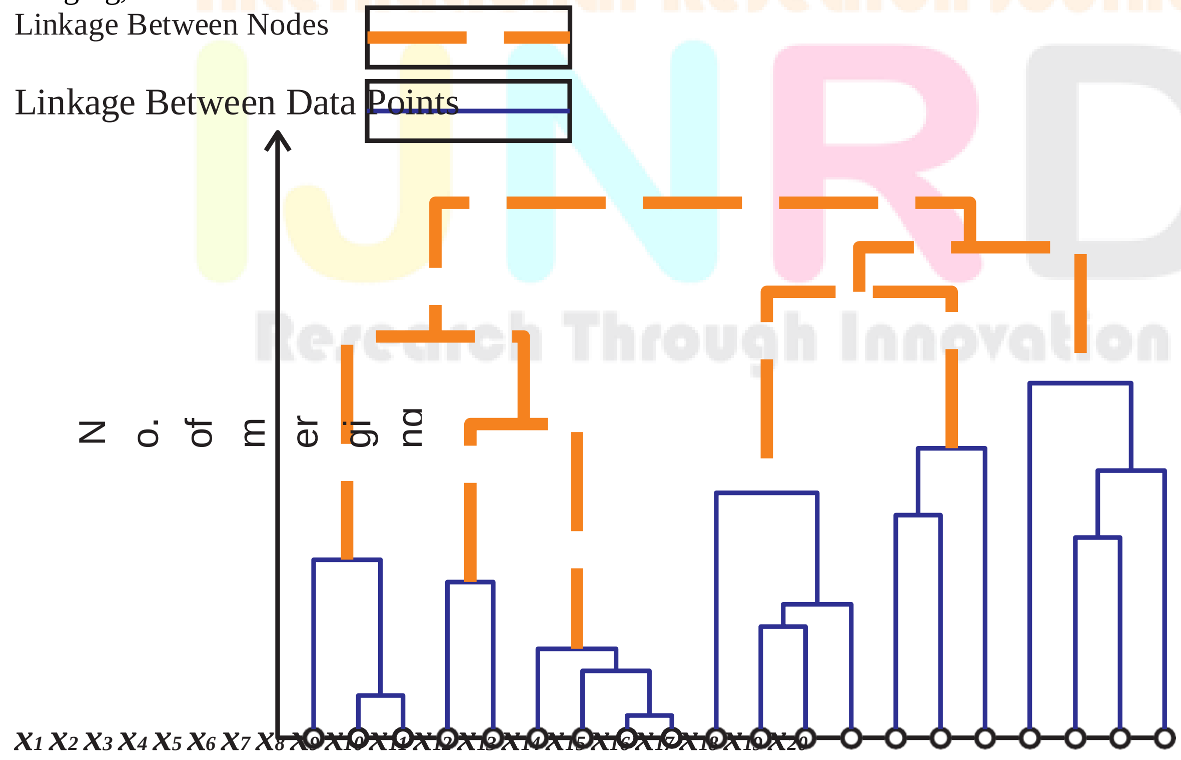

Linkage Between Nodes
Linkage Between Data Points
N o. of m er gi na
X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20

Figure 5.8: Hierarchy at the end of Stage 3.

produce a new candidate among the objects with the same parent node as the merged ones, and 3) CL can be used in the same manner as SL. When using these three linkages to merge two object groups, two modes are firstly selected from the two groups in the same way as k-modes clustering algorithm [63], and then the selected modes are treated as data objects for merging. Since SL is the simplest linkage with lower computation cost, we choose to adopt SL for the MST construction of the proposed GMTT framework.

Incremental Hierarchical Clustering Based on GMTT

Streaming data processing is a significant challenge for hierarchical clustering ap- proaches [99] [117] [114]. To make the GMTT framework feasible for the processing of streaming data, we present its incremental version. We firstly train a coarse topology through GMTT using the former part of inputs. After that, for each new input, the coarse topology is dynamically updated through the incremental version of GMTT, which is abbreviated as IGMTT. Specifically, for a streaming data set

Algorithm 11 MST-Hierarchy Transformation

```
Input: MST obtained through Algorithm 10, topology T trained through Algorithm 9.
Output: Hierarchy H.
Generate LMQs and LDQs for the object group indicated by each leaf node;
Generate merging candidates C;
while C is not empty do
    if new ACP occurs then
        Generate LMQ and LDQ for the ACP;
        Update C according to the LMQ and LDQ of the new ACP;
    end if
    Merge the closest pair in C;
    Remove the merged pair from C, and remove the corresponding linking distance from D;
    if the LMQ of the merged pair is not empty then
        Move the present closest pair from the LMQ to C, and move the corresponding linking distance from LDQ to D;
    end if
end while
```

X with N data objects, the coarse topology is trained through the GMTT algorithm using the former R streaming inputs of **X** with the upper limitation U_L and branching factor B. After that, for each new input $\mathbf{x}_i$, its most similar object group indicated by v_h is found by searching **T** from top to bottom according to the lineal consanguinity relationship. Object group $\mathbf{X}_h$ indicated by v_h incorporates the new input, and the size N_h of $\mathbf{X}_h$ is updated by $N_h^{(new)} = N_h^{(old)} + 1$. If v_h is judged as a coarse node, the updating is triggered to update **T** by creating B or B_s new child nodes for v_h, and training B or B_s new object groups. To make the IGMTT algorithm more efficient, we choose a reasonable and efficient updating trigger mechanism. That is, the updating is only triggered when a full-coarse node is detected. Otherwise, the algorithm will directly process the next input.

Specifically, if a node v_h becomes a full-coarse node after adopting a new input $\mathbf{x}_i$, we should update the mode of the object group indicated by v_h. If the mode is changed after updating, all the sub-MSTs and modes of the groups indicated by the nodes that are lineal consanguinity of v_h should be updated. Then, B new child nodes are created for v_h, and $\mathbf{X}_h$ is partitioned by training B new object groups. After that, sub-MSTs of each of the new groups indicated by the new child nodes, and sub-MST linking the new child nodes are produced in the same way of GMTT.

Since most of the existing categorical data distance metrics define distances based on the statistics of the whole data set, the defined distances should also be dynamically updated for streaming data. Therefore, the same updating trigger mechanism is adopted for the updating of the distance matrices of each attribute. Specifically, when a full-coarse node is detected, statistics of the whole data set is updated by counting the new inputs. Then, for each attribute, the distance matrix recording the distances between each pair of the categories are updated. This distance matrix updating mechanism is suitable for ADM, ABDM, CBDM, and UEBDM metrics. In addition, since HDM does not rely on the statistics for distance measurement, it can be directly used without maintaining distance matrices for the attributes.

The result of the IGMTT framework can also be transformed into a hierarchy according to Algorithm 11. To better explain the details of the IGMTT framework, we summarise it in Algorithm 12.

Time Complexity Analysis

Since most of the distance metrics that are applicable to GMTT frameworks (i.e., ADM, ABDM, CBDM, and the proposed UEBDM) have the same time complex- ity for distance measurement and dynamic distance matrices updating, we choose UEBDM as the distance metric for the time complexity analysis of GMTT and IGMTT frameworks. Time complexity of GMTT-UEBDM, IGMTT-UEBDM, and the MST-Hierarchy transformation algorithm are analysed in this section.

Algorithm 12 IGMTT Hierarchical Clustering Framework

```
Input: Streaming data set $\mathbf{X}$.
Output: MST.
Train a coarse topology $\mathbf{T}$ using the former $R$ inputs;
for $i = R + 1$ to $N$ do
    Search to find the closest object group (indicated by $v_h$) for $x_i$;
    $\mathbf{X}_h = \mathbf{X}_h \cup \mathbf{x}_i$ and $N^{(new)} = N^{(old)} + 1$;
    if $v_h$ is a full-coarse node then
        Update the mode of the object group indicated by $v_h$;
        if the mode is changed after updating then
            All the sub-MSTs and modes of the groups indicated by the nodes that are lineal consanguinity of $v_h$ should be updated;
        end if
        Generate $B$ new nodes and train $B$ new object groups for $\mathbf{X}_h$ through Algorithm 8;
        Form a sub-MST to link the new child nodes;
        Form sub-MSTs to link the objects inside the groups indicated by the new child nodes;
        if the adopted metric is not HDM then
            Update the statistics of the whole data set;
            Update the distance matrices of each attribute;
        end if
    end if
end for
```

Time Complexity Analysis for GMTT-UEBDM

We prove that the time complexity of the GMTT-UEBDM algorithm can be opti- mized to $O(N^{1.5})$, which is lower than the $O(N^2)$ of traditional approaches.

Theorem 1. GMTT-UEBDM hierarchical clustering algorithm has time complexity $O(N^{1.5})$ if the upper limitation U_L is set at $\sqrt{N}$.

Proof. When the topology $\mathbf{T}$ trained through GMTT is a total-imbalanced tree, we will have the worst-case time complexity. In this case, the number of non-leaf nodes is $u_{nl} = \frac{n-U_L}{(B-1)U_L}$. From the top to the bottom of **T**, the numbers of data objects for training the non-leaf nodes can be viewed as an arithmetic sequence $\{N, N-(B-U_L, N-2(B-1)U_L, ..., N-(u_{nl}-1)(B-1)U_L\}$. Therefore, total number of data objects for training all the non-leaf nodes is s_n $= N \cdot u_{nl}$ $\frac{}{nl}$ $\frac{}{} \frac{(}{B-1)U_L(u \quad + u_{nl})}$. object that will be trained, B nodes should be considered to find the winner node using Eq.(5.3.2), and the corresponding time complexity if $O(Bdv_{max})$ according to Eq. (5.3.3), where v_{max} has been defined in Chapter 4 for the time complexity of UEBDM. For each of the non-leaf node, its training will be repeated I times for convergence. Therefore, the time complexity for the topology training (Algorithm 10, line 3) is $O(s_nBdv_{max}I)$.

According to the time complexity analysis of UEBDM in Chapter 4, time com- plexity for producing the distance matrices that record the distances between intra-attribute categories of d attributes is $O(Nd + v^2$ $d)$. $_{max}$

For each of the non-leaf nodes, a sub-MST should be constructed for its B child nodes. For u_{nl} non-leaf nodes in total, the time complexity is $O(u_{nl}B^2)$. For each of the leaf nodes, a sub-MST should be constructed for its corresponding U_L data objects. For u_l leaf nodes in total, the time complexity is $O(u_lU^2)$. Therefore, the time complexity for constructing the complete MST (Algorithm 10, line 4 - 5) is $O(u_{nl}B^2 + u_lU^2)$.

The overall time complexity of the proposed GMTT-UEBDM is $O(s_nBdv_{max}I +$ $Nd + v^2_{max}$ $d + u_{nl}B^2 + u_lU^2_L)$. I is a very small constant ranging from 2 to 10$_L$ according to the experiment. B is always set to a small positive integer (i.e., 2 - 4) in the experiments. v_{max} satisfying v^2_{max} $< N$, which is a small constant for real categorical data. d is also a small constant for real categorical data. Therefore, when U_L is set at $\sqrt{N}$, the overall time complexity of GMTT-UEBDM can be optimized to $O(N^{1.5})$. Q

Time Complexity Analysis for IGMTT-UEBDM

We prove that the time complexity of IGMTT-UEBDM algorithm is still $O(N^{1.5})$.

Theorem 2. IGMTT-UEBDM hierarchical clustering algorithm has time complex- ity $O(N^{1.5})$ if the upper limitation U is set at $\sqrt{N}$. —

Proof. According to the proof of Theorem $_L$ 1, the time complexity for obtaining the coarse topology is $O(R^{1.5})$.

For N inputs, the time complexity for searching the closest object group (Algo- rithm 12, line 5) according to u_{nl} non-leaf nodes is $O(Bu_{nl}N)$.

For every $U_L(B-1)$ new inputs, a full-coarse node will be formed. Thus, line 8

- 18 of Algorithm 12 will be performed once if the mode of the group that adopts

the newest input is changed after the topology **T** adopts $U_L(B-1)$ new inputs. We assume that the mode is changed after the topology **T** adopting each $U_L(B-1)$ new inputs for time complexity analysis in the following. Therefore, the following operations will be triggered $\frac{N}{\quad}$ times in total during the hierarchical clustering.

$U_L(B{-}1)$

For each trigger, at most $\frac{N}{}$ sub-MSTs and modes should be updated with time

$_L$ U

complexity $O(B^2)$ and $O(U_L(B-1)d)$, respectively. Therefore, the time complexity

for the $\frac{N}{\quad}$ triggers is $O(\frac{N}{2}B + \frac{N}{2}d$).

$\overline{U_L(B{-}1)}$ U_L $_2$ U_L

For each trigger, B new groups should be trained for $U_L(B-1)$ data objects and the training will be repeated I times for convergence. Therefore, the time complexity

for the $\frac{N}{\quad}$ triggers is $O(BNdv_{max}I)$.

$U_L(B{-}1)$

For each trigger, B sub-MST linking at most U_L data objects should be formed with time complexity is $O(BU^2)$; A sub-MST should also be formed for the B new

nodes, which has time complexity $O(B^2)$. For $\frac{N}{\quad}$ triggers in total, the overall

$\overline{U_L(B{-}1)}$

time complexity for the sub-MSTs construction is $O(U_LN + \frac{BN}{}$).

$_L$ $_U$

For each trigger, we update the distance matrices with time complexity $O(Nd +$

v^2 $d)$ according to the time complexity analysis of UEBDM. Therefore, the overall

max

time complexity for the $\frac{N}{\quad}$ triggers is $O(\frac{N}{\quad} \cdot (Nd + v^2$ $d))$.

$U_L(B{-}1)$ $U_L(B{-}1)$ *max*

According to the above analysis, the overall time complexity of the IGMTT-UEBDM algorithm is $O(Bu$ $N + \frac{N}{2}B_{nl} + \frac{N}{2}d + BNdv$ $I + U$ $N + \frac{BN}{} + \frac{N}{\quad}$ ·

$\overline{U_L^2}$ $\overline{U_L}$ $_{max}$ $_L$ $_{U_L}$ $_{U_L(B-1)}$

$(Nd{+}v^2$ $_{max}$ $d))$. Similar to the time complexity analysis of GMTT-UEBDM, the time

complexity can be optimized to $O(N^{1.5})$ with $U_L = \sqrt{N}$. Q —

Time Complexity Analysis for MST-Hierarchy Transformation

Theorem 3. MST-Hierarchy transformation algorithm has time complexity $O(N^{1.5})$ if the upper limitation U is set at $\sqrt{N}$.

Proof. For a leaf node, the time complexity for forming LMQ for the corresponding
U_L objects is $O(U^2)$. For u_l leaf nodes, the time complexity is $O(U^2 u_l)$. For each
L L
merging, the distance between the first pairs in u_l LMQs should be compared to find the smallest one. For $N - 1$ merges, the time complexity is $O(u_l N)$. Therefore, the overall time complexity of the transformation algorithm is $O(U^2 u_l + u_l N)$. When we set $U_L = \sqrt{N}$, the time complexity can be optimized to $O(N^{1.5})$. Q

Discussions

We further discuss and analyse the potential limitations of the proposed GMTT and IGMTT frameworks in terms of dimensionality of data sets and coarse topology training.

Dimensionality: The GMTT algorithm extracts the data distribution structure by gradually creating necessary nodes. New nodes gradually split the data set to detect and represent the data distribution. Due to the curse of dimensionality, the distribution of data objects will be sparser for high-dimensional data. As a result, nodes trained through GMTT will be less representative for high- dimensional data, and the structural distribution information offered by the topology may have less contribution to improve the hierarchy quality.

Coarse Topology: The IGMTT algorithm trains a coarse topology using the former part of streaming data. Because it extracts the structural information of data set and allows fine training for the coarse topology according to the following inputs, the size of the former part of streaming inputs for coarse topology training will not significantly influence the clustering quality if the distribution of streaming data does not change with time. The case in which the data distribution changes over time is another challenging problem for streaming data hierarchical clustering, which is not considered in this thesis.

The above-mentioned discussions have been further justified by the experimental results in Section 5.5.

Experiments

Experiments were conducted in two parts: 1) evaluation of the proposed GMTT- UEBDM, and 2) evaluation of the incremental version of GMTT-UEBDM (IGMTT- UEBDM). All the experiments are performed on different types of benchmark and synthetic categorical data sets.

Experimental Settings

Data Sets

13 categorical data sets including 12 real and benchmark data sets and one synthetic data set are collected for the experiments. The 12 real and benchmark data sets include four ordinal, four nominal, and four mixed categorical data sets that are the same as the data sets used in the experiments of Chapter 4. The

synthetic data set has three attributes and 100,000 objects. Each object value is randomly selected from {1, 2, ..., 5}. Since the synthetic data set is just generated for execution time evaluation, objects in this data set have no label, and no physical meaning.

Counterparts

Since all the existing hierarchical clustering approaches are proposed for numerical data only, there is actually no counterpart in the field of categorical data hierarchi- cal clustering. Therefore, we choose the existing hierarchical clustering approaches that adopt a certain distance metric (e.g., Euclidean distance metric), and replace their adopted metrics with the existing categorical data metrics to form several counterparts for the evaluation of the proposed approaches.

Table 5.1: 15 counterparts.

	TSL	TAL	TCL
HDM	TSL-HDM	TAL-HDM	TCL-HDM
ADM	TSL-ADM	TAL-ADM	TCL-ADM
ABDM	TSL-ABDM	TAL-ABDM	TCL-ABDM
CBDM	TSL-CBDM	TAL-CBDM	TCL-CBDM
UEBDM	TSL-UEBDM	TAL-UEBDM	TCL-UEBDM

The selected hierarchical clustering frameworks are: the Traditional hierarchical clustering framework combined with Single Linkage (TSL), Average Linkage (TAL), and Complete Linkage(TCL).

The selected distance metrics are: the commonly used Hamming Distance Metric (HDM) [58], the state-of-the-art Ahmad's Distance Metric (ADM) [10], Association- Based Distance Metric (ABDM) [80], Context-Based Distance Metric (CBDM) [65], and the Unified Entropy-Based Distance Metric (UEBDM) proposed in Chaptere 4.

By combining the selected frameworks and metrics, 15 counterparts are formed, and we list them in Table 5.1. Potential-based, RP-based, and incremental hier- archical clustering approaches are not chosen because they are original proposed for numerical data, and cannot be modified for categorical data hierarchical clus- tering by simply replacing their adopted metrics. Jia's Distance Metric [72] is not chosen because its computation is relative laborious, and it cannot be adequately accelerated by maintaining inter-category distance matrices for each attribute.

Parameter Settings

According to the time complexity analysis in Section 5.4, we set $U_L = \sqrt{N}$ for different data se~~ts~~. According to the experiments, we set $B = 4$. In the following experiments, reasonableness of setting $B = 4$ is studied.

Table 5.2: Clustering performance of GMTT-based approaches on four ordinal data sets.

Approach	Internship	Photo	Employee	Lecturer
GMTT-HDM	0.5633±0.06	0.5091±0.07	0.1865±0.01	0.3419±0.02
GMTT-ADM	0.5389±0.04	0.4773±0.02	0.2055±0.01	0.3363±0.03
GMTT-ABDM	0.5456±0.06	0.5061±0.02	0.2042±0.01	0.3388±0.01
GMTT-CBDM	0.5067±0.01	0.5303±0.07	0.1948±0.01	0.3225±0.03
GMTT-UEBDM	0.6733±0.04	0.5788±0.05	0.2135±0.02	0.3546±0.04

Validity Index

Quality of the hierarchy produced by hierarchical clustering approaches is usually measured by Fowlkes Mallows Index (FMI) [46] that has been reviewed in Chapter 2.

Performance Evaluation of GMTT-UEBDM

We separately evaluate the proposed UEBDM metric and GMTT framework to investigate their effectiveness in categorical data hierarchical clustering. Because there are randomization procedures in the GMTT framework, all the approaches formed by combining GMTT and a metric are run 10 times, and we record their averaged FMI values as the final results.

Effectiveness Evaluation of UEBDM Metric

We combine the four counterpart categorical data distance metrics that are listed in Table 5.1 into GMTT to form four hierarchical clustering approaches. We compare the performance of them and the proposed GMTT-UEBDM on the 12 categorical data sets in Table 5.2 - 5.4 to illustrate the superiority of UEBDM in categorical data hierarchical clustering. In the following of this section, mixed categorical data is called mixed data interchangeably for simplicity.

It can be observed that GMTT-UEBDM obviously outperforms the other coun- terparts on most of the data sets, which illustrates the effectiveness of UEBDM in

Table 5.3: Clustering performance of GMTT-based approaches on four mixed data sets.

Approach	Assistant	Fruit	Hayes	Nursery
GMTT-HDM	0.5097±0.06	0.4880±0.04	0.3644±0.01	0.3910±0.05
GMTT-ADM	0.5056±0.04	0.5218±0.04	0.3598±0.01	-
GMTT-ABDM	0.5139±0.03	0.5160±0.04	0.3795±0.03	-
GMTT-CBDM	0.5333±0.04	0.5397±0.04	0.3758±0.03	-
GMTT-UEBDM	0.5528±0.04	0.5080±0.03	0.4477±0.04	0.4410±0.06

Table 5.4: Clustering performance of GMTT-based approaches on four nominal data sets.

Approach	Pillow	Solar	Voting	Tictac
GMTT-HDM	0.3080±0.02	0.4321±0.03	0.8506±0.03	0.5768±0.03
GMTT-ADM	0.3040±0.02	0.4452±0.03	0.8667±0.02	0.5842±0.03
GMTT-ABDM	0.3050±0.02	0.4390±0.04	0.8545±0.02	0.5742±0.03
GMTT-CBDM	0.3082±0.01	0.3981±0.03	0.8310±0.08	0.5678±0.04
GMTT-UEBDM	0.3136±0.02	0.4700±0.04	0.8545±0.03	0.5848±0.03

categorical data hierarchical clustering. Results of GMTT-ADM, GMTT-ABDM, and GMTT-CBDM are empty for Nursery data sets because ADM, ABDM, and CBDM metrics are unable to measure distances for a data sets with extremely low inter-attribute dependency like Nursery data set.

Effectiveness Evaluation of GMTT Framework

We compare the performance of different hierarchical clustering frameworks adopt- ing the same distance metric to illustrate the effectiveness of the proposed GMTT framework. The comparative results are shown in Table 5.5 - 5.7.

According to the experimental results, we found that the GMTT-UEBDM has very competitive performance, and the performance of all the other GMTT-based approaches that adopt the existing metrics are more robust (i.e., always not the worst

Table 5.5: Clustering performance of GMTT-based approaches and 15 counterparts on four ordinal data sets.

Approach	Internship	Photo	Employee	Lecturer
TSL-HDM	0.6667	0.4242	0.1970	0.4240
TAL-HDM	0.5111	0.5303	0.1910	0.3140
TCL-HDM	0.6778	0.5152	0.1760	0.3880
GMTT-HDM	0.5633	0.5091	0.1865	0.3419
TSL-ADM	0.6889	0.4545	0.1960	0.4110
TAL-ADM	0.5111	0.4697	0.2240	0.3330
TCL-ADM	0.6889	0.4697	0.2180	0.3320
GMTT-ADM	0.5389	0.4773	0.2055	0.3363
TSL-ABDM	0.5000	0.4545	0.2130	0.3910
TAL-ABDM	0.6889	0.4848	0.2070	0.3300
TCL-ABDM	0.6889	0.5303	0.2110	0.3160
GMTT-ABDM	0.5456	0.5061	0.2042	0.3388
TSL-CBDM	0.5222	0.5000	0.2080	0.3980
TAL-CBDM	0.5000	0.5152	0.2190	0.3110
TCL-CBDM	0.5222	0.5909	0.2030	0.3440
GMTT-CBDM	0.5067	0.5303	0.1948	0.3225
TSL-UEBDM	0.6889	0.4091	0.2150	0.4030
TAL-UEBDM	0.7444	0.6515	0.2160	0.3330
TCL-UEBDM	0.5778	0.7121	0.1960	0.3250
GMTT-UEBDM	0.6733	0.5788	0.2135	0.3546

in comparison with the corresponding TSL-, TAL-, and TCL-based approaches). The three more detailed observations are discussed as follows:

By comparing the results in Table 5.5 - 5.7, it can be found that GMTT frame- work can better boost the performance of traditional hierarchical clustering frameworks on mixed and nominal data sets. For the four ordinal data set- s, performance of GMTT framework cannot obviously boost the performance

Table 5.6: Clustering performance of GMTT-based approaches and 15 counterparts on four mixed data sets.

Approach	Assistant	Fruit	Hayes	Nursery
TSL-HDM	0.4028	0.3800	0.4015	0.3334
TAL-HDM	0.4306	0.4900	0.3636	0.3436
TCL-HDM	0.4583	0.3500	0.4773	0.3353
GMTT-HDM	0.5097	0.4880	0.3644	0.3910
TSL-ADM	0.4583	0.3700	0.3864	-
TAL-ADM	0.5000	0.5500	0.3561	-
TCL-ADM	0.5000	0.5300	0.4242	-
GMTT-ADM	0.5056	0.5218	0.3598	-
TSL-ABDM	0.4583	0.4400	0.3864	-
TAL-ABDM	0.5000	0.5200	0.3864	-
TCL-ABDM	0.5000	0.4600	0.3636	-
GMTT-ABDM	0.5139	0.5160	0.3795	-
TSL-CBDM	0.5000	0.3700	0.4091	-
TAL-CBDM	0.5000	0.4900	0.3409	-
TCL-CBDM	0.6250	0.5100	0.4621	-
GMTT-CBDM	0.5333	0.5397	0.3758	-
TSL-UEBDM	0.4167	0.3300	0.4015	0.3402
TAL-UEBDM	0.6111	0.5100	0.4167	0.5307
TCL-UEBDM	0.5278	0.5100	0.3636	0.4651
GMTT-UEBDM	0.5528	0.5080	0.4477	0.4410

because the GMTT framework use object-cluster distance for the topology construction, and thus it is more suitable to be combined with an ordinal da- ta distance metric for ordinal data clustering. In other words, by combining with HDM, ADM, ABDM, and CBDM, the positive impact on the clustering accuracy offered by GMTT is hampered. It is because that, even existing categorical data distance metrics are actually designed for nominal data, they

Table 5.7: Clustering performance of GMTT-based approaches and 15 counterparts on four nominal data sets.

Approach	Pillow	Solar	Voting	Tictac
TSL-HDM	0.3500	0.2786	0.6161	0.6524
TAL-HDM	0.2800	0.2972	0.6161	0.5052
TCL-HDM	0.3200	0.4396	0.6161	0.5752
GMTT-HDM	0.3080	0.4321	0.8506	0.5768
TSL-ADM	0.3000	0.2724	0.6161	0.6545
TAL-ADM	0.3200	0.3096	0.6161	0.6263
TCL-ADM	0.3300	0.3158	0.8345	0.5741
GMTT-ADM	0.3040	0.4452	0.8667	0.5842
TSL-ABDM	0.3200	0.2724	0.6161	0.6545
TAL-ABDM	0.3200	0.2601	0.8506	0.6409
TCL-ABDM	0.3100	0.3251	0.8782	0.5699
GMTT-ABDM	0.3050	0.4390	0.8545	0.5742
TSL-CBDM	0.2900	0.3127	0.6161	0.6566
TAL-CBDM	0.2800	0.3158	0.6161	0.5772
TCL-CBDM	0.3000	0.3220	0.8529	0.6263
GMTT-CBDM	0.3082	0.3981	0.8310	0.5678
TSL-UEBDM	0.3400	0.4892	0.6115	0.6524
TAL-UEBDM	0.2800	0.5015	0.8920	0.5678
TCL-UEBDM	0.3100	0.4830	0.8253	0.5595
GMTT-UEBDM	0.3136	0.4700	0.8545	0.5848

are still somewhat workable for the distinguishing of similarity levels between different pairs of objects. But when combined with GMTT, these existing nominal data metrics are utilized to more finely measure the distances between each attribute value of an object and the occurrence probability distributions of the attribute values in different clusters, which makes the unreasonableness of the nominal data metrics in ordinal data distance measurement has more significant impact on the clustering results. Therefore, GMTT framework is not suitable to be combined with improper categorical data metrics, especially for ordinal data.

GMTT-UEBDM is competitive on all the 12 data sets. It is because that the adopted UEBDM can more reasonably measure the distances for different types of categorical data. Based on UEBDM, the object-cluster distance mea- surement procedure of GMTT can more accurately assign data objects to more similar clusters, and thus GMTT-UEBDM can achieve competitive clustering performance. Although the performance of GMTT-UEBDM is not always the best, it is very robust to different data sets and it outperforms most approach- es formed by combining traditional hierarchical clustering frameworks and the existing metrics on most data sets.

The approaches formed by combining different hierarchical clustering frame- works with the proposed UEBDM metric (i.e., TSL-UEBDM, TAL-UEBDM, TCL-UEBDM, and GMTT-UEBDM) have obvious better performance than the approaches formed by combining different hierarchical clustering frame- works with existing nominal data metrics. This indicates that the adopted metric usually dominates the clustering performance of a hierarchical cluster- ing approach. Therefore, from the perspective of the quality of the produced hierarchy, it is obvious that the contribution of the proposed GMTT frame- work is to make the hierarchical clustering results more robust to different types of categorical data sets, and the contribution of the proposed UEBDM metric is to achieve more accurate hierarchical clustering results.

In short, the proposed GMTT framework can produce more robust hierarchical clustering results in comparison with the existing frameworks. Moreover, the pro- posed GMTT-UEBDM can produce more robust and accurate hierarchical clustering results than the counterparts.

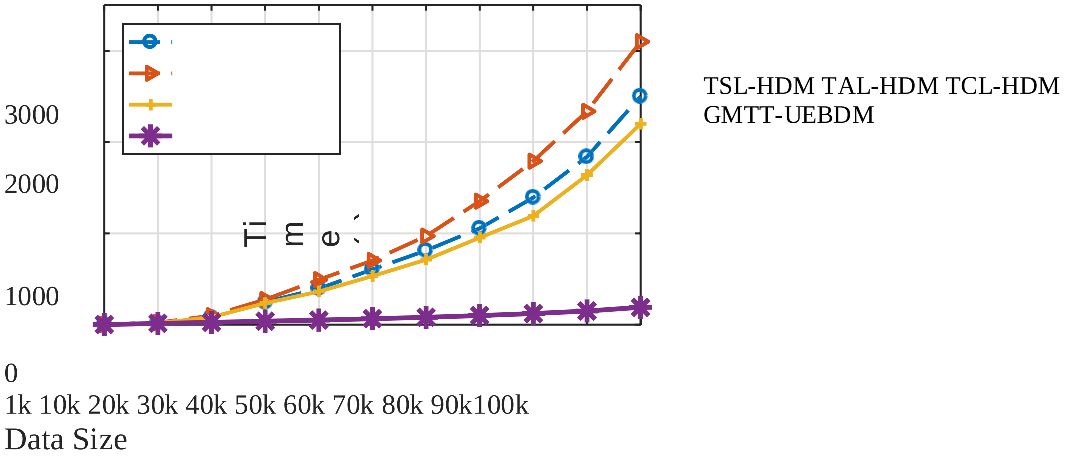


Figure 5.9: Execution time of GMTT-UEBDM, TSL-HDM, TAL-HDM, and TCL- HDM on synthetic data sets with different sizes.

Efficiency Evaluation of GMTT-UEBDM

To verify the efficiency of GMTT-UEBDM, the execution time of it is compared with several representative counterparts in Figure 5.9. For each compared approach, the execution time of it for the hierarchical clustering of a synthetic data set with differ- ent sampling rates are recorded as a “Time - Data Size” curve. The synthetic data set with 100,000 objects is sampled using the sampling rates {0.01, 0.1, 0.2, ..., 1} to produce 11 synthetic data sets with size 1k, 10k, 20k, ..., 100k.

From Figure 5.9, it can be observed that the execution time of TSL-HDM, TAL- HDM, and TCL-HDM increases dramatically with the increasing of the data size. Compared with them, the execution time of the proposed GMTT-UEBDM increases obviously slower, which is consistent with the time complexity analysis in Section 5.4. In short, GMTT-UEBDM is the most efficient among the compared categorical data hierarchical clustering approaches.

Performance Evaluation of IGMTT-UEBDM

Effectiveness Evaluation of IGMTT-UEBDM

Since there is no existing hierarchical clustering approaches that is applicable to streaming categorical data, we compare the proposed IGMTT-UEBDM with GMTT- UEBDM to validate its effectiveness. This experiment is also run 10 times on the 12 categorical data sets, and the averaged FMI values are recorded in Table 5.8 -

Table 5.8: Clustering performance of GMTT-UEBDM and IGMTT-UEBDM on four ordinal data sets.

Approach	Internship	Photo	Employee	Lecturer
GMTT-UEBDM	0.6733±0.08	0.5788±0.07	0.2135±0.02	0.3546±0.04
IGMTT-UEBDM	0.6427±0.06	0.5282±0.07	0.2177±0.05	0.3495±0.05

Table 5.9: Clustering performance of GMTT-UEBDM and IGMTT-UEBDM on four mixed data sets.

Approach	Assistant	Fruit	Hayes	Nursery
GMTT-UEBDM	0.5528±0.10	0.5080±0.06	0.4477±0.04	0.4410±0.09
IGMTT-UEBDM	0.5532±0.11	0.5281±0.08	0.4424±0.05	0.4404±0.07

Table 5.10: Clustering performance of GMTT-UEBDM and IGMTT-UEBDM on four nominal data sets.

Approach	Pillow	Solar	Voting	Tictac
GMTT-UEBDM	0.3136±0.02	0.4700±0.04	0.8545±0.03	0.5848±0.03
IGMTT-UEBDM	0.3028±0.02	0.4621±0.06	0.8497±0.06	0.5910±0.06

5.10.
It can be observed that the clustering performance of IGMTT-UEBDM is very close to that of GMTT-UEBDM on most of the data sets in general. Two detailed observations are discussed as follows:

IGMTT-UEBDM even outperforms GMTT-UEBDM on several data sets (i.e., Employee, Assistant, Fruit, and Tictac). If the former part of inputs do not contain noise and outliers, the coarse topology trained by them will be very close to or even better than the fine topology trained through GMTT. This is the reason why sometimes the performance of IGMTT-UEBDM is even better than that of GMTT-UEBDM.

The performance of IGMTT-UEBDM is obvious worse than that of GMTT-

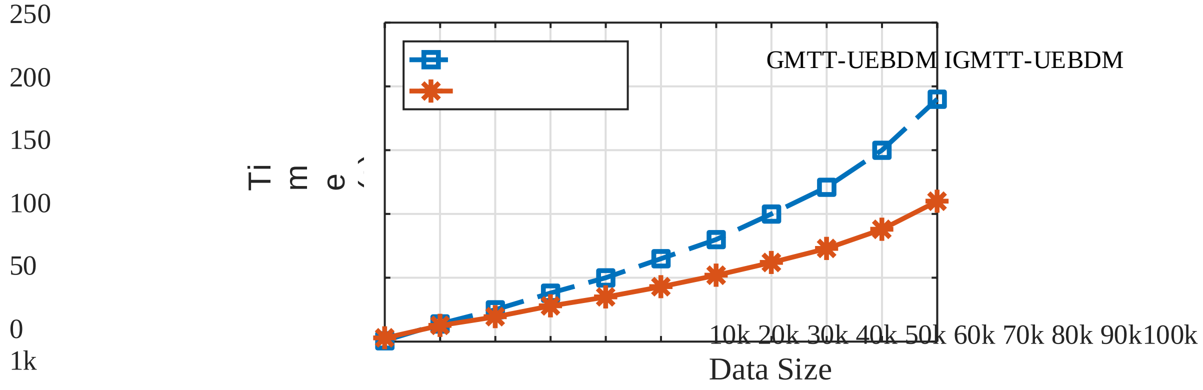


Figure 5.10: Execution time of GMTT-UEBDM and IGMTT-UEBDM on synthetic data sets with different sizes.

UEBDM on Internship and Photo data sets. The data size of Internship and Photo are very small in comparison with the other data sets. Therefore, the former part of inputs of them will be unstable in terms of the capability of training a representative coarse topology. Therefore, the performance of IGMTT-UEBDM on extreme small data sets will be unstable. However, since IGMTT-UEBDM is proposed for large-scale streaming categorical data, its unstable performance on small data sets will not hamper its effectiveness.

In conclusion, clustering accuracy of IGMTT-UEBDM is very competitive in the hierarchical clustering of streaming categorical data.

Efficiency Evaluation of IGMTT-UEBDM

Similar to the efficiency evaluation of GMTT-UEBDM, we run IGMTT-UEBDM on the same synthetic data set with different sampling rates. Since there is no existing counterparts, we still compare it with GMTT-UEBDM. The execution time of them are shown in Figure 5.10.

It can be observed that the increasing rate of IGMTT-UEBDM's and GMTT- UEBDM's execution time are similar. Moreover, IGMTT-UEBDM costs less exe- cution time than GMTT-UEBDM in general, because IGMTT-UEBDM adopts a lazy topology training scheme (i.e., only perform topology training when full-coarse nodes are detected). In addition, the execution time of IGMTT-UEBDM is higher than GMTT-UEBDM when the data size is less than 10k. This is caused by the coarse topology training procedure of IGMTT-UEBDM. When data size is not large enough, the efficiency of adopting coarse topology training is not obvious, and it may even increase the overall computation cost. However, since IGMTT-UEBDM is designed for large-scale streaming data, this phenomenon does not hamper the efficiency of IGMTT-UEBDM. In general, IGMTT-UEBDM is efficient for large-scale streaming categorical data hierarchical clustering.

Study of the Branching Factor

The two parameters (i.e., the branching factor *B* and the upper limitation U_L) influence the hierarchical clustering performance in different ways. Since our goal is to achieve a lower time complexity (i.e., $O(N^{1.5})$), the upper limitation U_L must be fixed at $\sqrt{N}$ for different data sets as analysed in Section 5.4. Therefore, we discuss how the branching factor *B* influences the performance of the proposed approaches as follows:

A too large *B* may cause a flat topology, which cannot offer rich structural information for forming the final hierarchy. Therefore, a too large value of *B* may influence the accuracy of hierarchical clustering.
A too small *B* may make the topology too deep (i.e., with too many layers). This will cause high computation cost for the topology training. Addition- ally, a too small *B* will also split data objects into large groups, which may incorrectly split benchmark clusters, and thus lead to poor clustering accura- cy. Therefore, a too small *B* may influence both the computation cost and accuracy of hierarchical clustering. To experimentally investigate the impact of *B*, we run GMTT-UEBDM 10 times with different *B* on each of the 12 data sets, and record the averaged results in Figure 5.11 - 5.13. In this experiment, the clustering performance when $B = 1$ is very poor because $B = 1$ makes the topology trained through GMTT never grow to better represent data set. Moreover, for each data set, we only evaluate the GMTT-

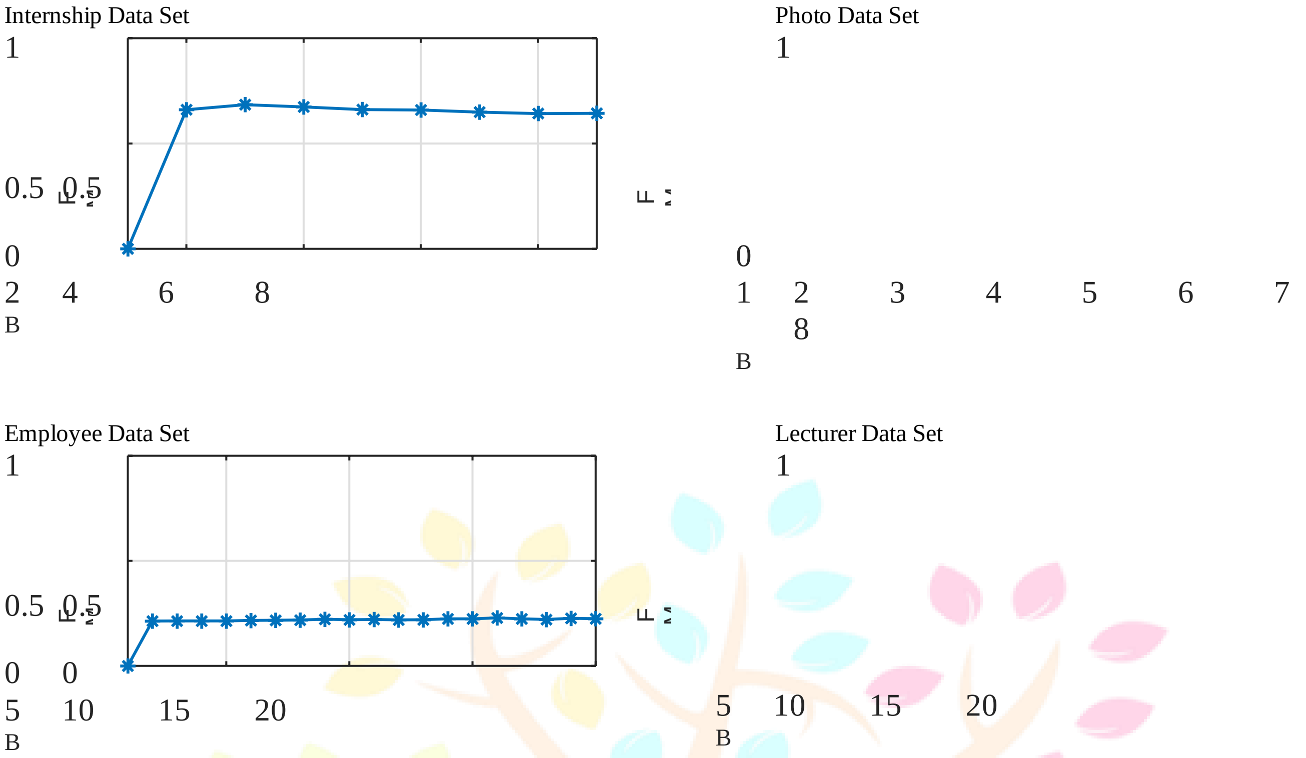


Figure 5.11: B - FM curves of GMTT-UEBDM on four ordinal data sets.

UEBDM performance when $B \le \sqrt{N}$ because $B > \sqrt{N}$ makes the clusters have less than $\sqrt{N}$ objects, which violates the setting $U = \sqrt{N_L}$. Since a too large B will lead to poor clustering accuracy as discussed before, for the data sets with $\sqrt{N} > 20$, we only evaluate the performance of GMTT-UEBDM with $B \in \{1, 2, ..., 20\}$ on them.

It can be observed that the clustering accuracy of GMTT-UEBDM is very robust to different B, excepting some extreme values (e.g., 1, 2, and $\sqrt{N}$), which is consistent with our discussions. Therefore, setting B = 4 is reasonable for GMTT-based approaches.

Summary

This chapter has presented a topology training framework for categorical data hierarchical clustering, called GMTT, which trains a multi-layer topology to describe the data set, and uses the topology to guide and accelerate the merging process of hierarchical clustering. Based on the GMTT, a hierarchical clustering approach called GMTT-UEBDM has been designed by combining GMTT with the proposed

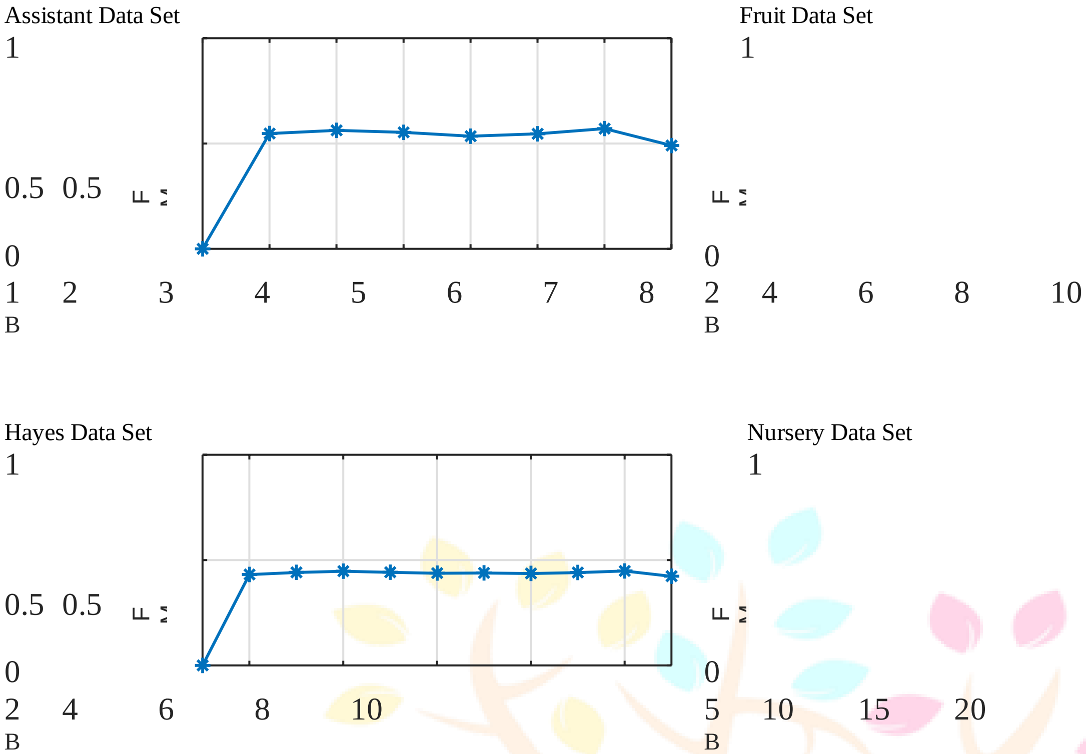


Figure 5.12: B - FM curves of GMTT-UEBDM on four mixed data sets.

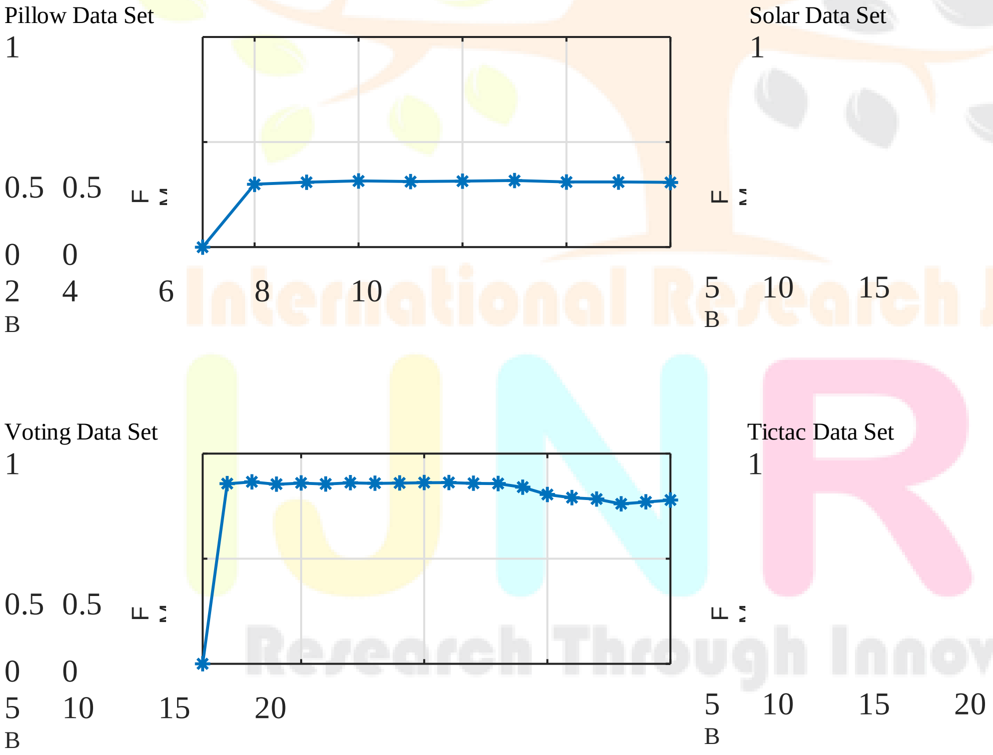


Figure 5.13: B - FM curves of GMTT-UEBDM on four nominal data sets.

UEBDM metric. This approach features lower time complexity and higher clustering accuracy compared to the counterparts. We have analysed that the GMTT-UEBDM improves the time complexity of existing applicable categorical data hierarchical clustering approaches to $O(N^{1.5})$. Although a parameter B should be set in advance, we have illustrated that the performance of GMTT-UEBDM is very robust to different values of this parameter. Furthermore, an incremental version of GMTT- UEBDM (i.e., IGMTT-UEBDM) has also been proposed to make the proposed ap- proach also applicable to large-scale streaming categorical data. IGMTT-UEBDM has the same time complexity as the GMTT-UEBDM, but can dynamically update the topology and successively incorporate new inputs to update the corresponding hierarchy. Experiments on different real, benchmark, and synthetic data sets have demonstrated that GMTT-based approaches improve the time complexity without sacrificing the quality of the constructed hierarchy.

Chapter 6

Conclusions and Future Work

Conclusions

This thesis has addressed four significant issues in categorical data clustering: 1) dis- tance measurement of ordinal attributes, 2) the unification of the distance definition of ordinal and nominal attributes, 3) the design of an applicable fast hierarchical clustering framework for categorical data, and 4) the design of an incremental hier- archical clustering framework for streaming categorical data.

First, an Entropy-Based Distance Metric (EBDM) was presented for ordinal data clustering. The EBDM uses cumulative entropy as a measure with which to quantify the distances between ordinal categories by considering both their order relationship and their statistics. Because the proposed EBDM appropriately uses the order information for distance measurement, it outperforms the existing categorical data distance metrics that were proposed under the hypothesis that categorical data comprise only nominal attributes in the clustering analysis of ordinal data. We also studied the relationship between the degree to which order information is exploited and the ordinal data clustering performance; we found that the clustering performance of an ordinal data clustering algorithm is dominated by the adopted metric and that the effectiveness of a metric is dominated by the degree to which it exploits the order information. The proposed EBDM is parameter-free and easy to use, and the experimental results demonstrated that its clustering performance obviously outperforms the existing applicable categorical data distance metrics.

The EBDM was then extended to a unified distance metric (i.e., the unified EBDM or UEBDM) to cope with the more complex mixed categorical data clustering problem. From the perspective of information theory, the UEBDM treats ordinal attributes and nominal attributes differently but unifies the concepts of the distance between categories and the importance of attributes, which avoids information loss during the distance measurement of mixed categorical data. For ordinal attributes, the order relationships between categories and their statistics are considered for the distance measurement, while for the nominal attributes, the statistics of the categories are used for distance measurement. Because the distance concepts of ordinal and nominal attributes are unified, it is not necessary to separately compute the distances for ordinal and nominal attributes and then weight and combine them to produce the final distances between data objects. Moreover, the UEBDM is still easy to use and non-parametric, and it can be easily applied for clustering analysis of any type of categorical data, including ordinal data, nominal data, and mixed categorical data. The experimental results showed that the UEBDM outperforms the existing categorical data metrics in the clustering of various types of categorical data.

Because hierarchical clustering is a useful but laborious type of clustering anal- ysis and the existing efficient hierarchical clustering approaches are designed only for numerical data, we developed a fast hierarchical clustering framework that is ap- plicable to both categorical and numerical data. According to our design, the time complexity of most hierarchical clustering approaches (i.e., $O(N^2)$) is reduced to $O(N^{1.5})$, where N is the number of data objects in the target data set. The proposed framework automatically trains a topology to describe the distribution structure of a data set, and the most computationally expensive hierarchical clustering process, that is, searching the most similar pair of data objects, is then accelerated under the guidance of the trained topology. More specifically, the global most similar pair searching problem is converted to a local problem by the topology, which results in significant savings in the computation cost. A fast and accurate categorical data hierarchical clustering approach can be obtained by combining this framework with the proposed UEBDM metric. According to the experiments, this approach features robust and competitive clustering performance and a low computation cost, and its only parameter is very easy to set.

Finally, we further extended the fast hierarchical clustering framework to an incremental version to tackle the problem of large-scale streaming categorical data hierarchical clustering, which is a significant problem that has yet to be solved. The proposed incremental framework dynamically incorporates new inputs to train the topology and updates the corresponding hierarchy when the topology's structure is changed by the new inputs. We also provided an incremental version of the proposed UEBDM metric that can be combined with the incremental framework for categorical data hierarchical clustering. Experiments on various real, benchmark, and synthetic data sets have shown the effectiveness and efficiency of the proposed incremental categorical data hierarchical clustering approach.

Future Work

Although many challenging problems in the field of categorical data clustering have been addressed in this thesis, many others remain for future studies; we discuss the future works along four directions as follows:

Unified inter-attribute dependence measurement. The relationships among at- tributes have not yet been considered. The degree of dependence between attributes may offer valuable information for distance measurement. How- ever, three types of inter-attribute relationships exist for mixed categorical data: those between ordinal attributes, those between nominal attributes, and those between ordinal and nominal attributes. This makes inter-attribute de- pendence measurement a very challenging problem. More specifically, if the definitions of the three types of relationships are not unified, information loss will occur during clustering analysis. If an appropriate inter-attribute depen- dence measure is adopted, the contributions of various types of attributes can be reasonably weighted, and the clustering performance can be improved ac- cordingly. Therefore, one main future direction of categorical data clustering may be to define a unified inter-attribute dependence measure.

Automatic attribute type recognition. This thesis assumes that the types of attributes were specified in advance. However, a common phenomenon exists in which the data collectors incorrectly mark the attribute types for the at- tributes of the collected data, such as marking the ordinal attributes as nominal ones, marking the nominal attributes as ordinal ones, or marking the ordinal attributes as numerical ones. Manual correction of the incorrectly marked at- tribute types is a laborious task for categorical data sets with a large number of attributes. Therefore, automatically distinguishing the types of categorical attributes would be a potential orientation.

Categorical data distance metric learning. The proposed UEBDM metric de- fines the distances for categorical data before the learning procedure for a clustering algorithm. From a practical view-point, however, the distances be- tween categories are inherently task-dependent and data-dependent. Hence, determining the distances from the data set and adapting them to fit the learn- ing task is a possible mean to achieve better clustering performance. However, because the distances among categorical data are not as well-defined as those of numerical data in general, learning the distances between each pair of categories for each attribute is a non-trivial task. Therefore, a meaningful future study could involve the design of an efficient distance metric learning approach for categorical data clustering analysis.

Categorical data summarisation. Most fast hierarchical clustering approaches proposed for numerical data adopt data summarisation techniques to represent the whole data set using a small number of representative data objects or seed points. In this way, hierarchical clustering can be efficiently performed on these representative objects or seed points to save considerable computation cost. Although the GMTT framework presented in this thesis also summarises categorical data objects using a trained topology, it requires the statistics of each object group to be maintained during the clustering procedures, which may cause high space complexity, especially for high-dimensional categorical data with large numbers of possible values of the attributes. To this end, the design of an efficient categorical data summarisation scheme for categorical data clustering that does not sacrifice clustering accuracy is also a meaningful direction for a future study.

BIBLIOGRAPHY

[CCT99] W.-T. Chan, F. Y. L. Chin, and H.-F. Ting. A faster algorithm for finding disjoint paths in grids. In *Proceedings of the 10th Annual International Symposium on Algorithms and Computation (ISAAC)*, pages 393–402, 1999. 67

[CCT03] W.-T. Chan, F. Y. L. Chin, and H.-F. Ting. Escaping a grid by edge-disjoint paths. *Algorithmica*, 36(4):343–359, 2003. Preliminary version in SODA 2000. 67

[CEN09] E. W. Chambers, , J. Erickson, and A. Nayyeri. Homology flows, cohomology cuts. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 273–282, 2009. 3, 35

[CKM+11] P. Christiano, J. A. Kelner, A. M˛ adry, D. A. Spielman, and S. Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 273–282, 2011. 64

[CR10] S. Cabello and G. Rote. Obnoxious centers in graphs. *SIAM Journal on Discrete Mathematics*, 24(4):1713–1730, 2010. 3

[CRZ96] I. Cox, S. Rao, and Y. Zhong. Ratio regions: A technique for image segmentation. *International Conference on Pattern Recognition*, 02:557, 1996. 35

[CX00] D.Z. Chen and J. Xu. Shortest path queries in planar graphs. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 469–478, 2000. 3, 53,

54
[Dan58] G. B. Dantzig. On the shortest route through a network. RAND Report P-1345, The Rand Corporation, Santa Monica, CA, April 1958. published in Management Science 6
(1960) 18–190. 14
[Dij59] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*,1:269–271, 1959. 10.1007/BF01386390. 14
[Dji96] Hristo Djidjev. Efficient algorithms for shortest path problems on planar digraphs. In *22nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*,
pages 151–165, 1996. 3, 53, 54, 61
[DPZ00] H. Djidjev, G. E. Pantziou, and C. D. Zaroliagis. Improved algorithms for dynamic shortest paths. *Algorithmica*, 28(4):367–389, 2000. 54
[DSST89] J. R. Driscoll, N. Sarnak, D. D. Sleator, and R. Tarjan. Making data structures persistent. *Journal of Computer and System Sciences*, 38:86– 124, 1989. 17
[Edm60] J. Edmonds. A combinatorial representation for polyhedral surfaces. *Notices of the American Mathematical Society*, 7:646, 1960. 7
99
[EFS56] P. Elias, A. Feinstein, and C. Shannon. A note on the maximum flow through a network. *IEEE Transactions on Information Theory*, 2(4):117–119, 1956. 27
[EG08] David Eppstein and Michael T. Goodrich. Studying (non-planar) road networks through an algorithmic lens. In *16th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems (ACM-GIS)*, page 16, 2008. 52
[EIT+90] D. Eppstein, G. Italiano, R. Tamassia, R. Tarjan, J. Westbrook, and M. Yung. Maintenance of a minimum spanning forest in a dynamic planar graph. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1 – 11, 1990.
12
[Epp97] D. Eppstein. Dynamic connectivity in digital images. *Information Processing Letters*, 62(3):121–126, May 1997. 2
[Epp99] D. Eppstein. Subgraph isomorphism in planar graphs and related problems. *Journal of Graph Algorithms and Applications*, 3(3):1–27, 1999. 54
[Eri10] J. Erickson. Maximum flows and parametric shortest paths in planar graphs. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages
794–804, 2010. 64
[Eri11] J. Erickson. personal communication, 2011. 35
[Eul41] L. Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, 8:128–140, 1741. 2
[FF56] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956. 27, 66
[FF57] L. R. Ford and D. R. Fulkerson. Construction of maximal dynamic flows in networks. RAND Report P-1079 (RM-1981), The Rand Corporation, Santa Monica, CA, May 1957.
published in Operations Research 6 (1958) 419–433. 16
[FMS91] E. Feuerstein and A. Marchetti-Spaccamela. Dynamic algorithms for shortest paths in planar graphs. In *17th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 187–197, 1991. 53
[For56] L. R. Ford. Network flow theory. RAND Report P-923, The Rand Corporation, SantaMonica, CA, August 1956. 15
[FR06] J. Fakcharoenphol and S. Rao. Planar graphs, negative weight edges, shortest paths, and near linear time. *J. Comput. Syst. Sci.*, 72(5):868–889, 2006. Preliminary version
in FOCS 2001. 3, 19, 21, 22, 34, 53, 54, 69, 70
100
[Fre87] G. N. Frederickson. Fast algorithms for shortest paths in planar graphs with applications. *SIAM Journal on Computing*, 16:1004–1022, 1987. Preliminary version in FOCS
1983. 14
[FT87] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization

algorithms. *J. ACM*, 34(3):596–615, 1987. 15, 34

[GBT84] H.N. Gabow, J.L. Bentley, and R.E. Tarjan. Scaling and related techniques for geometry problems. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing (STOC)*, pages 135–143, 1984. 21

[GG84] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian relation of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721– 742., 1984. 66[Gol95] A. V. Goldberg. Scaling algorithms for the shortest paths problem. *SIAM J. Comput.*, 24(3):494–504, 1995. 34

[Gol98] A. V. Goldberg. Recent developments in maximum flow algorithms. In Stefan Arnborg and Lars Ivansson, editors, *Algorithm Theory SWAT'98*, volume 1432 of *Lecture Notes in Computer Science*, pages 1–10. Springer Berlin / Heidelberg, 1998. 64