# UC-LTM: Unidimensional Clustering Using Latent Tree Models for Discrete Data[☆]

Leonard K. M. Poon[a], April H. Liu[b], Nevin L. Zhang[c]

[a] *Department of Mathematics and Information Technology, The Education University of Hong Kong, Hong Kong SAR, China*
[b] *School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai, China*
[c] *Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China*

**Abstract**

This paper is concerned with model-based clustering of discrete data. Latent class models (LCMs) are usually used for this task. An LCM consists of a latent variable and a number of attributes. It makes the overly restrictive assumption that the attributes are conditionally independent given the latent variable. We propose a novel method to relax this assumption. The key idea is to partition the attributes into groups such that correlations among the attributes in each group can be properly modeled by using a single latent variable. The latent variables for the attribute groups are then used to build a number of models, and one of them is chosen to produce the clustering results. The new method produces unidimensional clustering using latent tree models and is named UC-LTM. Extensive empirical studies were conducted to compare UC-LTM with several model-based and distance-based clustering methods. UC-LTM outperforms the alternative methods in most cases, and the differences are often large. Further, analysis on real-world social capital data further shows improved results given by UC-LTM over results given by LCMs in a previous study.

*Keywords:* Unidimensional clustering, Latent tree models, Latent class models, Probabilistic graphical models, Unsupervised learning

## 1. Introduction

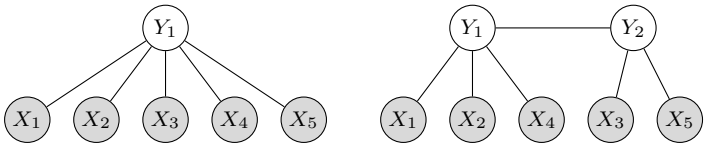
Cluster analysis is a classic research topic in artificial intelligence. A variety of approaches have been proposed, including distance/similarity-based algorithms such as K-means, kernel K-means, and spectral clustering [2], as well as model-based methods such as Gaussian mixture models (GMMs) [3] and latent class models (LCMs) [4]. Whereas GMMs are used to analyze continuous data, LCMs are used to deal with discrete data. LCMs have been used for cluster analysis in the social, behavioral, and health sciences [5].

This paper focuses on discrete data. An LCM for clustering discrete data consists of a latent variable and a set of discrete attributes (observed variables) that describe the data (Figure 1(a)). Each state of the latent variable represents a cluster to be identified, and the latent variable itself represents a partition of data to be obtained. The model assumes that the attributes are conditionally independent given the clustering latent variable. In other words, the attributes are mutually independent in each cluster. This assumption is hence referred to as the *local independence* assumption. In practice, it is often violated in practice and can lead to spurious clusters [6, 7].

In this paper, we propose a novel method to relax the local independence assumption of LCM and to detect and model local dependence properly so as to improve clustering quality. Our proposed method, named UC-LTM, can be divided into three main steps.

[☆] This paper is an extended version of [1].

*Email addresses:* kmpoon@eduhk.hk (Leonard K. M. Poon), liu.hua@mail.shufe.edu.cn (April H. Liu), lzhang@cse.ust.hk (Nevin L. Zhang)




(a) Model $m_1$ (LCM) (b) Model $m_2$ (LTM)

Figure 1: Examples of a latent class model ($m_1$) and a latent tree model ($m_2$). The shaded nodes represent observed variables (attributes), and the unshaded nodes represent latent variables. An LCM can have only one latent variable, whereas an LTM can have multiple latent variables connected in a tree structure. This figure also illustrates the models considered in the unidimensionality test (UD test). LCMs ($m_1$) and LTMs restricted to having at most two latent variables ($m_2$) are compared in the UD test to determine the unidimensionality of a set of attributes.

In Step 1, we extract latent features from data. More specifically, we partition the attributes into groups such that correlations among the attributes in each group can be properly modeled using a single latent variable, and we introduce a latent variable for each group. Each latent variable introduced can be intuitively understood as capturing one aspect of the data.

In Step 2, we construct a number of different models for clustering using the latent variables obtained in the previous step. The models constructed produce clustering with different emphases on the multiple aspects of the data. One model constructed uses all the latent variables from Step 1 as features and introduces a new latent variable for clustering. It produces a partition of data that is evenly based on all aspects of data and hence is called the *balanced model*. In each of the other models, attributes from one group and latent variables for the other groups are used as features for clustering. Those models produce partitions of data that primarily depend on only one aspect of the data and hence are called *unbalanced models*.

In Step 3, we select one of the models constructed in the previous step to produce the final clustering results. We choose among the different models using a model selection criterion. Both the AIC score [8] and the BIC score [9] are considered.

All the models considered contain multiple latent variables that are connected up to form a tree structure. Hence they are special cases of latent tree models (LTMs) [10, 11, 12] (Figure 1(b)). This paper differs from previous works on LTMs as follows. In the method proposed in this paper, one of the latent variables is designated as the clustering variable during model construction. The objective is to model local dependence in an LCM so as to improve clustering quality. In contrast, the objective of previous studies on LTMs was to optimize the fit to the data. None of the latent variables was designated as *the* clustering variable. They were sometimes all interpreted as clustering variables, leading to multiple partitions of data. Thus, previous studies on LTMs aimed to find the best way to cluster data simultaneously along multiple dimensions, whereas in the approach described in this paper, the aim is to find the best way to cluster data along a single dimension.

The remainder of the paper is organized as follows. We briefly review the basic concepts in Section 2. We then discuss the new method, named UC-LTM, in the following three sections, describing Step 1 in Section 3 and Steps 2 and 3 in Section 4 and analyzing the time complexity in Section 5. In Section 6, we discuss related works. After that, we present empirical results on synthetic data, real-world benchmark data, and social capital data in Sections 7–9. Finally, we draw conclusions in Section 10.

## 2. Review of Basic Concepts

We start by giving a brief review of LCMs and LTMs. A *latent tree model* (LTM) is a Markov random field over an undirected tree, where variables at leaf nodes are observed and variables at internal nodes are hidden. An example of an LTM is shown in Figure 1(b), where $Y_1$ and $Y_2$ are latent variables and $X_1$–$X_5$ are observed variables. For technical convenience, we often root an LTM at one of its latent nodes and regard it as a directed graphical model, i.e., a Bayesian network [13]. In the example, suppose that we root the model at the node $Y_1$. Then the numerical information of the model will include a marginal distribution

$P(Y_1)$ for the root and one conditional distribution for each edge (e.g., $P(X_1|Y_1)$ for $Y_1 \rightarrow X_1$ and $P(Y_2|Y_1)$ for $Y_1 \rightarrow Y_2$).

In general, suppose there are $p$ observed variables $X_1, \ldots, X_p$ and $q$ latent variables $Y_1, \ldots, Y_q$ in an LTM. Denote the parent of a variable $Z$ as $parent(Z)$ and let $parent(Z)$ be the empty set when $Z$ is the root. The LTM defines a joint distribution over $X_1, \ldots, X_p, Y_1, \ldots, Y_q$ as

$$P(X_1, \ldots, X_p, Y_1, \ldots, Y_q) = \prod_{Z \in \{X_1, \ldots, X_p, Y_1, \ldots, Y_q\}} P(Z|parent(Z)).$$

The BIC score [9] is usually used to evaluate an LTM $m$:

$$BIC(m|\mathcal{D}) = \log P(\mathcal{D}|m, \theta^*) - \frac{d(m)}{2} \log N,$$

where $\mathcal{D}$ is the data set, $\theta^*$ is the maximum likelihood estimate of the parameters, $d(m)$ is the number of free probability parameters in $m$, $N$ is the sample size, and the log function refers to the natural logarithm. In this paper, we also consider the AIC score [8]:

$$AIC(m|\mathcal{D}) = \log P(\mathcal{D}|m, \theta^*) - d(m).$$

In both scores, the first terms correspond to the likelihood and are the same. The second terms correspond to a penalty based on the model complexity and are different. By comparing the penalty terms, we see that the BIC score tends to favor simpler models. We empirically compare the use of both scores in our experiments. Interested readers are referred to other sources for theoretical comparisons such as the differences in their mathematical motivations and required assumptions [14, 15, 16].

A variety of algorithms have been proposed for learning the structure of LTMs [11]. Here we describe a search-based method called EAST [17] because of its relevance to the current work. EAST searches in the space of all LTMs to find the one with the highest BIC score. In each search step, EAST considers a number of candidate models generated from a base model by five possible search operators, namely node introduction (NI), node deletion (ND), state introduction (SI), state deletion (SD), and node relocation (NR). Given two sibling nodes $V_1$ and $V_2$ and their parent node $Y$, the NI operator creates a candidate model by adding a new node as a child of $Y$ and the parent of $V_1$ and $V_2$. The ND operator is the opposite of NI. The SI operator creates a candidate model by adding a state to the domain of one of the latent variables. The SD operator does the opposite. The NR operator creates a candidate model by relocating a node to another latent node as its new parent.

For the sake of efficiency, each iteration in the search is divided into three phases. The SI and NI operators are used in an expansion phase, the NR operator in an adjustment phase, and the SD and ND operators in a simplification phase. The EAST search repeats the three phases until none of them can find a better model in terms of the BIC score.

A *latent class model* (LCM) is an LTM with a single latent variable. An example is shown in Figure 1(a), where $Y_1$ is the latent variable and $X_1$–$X_5$ are observed variables. Suppose there is a data set on the observed variables. To learn an LCM from the data set means to determine the *cardinality* (i.e., the number of states) of $Y_1$ and the probability distributions $P(Y_1)$ and $P(X_i|Y_1)$ $(i = 1, \ldots, 5)$. To do so, we initially set the cardinality of $Y_1$ to 2 and optimize the probability parameters using the EM algorithm [18]. Then, the cardinality is gradually increased, and the parameters are re-estimated after each increase. The process stops when the model score ceases to increase. The final model is returned as the output. We refer to this procedure as `LearnLCM`$(\mathcal{D}, f)$, where $\mathcal{D}$ is the data set and $f$ is the model scoring function (i.e. AIC or BIC).

After an LCM is learned, we can calculate the posterior distribution $P(Y_1|X_1, \ldots, X_5)$ for each data case. The data case belongs to each state of $Y_1$ with some probability. Hence, the posterior distributions for all data cases give a *soft partition* of the data. If we assign each data case to the state of $Y_1$ with the maximum posterior probability, an operation known as *hard assignment*, then we obtain a *hard partition* of the data.

## 3. Extraction of Latent Features

In this section, we describe Step 1 of the new method. The *unidimensionality test*, or *UD test* for short, is a Bayesian statistical test that tests whether a subset $\boldsymbol{S}$ of attributes is *unidimensional*, meaning that correlations among the attributes in $\boldsymbol{S}$ can be properly modeled using a single latent variable [19]. The test takes unidimensionality as the default position and looks for evidence to reject it. To look for such evidence, the test searches for the models with the highest BIC scores among two restricted sets of LTMs. Let $m_1$ be the best model among the LTMs that contain a single latent variable and $m_2$ be the best among LTMs that contain no more than two latent variables (see Figure 1). The *UD test* rejects the default position if $m_2$ contains two latent variables and

$$BIC(m_2|\mathcal{D}') - BIC(m_1|\mathcal{D}') > \delta, \tag{1}$$

where $\delta$ is a threshold parameter. The left side of inequality (1) is an approximation of the natural logarithm of the Bayes factor $B_{21}$ for comparing $m_2$ with $m_1$ [20]:

$$\log B_{21} \coloneqq \log \frac{P(\mathcal{D}'|m_2)}{P(\mathcal{D}'|m_1)} \approx BIC(m_2|\mathcal{D}') - BIC(m_1|\mathcal{D}').$$

For this reason, only the BIC score is used in the UD test.

In our experiments, the threshold $\delta$ is by default set to 3 as suggested by Kass and Raftery [20]. Assuming equal prior odds $P(m_2)/P(m_1)$, the Bayes factor $B_{21}$ is equal to the posterior odds $P(m_2|\mathcal{D}')/P(m_1|\mathcal{D}')$. When $\log B_{21} > 3$, the posterior odds is greater than 20:1. This is analogous to the standard 5% significance level [21]. Therefore, it can be interpreted as strong evidence that $m_2$ should be selected over $m_1$. Furthermore, if $m_2$ contains two latent variables at the same time, the evidence supports that correlations among the attributes in $\boldsymbol{S}$ should be modeled by more than one latent variable. Consequently, the default position of unidimensionality should be rejected.

To perform the UD test in practice, we first project the original data set $\mathcal{D}$ onto $\boldsymbol{S}$ to obtain a smaller data set $\mathcal{D}'$. In other words, a weight is associated in $\mathcal{D}'$ with each distinct data case on $\boldsymbol{S}$ based on the count of the corresponding data cases in $\mathcal{D}$. The model $m_1$ is obtained using $\texttt{LearnLCM}(\mathcal{D}', f)$. The model $m_2$ is obtained using the EAST algorithm [17]. For the UD test, we restrict the search space to contain only LTMs with one or two latent variables. We refer to this restricted version of EAST as $\texttt{LearnLTM-2L}(\mathcal{D}', f)$, where LTM-2L stands for LTMs with at most two latent variables.

In the following, we present a method proposed by Liu et al. [19] for partitioning the attributes in a data set into unidimensional clusters, or *UD clusters* for short. The method relies on mutual information (MI) [22]. The mutual information $I(X;Y)$ between two variables $X$ and $Y$ is defined as

$$I(X;Y) = \sum_{X,Y} P(X,Y) \log \frac{P(X,Y)}{P(X)P(Y)},$$

where the summation is taken over all possible states of $X$ and $Y$. For this work, $P(X,Y)$ is the joint empirical distribution of the two variables estimated from data.

To determine the first UD cluster, one maintains a working set $\boldsymbol{S}$ of attributes that initially consists of the pair of attributes with the highest MI. The set is then expanded by adding other attributes one by one. At each step, the attribute that has the highest MI with the current set is added. The MI between a variable $X$ and a set $\boldsymbol{S}$ of variables is estimated as $I(X;\boldsymbol{S}) = \max_{Z\in\boldsymbol{S}} I(X;Z)$. Then the UD test is performed to determine whether correlations among the variables in $\boldsymbol{S}$ can still be properly modeled using a single latent variable. If the UD test is failed, the expansion process stops and the first UD cluster is picked.

To illustrate the process, let us suppose that the working set initially contains $X_1$ and $X_2$, and then $X_3$ and $X_4$ are added, and the UD test is passed in both cases. Now consider the addition of attribute $X_5$. Suppose models $m_1$ and $m_2$ learned for the attributes $\{X_1, X_2, X_3, X_4, X_5\}$ are as shown in Figure 1. Further, suppose the difference in the BIC scores between $m_2$ and $m_1$ exceeds the threshold $\delta$. Then the UD test is failed, and it is time to pick the first UD cluster.

**Algorithm 1** `AttributeGrouping`$(\mathcal{D}, \delta, f)$

**Input**: $\mathcal{D}$ — Data with attribute set $\boldsymbol{X}$, $\delta$ — Threshold for UD test, $f$ — Model scoring function.
**Output**: Set of latent features $\boldsymbol{Y}$.

$\boldsymbol{Y} \leftarrow \emptyset$.
Calculate the empirical MI between each pair of variables in $\boldsymbol{X}$.
**while** $|\boldsymbol{X}| > 0$ **do**
  **if** $(|\boldsymbol{X}| \leq 3)$ **then**
    $\boldsymbol{S} \leftarrow \boldsymbol{X}$, $\boldsymbol{X} \leftarrow \emptyset$.
  **else**
    $\boldsymbol{S} \leftarrow$ the pair of variables in $\boldsymbol{X}$ with the highest MI.
    **loop**
      $X \leftarrow$ the variable in $\boldsymbol{X} \setminus \boldsymbol{S}$ that has the highest MI with $\boldsymbol{S}$.
      $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{X\}$, $\boldsymbol{X} \leftarrow \boldsymbol{X} \setminus \{X\}$.
      $\mathcal{D}' \leftarrow$ projection of $\mathcal{D}$ onto $\boldsymbol{S}$.
      $m_1 \leftarrow$ `LearnLCM`$(\mathcal{D}', f)$, $m_2 \leftarrow$ `LearnLTM-2L`$(\mathcal{D}', f)$.
      **if** ($m_2$ contains two latent variables **and** $BIC(m_2|\mathcal{D}') - BIC(m_1|\mathcal{D}') > \delta$) **then**
        Let $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$ be the first and second UD clusters as explained in Section 3.
        $\boldsymbol{S} \leftarrow \boldsymbol{S_1}$ and $\boldsymbol{X} \leftarrow \boldsymbol{X} \cup \boldsymbol{S_2}$.
        **break**
      **else if** $(|\boldsymbol{X}| = 0)$ **then**
        **break**
      **end if**
    **end loop**
  **end if**
  $\mathcal{D}' \leftarrow$ projection of $\mathcal{D}$ onto $\boldsymbol{S}$.
  $m \leftarrow$ `LearnLCM`$(\mathcal{D}', f)$, $Y \leftarrow$ the latent variable in $m$.
  $\boldsymbol{Y} \leftarrow \boldsymbol{Y} \cup \{Y\}$.
**end while**
**return** $\boldsymbol{Y}$.

Model $m_2$ gives us two possible UD clusters $\{X_1, X_2, X_4\}$ and $\{X_3, X_5\}$. The first cluster is picked because it contains both of the two initial attributes, $X_1$ and $X_2$. In general, it might happen that none of the two clusters given by $m_2$ contain both of the two initial attributes. In such a case, we pick the one with more attributes and break ties arbitrarily.

After the first UD cluster is determined, attributes in the cluster are removed from the data set, and the process is repeated to find other UD clusters. This continues until all attributes are grouped into UD clusters.

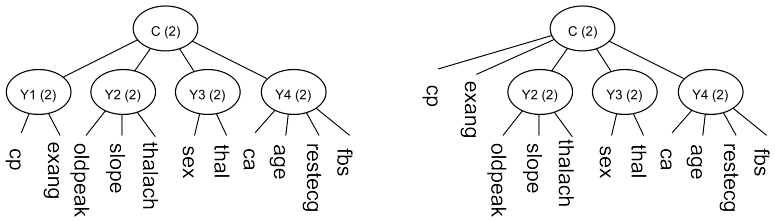
After attribute partitioning, an LCM is learned for attributes in each UD cluster using `LearnLCM`$(\mathcal{D}', f)$. Suppose there are $L$ variable clusters; then we will get $L$ LCMs. Denote the latent variables in the LCMs as $Y_1$, $Y_2$, ..., $Y_L$. These will be used as features for data clustering. We refer to the procedure that produces the $L$ latent variables and LCMs as `AttributeGrouping`$(\mathcal{D}, \delta, f)$, where $\mathcal{D}$ is the data set, $\delta$ is the threshold for the UD test, and $f$ is the model scoring function used when learning LCMs for the UD clusters. Its pseudo-code is given as Algorithm 1.

The subroutine `LearnLTM-2L`$(D', f)$ is used by `AttributeGrouping` for the UD test. It is given as Algorithm 2. In this algorithm, `pickBestModel`$(\mathcal{M}, f)$ returns the best model among a given set of models $\mathcal{M}$ according to the model scoring function $f$. $NI(m)$ and $SI(m)$ indicate the sets of candidates that can be generated from a model $m$ by the NI and SI operators. $NR(m', Y, Y')$ stands for the set of models that can be obtained from model $m'$ by relocating one neighbor of $Y$ to $Y'$. The subroutine uses only the expansion phase of EAST for the sake of efficiency.

**Algorithm 2** `LearnLTM-2L`$(\mathcal{D}', f)$

**Input**: $\mathcal{D}'$ — Data with attributes $\boldsymbol{S}$, $f$ — Model scoring function.
**Output**: A latent tree model over $\boldsymbol{S}$ that contains one or two latent variables.

1: $m \leftarrow$ LCM with observed variables $\boldsymbol{S}$ and one latent variable $Y$ with two values.
2: **loop**
3: **if** ($m$ has only 1 latent variable) **then**
4: $m' \leftarrow$ `pickBestModel`$(NI(m) \cup SI(m), f)$.
5: **else**
6: $m' \leftarrow$ `pickBestModel`$(SI(m), f)$.
7: **end if**
8: **if** ($m'$ was obtained from $m$ by introducing a new latent variable $Y'$) **then**
9: **loop**
10: $m'' \leftarrow$ `pickBestModel`$(NR(m', Y, Y'), f)$.
11: **if** $(f(m'' \mid \mathcal{D}') \leq f(m' \mid \mathcal{D}'))$, **break**.
12: $m' \leftarrow m''$.
13: **end loop**
14: **end if**
15: **if** $(f(m' \mid \mathcal{D}') \leq f(m \mid \mathcal{D}'))$, return $m$.
16: $m \leftarrow m'$.
17: **end loop**




(a) Balanced 3L-LTM (b) Unbalanced 3L-LTM

Figure 2: The balanced 3L-LTM and one of the unbalanced 3L-LTMs learned on the `heart-c` data set. The grouping of attributes is determined by `AttributeGrouping`. The model in (b) is constructed from the model in (a) by removing $Y_1$ and connecting the children of $Y_1$ to $C$. Three other unbalanced 3L-LTMs can be similarly constructed by removing one of $Y_2$, $Y_3$, or $Y_4$. The numbers in parentheses are the cardinalities of the latent variables estimated by our method.

## 4. Use of Latent Features for Data Clustering

In this section, we describe Steps 2 and 3 of the new method. We use an example for illustration. We have run `AttributeGrouping` on a version of the `heart-c` data set from the UCI repository [23]. Four UD clusters are detected. The first cluster consists of the attributes `cp` (chest pain type) and `exang` (exercise induced angina); and the second consists of `oldpeak` (ST depression induced by exercise), `slope` (slope of the peak exercise ST segment), and `thalach` (maximum heart rate achieved). The two clusters are clearly meaningful and capture two different aspects of the data. The other two UD clusters are not given for brevity.

An LCM is learned for each UD cluster of attributes. Denote the latent variables in the four LCMs as $Y_1$, $Y_2$, $Y_3$, and $Y_4$, respectively. They are the latent features detected by `AttributeGrouping`. They summarize the attributes connected to them. The question is how to use these latent features to cluster the data. We consider two variants of three-level LTMs (3L-LTM) for this purpose as follows.

One natural idea is to build an LCM using the four latent variables as features. This results in the model shown in Figure 2(a). The top part of the model is an LCM, where $C$ is a new latent variable that represents

**Algorithm 3** `UC-LTM`$(\mathcal{D}, \delta, f)$

**Input**: $\mathcal{D}$ — Data, $\delta$ — Threshold for UD test, $f$ — Model scoring function (either AIC or BIC).
**Output**: A three-level latent tree model (3L-LTM).

1: Run `AttributeGrouping`$(\mathcal{D}, \delta, f)$ to obtain a list of latent features $Y_1$, $Y_2$, ..., $Y_L$.
2: Build a balanced 3L-LTM using the latent variables $Y_1$, $Y_2$, ..., $Y_L$.
3: **for** $i = 1$ to $L$ **do**
4: Build an unbalanced 3L-LTM by deleting $Y_i$ from the balanced model and connecting the clustering variable $C$ directly to each child of $Y_i$.
5: **end for**
6: Among the $(L + 1)$ three-level models, pick the one that has the highest score according to the scoring function $f$.
7: **return** the selected model.

the data partition to be obtained, and is hence called the *clustering variable*. Attributes are added at the bottom because the variables at the middle level are latent and their values must be inferred from observed variables. The model is balanced because all UD clusters are treated the same as far as the model structure is concerned. It will be called the *balanced 3L-LTM*.

In the balanced 3L-LTM, the cardinalities of the latent variables $Y_1$, $Y_2$, $Y_3$, and $Y_4$ and the conditional distributions of their children are inherited from the LCMs produced by `AttributeGrouping` and are fixed. Those distributions define the latent features. If they were allowed to change, then we would not be using the features as they are.

We need to determine the cardinality of the clustering variable $C$, the marginal distribution $P(C)$, and the conditional distribution of each child of $C$ given $C$, i.e., $P(Y_i|C)$ $(i = 1, 2, 3, 4)$. This is done using a procedure similar to `LearnLCM`$(\mathcal{D}, f)$. Note that the data set $\mathcal{D}$ here is obtained by computing the posterior probability of $Y_i$ on the original data using the LCMs.

In addition to the balanced 3L-LTM, we also consider a number of unbalanced models in which the attributes from one UD cluster, together with latent variables for other UD clusters, are used as features. One example is shown in Figure 2(b), where the attributes `cp` and `exang` from one UD cluster and the latent variables $Y_2$, $Y_3$ and $Y_4$ for the other UD clusters are used as features for the LCM at the top. Such a model is desirable if the "true clustering" primarily depends on only one aspect of the data.

In the unbalanced model, the cardinalities of $Y_2$, $Y_3$, and $Y_4$ and the conditional distributions of their children are inherited from the LCMs produced by `AttributeGrouping` and are fixed. We need to determine the cardinality of $C$, the marginal distribution $P(C)$, and the conditional distribution of each child of $C$ given $C$. This is accomplished using a procedure similar to `LearnLCM`$(\mathcal{D}, f)$ as with the balanced model.

Note that we do not consider the use of attributes from multiple UD clusters directly as features because that would introduce local dependence.

Suppose the subroutine `AttributeGrouping` produces $L$ latent features. Using these features, we can construct one balanced 3L-LTM, and $L$ unbalanced 3L-LTMs in Step 2. Among the $L + 1$ models, we pick one best model as the final output in Step 3. Here we try both BIC and AIC as the criteria for model selection. After the model is learned, one can compute the posterior distribution $P(C|d_i)$ of the clustering variable $C$ for each data case $d_i$. This gives a soft partition of the data. To obtain a hard partition, one can assign each data case to the state of $C$ that has the maximum posterior probability.

Algorithm 3 shows the pseudo-code for our algorithm. It is called *UC-LTM*, which stands for Unidimensional Clustering using Latent Tree Models.

To further illustrate the difference between balanced and unbalanced 3L-LTMs, let us consider an example of a data set with 12 attributes $X_1$–$X_{12}$. Suppose the attributes in each of the attribute groups $\{X_1, X_2, X_3\}$, $\{X_4, X_5, X_6\}$, $\{X_7, X_8, X_9\}$, and $\{X_{10}, X_{11}, X_{12}\}$ are highly correlated such that they always take the same value in each data case. Given sufficient data, the subroutine `AttributeGrouping` should detect the four attribute groups. Denote the root variables of the four LCMs built on the attribute groups by $Y_1$, $Y_2$, $Y_3$,

and $Y_4$, respectively. Figure 4(a) shows the balanced 3L-LTM that is constructed, Figure 4(b) shows the unbalanced 3L-LTM constructed by removing $Y_1$.

Now consider the likelihood functions of the LCMs at the top of the two models. Use $m_b$ and $m_u$ to denote the two LCMs in the balanced model and unbalanced model, respectively. The likelihood function for $m_b$ is:

$$P(Y_1, Y_2, Y_3, Y_4|m_b) = \sum_C \prod_{i=1}^{4} P(Y_i|C)P(C),$$

and the likelihood function for $m_u$ is

$$\begin{aligned} P(X_1, X_2, X_3, Y_2, Y_3, Y_4|m_u) &= \sum_C \prod_{i=1}^{3} P(X_i|C) \prod_{i=2}^{4} P(Y_i|C)P(C) \\ &= \sum_C (P(Y_1|C))^3 \prod_{i=2}^{4} P(Y_i|C)P(C). \end{aligned}$$

The second equation follows because the values of $X_1$–$X_3$ are always the same, and thus $Y_1$ should take that same value as well. Comparing the two likelihood functions, we see that the one corresponding to the unbalanced model $m_u$ has a stronger dependence on $P(Y_1|C)$. As a result, $C$ has a stronger dependence on $Y_1$ in the maximum likelihood case. In this example, we may consider $Y_1$ as a latent feature or an aspect of the data corresponding to the attributes $X_1$–$X_3$. The example explains how the unbalanced 3L-LTM primarily depends on that aspect of the data.

## 5. Complexity Analysis

In this section, we analyze the time complexity of the new method, UC-LTM. Algorithm 3 spends time mostly on running `AttributeGrouping` (line 1) and building the 3L-LTMs (lines 2–5). We analyze them in terms of the following quantities. Let $N$ be the sample size, $p$ be the number of attributes, $L$ be the number of latent features discovered, $c$ be the maximum cardinality of a variable, $k$ be the maximum number of observed variables in a working set $\boldsymbol{S}$ in `AttributeGrouping`, and $r$ be the maximum number of iterations of EM.

We first look at `AttributeGrouping` (Algorithm 1). Its run time is dominated by calls to subroutine `LearnLTM-2L` at line 12. In each call (Algorithm 2), lines 4 and 6 are executed no more than $2(c-2)$ times in total; each time, no more than $\binom{k}{2} < k^2$ candidate models are considered. Line 10 is executed no more than $k-2$ times; each time, no more than $k$ candidates are considered. Therefore, one call to `LearnLTM-2L` involves no more than $2(c-2)k^2 + (k-2)k < 2ck^2$ candidate models.

To evaluate each candidate model, we need to run EM once to optimize its parameters. A candidate model contains no more than $k+2$ variables. There are $N$ samples. Since inference in trees takes linear time in the number of nodes and is quadratthe ic in maximum cardinality of the variables, each EM iteration takes $O((k+2)c^2N)$ time. Consequently, the time it takes to evaluate one candidate model is $O(rkc^2N)$.

The while loop of `AttributeGrouping` is executed $L$ times. In each pass through the while loop, `LearnLTM-2L` is called no more than $k-2$ times. Putting everything together, we see that the total time of all calls to `LearnLTM-2L` is $O(L \cdot (k-2) \cdot 2ck^2 \cdot rkc^2N) = O(LNrc^3k^4)$. It is linear in the sample size $N$ and the number of latent features $L$. In a more careful analysis, $N$ can be replaced by the number of distinct data cases one obtains by projecting $N$ samples onto the working set $\boldsymbol{S}$; this number can be much smaller than $N$. The maximum number of EM iterations $r$ can be controlled. The terms $c^3$ and $k^4$ may look bad. Fortunately, however, $c$ is usually very small relative to $N$ and $L$, and $k$ is usually much smaller than the total number of observed variables $p$.

We now consider the run time for building the 3L-LTMs. To learn their parameters, we run EM only on the top part of the model, which is an LCM. For the balanced model, the LCM has $L$ leaf variables. For the $L$ unbalanced models, the LCM has no more than $(L-1+k)$ leaf variables. Hence, each LCM has at

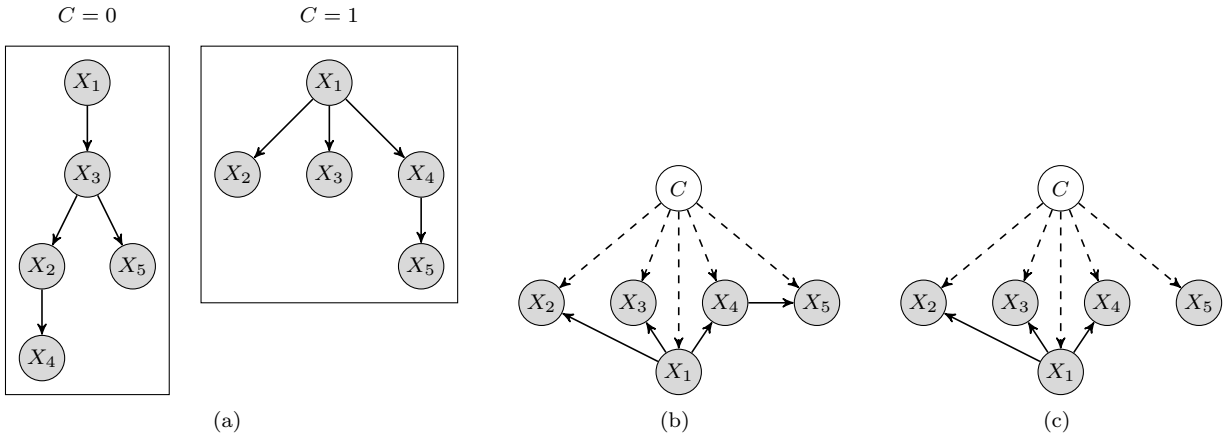



Figure 3: Illustration of the models based on the tree-augmented naive (TAN) Bayes model for clustering [34]: (a) Chow and Liu multinet classifier (CLMNC); (b) tree-augmented naive Bayes classifier (TANBC); and (c) simple Bayesian network classifier (SBNC). The clustering variable is denoted by $C$. In (b) and (c), the dashed lines are for the naive Bayes model, and the solid lines are connections among attributes for relaxing the assumption of local independence.

most $L + k$ variables. It takes $O(r(L + k)c^2N)$ time to run EM on each of the model. Therefore, the time for building these models is $O(L \cdot r(L + k)c^2N) = O(L^2Nrc^2 + LNrc^2k)$. The first term is linear in sample size $N$ and quadratic in the number of latent features $L$. The second term is dominated by the time for `AttributeGrouping`.

## 6. Related Work

LTMs have been proposed to relax the local independence assumption in LCMs [10]. A number of structural learning algorithms have been developed. Following Mourad et al. [11], these algorithms can be classified into search-based methods [10, 17], methods based on variable clustering [24, 25, 19, 26], and distance-based methods [27, 28, 29].

Among the four methods based on variable clustering, the method of Wang et al. [24] and that of Harmeling and Williams [25] consider only binary trees. In other words, each latent node can have only two children. In contrast, the method of Liu et al. [19], called Bridged-Islands (BI), and the method of Chen et al. [26], called hierarchical latent tree analysis (HLTA), are similar to UC-LTM. BI, HLTA, and UC-LTM all use the UD test to determine the members of UD clusters and add a latent node as the parent of each UD cluster. These three methods allow a latent node to have more than two children.

BI, HLTA, and UC-LTM use different strategies for building LTMs based on the UD clusters. They produce models with different structures. BI builds a model by connecting the parents of UD clusters using a tree structure. It results in a flat model in which every latent variable is directly connected to at least one attribute. HLTA uses a recursive process and builds an LTM layer by layer; the parents of UD clusters on one layer are treated as attributes for finding the UD clusters on the layer above. The resulting tree structure may have more than three levels. UC-LTM uses the parent nodes of UD clusters as features for building a number of three-level LTMs with specific structures. It outputs one of these three-level LTMs based on a model selection score.

The discrete latent variables in LTMs can be used for clustering (e.g. [17, 30, 31]). Recently, Asbeh and Lerner [32, 33] studied a class of latent variable models called multiple indicator models (MIMs) on discrete data. They showed that LTMs are special cases of MIMs. Both MIMs and LTMs can have multiple latent variables for clustering, but little work has been done to show which of those latent variables should be used. In contrast, UC-LTM designates a latent variable for clustering during model construction.

Technically, LCMs are closely related to the naive Bayes (NB) model for classification. An LCM can be viewed as an NB model in which the class variable is not observed. Several methods have been proposed

to relax the local independence assumption in the NB model for clustering. Peña et al. [35] allowed some attributes to be grouped under the same node as fully correlated attributes. Pham and Ruz [34] proposed three models based on the tree-augmented naive (TAN) Bayes model for clustering (Figure 3). The TAN model [36] is similar to the NB model but allows direct connections among the attributes. For the sake of computational efficiency, those connections are assumed to form a tree. The first model proposed by Pham and Ruz [34] is called the Chow and Liu multinet classifier (CLMNC). It uses different tree-structured models for different clusters. The second model is called the tree-augmented naive Bayes classifier (TANBC). It uses the same tree structure for all clusters but allows the parameters to take different values for different clusters. The third model is called the simple Bayesian network classifier (SBNC). It allows some edges among the attributes to be dropped from the TANBC model.

UC-LTM learns the model parameters with maximum likelihood estimates. Other principles have also been used for parameter estimation in clustering. Bayesian model averaging has been used for learning by NB models [37] and the TAN models [38]. Pham and Ruz [34] estimated model parameters based on classification maximum likelihood. Essentially, in each iteration of an EM-like algorithm, their method partitions the data using the clustering variable and computes the maximum likelihood estimates of the parameters for a cluster using only the subset of data corresponding to that cluster.

## 7. Empirical Evaluation on Synthetic Data

In this section, we empirically evaluate UC-LTM on synthetic data. Two versions of UC-LTM were used in the experiments. The two versions use the AIC and BIC scores for model selection, respectively. The synthetic data are used to demonstrate that UC-LTM can detect and model local dependence properly. They are also used to evaluate some design choices in the algorithm.

A common way to evaluate a clustering algorithm is to start with labeled data, remove the class labels, perform cluster analysis to obtain a hard partition of data, and compare the partition obtained with the partition induced by the class labels. We refer to these two partitions as the cluster partition and the true data partition, respectively, and denote them by $C$ and $C_t$. The quality of the cluster partition is measured using the *normalized mutual information* (NMI) between the two partitions [39], given by

$$NMI(C; C_t) = \frac{I(C; C_t)}{\sqrt{H(C)H(C_t)}},$$

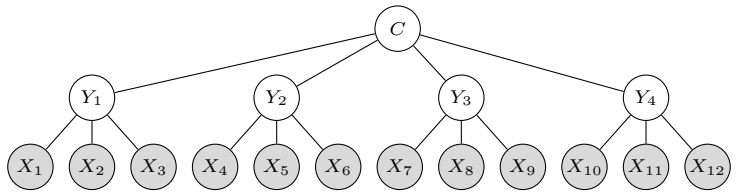
where $I(C; C_t)$ is the MI between $C$ and $C_t$ and $H(\cdot)$ stands for entropy [22]. These quantities can be computed from the empirical joint distribution $P(C, C_t)$ of $C$ and $C_t$. *NMI* ranges from 0 to 1, with a larger value indicating a closer match between $C$ and $C_t$. In the results below, we report the averages and standard deviations of NMI values over 10 repetitions.

### *7.1. Clustering Performance on Synthetic Data*

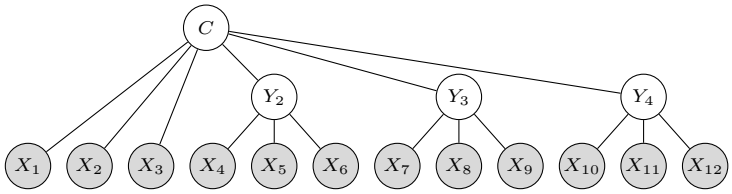
The synthetic data were generated from the two models shown in Figure 4. In the models, all variables have two possible states except for the root variable, which has three. Model parameters were randomly generated. Three data sets were sampled from each model. The sample sizes were 1,000, 5,000 and 10,000, respectively. Each sample contained values for the observed variables and the class variable $C$. The values of $C$ were removed before running the clustering algorithms.

Because of the way the data were generated, the correlations among the attributes cannot be properly modeled using a single latent variable. In other words, local dependence exists. UC-LTM was able to recover the generative structure perfectly in all cases. This shows that UC-LTM is effective in detecting local dependence and modeling it properly.

Table 1 shows the quality of the clustering results produced by UC-LTM and LCM as measured by the NMI with the true class partitions. UC-LTM markedly outperformed LCM regardless of the model scoring function used. This shows the benefits of modeling local dependence. Moreover, UC-LTM outperformed balanced 3L-LTM on the last three data sets. This shows the benefits of including unbalanced models when there is a predominant aspect in the data.




(a) Balanced model $m_1$




(b) Unbalanced model $m_2$

Figure 4: Generative models for the synthetic data. The cardinality of the class variable $C$ is 3. The other variables are binary variables. Model parameters are generated randomly.

| | LCM | Balanced 3L-LTM | UC-LTM-AIC | UC-LTM-BIC |
|---|---|---|---|---|
| syn-b-1k | 0.48±.00 | **0.65±.00** | **0.65±.00** | **0.65±.00** |
| syn-b-5k | 0.48±.00 | **0.65±.00** | **0.65±.00** | **0.65±.00** |
| syn-b-10k | 0.48±.00 | **0.64±.00** | **0.64±.00** | **0.64±.00** |
| syn-ub-1k | 0.15±.00 | 0.25±.01 | **0.32±.01** | **0.32±.01** |
| syn-ub-5k | 0.15±.00 | 0.25±.01 | **0.32±.01** | **0.32±.01** |
| syn-ub-10k | 0.15±.00 | 0.19±.05 | **0.32±.01** | **0.32±.01** |

Table 1: Performance of various methods on synthetic data. The abbreviations "b" and "ub" denote data generated from the balanced and unbalanced models, respectively, shown in Figure 4. The numbers 1k, 5k, and 10k refer to the number of samples. UC-LTM attains the highest NMI values.

### *7.2. Sensitivity Study on $\delta$*

The UC-LTM algorithm has one parameter that the user must set, namely the threshold $\delta$ for the UD test. To gain some intuition about the impact of the parameter, take a look at Algorithm 1. We see that the larger the value of $\delta$, the harder it is to satisfy the condition at line 9, and the longer the set $\boldsymbol{S}$ would keep expanding, which often implies larger sibling clusters.

In the above experiments, $\delta$ was set to 3 as suggested by Kass and Raftery [20]. In the context of this paper, the use of the value 3 implies that we would conclude the correlations among attributes in the set $\boldsymbol{S}$ can be properly modeled using a single latent variable if there is no *strong* evidence to the contrary. Two other possible values for $\delta$, 1 and 5, were also suggested by Kass and Raftery [20]. The use of those values would mean, respectively, to draw the same conclusion when there is *no positive* or *very strong* evidence to the contrary.

To investigate the impact of $\delta$ on our algorithm, we tried $\delta$ values of 1 and 5 in addition to 3. The value 10 was also included as a reference. The results are shown in Table 2. We see that the choice of $\delta$ did not influence the performance of our algorithm considerably in terms of NMI values.

### *7.3. Alternative Method for Latent Feature Extraction*

The UC-LTM algorithm extracts latent features using `AttributeGrouping` based on the UD test. One advantage of using AttributeGrouping is that it focuses on only a subset of variables and leads to higher efficiency.

Alternatively, the latent features can be extracted from an LTM built using the whole data set. Specifically, we build an LTM without any restrictions using the EAST algorithm. We then group the attributes

| | $\delta = 1$ | $\delta = 3$ | $\delta = 5$ | $\delta = 10$ |
|---|---|---|---|---|
| syn-b-1k | **0.65±.00** | **0.65±.00** | 0.63±.00 | 0.63±.00 |
| syn-b-5k | 0.65±.00 | 0.65±.00 | 0.65±.00 | 0.65±.00 |
| syn-b-10k | **0.65±.00** | 0.64±.00 | 0.64±.00 | 0.64±.00 |
| syn-ub-1k | **0.32±.00** | **0.32±.00** | 0.31±.00 | 0.30±.00 |
| syn-ub-5k | **0.32±.00** | **0.32±.00** | 0.31±.00 | 0.30±.00 |
| syn-ub-10k | **0.32±.00** | **0.32±.00** | **0.32±.00** | 0.31±.00 |

Table 2: Impact of $\delta$ on the performance of UC-LTM-AIC.

| | Alternative Method based on EAST (AIC) | Alternative Method based on EAST (BIC) | UC-LTM-AIC | UC-LTM-BIC |
|---|---|---|---|---|
| syn-b-1k | **0.66±.00** | **0.66±.00** | 0.65±.00 | 0.65±.00 |
| syn-b-5k | 0.65±.00 | 0.65±.00 | 0.65±.00 | 0.65±.00 |
| syn-b-10k | **0.65±.00** | **0.65±.00** | 0.64±.00 | 0.64±.00 |
| syn-ub-1k | **0.34±.00** | **0.34±.00** | 0.32±.00 | 0.32±.00 |
| syn-ub-5k | 0.32±.00 | 0.32±.00 | 0.32±.00 | 0.32±.00 |
| syn-ub-10k | 0.32±.00 | 0.32±.00 | 0.32±.00 | 0.32±.00 |

Table 3: Clustering performance (NMI) obtained by using two different methods for latent feature extraction. The alternative method is explained in Section 7.3.

by their parent latent variable in the resulting LTM. Finally, an LCM is learned on each group of attributes using `LearnLCM`$(\mathcal{D}, f)$. The latent variables of the LCMs can be used as latent features for data clustering as explained in Section 4.

We used the synthetic data to compare the clustering performance using the two different methods for latent feature extraction. The results are shown in Table 3. We see that the clustering performance obtained using the alternative method for latent feature extraction is equal to or slightly better than that using `AttributeGrouping` (UC-LTM). This suggests that the latent features extracted from the LTMs obtained by EAST tend to have higher quality. This is reasonable since EAST considers the whole attribute set whereas the UD test considers only subsets of attributes one at a time.

One drawback of using EAST for feature extraction is that considerable time is spent building LTMs before features can be extracted. To test the computation time, we generated several synthetic data sets from models similar to the ones in Figure 4 but having different numbers of attributes. We generated data

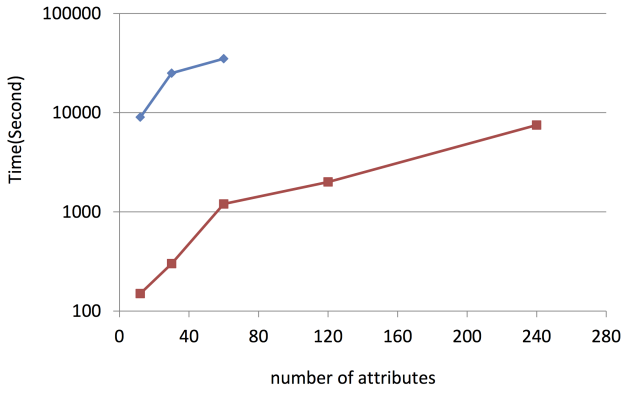



Figure 5: Run time (s) for UC-LTM using `AttributeGrouping` and the alternative method based on EAST for latent feature extraction. The upper blue curve corresponds to using the alternative method for feature extraction, and the lower red curve corresponds to using `AttributeGrouping` for feature extraction.

| | # Attributes | # Samples | # Classes | UC-LTM CPU Time (s) |
|---|---|---|---|---|
| australian | 14 | 690 | 2 | 484 ± 23 |
| autos | 22 | 159 | 7 | 336 ± 10 |
| breast-cancer | 9 | 277 | 2 | 197 ± 9 |
| breast-w | 9 | 683 | 2 | 301 ± 26 |
| corral | 6 | 128 | 2 | 52 ± 3 |
| credit-a | 15 | 653 | 2 | 612 ± 38 |
| credit-g | 15 | 1000 | 2 | 1036 ± 81 |
| diabetes | 6 | 768 | 2 | 67 ± 5 |
| flare | 7 | 1066 | 2 | 200 ± 41 |
| glass | 7 | 214 | 15 | 66 ± 5 |
| glass2 | 5 | 163 | 2 | 42 ± 4 |
| heart-c | 11 | 296 | 5 | 170 ± 18 |
| heart-statlog | 9 | 270 | 2 | 101 ± 9 |
| hypothyroid | 25 | 2645 | 4 | 2849 ± 187 |
| ionosphere | 33 | 351 | 2 | 3042 ± 112 |
| iris | 4 | 150 | 3 | 32 ± 3 |
| kr-vs-kp | 36 | 3196 | 2 | 25526 ± 3420 |
| lymph | 18 | 148 | 4 | 241 ± 10 |
| mofn-3-7-10 | 10 | 1324 | 2 | 423 ± 14 |
| mushroom | 21 | 5644 | 2 | 38902 ± 19707 |
| pima | 6 | 768 | 2 | 68 ± 6 |
| segment | 18 | 2310 | 7 | 16888 ± 2771 |
| shuttle-small | 8 | 5800 | 7 | 3397 ± 1465 |
| sonar | 21 | 208 | 2 | 435 ± 25 |
| soybean | 35 | 562 | 22 | 3441 ± 445 |
| vehicle | 18 | 846 | 4 | 1565 ± 309 |
| vote | 16 | 232 | 2 | 165 ± 14 |
| vowel | 13 | 990 | 11 | 907 ± 80 |
| waveform-21 | 19 | 5000 | 3 | 69582 ± 5634 |
| zoo | 17 | 101 | 7 | 207 ± 5 |

Table 4: Properties of real-world benchmark data sets. The rightmost column shows the average CPU times of UC-LTM with their standard deviations in seconds.

using models with 12, 30, 60, 120, and 240 attributes. Figure 5 shows the run time for the two versions of UC-LTM using `AttributeGrouping` and the alternative method, respectively, for feature extraction. The experiment was conducted on a modest desktop computer. We see that feature extraction based on the UD test can be faster than that based on EAST by one or two orders of magnitude. Furthermore, EAST could not finish within a reasonable amount of time on data sets with more than one hundred attributes. Since the method based on the UD test is only slightly worse in clustering performance but is drastically faster, we adopt the method based on the UD test for latent feature extraction in UC-LTM.

## 8. Empirical Evaluation on Real-World Data

In this section, we empirically evaluate UC-LTM on real-world data. The real-world data are used to show the benefits of modeling local dependence.

### *8.1. Unknown Number of Clusters*

The real-world data sets were obtained from the UCI machine learning repository [23]. We chose 30 labeled data sets often used studies in the literature. The data are from various domains such as medical

| | LCM | CLMNC | TANBC | UC-LTM-BIC | UC-LTM-AIC |
|---|---|---|---|---|---|
| australian | .16±.00 | .29±.06 | .30±.05 | .31±.00 | **.44±.00** |
| autos | .21±.01 | .20±.02 | .20±.01 | .17±.00* | **.23±.00*** |
| breast-cancer | .09±.00 | .01±.00 | .05±.04 | .09±.00 | **.10±.00*** |
| breast-w | .68±.00 | **.75±.09** | .74±.08 | .68±.00 | .68±.00 |
| corral | .19±.00 | .20±.09 | **.22±.12** | .19±.00 | .19±.00 |
| credit-a | .11±.02 | .26±.07 | **.31±.04** | .12±.00 | .13±.01 |
| credit-g | **.01±.00** | .00±.00 | .00±.00 | **.01±.00** | **.01±.00** |
| diabetes | .12±.00 | .09±.03 | .08±.02 | **.15±.02*** | **.15±.02*** |
| flare | **.07±.00** | .05±.02 | .05±.02 | **.07±.00** | **.07±.00** |
| glass | .47±.02 | .26±.08 | .32±.07 | **.48±.00** | **.48±.00** |
| glass2 | **.31±.00** | .19±.08 | .22±.05 | **.31±.00** | **.31±.00** |
| heart-c | .30±.00 | .23±.05 | .28±.03 | .29±.00 | **.33±.00** |
| heart-statlog | .30±.00 | .20±.03 | .22±.04 | **.35±.00** | **.35±.00** |
| hypothyroid | .18±.00 | .03±.03 | .04±.04 | **.21±.00*** | **.21±.00*** |
| ionosphere | .38±.00 | .45±.08 | **.46±.08** | .41±.05 | .44±.03 |
| iris | **.83±.00** | .76±.15 | .63±.30 | **.83±.00** | **.83±.00** |
| kr-vs-kp | **.06±.01** | .05±.02 | .05±.02 | .04±.01 | .04±.01 |
| lymph | .22±.00 | .11±.07 | .11±.05 | .17±.00 | **.29±.00** |
| mofn-3-7-10 | .04±.03 | **.05±.02** | **.05±.02** | **.05±.02*** | **.05±.02*** |
| mushroom | .49±.05 | .22±.17 | .23±.17 | **.52±.01** | **.52±.01** |
| pima | .12±.00 | .08±.03 | .08±.01 | **.15±.03*** | **.15±.03*** |
| segment | **.68±.01** | .19±.10 | .21±.06 | .63±.03 | .63±.03 |
| shuttle-small | .48±.01 | .29±.07 | .35±.07 | **.49±.03*** | **.49±.03*** |
| sonar | **.25±.00** | .23±.03 | .22±.05 | .23±.00 | .23±.00 |
| soybean | **.66±.02** | .44±.03 | .42±.06 | .63±.02* | .63±.02* |
| vehicle | **.31±.01** | .11±.05 | .10±.04 | .30±.01 | .30±.01 |
| vote | **.43±.00** | .37±.07 | .37±.08 | .41±.00 | .41±.00 |
| vowel | .18±.01 | .05±.03 | .05±.04 | **.21±.02** | **.21±.02** |
| waveform-21 | .43±.00 | .31±.05 | .29±.03 | **.48±.00*** | **.48±.00*** |
| zoo | .64±.00 | .23±.22 | .34±.17 | **.72±.07*** | **.72±.07*** |
| win/tie/loss | 6/6/18 | 5/2/23 | 5/1/24 | 0/23/7 | - |
| p-value | 0.0123 | 0.0001 | 0.0002 | 0.0223 | - |

Table 5: Performance of UC-LTM and alternative methods on 30 real-world data sets in the setting of an unknown number of clusters. The averages and standard deviations of NMI values for 10 runs are reported. The highest values are shown in bold. The last two rows compare each alternative method against UC-LTM-AIC. The second-last row summarizes the number of data sets on which the alternative method wins, ties, and loses compared with UC-LTM-AIC. The last row shows the p-value of the Wilcoxon signed-rank test comparing the alternative method and UC-LTM-AIC. In the columns for UC-LTM, an asterisk indicates that the performance is improved with the inclusion of unbalanced 3L-LTMs.

diagnosis, handwriting recognition, and biology. The number of attributes ranges from 4 to 36, the number of classes ranges from 2 to 22, and the sample size ranges from 101 to 5,800. Their properties are summarized in Table 4. Continuous data were discretized using the method of Fayyad and Irani [40].

The class labels were removed, and then different methods were used to recover the class partition from the resulting unlabeled data. The results are shown in Table 5. Instead of comparing each pair of methods, we use UC-LTM-AIC as a pivot and compare it with each of the other methods. We count the number wins, ties, and losses of each alternative method against UC-LTM-AIC. We show also the p-values of the Wilcoxon signed-rank test to check whether the difference in performances of two methods is statistically significant [41]. In the following, we assume a significance level of 5%.

*8.1.1. Findings about UC-LTM*

There are two questions regarding UC-LTM itself. First, what is the impact of the inclusion of unbalanced models? To answer this question, we ran another version of UC-LTM that uses only balanced models. It turns out that the inclusion of unbalanced models never has a negative impact on the performance. It improved the performance on a number of data sets; these are marked with asterisks in Table 5.

The second question is whether the choice of model selection criterion has a significant impact on the performance. We see in Table 5 that UC-LTM-AIC beats UC-LTM-BIC on seven data sets. They obtained the same results on all other data sets. Overall, the performance of UC-LTM-AIC is better. In the following, we compare UC-LTM-AIC with other methods.

*8.1.2. CPU Time of UC-LTM*

The CPU time taken by UC-LTM is reported in the rightmost column of Table 4. Our implementation was written in Java. In our implementation, the final models selected by AIC and BIC were output together in the same run. Hence, we do not report the time for the two versions separately. The CPU time of UC-LTM ranged from 32 seconds on the `iris` data to 19 hours on the `waveform-21` data. In practice, the actual run times were several times smaller than those shown as our implementation utilized multiple CPU cores for parallel computation.

*8.1.3. Comparisons with Alternative Methods*

Table 5 also slows the performance of the LCM, CLMNC, and TANBC methods for comparison. Pham and Ruz [34] assumed the number of clusters are known in their study on CLMNC and TANBC. In the current setting of an unknown number of clusters, we determine the number of clusters using the BIC score based on a procedure similar to `LearnLCM`.

UC-LTM-AIC outperformed LCM on 18 of the 30 data sets. In contrast, LCM outperformed UC-LTM-AIC on 6 data sets. Their performance difference is statistically significant as indicated by the p-value. Overall, the performance of UC-LTM-AIC is superior to that of LCM. The results indicate that it is beneficial to detect and model local dependence, and UC-LTM-AIC is an effective way to do that.

UC-LTM-AIC performed better than CLMNC and TANBC on 23 and 24 data sets, respectively. The p-values show that UC-LTM-AIC is significantly better than those two methods. Intuitively, the clustering of discrete data is based on correlations among the attributes. In CLMNC and TANBC, much of the correlation is explained by the edges among the attributes themselves. Hence, they result in poor clustering performance.

*8.1.4. A Remark*

A careful reader might have noticed that whereas the performance of LCM and UC-LTM is good on data sets such as `iris` and `breast-w`, they are very poor on data sets such as `credit-g` and `kr-vs-kp`. This phenomenon can be understood by considering how closely related the attributes are to the class variables. For a given data set, calculate the average empirical NMI between the class variable $C$ and the attributes as follows:

$$A\text{–}C \text{ correlation strength} = \sum_{A \in \boldsymbol{A}} NMI(A, C)/|\boldsymbol{A}|,$$

| | A–C Correlation | LCM | UC-LTM-BIC | UC-LTM-AIC |
|---|---|---|---|---|
| credit-g | 0.02 | 0.01±.00 | 0.01±.00 | 0.01±.00 |
| kr-vs-kp | 0.02 | 0.06±.01 | 0.04±.01 | 0.04±.01 |
| breast-cancer | 0.04 | 0.09±.00 | 0.09±.00 | 0.10±.00 |
| mofn-3-7-10 | 0.05 | 0.04±.03 | 0.05±.02 | 0.05±.02 |
| flare | 0.05 | 0.07±.00 | 0.07±.00 | 0.07±.00 |
| hypothyroid | 0.05 | 0.18±.00 | 0.21±.00 | 0.21±.00 |
| diabetes | 0.07 | 0.12±.00 | 0.15±.02 | 0.15±.02 |
| pima | 0.07 | 0.12±.00 | 0.15±.03 | 0.15±.03 |
| credit-a | 0.09 | 0.11±.02 | 0.12±.00 | 0.13±.01 |
| glass | 0.30 | 0.47±.02 | 0.48±.00 | 0.48±.00 |
| segment | 0.33 | 0.68±.01 | 0.63±.03 | 0.63±.03 |
| soybean | 0.35 | 0.66±.02 | 0.63±.02 | 0.63±.02 |
| zoo | 0.40 | 0.64±.00 | 0.72±.07 | 0.72±.07 |
| breast-w | 0.44 | 0.68±.00 | 0.68±.00 | 0.68±.00 |
| iris | 0.60 | 0.83±.00 | 0.83±.00 | 0.83±.00 |

Table 6: Strength of attribute–class (A–C) correlation and performance of clustering algorithms.

where $\boldsymbol{A}$ stands for the set of all attributes. Call the quantity *A–C correlation strength*. Table 6 shows several data sets, where those with the lowest A–C correlation strength are at the top and those with highest strength are at the bottom. It is clear that the performance of LCM and UC-LTM-AIC is good when the A-C correlation strength is high. When the A-C correlation strength is low, on the other hand, there is little information about the class variable in the attributes. It is hence unlikely to obtain a cluster partition based on the attributes that matches the true class partition well, regardless of the clustering method that is used. Consequently, both LCM and UC-LTM-AIC have poor performances.

### *8.2. Known Number of Clusters*

We next compare the methods in the setting of a known number of clusters. Here we include several methods that are not model-based and require that the number of clusters be given, namely K-Means, kernel K-Means (kkmeans) and spectral clustering (SC). For all methods, we use the number of classes given in the data as the number of clusters. The results are given in Table 7.

In this setting, UC-LTM-AIC outperformed LCM on 11 of the 30 data sets, whereas LCM outperformed UC-LTM-AIC on 5 data sets. UC-LTM-AIC outperformed CLMNC on 25 data sets, whereas CLMNC outperformed UC-LTM-AIC on 3 data sets. The comparisons of UC-LTM-AIC versus TANBC, K-means, kernel K-Means, and spectral clustering are similar. Overall, UC-LTM-AIC has a performance significantly superior to all the alternative methods.

Note that, as we move from the setting of an unknown number of clusters to the setting of a known number of clusters, the performance of UC-LTM improves on several data sets, such as `autos` and `breast-w`. However, its performance degrades on several other data sets, such as `glass2` and `heart-c`. This is probably due to the fact that the some classes in the ground-truth class partition have a very small number of samples. When the number of clusters is unknown, UC-LTM would group them together with large clusters. When the number of clusters is given, on the other hand, UC-LTM would tend to balance the sizes of all clusters and therefore produce an inferior partition. Take the data set `heart-c` as an example. There are five true clusters with sizes 160, 136, 0, 0 and 0. When the number of clusters is not given, UC-LTM produces two clusters, which is ideal given that there are only two non-empty true clusters. When the number of clusters is set to 5, on the other hand, UC-LTM partitions the data into five clusters of sizes 93, 56, 55, 23, 69. The clusters are now more balanced in size, but the partition differs more from the true partition than the partition for the setting of an unknown number of clusters.

| | K-means | kkmeans | SC | LCM | CLMNC | TANBC | UC-LTM-BIC | UC-LTM-AIC |
|---|---|---|---|---|---|---|---|---|
| australian | .30±.00 | .03±.02 | .07±.01 | .16±.00 | .28±.06 | .29±.05 | .31±.00 | **.44±.00** |
| autos | .36±.03 | .23±.02 | .25±.04 | .36±.02 | .29±.03 | .28±.04 | .26±.01 | **.37±.02** |
| breast-cancer | .00±.00 | .03±.01 | .05±.02 | **.09±.00** | .02±.03 | .02±.02 | **.09±.00** | **.09±.00** |
| breast-w | .83±.00 | .44±.10 | .83±.00 | **.85±.00** | .80±.02 | .79±.02 | **.85±.00** | **.85±.00** |
| corral | .19±.00 | .13±.05 | **.37±.05** | .19±.00 | .21±.14 | .15±.04 | .19±.00 | .19±.00 |
| credit-a | .24±.00 | .01±.01 | .02±.00 | .15±.00 | .26±.08 | **.29±.04** | .10±.01 | .13±.05 |
| credit-g | **.03±.00** | .02±.03 | .00±.00 | .01±.00 | .00±.00 | .00±.00 | .01±.00 | .01±.00 |
| diabetes | .08±.00 | .09±.05 | .11±.03 | .09±.00 | .09±.03 | .08±.04 | **.16±.00** | **.16±.00** |
| flare | .02±.00 | **.05±.02** | **.05±.00** | **.05±.00** | .04±.03 | .04±.04 | **.05±.00** | **.05±.00** |
| glass | .44±.00 | .35±.03 | .37±.06 | **.48±.01** | .46±.02 | .44±.01 | .43±.01 | .46±.01 |
| glass2 | .15±.00 | .08±.10 | .13±.07 | .20±.00 | .16±.11 | **.21±.06** | .20±.00 | .20±.00 |
| heart-c | .26±.00 | .23±.02 | .25±.02 | **.28±.01** | .19±.02 | .21±.02 | .26±.01 | .21±.00 |
| heart-statlog | .34±.00 | .32±.02 | .34±.00 | .30±.00 | .25±.02 | .26±.03 | **.35±.00** | **.35±.00** |
| hypothyroid | .05±.08 | .08±.02 | .11±.06 | .22±.00 | .04±.03 | .04±.04 | **.25±.00** | **.25±.00** |
| ionosphere | .11±.00 | .26±.01 | .04±.00 | .48±.00 | .48±.08 | .48±.06 | **.54±.01** | **.54±.01** |
| iris | .76±.07 | .59±.14 | **.83±.00** | **.83±.00** | .74±.20 | .82±.09 | **.83±.00** | **.83±.00** |
| kr-vs-kp | .00±.00 | .00±.01 | .00±.00 | .00±.00 | .00±.00 | .00±.00 | .00±.00 | .00±.00 |
| lymph | .23±.00 | .09±.02 | .07±.01 | .23±.02 | .18±.05 | .21±.04 | **.30±.01** | .24±.00 |
| mofn-3-7-10 | **.06±.00** | .05±.03 | .01±.00 | **.06±.00** | .04±.03 | .04±.04 | **.06±.00** | **.06±.00** |
| mushroom | .15±.00 | .09±.05 | .04±.00 | **.48±.00** | .03±.05 | .03±.04 | **.48±.00** | **.48±.00** |
| pima | .08±.00 | .07±.06 | .09±.03 | .09±.00 | .05±.04 | .07±.03 | **.16±.00** | **.16±.00** |
| segment | .59±.02 | .64±.04 | **.72±.03** | .65±.03 | .50±.08 | .44±.07 | .65±.06 | .67±.05 |
| shuttle-small | .30±.04 | .30±.03 | **.54±.08** | .41±.01 | .30±.08 | .35±.08 | .50±.03 | .50±.03 |
| sonar | .32±.00 | .33±.03 | **.35±.00** | .31±.00 | .24±.06 | .28±.05 | .27±.00 | .27±.00 |
| soybean | .66±.01 | .62±.04 | .68±.03 | **.76±.03** | .60±.03 | .62±.04 | **.76±.01** | **.76±.01** |
| vehicle | .11±.00 | .19±.02 | .21±.04 | .21±.00 | **.22±.04** | .20±.04 | .20±.01 | .20±.01 |
| vote | .54±.00 | .50±.03 | .51±.00 | .51±.00 | .39±.16 | .33±.04 | **.58±.01** | **.58±.01** |
| vowel | .20±.01 | .23±.03 | .22±.03 | .22±.03 | .19±.02 | .20±.02 | **.26±.01** | **.26±.01** |
| waveform-21 | **.37±.00** | .36±.01 | .36±.00 | **.37±.00** | .25±.04 | .31±.04 | **.37±.00** | **.37±.00** |
| zoo | .85±.02 | .14±.01 | .18±.07 | **.86±.00** | .70±.11 | .73±.07 | **.86±.00** | **.86±.00** |
| win/tie/loss | 4/4/22 | 3/2/25 | 6/3/21 | 5/14/11 | 3/2/25 | 3/3/24 | 2/23/5 | - |
| p-value | 0.0012 | 0.0000 | 0.0038 | 0.0442 | 0.0000 | 0.0001 | 0.4469 | - |

Table 7: Performance of UC-LTM and alternative methods on 30 real-world data sets in the setting of a known number of clusters. See Table 5 for the explanation of the notation.

| Attribute | Description | Percentage |
|---|---|---|
| `fair` | People are fair (rather than try to take advantage of others) | 59.72 |
| `trust` | People can generally be trusted | 39.87 |
| `church` | Membership in church group | 34.58 |
| `farm` | Membership in farm organization | 3.71 |
| `fraternal` | Membership in fraternal group | 9.40 |
| `greek` | Membership in school fraternity | 4.80 |
| `hobby` | Membership in hobby club | 9.31 |
| `literary` | Membership in literary or art group | 8.78 |
| `nationality` | Membership in nationality group | 3.30 |
| `other` | Membership in any other group | 10.43 |
| `political` | Membership in political club | 3.97 |
| `professional` | Membership in professional society | 14.70 |
| `school` | Membership in school service | 13.08 |
| `service` | Membership in service group | 9.72 |
| `sport` | Membership in sports club | 19.50 |
| `union` | Membership in labor union | 13.20 |
| `veteran` | Membership in veteran group | 7.04 |
| `youth` | Membership in youth group | 9.48 |

Table 8: Summary of the social capital data. The data set consists of 18 attributes and 14,527 samples. All attributes are binary. The last column shows the percentage of "Yes" values for each attribute.

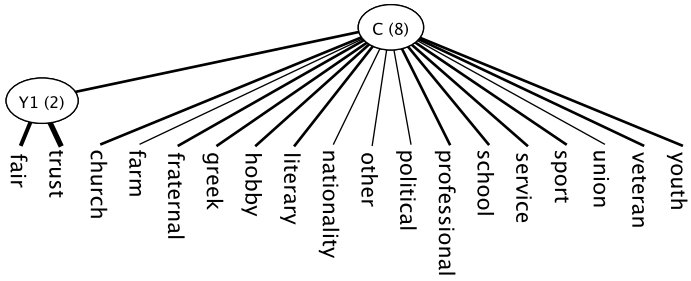



Figure 6: Model obtained from UD-LTM-AIC on the social capital data.

## 9. Unidimensional Clustering on Social Capital Data

*Social capital* is a useful concept in the economics literature. The number of journal articles using the term "social capital" has multiplied rapidly in the last two decades [42]. However, Owen and Videras [43] note that many researchers consider the concept ill-defined and imperfectly measured. Therefore, they propose measuring social capital using LCMs. In this section, we apply UC-LTM to the same social capital data set and aim to demonstrate its advantages over LCMs.

The data for measuring social capital were collected from the General Social Survey (GSS).[1] The data set contains responses from 14,527 individuals to 18 questions that are considered common proxies for social capital. Sixteen questions asked whether the respondent was a member of various kinds of voluntary organizations. The other two questions asked whether the respondents thought other people could generally be trusted and whether other people were fair (rather than tried to take advantage of others). All responses are binary. The attributes are summarized in Table 8.

The model we obtained with UC-LTM-AIC on the data set is shown in Figure 6. Compared with an LCM, our model contains an additional latent variable as the parent of the two attributes that indicate how

[1]We thank Ann Owen for sharing with us the data she used in her work. The data were originally provided by Bruce Sacerdote.

| Model | #Classes | LL | AIC | BIC | #Parameters |
|---|---|---|---|---|---|
| UD-LTM-AIC | 8 | **-87093** | **-87240** | **-87798** | 147 |
| LCM | 8 | -87322 | -87473 | -88045 | 151 |
| LCM (1 direct effect) | 8 | -87273 | -87425 | -88002 | 152 |

Table 9: Comparison of the model obtained by UD-LTM-AIC and the two models obtained by Owen and Videras [43].

| Attribute | Description | Cardinality |
|---|---|---|
| `education` | Years of education | 22 |
| `income` | Annual income | 14 |
| `black` | Black if value is 1 | 2 |
| `female` | Female if value is 1 | 2 |
| `married` | Married if value is 1 | 3 |
| `east` | In east if value is 1 | 2 |
| `south` | In south if value is 1 | 2 |
| `west` | In west if value is 1 | 2 |
| `young` | Between 18 and 29 if value is 1 | 2 |
| `thirties` | Between 30 and 39 if value is 1 | 2 |
| `forties` | Between 40 and 49 if value is 1 | 2 |
| `baptist` | Baptist if value is 1 | 2 |
| `methodist` | Methodist if value is 1 | 2 |
| `lutheran` | Lutheran if value is 1 | 2 |
| `presbyterian` | Presbyterian if value is 1 | 2 |
| `episcopalian` | Episcopalian if value is 1 | 2 |
| `oprot` | Belonging to other Protestant religion if value is 1 | 2 |
| `nondp` | Non-denominational Protestant if value is 1 | 2 |
| `jewish` | Jewish if value is 1 | 2 |
| `catholic` | Catholic if value is 1 | 2 |
| `reloth` | Having other religious affiliation if value is 1 | 2 |

Table 10: Additional attributes for the social capital data. These are used as explanatory variables for the clustering variable in the LCM of Owen and Videras [43]. The binary attributes can take a value of 0 or 1. The possible values of the other attributes are shown in Table 11.

respondents thought about others. The model structure looks reasonable. Owen and Videras [43] obtained an LCM with eight classes in their analysis. They also tried to model the local dependence between the two attributes with the largest bivariate residual. The resulting model was constructed based on the eight-class LCM but includes one direct effect between the attributes `fraternal` and `greek`. Table 9 compares these two models obtained by Owen and Videras and the model obtained by UC-LTM-AIC. Our model produces eight clusters as the other two models did. However, our model fits the data better in terms of log likelihood (LL), the BIC score, and the AIC score. Our model also contains fewer parameters. The results show that it is useful to model local dependence and that LC-LTM-AIC is more effective in doing so than including a direct effect in an LCM.

In their study, Owen and Videras [43] included additional attributes as explanatory variables in a regression model for predicting the clustering variable. Some of these attributes are listed in Table 10. In our next experiment, we used these explanatory attributes in the same model for unidimensional clustering. Specifically, we ran `AttributeGrouping` on the additional attributes to extract the latent features. Then, we extended the model obtained by UC-LTM-AIC on the original attributes (Figure 6) with those latent features and additional attributes. Finally, we learned the cardinality of the root variable as in an unbalanced 3L-LTM. We constructed the model in this way for the following reason. The membership indicators serve as common proxies for social capital, and we want to cluster primarily based on them to obtain a clustering related to social capital. At the same time, we want the clustering to consider also the background of the

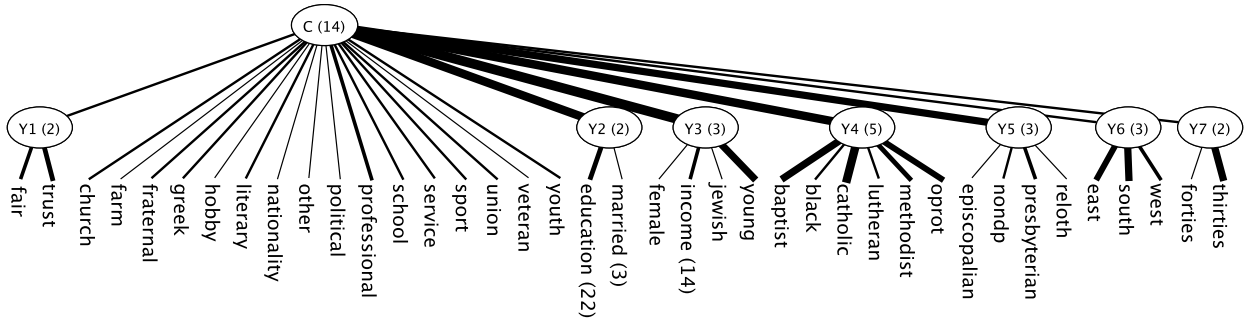



Figure 7: An unbalanced 3L-LTM obtained from the social capital data with additional attributes. The model is an extension of the one in Figure 6. It clusters primarily on the membership indicators but with consideration of the supplementary information from the explanatory attributes. The edge width indicates the strength of the probabilistic dependency between two vertices in terms of mutual information.

individuals as observed from the additional explanatory attributes.

Figure 7 shows the model that we obtained. The model contains seven latent features, $Y_1$–$Y_7$. Table 11 shows the probability tables of their child attributes conditional on those latent features. Note that the latent states are labeled based on our own interpretation. The conditional probability tables show that the latent features are meaningful. For example, $Y_1$ indicates whether a respondent thinks other people are fair and trustworthy. The model shows that 55% of respondents are more suspicious and the other 45% of respondents are more trusting. $Y_2$ divides respondents into two groups based on their education level. Most of the respondents in the first group have 12 or fewer years of education, whereas most in the second group have 13 or more years of education. Whether one is married is less important in the grouping. $Y_3$ classifies respondents into three groups. The first group has lower income, the second group has higher income, and the third group consists of only young people. Similarly, $Y_4$ and $Y_5$ group respondents based on their religion, and $Y_6$ based on their location. $Y_7$ indicates whether or not respondents are in their thirties.

The clustering variable of the model obtained gives 14 classes. The probabilities of attribute values conditional on the 14 classes are listed in Table 12. A cell is shaded in red if its conditional probability value is higher than the unconditional probability; it is shaded in blue if lower. From the table, we see that individuals in the first four classes have low probabilities of membership in most types of organizations. In addition, they are less trusting and have lower levels of education. They can be considered to have a low level of social capital. Classes 1, 3, and 4 tend to have lower-income individuals and more females, and they comprise mostly Catholics and Protestants. Class 2 consists only of young people.

Individuals in class 5 have relatively high probabilities of membership in labor unions and veteran groups. They are less trusting and less educated but have higher income. Those in class 6 have relatively low probabilities of membership in most groups but are more educated and have higher income. Classes 7 and 8 are similar to class 6. However, class 7 comprises young people only, and they have a higher level of membership in sport clubs. Class 8 comprises individuals of some other religions such as Catholicism and other Protestant religions.

Individuals in classes 9 and 10 are more trusting and have relatively high probabilities of membership in many groups. Those in class 9 are less educated, whereas those in class 10 are more educated, tend to have higher income, and have higher membership in professional societies, school fraternities, and school service organizations. Classes 11 and 12 consist of less trusting individuals with higher probabilities of membership in church groups, sport clubs, and youth groups. Those in class 11 are young people, and many in class 12 are in their thirties. Individuals in the classes 13 and 14 are more trusting and more educated and have higher income and have high probabilities of membership in most groups. They are considered to have a high level of social capital. The two classes consist of individuals of different religions.

The above classification shows that the model obtained by UC-LTM can provide an interesting clustering on the social capital data. The model clusters individuals not only based on their membership of various

| | | $Y_1$ | |
|---|---|---|---|
| | | suspicious | trusting |
| class size | | 0.55 | 0.45 |
| `fair` | YES | 0.36 | 0.89 |
| `trust` | YES | 0.08 | 0.80 |

(a) $Y_1$

| | | $Y_7$ | |
|---|---|---|---|
| | | others | thirties |
| class size | | 0.77 | 0.23 |
| `forties` | 1 | 0.21 | 0.00 |
| `thirties` | 1 | 0.00 | 1.00 |

(b) $Y_7$

| | | $Y_4$ | | | | |
|---|---|---|---|---|---|---|
| | | others | baptist | methodist | oprot | catholic |
| class size | | 0.30 | 0.21 | 0.10 | 0.14 | 0.24 |
| `baptist` | 1 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 |
| `black` | 1 | 0.06 | 0.36 | 0.12 | 0.14 | 0.04 |
| `catholic` | 1 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |
| `lutheran` | 1 | 0.23 | 0.00 | 0.00 | 0.00 | 0.00 |
| `methodist` | 1 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 |
| `oprot` | 1 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 |

(c) $Y_4$

| | | $Y_6$ | | |
|---|---|---|---|---|
| | | E | W | others |
| class size | | 0.25 | 0.39 | 0.37 |
| `east` | 1 | 0.82 | 0.00 | 0.00 |
| `south` | 1 | 0.00 | 0.88 | 0.00 |
| `west` | 1 | 0.00 | 0.00 | 0.49 |

(d) $Y_6$

| | | $Y_2$ | |
|---|---|---|---|
| | | less educated | educated |
| class size | | 0.62 | 0.38 |
| `education` | 0 | 0.00 | 0.00 |
| | 1 | 0.00 | 0.00 |
| | 2 | 0.00 | 0.00 |
| | 3 | 0.01 | 0.00 |
| | 4 | 0.01 | 0.00 |
| | 5 | 0.01 | 0.00 |
| | 6 | 0.02 | 0.00 |
| | 7 | 0.03 | 0.00 |
| | 8 | 0.09 | 0.00 |
| | 9 | 0.06 | 0.00 |
| | 10 | 0.09 | 0.00 |
| | 11 | 0.10 | 0.01 |
| | 12 | 0.40 | 0.20 |
| | 13 | 0.07 | 0.09 |
| | 14 | 0.06 | 0.14 |
| | 15 | 0.02 | 0.08 |
| | 16 | 0.02 | 0.26 |
| | 17 | 0.00 | 0.07 |
| | 18 | 0.00 | 0.07 |
| | 19 | 0.00 | 0.03 |
| | 20 | 0.00 | 0.04 |
| | MISSING | 0.00 | 0.00 |
| `married` | 0 | 0.46 | 0.38 |
| | 1 | 0.54 | 0.62 |
| | MISSING | 0.00 | 0.00 |

(e) $Y_2$

| | | $Y_5$ | | |
|---|---|---|---|---|
| | | none | presbyterian | others |
| class size | | 0.76 | 0.05 | 0.20 |
| `episcopalian` | 1 | 0.00 | 0.00 | 0.12 |
| `nondp` | 1 | 0.00 | 0.00 | 0.20 |
| `presbyterian` | 1 | 0.00 | 1.00 | 0.00 |
| `reloth` | 1 | 0.00 | 0.00 | 0.11 |

(f) $Y_5$

| | | $Y_3$ | | |
|---|---|---|---|---|
| | | poorer | richer | young |
| class size | | 0.38 | 0.40 | 0.23 |
| `female` | 1 | 0.69 | 0.45 | 0.54 |
| `income` | LT $1000 | 0.02 | 0.00 | 0.02 |
| | $1000 TO | 0.05 | 0.00 | 0.04 |
| | $3000 TO | 0.06 | 0.00 | 0.03 |
| | $4000 TO | 0.06 | 0.00 | 0.03 |
| | $5000 TO | 0.06 | 0.00 | 0.03 |
| | $6000 TO | 0.05 | 0.00 | 0.03 |
| | $7000 TO | 0.05 | 0.01 | 0.03 |
| | $8000 TO | 0.07 | 0.02 | 0.06 |
| | $10000 TO | 0.18 | 0.07 | 0.17 |
| | $15000 TO | 0.10 | 0.10 | 0.13 |
| | $20000 TO | 0.08 | 0.12 | 0.11 |
| | $25000 O | 0.09 | 0.63 | 0.26 |
| | REFUSED | 0.05 | 0.04 | 0.02 |
| | MISSING | 0.07 | 0.01 | 0.05 |
| `jewish` | 1 | 0.01 | 0.03 | 0.01 |
| `young` | 1 | 0.03 | 0.00 | 1.00 |

(g) $Y_3$

Table 11: Conditional probability tables for the attributes of the latent features $Y_1$–$Y_7$ in Figure 7. In each table, the first two rows show the parent variable and its states. The third row shows the class sizes (i.e., marginal probabilities) for the parent variable. The first two columns show the child variables and their states. When a child variable has only two states, only one state is shown since the value of the other state can be derived easily. The states of the parent variables are labeled based on our interpretation.

| | | C | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| | class size | .08 | .08 | .18 | .10 | .06 | .08 | .05 | .09 | .07 | .04 | .04 | .04 | .05 | .04 |
| fair | YES | .55 | .44 | .44 | .55 | .53 | .72 | .69 | .77 | .74 | .70 | .50 | .54 | .78 | .72 |
| trust | YES | .33 | .19 | .19 | .34 | .31 | .56 | .53 | .64 | .59 | .54 | .27 | .32 | .64 | .57 |
| church | YES | .12 | .02 | .38 | .21 | .10 | .18 | .12 | .39 | .64 | .52 | .62 | .77 | .58 | .77 |
| farm | YES | .00 | .01 | .02 | .03 | .02 | .03 | .02 | .01 | .10 | .05 | .06 | .10 | .08 | .12 |
| fraternal | YES | .00 | .00 | .02 | .07 | .14 | .06 | .05 | .10 | .26 | .16 | .03 | .05 | .29 | .36 |
| greek | YES | .00 | .00 | .00 | .00 | .00 | .03 | .12 | .06 | .02 | .17 | .03 | .02 | .22 | .32 |
| hobby | YES | .02 | .02 | .02 | .04 | .04 | .08 | .11 | .11 | .14 | .12 | .17 | .18 | .30 | .32 |
| literary | YES | .00 | .00 | .01 | .02 | .00 | .09 | .10 | .12 | .11 | .07 | .15 | .11 | .38 | .49 |
| nationality | YES | .02 | .00 | .00 | .01 | .04 | .02 | .04 | .04 | .03 | .05 | .04 | .03 | .12 | .17 |
| other | YES | .07 | .04 | .05 | .09 | .04 | .18 | .10 | .12 | .21 | .09 | .14 | .09 | .18 | .21 |
| political | YES | .00 | .00 | .00 | .02 | .02 | .03 | .03 | .03 | .07 | .05 | .04 | .04 | .16 | .26 |
| professional | YES | .00 | .00 | .01 | .01 | .00 | .26 | .34 | .33 | .02 | .44 | .10 | .04 | .52 | .67 |
| school | YES | .03 | .03 | .04 | .03 | .03 | .09 | .07 | .20 | .03 | .21 | .23 | .59 | .36 | .56 |
| service | YES | .00 | .00 | .01 | .02 | .05 | .04 | .09 | .09 | .17 | .16 | .10 | .11 | .43 | .56 |
| sport | YES | .02 | .14 | .02 | .09 | .20 | .21 | .36 | .30 | .09 | .22 | .49 | .45 | .42 | .57 |
| union | YES | .06 | .11 | .04 | .09 | .58 | .15 | .11 | .13 | .15 | .04 | .14 | .16 | .10 | .20 |
| veteran | YES | .03 | .01 | .02 | .07 | .17 | .03 | .02 | .05 | .23 | .13 | .03 | .05 | .12 | .15 |
| youth | YES | .00 | .01 | .01 | .03 | .03 | .04 | .02 | .08 | .03 | .07 | .38 | .52 | .26 | .52 |
| education | 0–12 | .83 | .80 | .83 | .83 | .80 | .22 | .22 | .22 | .83 | .22 | .62 | .71 | .22 | .22 |
| | 13–20 | .16 | .20 | .16 | .17 | .20 | .78 | .78 | .78 | .16 | .78 | .37 | .29 | .78 | .78 |
| married | 1 | .54 | .54 | .54 | .54 | .54 | .62 | .62 | .62 | .54 | .62 | .57 | .56 | .62 | .62 |
| female | 1 | .67 | .54 | .67 | .62 | .46 | .46 | .54 | .45 | .63 | .48 | .54 | .56 | .48 | .49 |
| income | LT $15000 | .56 | .43 | .56 | .49 | .11 | .12 | .43 | .10 | .46 | .18 | .43 | .32 | .18 | .21 |
| | $15000 O | .32 | .50 | .32 | .41 | .84 | .83 | .50 | .85 | .43 | .77 | .50 | .60 | .76 | .73 |
| jewish | 1 | .01 | .01 | .01 | .01 | .03 | .03 | .01 | .03 | .01 | .03 | .01 | .02 | .03 | .03 |
| young | 1 | .02 | 1.00 | .10 | .23 | .00 | .00 | 1.00 | .00 | .02 | .24 | 1.00 | .01 | .15 | .18 |
| baptist | 1 | .00 | .26 | .62 | .00 | .23 | .00 | .01 | .05 | .15 | .54 | .27 | .34 | .00 | .15 |
| black | 1 | .06 | .14 | .27 | .06 | .14 | .06 | .06 | .09 | .13 | .25 | .17 | .18 | .06 | .13 |
| catholic | 1 | .73 | .36 | .00 | .00 | .43 | .00 | .40 | .61 | .26 | .05 | .26 | .21 | .00 | .35 |
| lutheran | 1 | .02 | .04 | .00 | .23 | .03 | .23 | .11 | .00 | .03 | .00 | .01 | .02 | .23 | .02 |
| methodist | 1 | .10 | .08 | .13 | .00 | .11 | .00 | .05 | .13 | .25 | .32 | .10 | .09 | .00 | .21 |
| oprot | 1 | .09 | .12 | .25 | .00 | .10 | .00 | .05 | .21 | .22 | .10 | .31 | .28 | .00 | .22 |
| episcopalian | 1 | .00 | .00 | .00 | .10 | .00 | .11 | .04 | .00 | .00 | .00 | .00 | .00 | .09 | .00 |
| nondp | 1 | .00 | .00 | .00 | .16 | .00 | .18 | .06 | .00 | .00 | .00 | .00 | .00 | .14 | .00 |
| presbyterian | 1 | .00 | .00 | .00 | .21 | .00 | .11 | .03 | .00 | .00 | .00 | .00 | .00 | .31 | .00 |
| reloth | 1 | .00 | .00 | .00 | .08 | .00 | .09 | .03 | .00 | .00 | .00 | .00 | .00 | .07 | .00 |
| east | 1 | .43 | .21 | .06 | .22 | .28 | .22 | .29 | .33 | .14 | .01 | .15 | .11 | .23 | .17 |
| south | 1 | .17 | .35 | .69 | .26 | .18 | .15 | .15 | .12 | .23 | .83 | .36 | .45 | .29 | .30 |
| west | 1 | .14 | .17 | .07 | .22 | .23 | .28 | .24 | .23 | .28 | .03 | .20 | .18 | .20 | .22 |
| forties | 1 | .16 | .21 | .17 | .19 | .14 | .10 | .21 | .10 | .21 | .17 | .21 | .08 | .17 | .16 |
| thirties | 1 | .25 | .00 | .20 | .13 | .35 | .52 | .00 | .51 | .00 | .21 | .00 | .62 | .22 | .25 |

Table 12: Probabilities of attribute values conditional on the clustering variable. The clustering variable $C$ has 14 states. The third row lists the sizes of the 14 classes. A cell is shaded in red (blue) if its conditional probability value is higher (lower) than the unconditional probability. For brevity, some states are omitted and some states are aggregated. The attributes are divided into eight groups in accordance with the model structure.

groups, but also taking into account other factors such as their education level, income level, religion, and whether they are young.

## 10. Concluding Remarks

When performing cluster analysis on discrete data using latent class models (LCMs), local dependence is an issue that should not be ignored. A method for detecting and modeling local dependence called UC-LTM has been proposed in this paper. In empirical studies, UC-LTM outperforms LCM in most cases, especially in the setting of an unknown number of clusters. The improvements are often large (exceeding 10%). In the setting of a known number of clusters, UC-LTM is also superior to popular distance/similarity-based methods. In the experiment on social capital data, the model given by UC-LTM has better quality than the LCM. UC-LTM can also produce interesting clustering based on the social capital indicators and other factors at the same time.

It would be interesting to carry out similar research for continuous data. To do so, one can either make no independence assumptions at all and work with full covariance matrices, or make the same independence assumption as in LCMs and work with diagonal covariance matrices. There have already been efforts described in the literature to explore middle grounds between the two extremes by working with block-diagonal covariances. The concept of unidimensionality testing and the procedure for dividing attributes into unidimensional clusters from this paper can be helpful in further work in that direction.

## Acknowledgment

Research on this article was supported in part by The Education University of Hong Kong under project RG90/2014-2015R and the Hong Kong Research Grants Council under grant 16202515. We are grateful for the comments and suggestions given by the anonymous reviewers.

## References

[1] A. H. Liu, L. K. M. Poon, N. L. Zhang, Unidimensional Clustering of Discrete Data Using Latent Tree Models, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2771–2777, 2015.
[2] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A Survey of Kernel and Spectral Methods for Clustering, Pattern Recognition 41 (1) (2008) 176–190.
[3] G. J. McLachlan, D. Peel, Finite Mixture Models, Wiley, New York, 2000.
[4] D. J. Bartholomew, M. Knott, Latent Variable Models and Factor Analysis, Arnold, 2nd edn., 1999.
[5] L. M. Collins, S. T. Lanza, Latent Class and Latent Transition Analysis: With Applications in the Social Behavioral, and Health Sciences, Wiley, 2010.
[6] E. S. Garrett, S. L. Zeger, Latent Class Model Diagnosis, Biometrics 56 (4) (2000) 1055–1067.
[7] K. J. Vermunt, J. Magidson, Latent Class Cluster Analysis, in: Applied Latent Class Analysis, Cambridge University Press, 89–106, 2002.
[8] H. Akaike, A New Look at the Statistical Model Identification, IEEE Transactions on Automatic Control 19 (6) (1974) 716–723.
[9] G. Schwarz, Estimating the Dimension of a Model, The Annals of Statistics 6 (2) (1978) 461–464.
[10] N. L. Zhang, Hierarchical Latent Class Models for Cluster Analysis, Journal of Machine Learning Research 5 (2004) 697–723.
[11] R. Mourad, C. Sinoquet, N. L. Zhang, T. Liu, P. Leray, A Survey on Latent Tree Models and Applications, Journal of Artificial Intelligence Research 47 (2013) 157–203.
[12] N. L. Zhang, L. K. M. Poon, Latent Tree Analysis, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 4891–4897, 2017.
[13] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, San Mateo, California, 1988.
[14] K. P. Burnham, D. R. Anderson, Multimodel inference: understanding AIC and BIC in model selection, Sociological Methods and Research 33 (2) (2004) 261–304.
[15] S. I. Vrieze, Model Selection and Psychological Theory: A Discussion of the Differences between the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC), Psychological Methods 17 (2) (2012) 228–243.
[16] K. Aho, D. Derryberry, T. Peterson, Model Selection for Ecologists: The Worldviews of AIC and BIC, Ecology 95 (3) (2014) 631–636.

[17] T. Chen, N. L. Zhang, T. Liu, K. M. Poon, Y. Wang, Model-Based Multidimensional Clustering of Categorical Data, Artificial Intelligence 176 (2012) 2246–2269.
[18] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum Likelihood from Incomplete Data Via the EM Algorithm, Journal of the Royal Statistical Society. Series B (Methodological) 39 (1) (1977) 1–38.
[19] T.-F. Liu, N. L. Zhang, P. Chen, A. H. Liu, L. K. Poon, Y. Wang, Greedy Learning of Latent Tree Models for Multidimensional Clustering, Machine Learning 98 (1–2) (2015) 301–330.
[20] R. E. Kass, A. E. Raftery, Bayes factor, Journal of American Statistical Association 90 (430) (1995) 773–795.
[21] A. E. Raftery, Bayesian Model Selection in Social Research, Sociological Methodology 25 (1995) 111–163.
[22] T. M. Cover, J. A. Thomas, Elements of Information Theory, Wiley, 2006.
[23] M. Lichman, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, http://archive.ics.uci.edu/ml, 2010.
[24] Y. Wang, N. L. Zhang, T. Chen, Latent Tree Models and Approximate Inference in Bayesian Networks, Journal of Artificial Intelligence Research 32 (2008) 879–900.
[25] S. Harmeling, C. K. I. Williams, Greedy Learning of Binary Latent Trees, IEEE Transaction on Pattern Analysis and Machine Intelligence 33 (6) (2011) 1087–1097.
[26] P. Chen, N. L. Zhang, T. Liu, L. K. M. Poon, Z. Chen, F. Khawar, Latent Tree Models for Hierarchical Topic Detection, Artificial Intelligence 250 (2017) 105–124.
[27] M. J. Choi, V. Y. F. Tan, A. Anandkumar, A. S. Willsky, Learning Latent Tree Graphical Models, Journal of Machine Learning Research 12 (2011) 1771–1812.
[28] A. Anandkumar, K. Chaudhuri, D. Hsu, S. M. Kakade, L. Song, T. Zhang, Spectral Methods for Learning Multivariate Latent Tree Structure, in: Advances in Neural Information Processing Systems 24, 2025–2033, 2012.
[29] L. Song, H. Liu, A. Parikh, E. Xing, Nonparametric Latent Tree Graphical Models: Inference, Estimation, and Structure Learning, Journal of Machine Learning Research 12 (2017) 663–707.
[30] L. K. M. Poon, N. L. Zhang, T. Liu, A. H. Liu, Model-Based Clustering of High-Dimensional Data: Variable Selection versus Facet Determination, International Journal of Approximate Reasoning 54 (1) (2013) 196–215.
[31] L. K. M. Poon, Clustering with Multidimensional Mixture Models: Analysis on World Development Indicators, in: Advances in Neural Network (ISNN 2017), 2017.
[32] N. Asbeh, B. Lerner, Learning Latent Variable Models by Pairwise Cluster Comparison: Part I – Theory and Overview, Journal of Machine Learning Research 17 (224) (2016) 1–52.
[33] N. Asbeh, B. Lerner, Learning Latent Variable Models by Pairwise Cluster Comparison: Part II – Algorithm and Evaluation, Journal of Machine Learning Research 17 (233) (2016) 1–45.
[34] D. T. Pham, G. A. Ruz, Unsupervised Training of Bayesian Networks for Data Clustering, Proceedings of The Royal Society A 465 (2109) (2009) 2927–2948.
[35] J. Peña, J. Lozano, P. Larrañaga, Learning Bayesian Networks for Clustering by Means of Constructive Induction, Pattern Recognition Letters 20 (1999) 1219–1230.
[36] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian Network Classifiers, Machine Learning 29 (1997) 131–163.
[37] G. Santafé, J. A. Lozano, P. Larrañaga, Bayesian Model Averaging of Naive Bayes for Clustering, IEEE Tractions on Systems, Man, and Cybernetics—Part B: Cybernetics 36 (5) (2006) 1149–1161.
[38] G. Santafé, J. A. Lozano, P. Larrañaga, Bayesian Model Averaging of TAN Models for Clustering, in: Proceedings of Europen Workshop on Probabilistic Graphical Models, 2006.
[39] S. Zhong, J. Ghosh, A Unified Framework for Model-Based Clustering, Journal of Machine Learning Research 4 (2003) 1001–1037.
[40] U. M. Fayyad, K. B. Irani, Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, in: Thirteenth International Joint Conference on Articial Intelligence, vol. 2, 1022–1027, 1993.
[41] J. Demšar, Statistical Comparisons of Classifiers over Multiple Data Sets, Journal of Machine Learning Research 7 (2006) 1–30.
[42] S.-W. Kwon, P. S. Adler, Social Capital: Maturation of a Field of Research, Academy of Management Review 39 (4) (2014) 412–422.
[43] A. L. Owen, J. Videras, Reconsidering Social Capital: A Latent Class Approach, Empirical Economics 37 (3) (2009) 555–582.