Unidimensional Clustering of Discrete Data Using Latent Tree Models

April H. Liu[1] and **Leonard K.M. Poon**[2] and **Nevin L. Zhang**[1]
[1] Department of Computer Science and Engineering
The Hong Kong University of Science and Technology, Hong Kong
{aprillh, lzhang}@cse.ust.hk
[2] Department of Mathematics and Information Technology
The Hong Kong Institute of Education, Hong Kong
kmpoon@ied.edu.hk

Abstract

This paper is concerned with model-based clustering of discrete data. Latent class models (LCMs) are usually used for the task. An LCM consists of a latent variable and a number of attributes. It makes the overly restrictive assumption that the attributes are mutually independent given the latent variable. We propose a novel method to relax the 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 one 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. Extensive empirical studies have been conducted to compare the new method with LCM and several other methods (K-means, kernel K-means and spectral clustering) that are not model-based. The new method outperforms the alternative methods in most cases and the differences are often large.

Introduction

Cluster analysis is a classic research topic in AI. A variety of approaches have been proposed, including distance/similarity-based algorithms such as K-means, kernel K-means and spectral clustering (Filippone et al. 2008), as well as model-based methods such as Gaussian mixture models (GMMs) (McLachlan and Peel 2000) and latent class models (LCMs) (Bartholomew and Knott 1999). While GMMs are used to analyze continuous data, LCMs are used to deal with discrete data.

This paper focuses on LCMs. An LCM consists of a latent variable and a set of discrete attributes (observed variables) that describe the data. 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 mutually independent given the clustering latent variable. In other words, the attributes are mutually independent in each cluster. The assumption is hence referred to as the *local independence* assumption. It is often violated in practice and can lead to spurious clusters (Garrett and Zeger 2000; Vermunt and Magidson 2002).

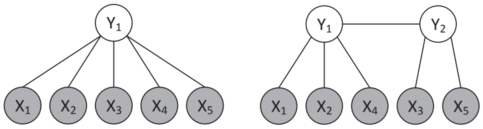
In this paper, we propose a novel method to relax the local independence assumption of LCM and to detect and model local dependence properly so as to improve clustering quality. The idea is to: (1) Partition the attributes into groups such that correlations among the attributes in each group can be properly modeled using one single latent variable, and introduce a latent variable for each group; (2) Construct several models for clustering using the latent variables from Step 1; and (3) Select of one the models to produce the final results.

Intuitively, each latent variable introduced in Step 1 can be understood as capturing one aspect of the data.

One of the models constructed in Step 2 uses the latent variables from Step 1 as features and introduces a new latent variable for clustering. It produces a partition of data that is based on all aspects of data evenly 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. They produce partitions of data that predominantly depends on only one aspect of data and hence are called *unbalanced models*. We choose among the different models using a model selection criterion. Both the AIC score (Akaike 1974) and the BIC score (Schwarz 1978) are considered.

All the models contain multiple latent variables that are connected up to form a tree structure. Hence they are special latent tree models (LTMs) (Zhang 2004; Mourad et al. 2014). This paper differs from previous works on LTMs in that one of the latent variables is designated as the clustering variable during model construction. The objective is to model local dependence in LCM so as to improve clustering quality. In contrast, the objective of previous works on LTMs is to optimize fitness to data. None of the latent variables is designated as *the* clustering variable. They are sometimes all interpreted as clustering variables, leading to multiple partitions of data. So, previous works on LTMs aim at finding the best way to cluster data simultaneously along multiple dimensions, while this paper focuses on finding the best way to cluster data along a single dimension.

We will start with a brief review of the basic concepts. In the bulk of the paper, we will describe the new method, first Step 1 and then Steps 2 and 3. After that, we will discuss related works, present empirical results and draw conclusions.



(a) Model m_1 (b) Model m_2

Figure 1: A latent class model and a latent tree model.

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 LTM is shown in Figure 1((b)), where Y_1 and Y_2 are latent variables, while 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 (Pearl 1988). In the example, suppose that we root the model at the node Y_1. Then the numerical information of the model includes 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 and $P(Y_2|Y_1)$ for $Y_1 \rightarrow X_2$).

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 (Schwarz 1978) is usually used to evaluate an LTM m: $BIC(m|D) = \log P(D|m, \theta^*) - \frac{d(m)}{2} \log N$, where D is the data set, θ^* is the maximum likelihood estimate of the parameters, $d(m)$ is the number of free probability parameters in m, and N is the sample size. In this paper, we sometimes also consider the AIC score (Akaike 1974): $AIC(m|D) = \log P(D|m, \theta^*) - d(m)$.

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 at 2 and optimized the probability parameters using the EM algorithm (Dempster, Laird, and Rubin 1977). Then the cardinality is gradually increased and the parameters are re-estimated after each increase. The process stops when model score ceases to increase. The final model is returned as the output. We will refer to this procedure as $\texttt{LearnLCM}(D, f)$, where D is the data set and f is a model scoring function.

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.

Extraction of Latent Features

In this section, we describe Step 1 of the new method. The *unidimensionality test* (shortened the *UD-test*), is a Bayesian statistical test which tests whether correlations among a subset $\boldsymbol{S}$ of attributes can be properly modeled using one single latent variable (Liu et al. 2013). Let m_1 and m_2 be the models with the highest BIC scores among LTMs for $\boldsymbol{S}$ that contain a single latent variable or contain no more than two latent variables respectively. The *UD-test passes* if and only if one of these two conditions is satisfied : (1) m_2 contains only one latent variable, or (2) m_2 contains two latent variables and

$$BIC(m_2|D') - BIC(m_1|D') \leq \delta, \qquad (1)$$

where δ is a threshold parameter. The left hand side of equation (1) is an approximation to the logarithm of the Bayes factor (Kass and Raftery 1995) for comparing m_2 with m_1. For this reason, only the BIC score is used in the UD-test.

In our experiments, the threshold δ is set at 3 as suggested by Kass and Raftery (1995). This means that we would conclude the correlations among a set of attributes can be properly modeled using one single latent variable if there is no strong evidence pointing to the opposite. If the UD-test passes, we say that the set $\boldsymbol{S}$ of attributes is *unidimensional*.

To perform the UD-test in practice, we first project the original data set D onto $\boldsymbol{S}$ to get a smaller data set D'. The model m_1 is obtained using $\texttt{LearnLCM}(D', f)$. The model m_2 is obtained using the EAST algorithm (Chen et al 2012). EAST searches in the space of all LTMs to find the one with the highest BIC score. For UD-test, we restrict the space to contain only LTMs with one or two latent variables.

In the following, we present a method proposed by Liu et al. (2013) for partitioning the attributes in a data set into unidimensional clusters, or *UD clusters* for short. The method relies on mutual information (MI) (Cover and Thomas 1991). 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 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, one adds the attribute that has the highest MI with the current set. 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 one single latent variable. If the UD-test fails, the expansion process stops and the first UD cluster is picked.

To illustrate the process, 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 passes in both cases. Now consider the addition of attribute X_5. Suppose the 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 of BIC scores between m_2 and m_1 exceeds the threshold δ. Then the UD-test fails and it is time to pick the first UD cluster. The 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 partition, an LCM is learned for attributes in each UD cluster using `LearnLCM`(D, f). Suppose there are L variable clusters. Then we get L LCMs. Denote the latent variables in the LCMs as Y_1, Y_2, ..., Y_L. They will be used as features for data clustering. We will refer to the procedure that produces the L latent variables and LCMs as `AttributeGrouping`(D, δ, f), where D is the data set, δ is the threshold for UD-test, and f is the model scoring function used when learning LCMs for the UD clusters.

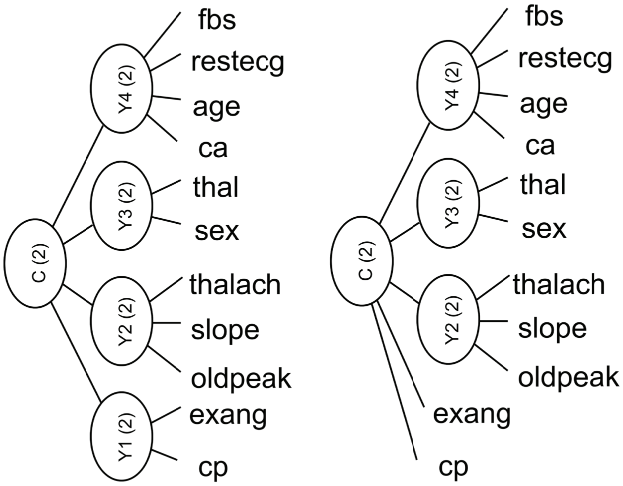
Use of Latent Features for Data Clustering

In this section, we describe Steps 2 and 3 of the new method using an example. We have run `AttributeGrouping` on a version of the `heart-c` data set from the UCI repository (Bache and Lichman 2013). 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 are not given to save space.

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`. The question is how to use the latent variables to cluster the data.

One natural idea is to build an LCM using the four latent variables as features. The 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 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 in 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 there 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`(D, f).



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

Figure 2: The balanced and unbalanced 3L-LTMs learned on the `heart-c` data set. The numbers in parenthesis are the cardinalities of the latent variables estimated by our method.

In addition to the balanced 3L-LTM, we also consider a number of unbalanced models where 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, 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 done using a procedure similar to `LearnLCM`(D, f). 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 those features, we can construct one balanced 3L-LTM, and L unbalanced 3L-LTMs. Among the $L + 1$ models, we pick one best model as the final output. Here we try both BIC and AIC as the criterion for model selection. After the model is learnt, 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 1 shows the pseudo-code for our algorithm. It

Algorithm 1 UC-LTM(D, δ, f)

Input: D - Data, δ - Threshold for UD-test,
f - Model scoring function, either BIC or AIC .
Output: A 3-level latent tree model (3L-LTM).

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

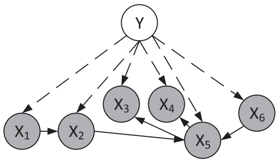


Figure 3: Illustration of the TAN-LCM. The dashed lines are for the LCM, while the solid lines are connections among attributes determined by Chow-Liu's algorithm.

is called *UC-LTM*, which stands for Unidimensional Clustering using Latent Tree Models.

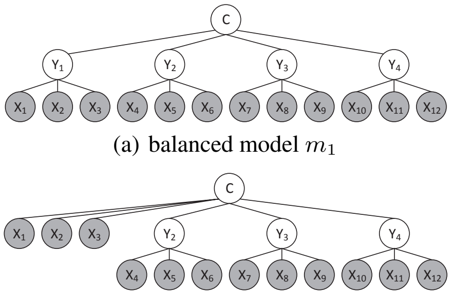
Related Work

Technically LCMs are closely related to the Naive Bayes (NB) model for classification. An LCM can be viewed as a NB model where the class variable is not observed. In the context of NB, a well-known method for relaxing the local independence assumption is the tree-augmented naive (TAN) Bayes (Friedman 1997). The idea is to allow direct connections among the attributes. For the sake of computational efficiency, those connections are assumed to form a tree. The same idea can applied to LCM to relax the local independence assumption. We call this method TAN-LCM.

Figure 3 illustrates the structure of the model that TAN-LCM uses. To build such a model, we first learn a tree model for the attributes using Chow-Liu's algorithm (Chow and Liu 1968). Specifically, we create a complete weighted graph over the attributes with empirical mutual information between attributes as edge weights, and hence find a spanning tree for the graph. After that, we introduce the clustering variable C and connects it to each of the attributes. The cardinality of C, the marginal distribution $P(C)$, and the conditional distribution of each attribute given C are determined using a procedure similar to LearnLCM(D, f).

Table 1: The performance on synthetic data of various methods. The abbreviations b and ub denote data sets generated from the balanced and unbalanced models, respectively, in Figure 4. UC-LTM attains the highest NMI values.

	LCM	Balanced 3L-LTM	UC-LTM-AIC	UC-LTM-BIC
syn-b-5k	.48±.00	**.65±.00**	**.65±.00**	**.65±.00**
syn-b-10k	.48±.00	**.64±.00**	**.64±.00**	**.64±.00**
syn-ub-5k	.15±.00	.25±.01	**.32±.01**	**.32±.01**
syn-ub-10k	.15±.00	.19±.05	**.32±.01**	**.32±.01**



(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.

Empirical Results

In this section we empirically evaluate UC-LTM on both synthetic data and real-world data. Two versions of UC-LTM were used in the experiments, which 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. The real-world data are used to show the benefits of modeling local dependence.

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 those 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*$(C; C_t)$ between the two partitions (Zhong and Ghosh 2003), given by: *NMI*$(C; C_t) = I(C; C_t)/\sqrt{H(C)H(C_t)}$, where $I(C; C_t)$ is the MI between C and C_t and $H(.)$ stands for entropy (Cover and Thomas 1991). 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 meaning a closer match between C and C_t.

Results 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 that the root variable has three. Model parameters were randomly generated. Two data sets were sampled from each model. The sample sizes were 5,000 and

Table 2: The performances of UC-LTM and alternative methods on 30 real-world data sets. UC-LTM-AIC is used as the pivot against which other methods are compared. The result of an alternative method on a data set is underlined if it is worse than that of UC-LTM-AIC, and it is marked blue if the difference exceeds 10%. Bold face fonts and red color are used to indicate the opposite. The last row summarizes the number of data sets on which an alternative method wins, ties or loses compared with UC-LTM-AIC. The ± values indicate the standard deviation. An asterisk indicates that the performance of UC-LTM is improved with the inclusion of unbalanced 3L-LTMs.

	Unknown Number of Clusters				Known Number of Clusters					
	LCM	TAN-LCM	UC-LTM-BIC	UC-LTM-AIC	k-means	kkmeans	specc	LCM	UC-LTM-BIC	UC-LTM-AIC
australian	.16±.00	.16±.00	.31±.00	.44±.00	.30±.00	.03±.02	.07±.01	.16±.00	.31±.00	.44±.00
autos	.21±.01	.08±.04	.17±.00*	.23±.00*	.36±.03	.23±.02	.25±.04	.36±.02	.26±.01	.37±.02
breastcancer	.09±.00	.01±.00	.09±.00	.10±.00*	.00±.00	.03±.01	.05±.02	.09±.00	.09±.00	.09±.00
breast-w	.68±.00	.58±.10	.68±.00	.68±.00	.83±.00	.44±.10	.83±.00	.85±.00	.85±.00	.85±.00
corral	.19±.00	.01±.01	.19±.00	.19±.00	.19±.00	.13±.05	**.37±.05**	.19±.00	.19±.00	.19±.00
credit-a	.11±.02	**.25±.06**	.12±.00	.13±.01	**.24±.00**	.01±.01	.02±.00	**.15±.00**	.10±.01	.13±.05
credit-g	.01±.00	.00±.00	.01±.00	.01±.00	**.03±.00**	**.02±.03**	.00±.00	.01±.00	.01±.00	.01±.00
diabetes	.12±.00	.05±.05	.15±.02*	.15±.02*	.08±.00	.09±.05	.11±.03	.09±.00	.16±.00	.16±.00
flare	.07±.00	.02±.02	.07±.00	.07±.00	.02±.00	.05±.02	.05±.00	.05±.00	.05±.00	.05±.00
glass	.47±.02	.23±.01	.48±.00	.48±.00	.44±.00	.35±.03	.37±.06	**.48±.01**	.43±.01	.46±.01
glass2	.31±.00	.17±.00	.31±.00	.31±.00	.15±.00	.08±.10	.13±.07	.20±.00	.20±.00	.20±.00
heart-c	.30±.00	.29±.01	.29±.00	.33±.00	**.26±.00**	**.23±.02**	**.25±.02**	**.28±.01**	**.26±.01**	.21±.00
heartStatlog	.30±.00	.34±.00	.35±.00	.35±.00	.34±.00	.32±.02	.34±.00	.30±.00	.35±.00	.35±.00
hypothyroid	.18±.00	.00±.00	.21±.00*	.21±.00*	.05±.08	.08±.02	.11±.06	.22±.00	.25±.00	.25±.00
ionosphere	.38±.00	.41±.14	.41±.05	.44±.03	.11±.00	.26±.01	.04±.00	.48±.00	.54±.01	.54±.01
iris	.83±.00	.18±.06	.83±.00	.83±.00	.76±.07	.59±.14	.83±.00	.83±.00	.83±.00	.83±.00
kr-vs-kp	**.06±.01**	.01±.00	.04±.01	.04±.01	.00±.00	.00±.01	.00±.00	.00±.00	.00±.00	.00±.00
lymph	.22±.00	.09±.02	.17±.00	.29±.00	.23±.00	.09±.02	.07±.01	.23±.02	**.30±.01**	.24±.00
mofn3-7-10	.04±.03	.03±.03	.05±.02*	.05±.02*	.06±.00	.05±.03	.01±.00	.06±.00	.06±.00	.06±.00
mushroom	.49±.05	.15±.09	.52±.01	.52±.01	.15±.00	.09±.05	.04±.00	.48±.00	.48±.00	.48±.00
pima	.12±.00	.04±.05	.15±.03*	.15±.03*	.08±.00	.07±.06	.09±.03	.09±.00	.16±.00	.16±.00
segment	**.68±.01**	.17±.05	.63±.03	.63±.03	.59±.02	.64±.04	**.72±.03**	.65±.03	.65±.06	.67±.05
shuttleSmall	.48±.01	.24±.06	.49±.03*	.49±.03*	.30±.04	.30±.03	**.54±.08**	.41±.01	.50±.03	.50±.03
sonar	**.25±.00**	.17±.14	.23±.00	.23±.00	**.32±.00**	**.33±.03**	**.35±.00**	**.31±.00**	.27±.00	.27±.00
soybean	**.66±.02**	.34±.03	.63±.02*	.63±.02*	.66±.01	.62±.04	.68±.03	.76±.03	.76±.01	.76±.01
vehicle	**.31±.01**	.16±.03	.30±.01	.30±.01	.11±.00	.19±.02	**.21±.04**	**.21±.00**	.20±.01	.20±.01
vote	**.43±.00**	.02±.00	.41±.00	.41±.00	.54±.00	.50±.03	.51±.00	.51±.00	.58±.01	.58±.01
vowel	.18±.01	.04±.03	.21±.02	.21±.02	.20±.01	.23±.03	.22±.03	.22±.03	.26±.01	.26±.01
waveform21	.43±.00	.26±.01	.48±.00*	.48±.00*	.37±.00	.36±.01	.36±.00	.37±.00	.37±.00	.37±.00
zoo	.64±.00	.04±.02	.72±.07*	.72±.07*	.85±.02	.14±.01	.18±.07	.86±.00	.86±.00	.86±.00
win/tie/loss	6/6/18	1/0/29	0/23/7	-/-/-	4/4/22	3/2/25	6/3/21	5/14/11	2/23/5	-/-/-

10,000 respectively. Each sample contains 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 significantly 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 two data sets. This shows the benefits of including unbalanced models when there is a predominant aspect in data.

Real-world Data: Unknown Number of Clusters

The real-world data sets were from the UCI machine learning repository. We chose 30 labeled data sets that have been often used in the literature. The data are from various domains such as medical diagnosis, biology, etc. 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. Continuous data were discretized using the method by Fayyad and Irani (1993). The class labels were first removed and then different methods were used to recover the class partition from the resulting unlabeled data. The results are shown in the left half of Table 2. We use UC-LTM-AIC as a pivot and compare it with each of the other methods. For each data set, we check the NMI for an alternative method and UC-LTM-AIC. The NMI of the alternative method is marked bold if it outperforms UC-LTM-AIC, and colored red if the difference exceeds 10%; the value is underlined and colored blue if it is the other way around.

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 have run another version of UC-LTM that uses only balanced models. It turns out that the inclusion of unbalanced models never impact negatively on the performance. It improved the performance on a number of data sets, which are marked with asterisks. The second question is whether the choice of model selection criteria has a significant impact on the performance. We see in the left half of Table 2 that UC-LTM-AIC beats UC-LTM-BIC on 7 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.

Comparisons with Alternative Methods UC-LTM-AIC outperforms LCM on 18 of the 30 data sets (those underlined). The differences exceed 10% on 14 data sets (those colored blue). In contrast, LCM outperforms UC-LTM-AIC on 6 data sets (those bold-faced). The difference exceeds 10% on only 1 data set (the one colored red). Overall, the performance of UC-LTM-AIC is superior to that of LCM. The results indicate that UC-LTM-AIC is beneficial to detect and model local dependence.

The performance of TAN-LCM is worse than UC-LTM-AIC on all but one data set. The differences are often drastic. Intuitively, the clustering of discrete data is based on correlations among the attributes. In TAN-LCM, much of the correlations are explained by the edges among the attributes themselves, which results in poor clustering performance.

A Remark A careful reader might have noticed that while the performances of LCM and UC-LTM are 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`. The phenomenon can be explained 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:

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

where $\boldsymbol{A}$ stands for the set of all attributes. Call the quantity *A-C correlation strength*. Table 3 shows several data sets with either strongest or weakest A-C correlation strength. It is clear that the performances of LCM and UC-LTM-AIC are good when the A-C correlation strength is strong. When the A-C correlation strength is weak, on the other hand, there is little information about the class variable in the attributes. It is hence unlikely to, based on the attributes, obtain a cluster partition that matches the true class partition well, no matter what clustering method is used. Consequently, both LCM and UC-LTM-AIC have poor performances.

Real-world Data: Known Number of Clusters

We next compare the methods in the setting of known number of clusters. Here we include several methods that are not model-based and require the number of clusters be given, namely K-Means, kernel K-Means and spectral clustering. The results are given on the right half of Table 2.

Table 3: Strength of attribute-class correlation and performances of clustering algorithms.

	A-C correlation	LCM	UC-LTM-AIC
credit-g	0.02	0.01	0.01
kr-vs-kp	0.02	0.06	0.04
breastcancer	0.04	0.09	0.10
mofn3-7-10	0.05	0.04	0.05
flare	0.05	0.07	0.07
hypothyroid	0.05	0.18	0.21
diabetes	0.07	0.12	0.15
pima	0.07	0.12	0.15
credit-a	0.09	0.11	0.13
glass	0.30	0.47	0.48
segment	0.33	0.68	0.63
soybean	0.35	0.66	0.63
zoo	0.40	0.64	0.72
breast-w	0.44	0.68	0.68
iris	0.60	0.83	0.83

In this setting, UC-LTM-AIC outperforms LCM on 11 of the 30 data sets (those underlined), and the differences exceed 10% on 10 data sets (those colored blue). On the other hand, LCM outperforms UC-LTM-AIC on 5 data sets (those bold-faced), and the differences exceed 10% on 3 data sets (those colored red). UC-LTM-AIC outperforms K-Means on 22 data sets, and the differences exceed 10% on 14 data sets. On the other hand, K-Means outperforms UC-LTM-AIC on 4 data sets, and the difference exceed 10% on all of them. The comparisons of UC-LTM-AIC versus kernel K-Means and spectral clustering are similar. Overall, the performance of UC-LTM-AIC is superior to all the alternative methods.

Note that, as we move from the setting of unknown number of clusters to the setting of known number of clusters, the performance of UC-LTM improves on several data sets such as `autos` and `breast-w`. However, the performance degrades on several other data sets such as `glass2` and `heart-c`. This is probably due to fact that the some clusters in the true class partition are very small in size. When the number of clusters is unknown, UC-LTM would group them together with big 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 5 true clusters with sizes 160, 136, 0, 0 and 0. When the number of clusters is not given, UC-LTM produces 2 clusters, which is ideal (i.e., only 2 non-empty true clusters). When the number of clusters is set at 5, on the other hand, UC-LTM partitions the data into 5 clusters of sizes 93, 56, 55, 23, 69. The clusters are now more balanced in size, but the partition is more different from the true partition than the partition for the setting of unknown number of clusters.

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 unknown number of clusters. The improvements are often large (exceed 10%). In the setting of known number of clusters, UC-LTM is also superior to popular distance/similiarity-based methods.

It would be interesting to carry out similar research for continuous data. Here one can either make no independence assumptions 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 in the literature to explore middle grounds between the two extremes by working with block-diagonal covariances. The concept of UD-test and the procedure for dividing attributes into unidmensional clusters from this paper can be helpful in further work in the direction.

References

Akaike, H. 1974. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on* 19(6):716–723.

Bache, K., and Lichman, M. 2013. UCI machine learning repository. http://archive.ics.uci.edu/ml.

Bartholomew, D. J., and Knott, M. 1999. *Latent Variable Models and Factor Analysis*. Arnold, 2nd edition.

Chow, C. K., and Liu, C. N. 1968. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory* 14(3):462–467.

Cover, T. M., and Thomas, J. A. 1991. *Elements of information theory*. New York, NY, USA: Wiley-Interscience.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 39(1):1–38.

Fayyad, U. M., and Irani, K. B. 1993. Multi-interval discretization of continuousvalued attributes for classification learning. In *Thirteenth International Joint Conference on Articial Intelligence*, volume 2, 1022–1027.

Filippone, M.; Camastra, F.; Masulli, F.; and Rovetta, S. 2008. A survey of kernel and spectral methods for clustering. *Pattern Recognition* 41(1):176–190.

Friedman, N. 1997. Bayesian network classifiers. *Machine Learning* 29(2-3):131–163.

Garrett, E., and Zeger, S. 2000. Latent class model diagnosis. *Biometrics* 56:1055–1067.

Kass, R. E., and Raftery, A. E. 1995. Bayes factor. *Journal of American Statistical Association* 90(430):773–795.

Liu, T.-F.; Zhang, N.; Chen, P.; Liu, A.; Poon, L.; and Wang, Y. 2013. Greedy learning of latent tree models for multidimensional clustering. *Machine Learning* 1–30.

McLachlan, G. J., and Peel, D. 2000. *Finite Mixture Models*. New York: Wiley.

Mourad, R.; Sinoquet, C.; Zhang, N. L.; Liu, T.; and Leray, P. 2014. A survey on latent tree models and applications. *CoRR* abs/1402.0577.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, California: Morgan Kaufmann Publishers.

Schwarz, G. 1978. Estimating the dimension of a model. *Annals of Statistics* 6:461–464.

Vermunt, J., and Magidson, J. 2002. Latent class cluster analysis. In Hagenaars, J., and A.L., M., eds., *Applied latent class analysis*. Cambridge University Press. 89–106.

Zhang, N. L. 2004. Hierarchical latent class models for cluster analysis. *Journal of Machine Learning Research* 5:697–723.

Zhong, S., and Ghosh, J. 2003. A unified framework for model-based clustering. *J. Mach. Learn. Res.* 4:1001–1037.