Hierarchical Latent Class Models for Cluster Analysis

*Nevin L. Zhang**

Technical Report HKUST-CS02-02

*Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

Revised April 2002

This work is completed while the author is on leave at Department of Computer Science, Aalborg University, Denmark.

Abstract

Latent class models are used for cluster analysis of categorical data. Underlying such a model is the assumption that the observed variables are mutually independent given the class variable. A serious problem with the use of latent class models, known as local dependence, is that this assumption is often untrue. In this paper we propose hierarchical latent class models as a framework where the local dependence problem can be addressed in a principled manner. We develop a search-based algorithm for learning hierarchical latent class models from data. The algorithm is evaluated using both synthetic and real-world data.

Keywords: Model-based clustering, latent class models, local dependence, Bayesian networks, learning.

1 Introduction

Cluster analysis is the partitioning of similar objects into meaningful classes, when both the number of classes and the composition of the classes are to be determined (Kaufman and Rousseeuw 1990; Everitt 1993). In model-based clustering, it is assumed that the objects under study are generated by a mixture of probability distributions, with one component corresponding to each class. When the attributes of objects are continuous, cluster analysis is sometimes called *latent profile analysis* (Gibson 1959; Lazarsfeld and Henry 1968; Bartholomew and Knott 1999; Vermunt and Magidson 2002). When the attributes are categorical, cluster analysis is sometimes called *latent class analysis (LCA)* (Lazarsfeld and Henry 1968; Goodman 1974b; Bartholomew and Knott 1999; Uebersax 2001). There is also cluster analysis of *mixed-mode data* (Everitt 1993) where some attributes are continuous while others are categorical.

This paper is concerned with LCA, where data are assumed to be generated by a *latent class (LC)* model. An LC model consists of a class variable that represents the clusters to be identified and a number of other variables that represent attributes of objects [1]. The class variable is not observed and hence said to be *latent*. On the other hand, the attributes are observed and are called *manifest variables*.

LC models assume *local independence*, i.e. manifest variables are mutually independent in each latent class, or equivalently, given the latent variable. A serious problem with the use of LCA, known as *local dependence*, is that this assumption is often violated. If one does not deal with local dependence explicitly, one implicitly attributes it to the latent variable. This can lead to spurious latent classes and poor model fit. It can also degenerate the accuracy of classification because locally dependent manifest variables contain overlapping information (Vermunt and Magidson 2002).

The local dependence problem has attracted some attention in the LCA literature (Espeland & Handelman 1989; Garrett & Zeger 2000; Hagenaars 1988; Vermunt & Magidson 2000). Methods for detecting and modeling local dependence have been proposed. To detect local dependence, one typically compares observed and expected cross-classification frequencies for pairs of manifest variables. To model local dependence, one can join manifest variables, introduce multiple latent variables, or reformulate LC models as loglinear models and then impose constraints on them. All existing methods are preliminary proposals and suffer from a number of deficiencies (Section 2).

1.1 Our work

This paper describes the first systematic approach to the problem of local dependence. We address the problem in the framework of *hierarchical latent class (HLC) models*. HLC models are Bayesian networks whose structures are rooted trees and where the leaf nodes are observed while all other nodes are latent. This class of models is chosen for two reasons. First it is significantly larger than the class of LC models and can accommodate local dependence. Second inference in an HLC model takes time linear in model size, which makes it computationally feasible to run EM.

We develop a search-based algorithm for learning HLC models from data. The algorithm systematically searches for the optimal model by hill-climbing in a space of HLC models with the guidance of a model selection criterion. When there is no local dependence, the algorithm returns an LC model. When local dependence is present, it returns an HLC model where local dependence is appropriately modeled. It should be noted, however, that the algorithm might not work well on data generated by models that neither are HLC models nor can be closely approximated by HLC models.

The motivation for this work originates from an application in traditional Chinese medicine. In the application, model quality is of utmost importance and it is reasonable to assume abundant data and computing resources. So we take a principled (as opposite to heuristic) approach when designing our algorithm and we empirically show that the algorithm yields models of good quality. In subsequent work, we will explore ways to scale up the algorithm.

[1] Latent class models are sometimes also referred to as Naive Bayes models. We suggest that the term "naive Bayes models" be used only in the context of classification and the term "latent class models" be used in the context of clustering.

1.2 Related literature

This paper is an addition to the growing literature on hidden variable discovery in Bayesian networks (BN). Here is a brief discussion of some of this literature. Elidan *et al.* (2001) discuss how to introduce latent variables to BNs constructed for observed variables by BN structure learning algorithms. The idea is to look for structural signatures of latent variables. Elidan and Friedman (2001) give a fast algorithm for determining the cardinalities — the numbers of possible states — of latent variables introduced this way. Meila-Predoviciu (1999) studies how the so-called tree H models can be induced from data, where a tree H model is basically an LC model with each observed variable replaced by an undirected tree of observed variables. This work is based on the method of approximating joint probability distributions with dependence trees by Chow and Liu (1968).

The algorithms described in Connolly (1993) and Martin and VanLehn (1994) are closely related the algorithm presented in this paper. They all aim at inducing from data a latent structure that explains correlations among observed variables. The algorithm by Martin and VanLehn (1994) builds a two-level Bayesian network where the lower level consists of observed variables while the upper level consists of latent variables. The algorithm is based on tests of association between pairs of observed variables. The algorithm by Connolly (1993) constructs exactly what we call HLC models. Mutual information is used to group variables, a latent variable is introduced for each group, and the cardinality of the latent variable is determined using a technique called conceptual clustering. In comparison with Connolly's method, our method is more principled in the sense that it determines model structure and cardinalities of latent variables using one criterion, namely (some approximation) of the marginal likelihood.

The task of learning HLC models is similar to the reconstruction of phylogenetic trees, which is a major topic in biological sequence analysis (Durbin *et al*. 1998). As a matter of fact, phylogenetic trees are special HLC models where the model structures are binary (bifurcating) trees and all the variables share the same set of possible states. However, phylogenetic trees cannot be directly used for general cluster analysis because the constraints imposed on them. And techniques for phylogenetic tree reconstruction do not necessarily cover over to HLC models. For example, the structural EM algorithm for phylogenetic tree reconstruction by Friedman *et al*. (2002) does not work for HLC models because we do not know, a priori, the number of latent variables and their cardinalities.

HLC models should not be confused with model-based hierarchical clustering (e.g. Hanson *et al*. 1991, Fraley 1998). In an LC model (or similar models with continuous manifest variables), there is only one latent variable and each state of the variable corresponds to a class. HLC models generalize LC models by allowing multiple latent variables. An HLC model contains a hierarchy of latent variables; In model-based hierarchical clustering, on the other hand, one has a hierarchy of classes. Conceptually there is only one latent variable. Classes at different levels of the hierarchy correspond to states of the variable at different levels of granularity.

1.3 Organization of paper

The rest of this paper is organized as follows. In the next section we give a brief review of latent class models and survey previous work on local dependence. In Section 3 we formally define HLC models and study a number of theoretical issues related to the task of learning HLC models. A hill-climbing algorithm for inducing regular HLC models from data is described in Section 4. Section 5 reports empirical results on synthetic data and Section 6 discusses experiments with real-world data. Conclusions and remarks about future directions are provided in the final section.

2 Latent class models and local dependence

A *latent class (LC)* model involves a latent variable X and a number of manifest variables $Y_1, Y_2, \ldots, Y_n$. All the variables are categorical and the relationships among them are described by the simple Bayesian network shown in Figure 1. In applications, the latent variable X represents concepts such as "depression" that cannot be directly measured (Eaton *et al.* 1989). States of the latent variable correspond to classes of individuals in a population. The manifest variables Y_i represent manifestations of the latent concept such as "loss of appetite", "trouble falling asleep", "thoughts of death", and so on.

The latent variable influences all the manifest variables at the same time and hence renders them correlated. The essence of *latent class analysis* (LCA) is to characterize the latent concept by analyzing those correlations. This is possible due to the assumption that the manifest variables are mutually independent given the latent variable, which can be intuitively interpreted as saying that the latent variable is the only reason for the correlations. Since

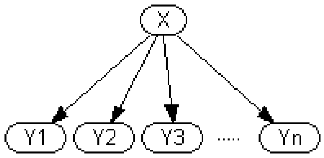

Figure 1: Structure of LC models.

each state of the latent variable corresponds to a class of individuals in a population, the conditional independence assumption can be restated as that the manifest variables are independent within each latent class. Because of this, it is sometimes called the *local independence* assumption.

Learning an LC model from data means to (1) determine the cardinality for variable X, i.e. the number of latent classes; and (2) estimate the model parameters $P(X)$ and $P(Y_i|X)$. Parameters are usually estimated using the EM algorithm (Dempster *et al.* 1977). The cardinality of X is determined by comparing alternatives using goodness-of-fit indices or scoring metrics. The most commonly used scoring metric is BIC (Schwarz 1978). Equivalent to the MDL score (Lanterman 2001), the BIC score is an approximation of the marginal likelihood that is derived in a setting when all variables are observed. Geiger *et al.* (1998) have recently cautioned against its use in LC models. Extensive experiments by Chickering and Heckerman (1997) show that BIC is less accurate than other efficient approximations of marginal likelihood such as the Cheeseman-Stutz (CS) score (Cheeseman and Stutz 1995).

A serious problem with the use of LCA is that the local independence assumption is often violated. The term *local dependence* is used to refer to this problem. Previous methods for dealing with local independence are surveyed by Uebersax (2000). In this survey, Uebersax distinguishes between two subtasks, namely the diagnosis and modeling of local dependence.

Diagnostic methods compare observed and expected frequencies for pairs of manifest variables. For concreteness, consider two manifest variables A and B in an LC model. Denote the observed and expected frequencies on A and B by $O(A,B)$ and $E(A,B)$ respectively. For any state a of A and b of B, $O(a,b)$ is the number of records where A is in state a and B is in state b. [2] On the other hand, $E(a,b) = P(a,b) * N$, where $P(A,B)$ is the joint probability of A and B in the LC model and N is the total number of records. Hagenaars (1988) suggests that one examines the standardized residuals

$$R(a,b) = \frac{O(a,b) - E(a,b)}{\sqrt{E(a,b)}}$$

for each combination (a,b) of states of A and B. If the residuals deviate from zero significantly, one concludes that A and B are locally dependent. Espeland and Handelman (1988) propose to compute the likelihood ratio statistic

$$L(A,B) = \sum_{a,b} 2O(a,b) log \frac{O(a,b)}{E(a,b)}.$$

The larger the statistic, the stronger the evidence for local dependence between A and B. When A and B are binary variables, we denote the possible states for the two variables by a, $\neg a$, b, and $\neg b$. Garret and Zeger (2000) recommend to compare the observed and expected log odd ratio

$$log \frac{O(\neg a,b)/O(a,b)}{O(\neg a,\neg b)/O(a,\neg b)}, log \frac{E(\neg a,b)/E(a,b)}{E(\neg a,\neg b)/E(a,\neg b)}.$$

Again larger differences indicate stronger evidence for local dependence.

An obvious way to model local dependence is to introduce joint variables. Consider the LC model M1 in Figure 2. If variables B and C are locally dependent, we can combine those two variables and introduce a joint variable BC. This lead to the model M2. The second method is to introduce new latent variables (Goodman 1974a). Uebersax calls it the *multiple indicator method.* To account for the local dependence between B and C in M1, for instance, we can introduce a new latent variable X_1 and thereby get model M3. By doing this, we are

[2] We use upper case letters for variable names and the corresponding lower case letters for their states.

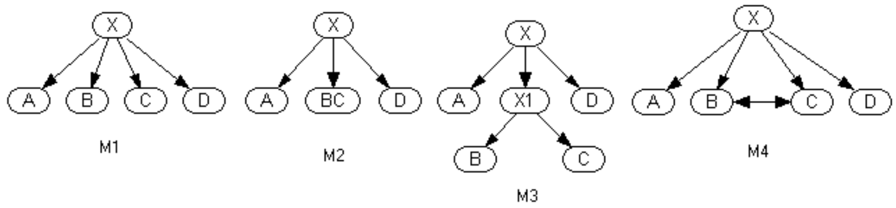

Figure 2: Modeling of local dependence.

assuming that the reason for B and C being locally dependent is that they are jointly influenced by a latent variable X_1 that is not completely determined by the latent variable X. In the third approach (Hagenaars 1988), one views LC models as special loglinear models. When two manifest variables are locally dependent, one simply adds a direct effect between them. In model M1, adding a direct effect between B and C yields the model M4. Note that M4 is no longer a Bayesian network. It is the path-diagram (see Bohrnstedt and Knoke 1994, Chapter 11) for a loglinear model.

Previous work is not sufficient for a number of reasons. First, the criteria for detecting local dependence is heuristic in nature. Judgments are required as to how the various thresholds should be set. Second, there are no criteria for making the trade-off between increasing the cardinalities of existing latent variables versus increasing the complexity of model structure. In Hagenaars (1988) and Uebersax (2000), cardinalities of all latent variables are fixed at 2 while model structures are allowed to change. In most other works the standard one-latent-variable structure is assumed and fixed, while the cardinality of the latent variable is allowed to change. Third, the search for the best model is carried out manually. Typically only a few simple models are considered (Goodman 1974a, Hagenaars 1988). The search space for the multiple indicator method is not even clearly defined. Finally, when there are multiple pairs of locally dependent manifest variables, it is not clear which pair should be tackled first or should all pairs be handled simultaneously.

The purpose of this paper is to develop a principled and systematic method for dealing with local dependence. In the next section, we describe the models that serve as the framework for our work.

3 Hierarchical latent class models

A *hierarchical latent class (HLC) model* is a Bayesian network where

1. The network structure is a rooted tree; and

2. The variables at the leaf nodes are observed and all the other variables are not [3].

Figure 3 shows an example of an HLC model. Following the LCA literature, we refer to the observed variables as *manifest variables* and all the other variables as *latent variables*. In this paper we do not distinguish between variables and nodes. So we sometimes speak also of *manifest nodes* and *latent nodes*. For technical convenience, we assume that there are at least two manifest variables.

We use θ to refer to the collection of parameters in an HLC model M and use m to refer to what is left when the parameters are removed from M. So we usually write an HLC model as a pair $M = (m, \theta)$. We sometimes refer to the first component m of the pair also as an HLC model. When it is necessary to distinguish between m and the pair (m, θ), we call m an *unparameterized HLC model* and the pair a *parameterized HLC model*. The term *HLC model structure* is reserved for what is left if information about cardinalities of latent variables are removed from an unparameterized model m. Model structures will be denoted by the letter S, possibly with subscripts.

3.1 Parsimonious HLC models

In this paper we study the learning of HLC models. We assume that there is a collection of identical and independently distributed (i.i.d.) samples generated by some HLC model. Each sample consists of states for all or

[3] The concept of a variable being observed is always w.r.t some given data set. A variable is *observed* in a data set if there is at least one record that contains the state for that variable.

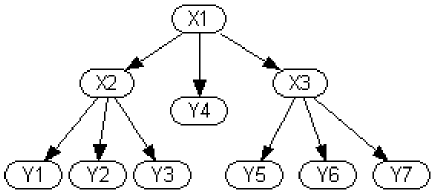

Figure 3: An example HLC model. The X_i's are latent variables and the Y_j's are manifest variables.

some of the manifest variables. The task is to reconstruct the HLC model from data. As will be seen later, not all HLC models can be reconstructed from data. It is hence natural to ask what models can be reconstructed. In this subsection we provide a partial answer to this question.

Consider two parameterized HLC models $M = (m, \theta)$ and $M' = (m', \theta')$ that share the same manifest variables $Y_1, Y_2, \ldots, Y_n$. We say that M and M' are *marginally equivalent* if the probability distribution over the manifest variables is the same in both models, i.e.

$$P(Y_1, \ldots, Y_n | m, \theta) = P(Y_1, \ldots, Y_n | m', \theta'). \tag{1}$$

Two marginally equivalent parameterized models are *equivalent* if they also have the same number of independent parameters. Two unparameterized HLC models m and m' are *equivalent* if for any parameterization θ of m there exists a parameterization θ' of m' such that (m, θ) and (m, θ') are equivalent and vice versa. Two HLC model structures S_1 and S_2 are *equivalent* if there are equivalent unparameterized models m_1 and m_2 whose underlying structures are S_1 and S_2 respectively.

A parameterized HLC model M is *parsimonious* if there does not exist another model M' that is marginally equivalent to M and that has fewer independent parameters than M. An unparameterized HLC model m is *parsimonious* if there exists a parameterization θ of m such that (m, θ) is parsimonious.

Let M be a parameterized HLC model and D be a set of i.i.d. samples generated by M. If M is not parsimonious, then there must exist another HLC model whose penalized loglikelihood score given D (Green 1998, Lanternman 2001) is greater than that of M. This means that, if one uses penalized loglikelihood for model selection, one would prefer this other parsimonious models over the non-parsimonious model M. The following theorem states that, to some extent, the opposite is also true, i.e. one would prefer M to other models if M is parsimonious.

Theorem 1 *Let M and M' be two parameterized HLC models with the same manifest variables. Let D be a set of i.i.d. samples generated from M.*

1. *If M and M' are not marginally equivalent, then the loglikelihood $l(M|D)$ of M is strictly greater than the loglikelihood $l(M'|D)$ of M' when the sample size is large enough.*

2. *If M is parsimonious and is not equivalent to M', then the penalized loglikelihood of M is strictly larger than that of M' when the sample size is large enough.*

Proof: Use P and P' to denote the marginal probability distributions over the manifest variables in M and M' respectively. Let N be the sample size. It follows from the central limit theorem that, as N goes to infinite, $[l(M|D) - l(M'|D)]/N$ approaches the Kullback-Leibler (KL) distance $I(P{:}P')$. The first part hence follows from the well-known property of the KL distance that $I(P{:}P') \geq 0$ and the equality is true only when P and P' are identical (e.g. Cover and Thomas 1991).

The second part can be divided into two cases. The first case is when M and M' are marginally equivalent and M' has more parameters than M. Here the statement is trivially true for all sample sizes. In the second case, M and M' are not marginally equivalent. According to the first part of the theorem, $l(M|D) - l(M'|D)$ is positive when N large enough. Moreover the quantity increases linearly with N. On the other hand, the penalty on model complexity increases logarithmically with N. Hence the statement is true when N is large enough. Q.E.D.

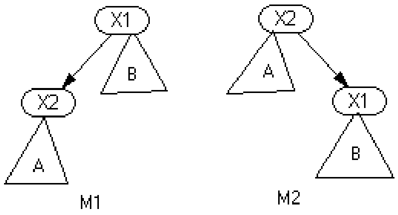

Figure 4: The operation of root walking.

3.2 Model equivalence

In this subsection we give an operational characterization of model equivalence. Let X_1 be the root of a parameterized HLC model M_1. Suppose X_2 is a child of X_1 and it is a latent node (see Figure 4). Define another HLC model M_2 by reversing the arrow $X_1 \rightarrow X_2$ and, while leaving the values for all other parameters unchanged, defining for $P_{M_2}(X_2)$ and $P_{M_2}(X_1|X_2)$ as follows:

$$\begin{aligned} P_{M_2}(X_2) &= \sum_{X_1} P_{M_1}(X_1) P_{M_1}(X_2|X_1) \\ P_{M_2}(X_1|X_2) &= \begin{cases} \frac{P_{M_1}(X_1)P_{M_1}(X_2|X_1)}{P_{M_2}(X_2)} & \text{if } P_{M_2}(X_2) > 0. \\ \frac{1}{|X_1|} & \text{otherwise.} \end{cases} \end{aligned}$$

We use the term *root walking* to refer to the process of obtaining M_2 from M_1. In the process, the root has walked from X_1 to X_2.

Theorem 2 [4] *Let M_1 and M_2 be two parameterized HLC models. If M_2 is obtained from M_1 by one or more steps of root walking, then M_1 and M_2 are equivalent.*

Proof: Without losing generality, assume M_2 is obtained from M_1 by one step of root walking and the models are as shown in Figure 4. Model M_2 is obtained by letting the root of M_1 walk from X_1 to X_2. Suppose A and B are the sets of variables in the subtrees rooted at X_2 and X_1 respectively. We have,

$$\begin{aligned} P_{M_1}(X_1, X_2, A, B) &= P_{M_1}(X_1)P_{M_1}(X_2|X_1)P_{M_1}(A|X_2)P_{M_1}(B|X_1) \\ &= P_{M_2}(X_2)P_{M_2}(X_1|X_2)P_{M_2}(A|X_2)P_{M_2}(B|X_1) \\ &= P_{M_2}(X_1, X_2, A, B). \end{aligned}$$

Consequently, M_1 and M_2 are marginally equivalent.

It is easy to see that $P_{M_1}(X_1)$ and $P_{M_1}(X_2|X_1)$ encapsulate $|X_1||X_2|-1$ parameters. The same is true for $P_{M_2}(X_2)$ and $P_{M_2}(X_1|X_2)$. Hence M_1 and M_2 have the same number of parameters. The theorem is therefore proved. Q.E.D

The two HLC models shown in Figure 5 are equivalent to the model in Figure 3. The model on the left is obtained by letting the root of the original model walk from X_1 to X_2, while the model on the right is obtained by letting the root walk from X_1 to X_3.

In general, the root of an HLC model can walk to any latent node. This implies the root node cannot be determined from data [5]. A question about the suitability of HLC models for cluster analysis naturally arises. We take the position that the root node can be determined from the objective in clustering and domain knowledge. Moreover we view the presence of multiple latent variables an advantage because it enables one to cluster data in multiple ways. Note that multiple clusterings due to multiple latent variables are very different from multiple clusterings in hierarchical clustering. In the latter case, a clustering at a lower level of the hierarchy is a refinement of a clustering at a higher level. The same relationship does not exist in the former case.

[4] A similar but different theorem was proved by Chickering (1996) for Bayesian networks with no latent variables. In Chickering (1996), model equivalence implies equal number of parameters. Here equal number of parameters is part of the definition of model equivalence.

[5] In the case of phylogenetic trees, this is a well-known fact (Durbin *et al.* 1998).

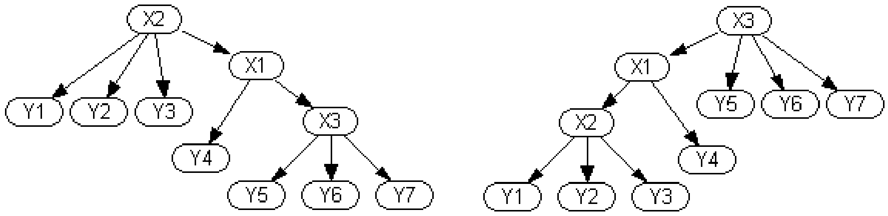

Figure 5: HLC models that are equivalent to the one in Figure 3.

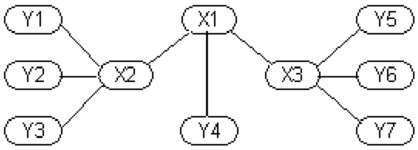

Figure 6: The unrooted HLC model that corresponds to the HLC model in Figure 3.

The inability of determining the root node from data also have some technical consequences. We can never induce HLC models from data. Instead we obtain what might be called unrooted HLC models. An *unrooted HLC model* is an HLC model with all directions on the edges dropped. Figure 6 shows the unrooted HLC model that corresponds to the HLC model in Figure 3. An unrooted HLC model represents a class of HLC models; members of the class are obtained by rooting the model at various nodes and by directing the edges away from the root. Semantically it is a Markov random field on an undirected tree. The leaf nodes are observed while the interior nodes are latent. The concepts of marginal equivalence, equivalence, and parsimony can be defined for unrooted HLC models in the same way as for rooted models.

From now on when we speak of HLC models we always mean unrooted HLC models unless it is explicitly stated otherwise.

3.3 Regular HLC models

In this subsection we first introduce the concept of regular HLC models and show that all parsimonious models must regular. We then show that the set of unparameterized regular HLC models for a given set of manifest variables is finite. This provides a search space for the learning algorithm to be developed in the next section.

For any variable X, use Ω_X and $|X|$ to denote its domain and cardinality respectively. For a latent variable Z in an HLC model, enumerate its neighbors as $X_1, X_2, \ldots, X_k$. An HLC model is *regular* if for any latent variable Z,

$$|Z| \leq \frac{\prod_{i=1}^{k} |X_i|}{\max_{i=1}^{k} |X_i|}, \tag{2}$$

and when Z has only two neighbors,

$$|Z| \leq \frac{|X_1||X_2|}{|X_1| + |X_2| - 1}. \tag{3}$$

Note that this definition applies to parameterized as well as unparameterized models. The first condition was suggested by Kocka and Zhang (2002).

Theorem 3 *Irregular HLC models are not parsimonious*

Proof: It is clear that if we can prove the theorem for parameterized models, it must be also true for unparameterized models. Let Z be a latent variable in a parameterized HLC model M. It suffices to show that if either (2) or (3) is violated, there exists another model M' that is marginally equivalent to M and that has fewer parameters than M.

Parameters of a rooted HLC model include the prior probability of the root and the conditional probability of each of the non-root nodes given its parent. On the other hand, parameters of an unrooted HLC model include a potential for each edge in the model structure. The potential is a function of the two variables connected by the edge. Let $f(Z, X_1, \ldots, X_k)$ be the multiplication of all potentials for edges between Z and its neighbors and let $g(X_1, \ldots, X_k) = \sum_Z f(Z, X_1, \ldots, X_k)$.

We first consider the case when (2) is violated, i.e. $|Z| > \prod_{i=1}^{k} |X_i| / \max_{i=1}^{k} |X_i|$. Without losing generality, suppose $|X_k| = \max_{i=1}^{k} |X_i|$. Let M' be the same as M except that the domain of Z is redefined to be $\prod_{i=1}^{k-1} \Omega_{X_i}$. An state of Z can be written as $<z_1, \ldots, z_{k-1}>$, where z_i is a state of X_i. For each $i = 1, \ldots, k-1$, set the potential $f_i(Z, X_i)$ for the edge between X_i and Z as follows:

$$f_i(<z_1, \ldots, z_{k-1}>, x_i) = \begin{cases} 1 & \text{if } z_i = x_i \\ 0 & \text{if } z_i \neq x_i. \end{cases}$$

Set the potential $f_k(Z, X_k)$ for the edge between X_k and Z as follows:

$$f_i(<z_1, \ldots, z_{k-1}>, x_k) = g(z_1, \ldots, z_{k-1}, x_k).$$

Then

$$\sum_Z \prod_{i=1}^{k} f_i(Z, X_i) = g(X_1, \ldots, X_k).$$

Hence M' is marginally equivalent to M. Because Z has fewer states in M' than in M, M' has fewer parameters than M. Therefore M is not parsimonious.

Now consider the case when (3) is violated. In this case, latent variable Z has two neighbors X_1 and X_2 and $|Z| > |X_1||X_2|/(|X_1| + |X_2| - 1)$. Let M' be the model obtained by eliminating Z from M [6]. Then M' is marginally equivalent to M. To calculate the difference in the number of independent parameters between M' and M, imagine rooting both models at X_1. Then it is easy to see that the difference is

$$|X_1|(|X_2| - 1) - [|X_1|(|Z| - 1) + |Z|(|X_2| - 1)] = |X_1||X_2| - |Z|(|X_1| + |X_2| - 1).$$

This is negative. Hence M' has fewer parameters than M. Therefore M is not parsimonious. The theorem is proved. Q.E.D

Theorem 4 *The set of all regular unparameterized HLC models for a given set of manifest variables is finite.*

Before proving this theorem, we need to introduce several lemmas, which are interesting on their own right. A latent node in an HLC model has at least two neighbors. A *singly connected* latent node is one that has exactly two neighbors.

Lemma 1 *In a regular HLC model, no two singly connected latent nodes can be neighbors.*

Proof: We see from (3) that the cardinality of a singly connect node is strictly smaller than those of its two neighbors. If two singly connected latent nodes Z_1 and Z_2 were neighbors, then we would have both $|Z_1| > |Z_2|$ and $|Z_1| < |Z_2|$. Therefore two singly connected latent nodes cannot be neighbors. Q.E.D.

This lemma inspires the following two definitions. We say that an HLC model structure is *regular* if no two singly connected latent nodes are neighbors. If there are no singly connected latent nodes at all, we say that the model structure is *strictly regular*.

Lemma 2 *Let S be an HLC model structure with n manifest variables. If S is regular, then there are fewer than $3n$ latent nodes. If S is strictly regular, then there are fewer than n latent nodes.*

[6] This means to (1) remove Z and connect X_1 and X_2 and (2) set the potential for the new edge between X_1 and X_2 to be $\sum_Z f_1(X_1, Z) f_2(X_2, Z)$, where f_1 and f_2 are the potentials for the edge between X_1 and Z and the edge between X_2 and Z respectively.

Proof: We prove the second part first [7]. Let h be the number of latent nodes. Then the total number of nodes is $n+h$. Hence the number of edges is $n+h-1$.

On the other hand, each manifest node appears in exactly one edge and, because of strict regularity, each latent node appears in at least three edges. Because each edge involves exactly two variables, there are at least $(n+3h)/2$ edges. Hence $n+h-1 \geq (n+3h)/2$. Solving this inequality yields $h \leq n-2 < n$.

To prove the first part, let m be the total number of nodes in a regular structure. Imagine that we root the structure at an arbitrary latent node. Then the child of a singly connected latent node is either a manifest node or another latent node that is not singly connected. Moreover, the children for different singly connected nodes are different. So if we eliminate all the singly connected latent nodes, the resulting structure will have at least $m/2$ nodes. The resulting structure is strictly regular. Hence $m/2 < 2n$. This implies that $m < 4n$. Since there are n manifest nodes, the number of latent is smaller than $3n$. Q.E.D

Lemma 3 *There are fewer than 2^{3n^2} different regular HLC model structures for a given set of n manifest nodes.*

Proof: Let $\mathcal{P}$ be the power set of the set of manifest nodes and let $\mathcal{V}$ be the collection of vectors that consist $3n$ elements of $\mathcal{P}$. Duplicates are allowed in any given vector. Since the cardinality of $\mathcal{P}$ is 2^n, the cardinality of $\mathcal{V}$ is $(2^n)^{3n} = 2^{3n^2}$.

Let $\mathcal{S}$ be the set of all regular HLC model structures for the given manifest nodes. Define a mapping from $\mathcal{S}$ to $\mathcal{V}$ as follows: For any given model structure in $\mathcal{S}$, first root the structure at the parent of the first manifest node. Second, arrange all the latent nodes into a vector according to the depth-first traversal order. According to Lemma 2, the length of the vector cannot exceed $3n$. Third, replace each latent node with the subset of manifest nodes in its subtree. Finally, add copies of the empty set to the end so that the length of the vector is $3n$. It is not difficult to see that the mapping is bijective. Therefore the cardinality of $\mathcal{S}$ cannot exceed that of $\mathcal{V}$, which is 2^{3n^2}. The theorem is proved. Q.E.D.

Proof of Theorem 4: According to Lemma 3, the number of regular model structures is finite. It is clear from (2) and (3), there number of unparameterized model for a given model structure must also be finite. The theorem is therefore proved. Q.E.D

4 Searching for optimal models

In this section we present a hill-climbing algorithm for learning HLC models. Hill-climbing requires a scoring metric for comparing candidate models. In this work we experiment with four existing scoring metrics, namely AIC (Akaike 1974), BIC (Schwarz 1978), the Cheeseman-Stutz (CS) score (Cheeseman and Stutz 1995), and the holdout logarithmic score (LS) (Cowell *et al.* 1999) .

Hill-climbing also requires the specification of a search space and search operators. According to Theorem 3, a natural search space for our task is the set of all regular (unparameterized) HLC models for the set of manifest variables that appear in data. By Theorem 4, we know that this space is finite.

Instead of searching this space directly, we structure the space into two levels according to the following two subtasks and we search those two levels separately:

1. Given a model structure, find optimal cardinalities for the latent variables.
2. Find an optimal model structure.

This search space restructuring is motivated by the fact that natural search operators exist for each of the two levels, while operators for the flat space are less obvious.

4.1 Estimating cardinalities of latent variables

The search space for the first subtask consists of all the regular models with the given model structure. To hill-climb in this space we start with the model where the cardinalities of all the latent variables are the minimum. In most cases, the minimum cardinality for a latent variable is 2. For a latent variable next to a singly connected latent node, however, the minimum possible cardinality is 4 because of the inequality (3). At each step, we modify the

[7]This proof is contributed by Tomas Kocha.

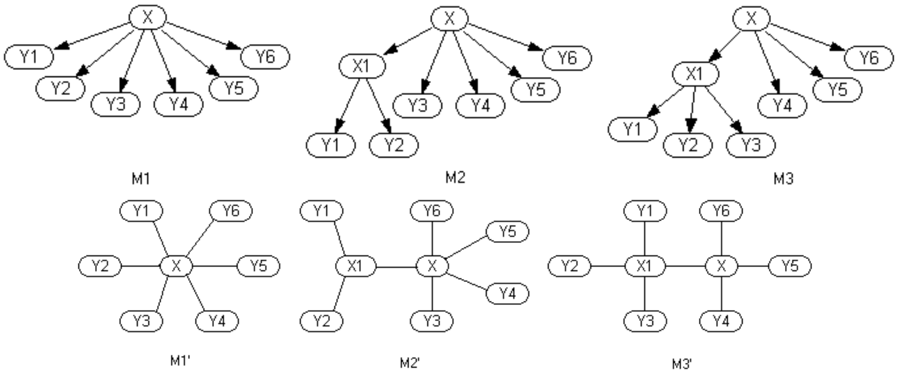

Figure 7: Illustration of structural search operators.

current model to get a number of new models. The operator for modifying a model is to increase the cardinality of a latent variable by one. Irregular new models are discarded. We then evaluate each of the new models and picks the best one to seed the next search step. To evaluate a model, one needs to estimate its parameters. We use the EM algorithm for this task.

4.2 Search for optimal model structures

The search space for the subtask of finding an optimal model structure consists of all the regular HLC model structures for the given manifest variables. To search this space, we start with the simplest HLC model structure, namely the LC model structure (viewed as an unrooted HLC model structure). At each step, we modify the current structure to construct a number of new structures. The new structures are then evaluated and the best structure is selected as the starting point for the next step. To evaluate a model structure, one needs to estimate the cardinalities of its latent variables. This issue is addressed in subtask 1.

We use three search operators to modify model structures, namely node-introduction, node-elimination, and neighbor-relocation.

4.2.1 Node introduction

To motivate the node-introduction operator, we need to go back to rooted models. Consider the rooted HLC model M_1 shown in Figure 7. Suppose variables Y_1 and Y_2 are locally dependent. A nature way to model this local dependence is to introduce a new parent for Y_1 and Y_2, as shown in M_2. This is precisely the idea behind the multiple indicator approach to local dependence that we mentioned in Section 2.

When translated to unrooted model structures, the new parent introduction operator becomes the *node-introduction* operator. Let X be a latent node in an unrooted model structure. Suppose X has more than two neighbors. Then for any two neighbors of X, say Z_1 and Z_2, we can introduce a new latent node Z to separate X from Z_1 and Z_2. Afterwards, X is no longer connected to Z_1 and Z_2. Instead X is connected to Z and Z is connected to Z_1 and Z_2. To see an example, consider the model structure M_1' in Figure 7. Introducing a new latent node X_1 to separate X from Y_1 and Y_2 results in the model structure M_2'.

In the case of rooted model structures, we do not consider introducing new parents for groups of three or more nodes for the sake of computational efficiency. This constraint implies that the model M_3 in Figure 7 cannot be reached from M_1 in one step. In the case of unrooted model structures, we do not allow the introduction of a new node to separate a latent node from three or more of its neighbors. This implies that we cannot reach M_3' from in Figure 7 cannot M_1' in one step.

Node-introduction is not allowed when it results in irregular model structures. This means that we cannot introduce a new node to separate a latent node X from two of its neighbors if it has only one other neighbor and that neighbor is a singly connected latent node. Moreover, we cannot introduce a new node to separate a singly connected latent node from its two neighbors. [8]

[8]Node-introduction is similar to an operator that PROMTL, a system for inferring phylogenetic trees, uses to search for optimal tree

4.2.2 Node elimination

The opposite of node-introduction is *node-elimination*. We notice that a newly introduced node has exactly three neighbors. Consequently we allow a latent node be eliminated only when it has three neighbors. Of course, node elimination cannot be applied if there is only one latent node.

4.2.3 Neighbor relocation

In Figure 7, we cannot reach M'_3 from either M'_1 or M'_2 using node-introduction and node-elimination. To overcome this difficult, we introduce the third search operator, namely *neighbor-relocation*. Suppose a latent node X has a neighbor Z that is also a latent node. Then we can relocate any of the other neighbors Z' of X to Z, which means to disconnect Z' from X and reconnect it to Z. To see an example, consider the model structure M'_2 in Figure 7. If we relocate the neighbor Y_3 of X to X_1, we reach M'_3.

For the sake of computational efficiency, we do not allow neighbor relocation between two non-neighboring latent nodes. In Figure 6, for example, we cannot relocate neighbors of X_2 to X_3 and vice versa. Moreover neighbor relocation is not allowed when it results in irregular model structures. To be more specific, suppose X is a latent node that has a latent node neighbor Z. We cannot relocate another neighbor Z' of X to Z if X has only three neighbors and the third neighbor is a singly connected latent node. The relocation is not allowed, of course, if X has only two neighbors. Finally note that the effects of any particular neighbor relocation can always be undone by another application of the operator. [9]

4.2.4 A property

Theorem 5 *Consider the collection of regular HLC model structures for a given set of manifest variables. One can go between any two structures in the collection using node-introduction, node-elimination, and neighbor-relocation.*

Proof: It suffices to show that any structure S in the collection can be reached from the (unrooted) LC model structure. We do this by induction on the number of latent nodes in S. If there is only one latent node, the proposition is trivially true. Suppose the proposition is true for the case of $n-1$ latent nodes where $n>1$. Consider the case of n. Because there are more than one latent node, there must be a latent node X that has 3 or more neighbors. We modify S by first relocating some of the neighbors of X to other (neighboring) latent nodes such that it has only 3 neighbors afterwards. We then eliminate X. Denote the resulting structure by S'. By the induction hypothesis, we can reach S' from the LC model structure using node-introduction, node-elimination, and neighbor-relocation. By the construction of S', we know that we can reach S from S' using those three operators. The theorem is therefore proved. Q.E.D

4.3 Complexity analysis

The description of our learning algorithm is now complete. In this subsection we analyze the worst case complexity of the algorithm.

Let n be the number of manifest variables. According to Lemma 2, a regular HLC model with n manifest variables have fewer than $3n$ latent nodes. The total number of nodes in the model is hence bounded by $4n-1$.

In each step of structural search, our algorithm applies node-introduction, node-elimination, and neighbor-relocation to the current model structure and produces a set of new structures. Each application of node-introduction involves a pair of neighbors of a latent nodes. Such pairs for different applications of the operator cannot be the same. Hence the number of new structures produced by node-introduction is bounded by $(4n-1)4n/2<8n^2$. The node-elimination operator produces no more than $3n$ new structures. An application of neighbor-relocation also involves a pair of nodes, a latent node and one of its neighbors. Such pairs for different applications of the operator are different. Hence the total number of new structures produced by neighbor relocation is bounded by the number of edges, which is no more than $4n$. The total number of new structures produced at each search step is hence

topologies via star decomposition (Kishino *et al.* 1990). The former is slightly less constrained than the latter in that it is allowed to create singly connected nodes as by-products.

[9] Neighbor relocation is related to but significantly different than an operator called branch swapping that PAUP, a system for inferring phylogenetic trees, uses to search for optimal tree topologies (Swofford 1998). The latter includes what are called nearest neighbor interchange; subtree pruning and regrafting; and tree bisection/reconnection .

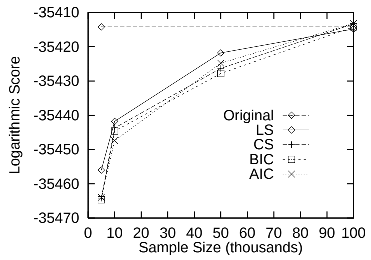

Figure 8: Logarithmic scores of learned models on testing data.

bounded by $8n^2+3n+4n=8n^2+7n$. If the entire search process takes N steps, then the total number of model structures that we need to examine is no more than $N(8n^2+7n)$.

For each model structure, we need to determine the cardinalities of all its latent variables. Our algorithm does so by hill-climbing. The search starts from the model where all latent variables have the minimum numbers of states possible and at each step a new model is generated for each latent variable by increasing its cardinality by one. Since the number of latent variables is no larger than $3n$, no more than $3n$ models can be generated at each step. Let k be the maximum number of states a variable can have in all models that we encounter. Then the search takes no more than $3nk$ steps. Consequently, the total number of models examined for an given model structure does not exceed $n * 3nk=3n^2k$.

For each model, we need to estimate its parameters using the EM algorithm. The complexity of inference in an HLC model is linear in the number of nodes. Since there are no more than $4n$ nodes, inference takes $O(4n)$ time. Suppose there are d distinct data records ($d \leq k^n$). Then each iteration of EM takes $O(4nd)$ time. Let M be the maximum number of EM iterations allowed on a model. Then parameter estimation for a given model takes $(4ndM)$ time.

Totaling up all the parts, we conclude that the time complexity of our algorithm is

$$O(N(8n^2+7n) * 3n^2k * 4ndM) = O(96MNkdn^5). \tag{4}$$

5 Empirical results on synthetic data

We have emipirically evaluated the algorithm described in the previous section using both synthetic and real-world data. This section discusses experiments with synthetic data. Two experiments were conducted. We report their results separately.

5.1 Experiment 1

In this experiment, the synthetic data were generated using the HLC model structure in Figure 3. The cardinalities of all variables were set at 3. The model was randomly parameterized. Four training sets with 5,000, 10,000, 50,000, and 100,000 records were sampled. A testing set of 5,000 records was also sampled. Each sample record consists of states for all the manifest variables.

We ran our learning algorithm on each of the four training sets, once for each of the four scoring metrics BIC, AIC, CS, and LS. There are 16 settings in total. For the LS scoring metric, 25% of the training data was set aside and used as validation data. Candidate models were compared using their logarithmic scores on the validation data. The EM termination threshold was set at 0.01 during model selection and at 0.0001 when estimating parameters for the final model. Irrespective of the threshold, EM was allowed to run no more than 200 iterations on any given model. For local maxima avoidance, we used the Chickering and Heckerman (1997) variant of the multiple-restart approach.

The logarithmic scores of the learned models on the testing data are shown in Figure 8. The scores are grouped into four curves according to the four scoring metrics. The score of the original model is also shown for comparison. We see that, in the relative sense, the scores of the learned models are quite close to that of the original model.

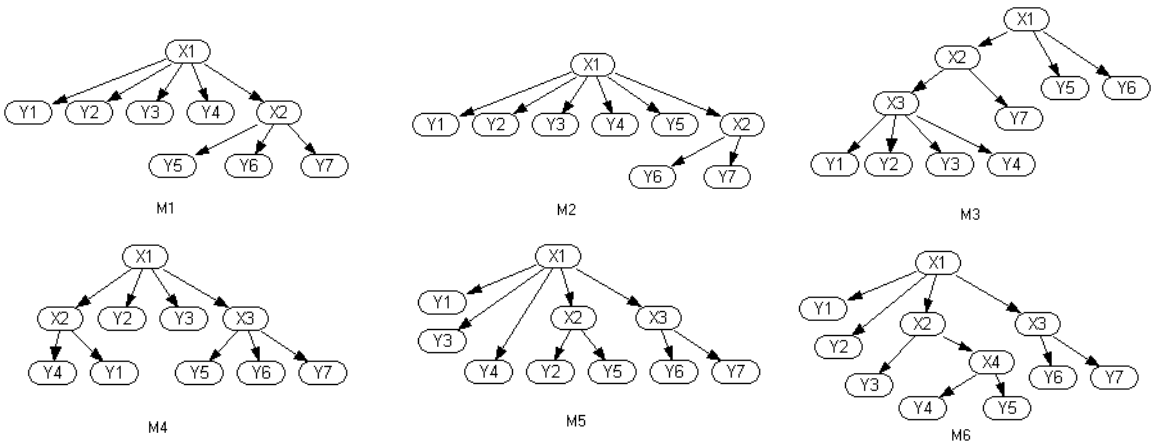

Figure 9: Structures of learned models.

This indicates that those models are as good as the original model when it comes to predicting the testing set. We also see that scores do not vary significantly across the scoring metrics.

The structures of the learned models do depend on the scoring metrics. There are 7 different model structures. The first one is the original structure and will be denoted by $M0$. The other six are shown in Figure 9. Model structures produced by our algorithm are unrooted. In this and the next section, we root them in certain ways for readability. Table 1 gives information about which structure was obtained in what setting and how far away the structures are from the original structure in terms the number of structural search operations.

	5k	10k	50k	100k
BIC	1 (M1)	1 (M1)	0 (M0)	0 (M0)
CS	1 (M1)	2 (M2)	0 (M0)	0 (M0)
LS	2 (M3)	4 (M6)	0 (M0)	0 (M0)
AIC	3 (M4)	4 (M5)	0 (M0)	0 (M0)

Table 1: Model structures found in the 16 settings and the numbers of operations it would take to reach the original structure.

We see that, when combined with either BIC or CS, our algorithm obtained, from the 50k and 100k training sets, the correct structure. In the other two training sets, the model structures found are quite close to the original structure. For example, the BIC scoring metric gave us M1. This model structure is very similar to the original structure. The only thing that our algorithm failed to recognize in M1 is that Y_4 and X_2 are not independent given X_1. This might be attributable to insufficient data.

When combined with AIC and LS, our algorithm also recovered the correct structure from the 50k and 100k training sets. In the other two training sets, however, it obtained structures that are significantly different from the original structure.

Although the correct model structure was recovered from the 50k and 100k training sets no matter which the scoring metric was used, different scoring metrics gave different estimates for the cardinalities of the latent variables. As can be seen from Table 2, BIC and CS tend to produce underestimates and with more data, they tend to give better estimates. On the other hand, AIC and LS tend to bring about overestimates and the estimates do not seem to improve with more data.

5.2 Experiment 2

The setup of this experiment is the same as that of Experiment 1 except the way model parameters were generated. Here we generated the parameters also in random fashion but ensured that each conditional probability distribution have one component with mass no smaller than 0.6. The objective is to see how our algorithm would perform in cases where the parameters are more extreme compared with those in Experiment 1.

In terms of logarithmic scores of learned models on testing data, results of this experiment are more or less the same as in Experiment 1. When it comes to structures and cardinalities of latent nodes, there are significant differences. With BIC and CS, our algorithm was able to recover the correct model structure from all four training sets. Moreover, the cardinalities of X_2 and X_3 were always estimated correctly. The cardinality of X_3 was estimated correctly in the 100k training sets, while was underestimated by 1 in all other training sets. These results are significantly better than those in Experiment 1.

When AIC and LS was used, on the other hand, the performance is worse than in Experiment 1. The algorithm was unable to recover the correct model structure even from the 50k and 100k training sets.

6 Empirical results on real-world data

This section reports empirical results on four data sets taken from the LCA literature, namely the Hannover rheumatoid arthritis data (Wasmus *et al.* 1989), the Coleman data (Coleman 1964), the HIV data (Alvord *et al.* 1988), and the housing building data (Hagenaars 1988). For the convenience of the reader, the data sets are reproduced in the appendix.

6.1 The Hannover rheumatoid arthritis data

The Hannover rheumatoid arthritis data was taken from a study by Wasmus *et al.* (1989) on the prevalence of rheumatoid arthritis in the adult population. A random sample of 25 to 74 year old German residents of Hannover, Germany was surveyed by means of a mailed questionnaire. Among others, this questionnaire contained five questions about the presence of five symptoms "today": back pain, neck pain, pain in one or several joints, joint swelling, and morning stiffness. Each item was to be answered in a simple yes/no response format. The data set consists of 7,162 records.

This data set has been analyzed by Kohlmann and Formann (1997). They conclude that the best model for this data set is a four class LC model. This model fits data well ($L = 8.2$, $df = 8$, $p = 0.414$) and is meaningful to epidemiologists.

Using scoring metrics BIC, CS, and AIC, our algorithm discovered exactly the same model. When LS was used, however, it computed a very different model that does not fit data well.

6.2 The Coleman data

The Coleman data summarize responses of 3,398 schoolboys, each was asked to respond to the following question and statement at two different points in time: (1) "Are you a member of the leading crowd?" (2) "If a fellow wants to be a part of the leading crowd around here, he sometimes has to go against his principles." There are four binary manifest variables A, B, C, and D. The variable A stands for the answer to the question in October 1957 and C that in May 1958. The variable B stands for the response to the statement in October 1957 and D that in May 1958. The value of 0 means "yes" and 1 means "no".

This data set has been previously analyzed by Goodman (1974a) and Hagenaars (1988). Goodman started with a 2-class LC model and found that it does not fit the data well ($L = 249.50$, $df = 6$, $p < 0.001$). He went on to consider the loglinear model that is represented by the path diagram M1 in Figure 10. In the model, both X_1

	50k				100k			
	BIC	CS	LS	AIC	BIC	CS	LS	AIC
X_1	2	2	2	2	2	2	2	2
X_2	3	3	3	3	3	3	3	4
X_3	2	2	4	4	3	3	4	4

Table 2: Cardinalities of latent variables in the learned models of Experiment 1 that have the correct structure. In the original model, all variables have 3 states.

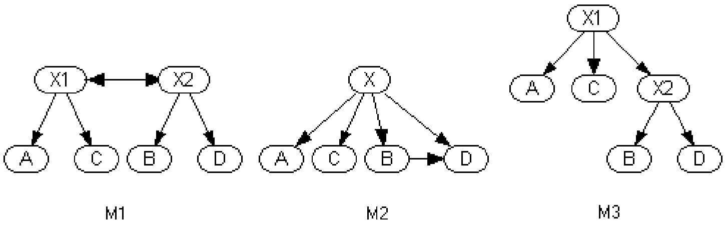

Figure 10: Models for the Coleman data.

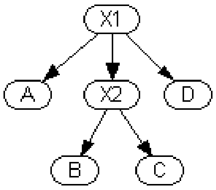

Figure 11: Model for the HIV data.

and X_2 are binary variables. This model fits data well ($L = 1.27$, $df = 4$, $p = 0.87$). Hagenaars examined several possible models and reached the conclusion that the loglinear model M2, where X is a binary variable, best explains the data. This model also fits the data very well ($L = 1.43$, $df = 5$, $p = 0.92$).

Using scoring metrics AIC, BIC, and CS, our algorithm found the model M3, where X_1 and X_2 are both binary variables. It's obvious that M3 is equivalent to M1 and hence fit data equally well. Our algorithm does not examine model M2 because it is not an HLC model. This is, however, no problem because M2 and M3 are almost identical as generative models for the manifest variables.

Using LS, our algorithm found a model that is the same as M3 except the cardinality of X_1 is 3. This model does not fit data well ($L = 1.27$, $df = 0$, $p = 0.0$).

6.3 The HIV test data

This data set consists of results on 428 subjects of four diagnostic tests for human HIV virus: "radioimmunoassay of antigen ag121" (A); "radioimmunoassay of HIV p24" (B); "radioimmunoassay of HIV gp120" (C); and "enzyme-linked immunosorbent assay" (D). A negative result is represented by 0 and a positive result by 1.

Alvord *et al.* (1988) reasoned that there should be two latent classes, corresponding to the presence and absence of the HIV virus. However, the two-class LC model does not fit data well ($L = 16.23$, $df = 6$, $p = 0.01$). This indicates the presence of local dependence.

The performance of our algorithm on this data set is similar to that on the Coleman data set. Using AIC, BIC, and CS, it found the model in Figure 11, where both latent variables are binary variables. The model is identical to one of the equivalent models Uebersax (2000) reached using some heuristic techniques. The model fit data well ($L = 3.056$, $df = 4$, $p = 0.548$).

With LS score, our algorithm produced the same model structure. However, the cardinalities of both latent variables are overestimated by 2. The model fits data poorly.

6.4 The house building data

The house building data are taken from a study by Hagenaars on people's view about what a new government should do. Again there are four binary manifest variables A, B, C, and D. Roughly speaking, A and C represent answers by respondents, in November and December 1970 respectively, to the question whether house building was an important problem. B and D represent the views of respondents, in November and December 1970 respectively, on how important house building was in relation to several other issues. We refer the reader to Hagenaars (1988) for the details.

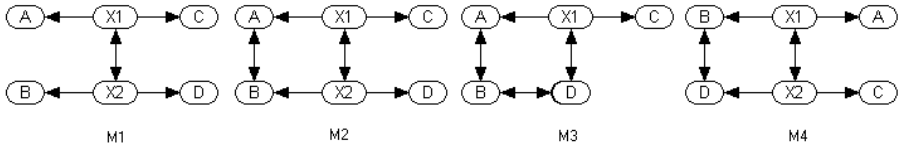

Figure 12: Models for the house building data.

When analyzing the data, Hagenaars started with the model M1 shown in Figure 12. In M1 and all the other three models, all latent variables are binary. The idea behind M1 is to capture the test-retest effects between the two interviews. This model does not fit data well ($L = 45.37$, $df = 4$, $p = 0.000$). Hagenaars went on to examine the standardized residuals and concluded that an direct effect should be added between A and B. This led to the model M2. This model fit data well ($L = 1.94$, $df = 3$, $p = 0.59$). Starting from some other initial models and adding direct effects properly among manifest variables, Hagenaars also derived the models M3 and M4, which fit data as well as M2.

It is clear that models M2, M3, and M4 are not close to any HLC models. Consequently, we cannot expect our algorithm to find satisfactory models for this data set. This turned out to be the case. Regardless of the scoring metric, our algorithm always found LC models. With BIC and CS it found 3-class LC models. With AIC and LS, it found 4-class LC models. None of the models fit data well.

To summarize the experiments with real-world data, we note that the performance of our algorithm on the Hannover rheumatoid arthritis data supports the claim made in the introduction that the algorithm would return LC models when there is no local dependence. The performances on the Coleman and HIV data sets support that claim that when local dependence is present, the algorithm would return HLC models with local dependence properly encoded. Finally, the performance on the housing building data shows the limitation of HLC models.

7 Conclusions and future directions

We have introduced a new class of models for cluster analysis, namely HLC models. HLC models are significantly more general than LC models and can accommodate local dependence. Yet they remain computationally attractive because of the simplicity in their structures.

A search-based algorithm has been developed for learning HLC models from data. Both synthetic and real-world data have been used to evaluate the algorithm with four different scoring metrics, namely AIC, BIC, CS, and LS. The results indicate that the algorithm works well with BIC and CS.

The models that our algorithm, with BIC or CS, reconstructed from synthetic data predicate testing data as well as the original model. Their structures are the same or equivalent to that of the original model, except in a couple of cases where minor differences exist probably due to insufficient data. The cardinalities of latent variables were, however, often underestimated. On three of the four real-world data sets, our algorithm found models that are considered optimal or close to optimal in the literature. However, it failed on the fourth data set, owing to the limitations of HLC models.

We end the paper with a list of issues that should be addressed. On top of the list is the issue of complexity. The focus of this paper has been on developing a principled search-based method for learning HLC models. Not much consideration was given to computational complexity. It is clear from Section 4.3 that, while not prohibitive, our algorithm is quite expensive computationally. To reduce the complexity one can use various heuristics to construct initial models for search that hopefully are close to the optimal. One can also replace hill-climbing that generates and evaluates a large number of candidate models at each step with heuristic search that inspects aspects of the current model that need improvement and constructs a model for the next step heuristically. Ideas from the vast literature on phylogenetic tree reconstruction, latent class analysis, and hidden variable discovery in Bayesian networks can all be brought to bear.

The second issue concerns scoring metric. Our experiments suggest that, among the four metrics examined, BIC and CS are the best for the task of learning HLC models. The derivation of CS depends on BIC. And the BIC score is derived based on the assumption that the likelihood function $P(D|m, \theta)$, when viewed as a function of parameters θ, is highly picked around the maximum likelihood estimate $\hat{\theta}$. As pointed out by Geiger *et al.*

(1996), this assumption is not necessarily true when latent variables are present. It does not seem to be true in our experiments. When running EM, we observed that the loglikelihood was hardly changing toward termination while the parameters were still changing significantly. Moreover, the use of BIC often leads to the underestimation of cardinalities of latent variables. Those observations raise the question whether there are scoring metrics for learning HLC models that are more suitable than BIC and CS. Some initial efforts in this direction are reported in Kocka and Zhang (2002).

In this paper we structure the search space into two levels. At the upper level the search is among different model structures, while at the lower the search is among different cardinalities for latent variables. One can also work directly with the natural one-level search space of all models. It would interesting to develop a search algorithm for the one-level search space and compare it with the one for the two-level space. Work in this direction might also lead to a proof of correctness for the algorithm developed in this paper.

Finally, we choose to study HLC models because they significantly generate LC models and are computationally attractive. As we have seen in Section 6, HLC models are well suited for some applications while inadequate for others. Sometimes more complex models are needed. The challenge is to keep computation feasible while considering more and more complex models.

Acknowledgement

Research was partially supported by Hong Kong Research Grants Council under grant HKUST6093/99E. I thank Tomas Kocka for insightful discussions on parsimonious and regular HLC models. I am also grateful to Finn V. Jensen, Thomas Nielsen, Kristian G. Olesen, Olav Bangso, Jose Pena, Jiri Vomlel, and Marta Vomlelova for valuable feedbacks on earlier versions of this paper.

References

[1] Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. Autom. Contr.*, 19, 716-723.

[2] Alvord, W. G., Drummond, J. E., Arthur. L.O., Biggar, R. J., Goedert, J. J., Levine, P. H., Murphy, E. L. Jr, Weiss, S. H., Blattner, W. A. (1988). A method for predicting individual HIV infection status in the absence of clinical information. *AIDS Res Hum Retroviruses*, 4(4):295-304.

[3] Bartholomew, D. J. and Knott, M. (1999). *Latent variable models and factor analysis*, 2nd edition. Kendall's Library of Statistics 7. London: Arnold.

[4] Bohrnstedt, G. W. and Knoke D. (1994). *Statistics for social data analysis (3rd Edition)*. F. E. Peacock Publishers Inc., Itasca, Illinois.

[5] Cheeseman, P. and Stutz, J. (1995). Bayesian classification (AutoClass): Theory and results. In Fayyad, U., Piatesky-Shaoiro, G., Smyth, P., and Uthurusamy, R. (eds.), *Advancesin Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA.

[6] Chickering, D. M. and Heckerman, D. (1997). Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning* 29(2-3): 181-212.

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

[8] Coleman, J. S. (1964). *Introduction to Mathematical Sociology*. London: Free Press.

[9] Connolly, D. (1993). Constructing hidden variables in Bayesian networks via conceptual learning. *ICML-93*, 65-72.

[10] Cover, T. M., Thomas, J. A. (1991). *Elements of Information Theory*, John Wiley & Sons.

[11] Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J. (1999). *Probabilistic networks and expert systems*, Springer.

[12] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B*, 39:1–38.

[13] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press.

[14] Eaton, W. W., Dryman, A., Sorenson, A., and McCutcheon, A. (1989). DSM-III Major depressive disorder in the community: A latent class analysis of data from the NIMH epidemiologic catchment area programme. *British Journal of Psychiatry*, 155, 48-54.

[15] Elidan, G., Lotner, N., Friedman, N. and Koller, D. (2000). Discovering hidden variables: A structure-based approach. *NIPS-01*.

[16] Elidan, G. and N. Friedman (2001). Learning the dimensionality of hidden variables. *UAI-01*.

[17] Espeland, M. A. and Handelman, S. L. (1989). Using latent class models to characterize and assess relative error in discrete measurements. *Biometrics*, 45, 587-599.

[18] Everitt, B. S. (1993). *Cluster Analysis*. London: Edward Arnold.

[19] Fraley, C. (1998). Algorithms for model-based Gaussian hierarchical clustering. *SIAM J. Sci. Comput.*, 20 (1), 270-281.

[20] Friedman, N., Ninio, M., Pe'er, I., and Pupko, T. (2002). A structural EM algorithm for phylogenetic inference. *Journal of Computational Biology*, to appear.

[21] Garrett, E. S. and Zeger, S. L. (2000). Latent class model diagnosis. *Biometrics*, 56, 1055-1067.

[22] Geiger, D., Heckerman, D., King, H., and Meek, C. (1998). Stratified exponential families: Graphical models and model selection. Technical Report MSR-TR-98-31, Microsoft Research.

[23] Gibson, W. A. (1959). Three multivariate models: Factor analysis, latent structure analysis, and latent profile analysis. *Psychometrika*, 24: 229-252.

[24] Goodman, L. A. (1974a). The analysis of systems of qualitative variables when some of the variables are unobservable. Part I-A Modified latent structure approach. *American Journal of Sociology*, 7(5), 1179-1259.

[25] Green, P. (1998). Penalized likelihood. In *Encyclopedia of Statistical Sciences*, Update Volume 2. John Wiley & Sons.

[26] Goodman, L. A. (1974b). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, 61, 215-231.

[27] Hagenaars, J. A. (1988). Latent structure models with direct effects between indicators: local dependence models. *Sociological Methods and Research*, 16, 379-405.

[28] Hanson, R., Stutz, J., and Cheeseman, P. (1991). Bayesian classification with correlation and inheritance. In *Proc. 12th. International Joint Conference on A.I. (IJCAI-91)*, 2, 692-698, Morgan Kaufmann.

[29] Kaufman, L. and Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis.* New York: John Wiley and Sons, Inc..

[30] Kishino, H., Miyata, T., and Hasegawa, M. (1990). Maximum likelihood inference of protein phylogeny and the origin of the chloroplasts. J. Mol. Evol. 31, 151-160.

[31] Kocha, T. and Zhang, N. L. (2002). Dimension correction for hierarchical latent class models. UAI-02, submitted.

[32] Kohlmann, T., and Formann, A. K. (1997). Using latent class models to analyze response patterns in epidemiologic mail surveys. Rost, J. and Langeheine, R. (eds.). *Applications of latent trait and latent class models in the social sciences*. Muenster: Waxman Verlag.

[33] Lanterman, A. D. (2001). Schwarz, Wallace, and Rissanen: Intertwining themes in theories of model order estimation. *International Statistical Review*, 69(2), 185-212.

[34] Lazarsfeld, P. F., and Henry, N.W. (1968). *Latent structure analysis*. Boston: Houghton Mifflin.

[35] Martin, J. and VanLehn, K. (1994). Discrete factor analysis: learning hidden variables in Bayesian networks. Technical Report LRGC_ONR-94-1, Department of Computer Science, University of Pittsburgh.

[36] Meila-Predoviciu, M. (1999). *Learning with mixtures of trees*, Ph.D. Dissertation, Department of Electrical Engineering and Computer Science, MIT.

[37] Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2), 461-464.

[38] Swofford, D. L. (1998). *PAUP* 4.0 - Phylogenetic Analysis Using Parsimony (*and Other Methods).*. Sinauer Assoc., Sunderland, MA.

[39] Uebersax, J. (2000). A practical guide to local dependence in latent class models. http://ourworld.compuserve.com/homepages/jsuebersax/condep.htm.

[40] Vermunt, J.K. and Magidson, J. (2000). *Latent GOLD User's Guide*. Belmont, Mass.: Statistical Innovations, Inc..

[41] Vermunt, J.K. and Magidson, J. (2002). Latent class cluster analysis. in Hagenaars, J. A. and McCutcheon A. L. (eds.), *Advances in latent class analysis*. Cambridge University Press.

[42] Wasmus, A., Kindel, P., Mattussek, S. and Raspe, H. H. (1989). Activity and severity of rheumatoid arthritis in Hannover/FRG and in one regional referral center. *Scandinavian Journal of Rheumatology*, Suppl. 79, 33-44.

Appendix: Real-world data sets

Back Pain	Neck Pain	Joint Pain	Swelling	Stiffness	Frequency
no	no	no	no	no	3,634
no	no	no	no	yes	73
no	no	no	yes	no	87
no	no	no	yes	yes	10
no	no	yes	no	no	440
no	no	yes	no	yes	89
no	no	yes	yes	no	106
no	no	yes	yes	yes	75
no	yes	no	no	no	295
no	yes	no	no	yes	25
no	yes	no	yes	no	15
no	yes	no	yes	yes	5
no	yes	yes	no	no	137
no	yes	yes	no	yes	42
no	yes	yes	yes	no	35
no	yes	yes	yes	yes	39
yes	no	no	no	no	489
yes	no	no	no	yes	37
yes	no	no	yes	no	23
yes	no	no	yes	yes	7
yes	no	yes	no	no	255
yes	no	yes	no	yes	116
yes	no	yes	yes	no	71
yes	no	yes	yes	yes	65
yes	yes	no	no	no	306
yes	yes	no	no	yes	48
yes	yes	no	yes	no	16
yes	yes	no	yes	yes	11
yes	yes	yes	no	no	229
yes	yes	yes	no	yes	162
yes	yes	yes	yes	no	44
yes	yes	yes	Yes	yes	176

Table 3: The Hannover rheumatoid arthritis data.

A	B	C	D	Coleman	HIV	House Building
0	0	0	0	458	170	193
0	0	0	1	140	15	44
0	0	1	0	110	0	26
0	0	1	1	49	0	34
0	1	0	0	171	6	103
0	1	0	1	182	0	77
0	1	1	0	56	0	15
0	1	1	1	87	0	58
1	0	0	0	184	4	58
1	0	0	1	75	17	16
1	0	1	0	531	0	32
1	0	1	1	281	83	48
1	1	0	0	85	1	84
1	1	0	1	97	4	54
1	1	1	0	338	0	60
1	1	1	1	554	128	283

Table 4: The Coleman, HIV, and housing building data sets.