Latent Structure Models for the Analysis of Gene Expression Data

Dong Hua
The George Washington University
Email: gwuhua@gwu.edu

Dechang Chen
Uniformed Services University of
the Health Sciences

Xiuzhen Cheng
The George Washington University
Email: cheng@seas.gwu.edu

Abdou Youssef
The George Washington University
Email: youssef@seas.gwu.edu

Abstract

Cluster methods have been successfully applied in gene expression data analysis to address tumor classification. By grouping tissue samples into homogeneous subsets, more systematic characterization can be developed and new subtypes of tumors be discovered. Central to cluster analysis is the notion of similarity between the individual samples. In this paper, we propose latent structure models as a framework where dependence among genes and thus relationship between samples can be modelled in a better way in terms of topology and flexibility. A latent structure model is a Bayesian network where the network structure contains at least a rooted tree including all variables, only variables at the leaf nodes are observed, and the structure after deleting all the observed variables is a rooted tree. The main gain in using latent structure models is that they provide a principled and systematic method to handle the dependence among genes. There are other benefits offered by latent structure models. They do not require any prior knowledge on the determination of tumor classes and choice of similarity metric, which are two important issues associated with the traditional clustering techniques. They are also computationally attractive due to the simplicity of their structures. We develop a search-based algorithm for learning latent structures model from microarrays. The effectiveness of the algorithm and the proposed models is demonstrated on publicly available microarray data.

1. Introduction

In model-based clustering, the objects under analysis are assumed to be generated by a finite mixture of probability distributions and one component corresponds to each class [8]. Cluster analysis is sometimes called latent class analysis [1, 4, 7, 9] when the attributes are categorical. Underlying such an analysis is the latent class (LC) model. Such a model is a simplest Bayesian network consisting of one class variable (unobserved, hence often called latent class variable) on the root with all the other observed variables on its children nodes as leaves. Each variable can take on several values. Each value of the latent class variable corresponds to one class, while each value of an observed variable corresponds to one of its states. The number of values a variable can take on is called its cardinality. Typically, one record of the observed data generated by the model consists of a combination of the values corresponding to certain states of some or all observed variables. Figure1 is an

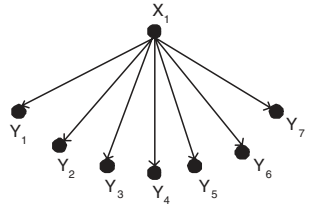

Figure 1. A LC model with one latent class variable X_1 and seven observed variables Y_1, $\cdots$, Y_7.

example of a LC model, which consists of one latent class variable X_1 and seven observed variables Y_i's.

One serious problem with the use of LC models, known as local dependence, is related to the assumption of conditional independence of observed variables given the latent class variable. This assumption often fails in reality and thus its use often weakens the performance of clustering analysis. It is clear that the better the local dependence is modelled, the higher the performance is achieved [9]. One recent study of modelling local dependence is conducted in [12]. The underlying model is hierarchical latent class

Proceedings of the Computational Systems Bioinformatics (CSB'03)
0-7695-2000-6/03 $17.00 © 2003 IEEE

(HLC) model, where the local dependence is handled by introducing new latent variables (Figure 2).

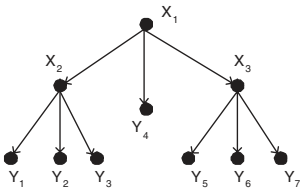

Figure 2. An HLC model obtained by introducing two additional latent variables X_2 and X_3 to the LC model in Figure 1.

When applying HLC models to gene expression data, the observed variables Y_i's denote the expression level of individual genes, and the latent variables denote other attributes (e.g., background variables) affecting the system. In doing so, HLC models may not be sufficient to model the dependence among genes. For example, there might be some interaction between the latent variable X_3 and the observed variable Y_3 in the model shown in Figure 2. But such an interaction effect cannot be addressed by the described HLC model.

In this paper, we propose latent structure models to analyze gene expression data. A latent structure model is a Bayesian network where the network structure contains at least a rooted tree including all variables, only variables at the leaf nodes are observed, and the structure after deleting all the observed variables is a rooted tree. Latent structure models allow latent variables to influence the observed variables interactively and include both LC and HLC models as the special cases. Latent structure models provide a principled and systematic method to handle the dependence among genes. They can lead to a better understanding of genes' expression and improve the analysis of microarrays.

The rest of paper is organized as follows. In Section 2, we introduce latent structure models and describe the techniques in learning the models. In Section 3, we present an application of latent structure models to clustering samples of the well-known leukemia data set. Our conclusion is given in Section 4.

2. Latent Structure Models

A *latent structure (LS) model* is a Bayesian network where

1. The network structure contains at least a rooted tree including all variables with the observed variables on the leaf nodes only;
2. The variables at the leaf nodes are observed while all the other variables are not;
3. The network structure after removing all the observed variables and their corresponding connections is a rooted tree[1]; and
4. No connections are allowed among leaf nodes.

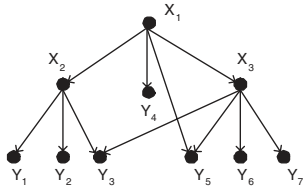

Figure 3. A LS model with three latent variables X_1, X_2, X_3 and seven observed variables $Y_1, \cdots, Y_7$.

An example of a LS model is shown in Figure 3. It contains a rooted tree (as in Figure 2) consisting of all the variables with only observed variables on the leaves. By removing all the leaf nodes and their corresponding connections, the rest of the network structure is also a rooted tree shown in Figure 4.

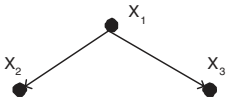

Figure 4. The network structure after removing all the observed variables and their corresponding connections from the LS model in Figure 3.

For convenience, a LS model is denoted by M, which contains the information of the topology and cardinalities of latent variables. We use $\mathbf{Core}(M) = \{C_1, C_2, \cdots, C_k\}$ to represent the set of HLC models contained by a LS model. For instance, $\mathbf{Core}(M)$ for the LS model in Figure 3 contains 4 HLC models. One of them is actually the HLC model in Figure 2. We also use internal(M) to indicate the remaining model after removing all the observed variables and the corresponding connections from M.

[1]The connection between two nodes is denoted by an arrow in the Bayesian network.

2.1. An Algorithm For Learning LS Models

This subsection describes a search-based algorithm for learning LS models from data. It distinguishes in two parts:

- **Part 1:** Learn the optimal $\mathbf{Core}(M)$ given observed variables;
- **Part 2:** Learn the optimal LS model based on the optimal $\mathbf{Core}(M)$.

The same algorithm for learning HLC models can be used for Part 1. As suggested by [12], the search space is structured into two levels according to the following two sub-tasks:

- **Sub-task 1**: Search the the optimal cardinalities for the latent variables given model structure;
- **Sub-task 2**: Search the optimal model.

A natural way of designing the search operators can be achieved by performing these two sub-tasks. Such restructuring of search space can also be applied in Part 2. For both parts, the EM algorithm and BIC score are used for parameter learning and model scoring respectively. The search control is the same for both parts. Figure 5 illustrates the

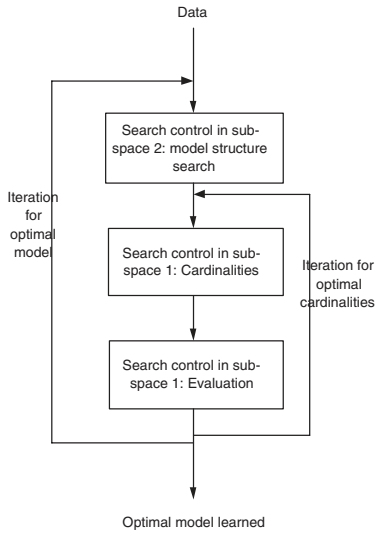

Figure 5. Search control in both Part 1 and Part 2.

sub-spaces corresponding to each of the two sub-tasks. The only difference between the two parts in terms of search control is that the search control of Part 1 starts from a LC model or some other HLC model, while the search control of Part 2 starts from $\mathbf{Core}(M)$ of Part 1.

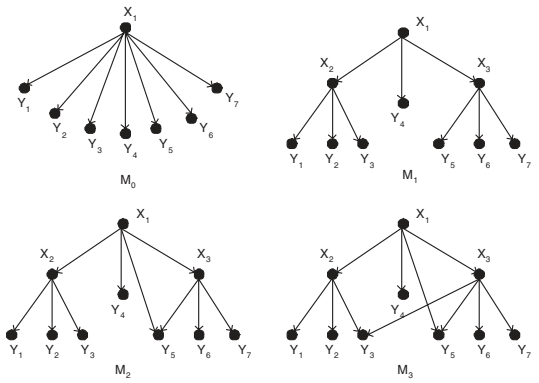

Figure 6. Illustration of the search operator *arrow addition.*

To conduct the search in Part 2, an operator called *arrow addition* may be introduced. Figure 6 illustrates how *arrow addition* works. To reach M_2 from M_1, one simply adds an arrow between X_1 and Y_5. This is what the search operator *arrow addition* does. Note that M_1 cannot reach M_3 by only one step. To reach M_3 from M_1, several steps are needed. We first reach M_2 from M_1 and then reach M_3 using *arrow addition* again.

For learning the optimal LS model based on $\mathbf{Core}(M)$, a search-based hill climbing method is typically used. In general, we only consider LS models where the upper bound of the number of parents for any leaf variable is prespecified k, namely only less than or equal to k connections to internal(M) for each leaf variable. We check each of the leaf variables and generate a number of candidate LS models. Such candidate models are generated by adding a connection (using the *arrow addition*) between the leaf variable under investigation and each of the variables in internal(M). The best candidate LS model will be selected as the seed for the next search, and we keep going until the optimal LS model is learned. Note that we only check the leaf variables whose connections to the internal(M) are bounded by k.

3. Analyzing Expression Data

In this section we investigate the effect of LS models on clustering gene expression data. We used the leukemia data set [6]. This data set contains gene expression levels from Affymetrix high-density oligonucleotide arrays. Included in the data set are 6817 genes, 47 cases of ALL, and 25 cases of AML. For a simple illustration, we employed gene selection method described in [3] to choose the top 40 genes by using the correlation threshold $\theta = 0.2$ and the Brown-Forsythe test statistic. Table 1 shows the genes selected in our experiment. Since we are dealing with the discrete

Table 1. The genes selected in this experiment

$gene2642$	$gene4050$	$gene4847$	$gene3938$
$gene1525$	$gene6581$	$gene451$	$gene312$
$gene4280$	$gene2528$	$gene2767$	$gene4194$
$gene3837$	$gene1005$	$gene1975$	$gene3122$
$gene5734$	$gene4662$	$gene7078$	$gene1582$
$gene128$	$gene2425$	$gene2434$	$gene2893$
$gene2365$	$gene2766$	$gene6857$	$gene1647$
$gene1139$	$gene3556$	$gene5773$	$gene4054$
$gene1708$	$gene3474$	$gene252$	$gene190$
$gene5327$	$gene2313$	$gene6297$	$gene3554$

LS models, data discretization is needed. To discretize the leukemia data, we use the median method described as follows. Choose one median for a record of data if we want to discretize it into two categories, say '1' and '0'. The value greater than or equal to the median is set to be '1' and '0' otherwise. Generally, $n-1$ medians are needed to discretize the data into n categories. In our experiment, we chose the eight categories.

We applied LS models to leukemia data set by choosing EM threshold to be 0.01 for model selection and $1e-8$ for the final model parameter learning. Out of the 72 samples, only one sample (AML) is misclassified. The correct rate is 98.6%. CLIFF [11] misclassified three ALLs. The corresponding correct rate is 95.8%. [10] analyzed this data and misclassified two ALLs, but only one AML was misclassified if utilizing the sample pathological phenotype. We have noted that the performance of LS models was affected by various factors such as the categories of discretization of the data and the number of genes selected.

4. Conclusion

We propose latent structure models for gene expression data analysis in this paper. The dependence among genes is naturally handled by these models. Latent structure models are generalization of both LC and HLC models. Experimental results show that latent structure models are effective in clustering gene data. Latent structure models are new models introduced for gene data analysis. We only provide some preliminary results in this paper. Further research will be continued.

References

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

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

[3] D. C. Chen, D. Hua, X. Z. Cheng, and J. Reifman. Gene selection for multi-class prediction of microarray data. IEEE Computer Society Bioinformatics Conference, accepted, 2003.

[4] P. R. Lazarsfeld, and N. W. Henry. *Latent Structure Analysis*. Boston:Houghton-Mifflin, 1968.

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

[6] T. R. Golub, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286, 531-537, 1999.

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

[8] Y. Y. Ka, F. Chris, M. Alejandro, E R. Adrian, and L R. Walter. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17:977-987, 2001.

[9] Uebersax. A Practical Guide to Local Dependence in Latent Class Models, 2000. our-world.compuserve.com/homepages/jsuebersax /condep.htm.

[10] W. J. Li, M. Fan, and M. M. Xiong. SamCluster:an integrated scheme for automatic discovery of sample classes using gene expression profile. *Bioinformatics*, Vol. 19, No. 7 2003, Pages 811-817, 2003.

[11] E. P. Xing, and R. Karp. CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, 17, S306-S315, 2001.

[12] N. L. Zhang. Hierarchical Latent Class Models for Cluster Analysis. *AAAI*, 230-237, 2002.