LPCC – Learning Latent Variable Models by Pairwise Cluster Comparisons, A Toolbox for Matlab

(based on Asbeh, N. and Lerner, B. (2012). "Learning latent variable models by pairwise cluster comparison", *4th Asian Conference on Machine Learning* (*ACML2012*), JMLR Workshop & Conference Proceedings, Singapore, Vol. 25, 33-48)

Written by Nuaman Asbeh

Version 1, Nov 23 2014

Prerequisites

LPCC requires the installation of two toolboxes:

1. FullBNT – A general Bayes Net toolbox, available at https://code.google.com/p/bnt/
2. SOM Toolbox – A Self-Organizing Map (SOM) algorithm toolbox, available at http://www.cis.hut.fi/somtoolbox/

How to use?

LPCC can be run in two ways that differ only by the applied approach of clustering, which is a pre-processing stage to LPCC. The first way is by manual clustering (performed prior to running LPCC) in which the user applies their favorite clustering algorithm, analyzes the clustering results, and extracts the major clusters to be inputted to LPCC. We have good experience with SOM, although any clustering algorithm that does not need to determine the number of clusters beforehand is good as well. The second way is by letting LPCC cluster the data itself. Clustering is by applying SOM to the raw data followed by clustering with *k*-means with different values for *k* (taking the trained SOM as an input). The *k*-means algorithm is run multiple times with random initializations for each *k*, and the one that reduces the sum of squared errors the most is selected for that *k*. Then, the Davies-Bouldin index is calculated for each clustering result to choose the best value of *k*. That is, because the SOM results need to be interpreted to identify the major clusters, we could to it either manually (first way) or automatically (second way), e.g., by using *k*-means.

Our own experience is that in automatic clustering, the results are less accurate, because *k*-means "forces" all data points to be assigned to a cluster, whereas in the manual clustering, some of the "noisy" points that are far from all centroids may be left out of the clustering result. However, automatic clustering using *k*-means makes LPCC's run easy and fast. Therefore, although we recommend manual clustering, we also recommend the user to consider both ways before deciding on that that is most appropriate for their needs

1. Running LPCC when clusters are inputted by the user

[pdag,Observed,Latent]=LPCC(data_file,observed_cardinality,clusters,clusters_size)

Input:

data_file: A text file in which rows and columns represent samples and values of *m* observed variables, respectively.

observed_cardinality: A vector with the cardinalities of the observed variables.

clusters: A matrix *n*X*m* that represents clusters' centroids.

cluster_size: A vector of length *n* with the clusters' sizes.

Output:

pdag: The learned pattern by LPCC, represented as a **V**x**V** matrix, where $\mathbf{V} = \mathbf{O} \cup \mathbf{L}$ is the set of the observed **O** and latent **L** variables in the learned pattern. The value of pdag in row *i* and column *j* is 1 if there is a directed edge from V_i into V_j in the learned pattern and 2 if there is an undirected edge between V_i and V_j in the learned pattern.

Observed: A vector of the observed variables in pdag, each is represented as a Matlab structure.

Latent: A vector of the latent variables in pdag, each is represented as a Matlab structure.

2. Running LPCC when clustering is performed by the code

[pdag,Observed,Latent]=LPCC(data_file,observed_cardinality)

Input/Output:

Similar to (1).

Examples

In the LPCC Toolbox, we provide three examples: LPCC_Example1.m, LPCC_Example2.m, and LPCC_Example3.m for running LPCC to learn graphs G1, G2, and G3 of Figure 1, respectively. For example, Figure 2 shows the pdag learned by LPCC for G1. All examples use simulated data sets of size 1,000, which were generated from the three graphs with binary variables and are provided in the toolbox. The priors on the exogenous latents were distributed uniformly, and the conditional probabilities between a latent L_k and each of its endogenous children EN_i (either a latent or an observed) are $P(EN_i = v|L_k = v)$ =0.8, $v = 0$ or 1, except of L4 in G3, for which $P(L_4 = 0|L_3L_5 = 00,01,10) = P(L_4 = 1|L_3L_5 = 11)$ =0.8.

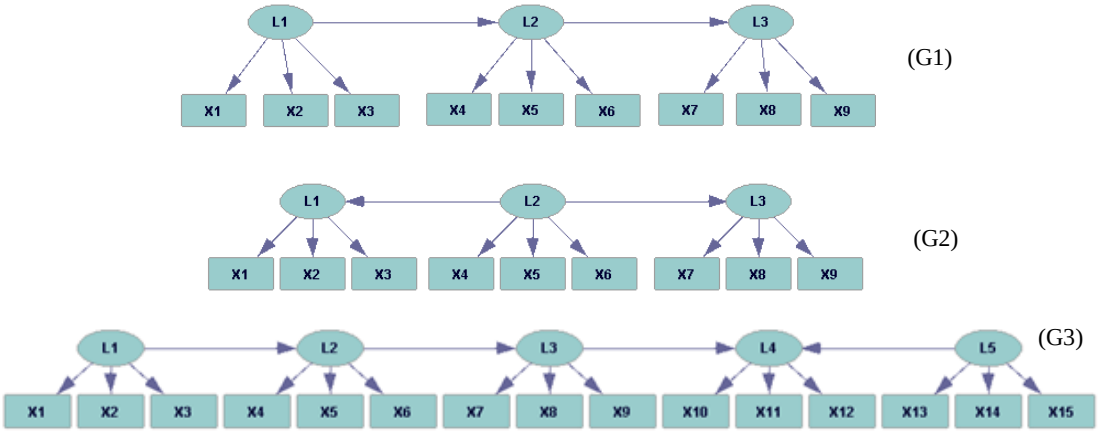

Figure 1: Three example latent variable models, which are multiple indicator models, learned using LPCC.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	0	0	0	0	0	0	0	2	0
11	0	0	0	1	1	1	0	0	0	2	0	2
12	0	0	0	0	0	0	1	1	1	0	2	0

Figure 2: The pdga learned by LPCC for G1 (Figure 1). Indices 1-9 correspond to the observed variables X1-X9, and indices 10-12 correspond to the latent variables L1-L3. Note that this pdag represents a pattern over the structural model (connections among latent variables) of G1, because this model is of a serial connection.