Algorithm for learning latent variable models

Yaniv Malowany, Dan Halbersberg, Boaz Lerner

In most domains, there are two types of variables — observed and unobserved (latent). While modeling with only the first type is straightforward, learning a model that involves both is much more difficult. Modeling with only observed variables in the presence of latent variables will likely lead to a wrong model that does not describe the true realm and may also be misleading. Hence, the identification of latent variables and the relationships between them and between those that are observed is crucial.

Latent variable models (LVMs) aim to demonstrate latent variables in the problem that are manifested by some indicators (observed variables). The learning pairwise clusters comparison (LPCC) algorithm [1,2] learns an LVM without prior knowledge about the number of latent variables in the model and their expected relations. However, since it is initialized using cluster analysis, its performance depends on the selected number of clusters. Thus, selecting the correct number of clusters is essential for the success of the algorithm.

In this work, we examined methods to find the optimal number of clusters and used two synthetic networks (Figure 1) for the methods' empirical evaluation. The goal was to find correspondence between a structural correctness measure (known as the structural Hamming distance, SHD), which evaluates the LPCC-derived structure (but depends on the true graph that is known only for synthetic problems), and a clustering evaluation method that does not depend on knowing the true graph. By evaluating this correspondence, we aimed at establishing a recommendation procedure to select among different clustering schemes the optimal one. We tested two types of methods and their accompanying performance measures: clustering performance measures represented by the Davies-Bouldin index [3] and Silhouette measures [4], and data-fitness performance measures represented by the log-likelihood function [5] and Kullback-Leibler (KL) divergence [6].

We performed two experiments. In the first experiment, we examined the correspondence between the four evaluated measures and the SHD with respect to the recommended number of clusters for a randomly initialized LPCC over a single dataset generated from each of the two networks of Figure 1. Figure 2 shows that the KL divergence for a thousand runs recommends for ten clusters in G1, which is equal to the lowest SHD value that we get, and fifteen clusters in G2, which is in the range of the lowest SHD. Moreover, we can see that as we increase the number of runs, the correspondence between the KL divergence and SHD score improves. Also, we can say that there is no correspondence between the other measures and the SHD, where they recommend for numbers of clusters that do not bring the best performance for the LPCC.

In the second experiment, we re-sampled 100 datasets from each of the two LVM structures of Figure 1, and repeated the evaluation of the LPCC performance over 100 algorithm initializations for a different number of clusters using the structural correctness SHD (ultimate) measure and the four evaluated measures. Figure 3 shows that the only measure that gave us a correspondence with the SHD is the KL. For G1, the correspondence is not perfect; while the best number of clusters we get from the SHD score is seven, the KL divergence recommends for six cluster. Although the correspondence is not perfect, we still in the range of the minimum value for the SHD score, and it is still a good result. In G2, we can see more convincing results, where we see a nearly perfect match between the KL and the SHD scores; the behavior of the two lines is almost identical, and the lowest number of clusters that recommended by the SHD is also the lowest number of clusters recommended by the KL.

Our empirical results show that for the two sampled networks, the best evaluation method of the clustering schemes is that based on the KL divergence. The KL divergence – measuring the probabilistic difference between the distribution derived from the learned graph and the sample distribution – showed in our experiments a strong correspondence with the LVM structural correctness measure, and therefore is recommended, in real-world problems in which the true graph is unknown, for the initialization of the LPCC algorithm.

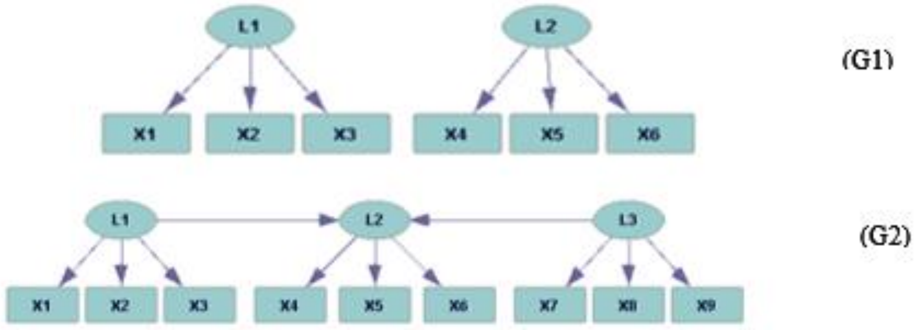

Figure 1: LVMs that are used in our experiment. Each is based on a pure measurement model and a structural model of different complexity, posing a different challenge to a learning algorithm.

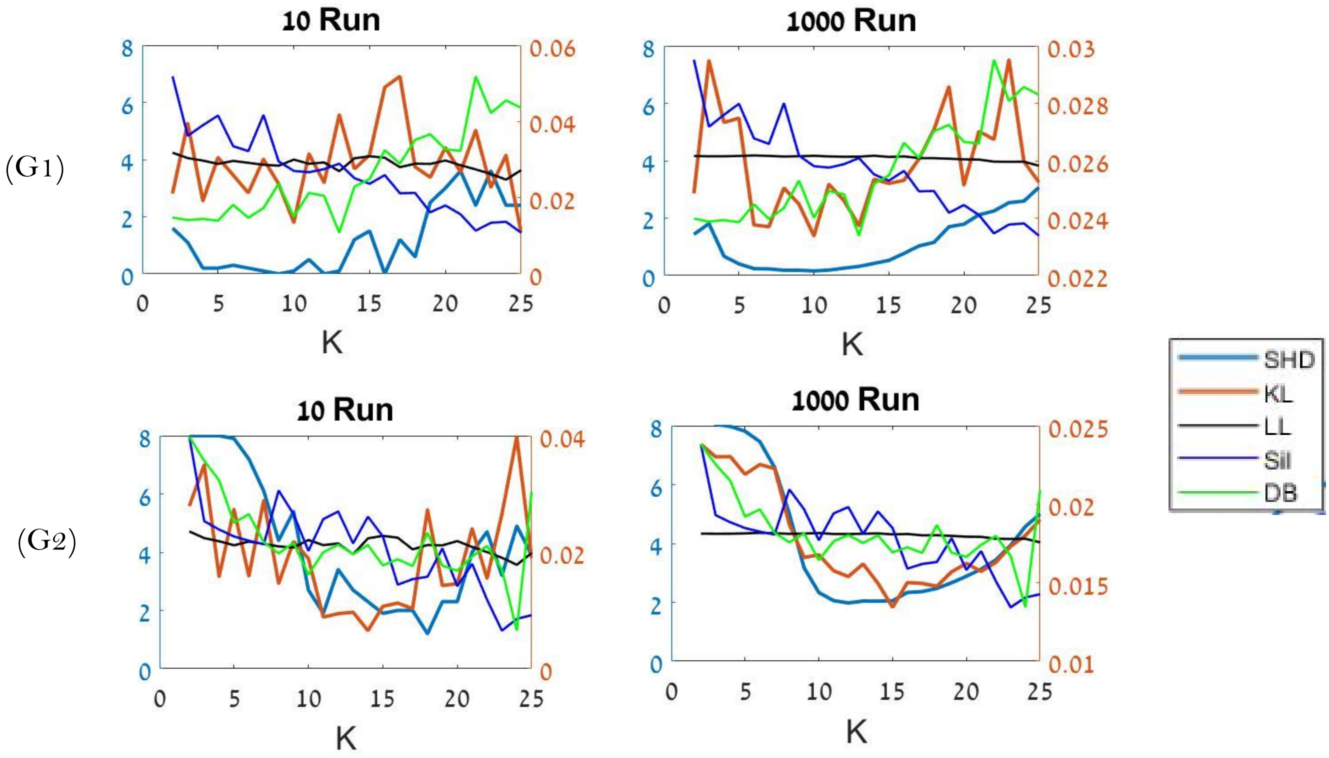

Figure 2: **Experiment 1 results for G1 *and G2***

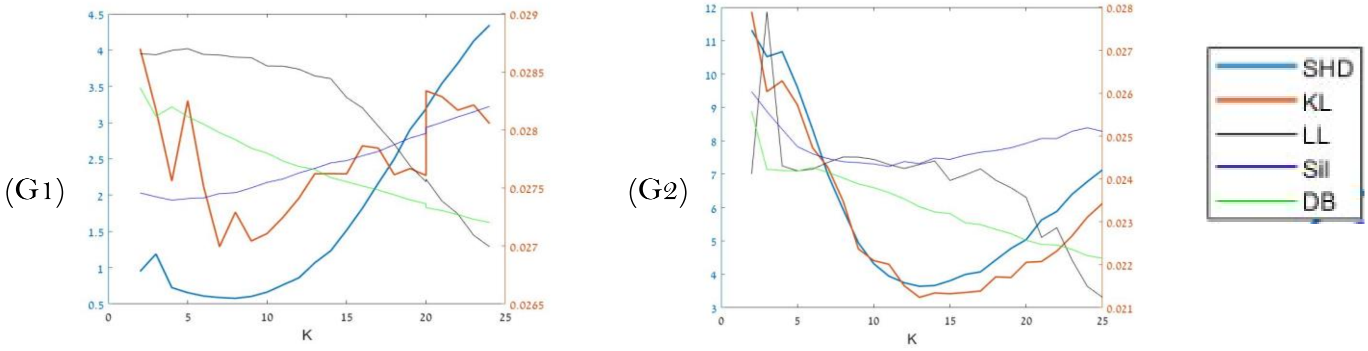

Figure 3: **Experiment 2 results for G1 *and G2***

[1] Asbeh, N., and Lerner, B. (2016). Learning latent variable models by pairwise cluster comparison: part I – theory and overview. *Journal of Machine Learning Research, vol. 17(224), pp. 1–52.*

[2] Asbeh, N., and Lerner, B. (2016). Learning latent variable models by pairwise cluster comparison: part II – Algorithm and Evaluation*. Journal of Machine Learning Research, vol. 17(233), pp. 1–45.*

[3] Davies, D.L., and Bouldin, D.W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, pp. 224–*227.

[4] Rousseeuw P (1987). Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis*. J. Comput. Appl. Math., 20, pp. 53-65*.

[5] Fisher, R. A. (1920). A Mathematical Examination of the Methods of Determining the Accuracy of an Observation by the Mean Error, and by the Mean Square Error.

[6] Seghouane, Abd-Krim, and Shun-Ichi Amari (2007). The AIC criterion and symmetrizing the Kullback–Leibler divergence. *IEEE Transactions on Neural Networks 18.1 97-106.*