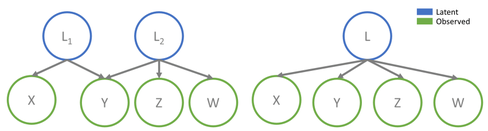
MCMC Estimation of Causal VAE Architectures With Applications to Spotify User Behavior

Alice Harting, supervised by Liam Solus

KTH Royal Institute of Technology

Background

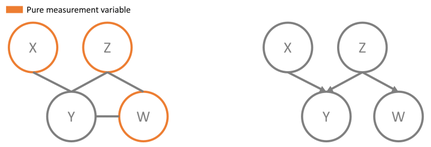
At Spotify, we have a set of metrics that reflect aspects of the user experience (e.g. number of tracks added to playlists). While often correlated, these are typically not directly causally related. Rather, we may assume that unobserved common causes explain covariance. This type of system is suitably modeled by *Neuro-Causal Factor Analysis (NCFA)* [4]. It is a Variational Autoencoder (VAE) [2] constrained to abide by observed marginal independence relations (Figure 1).



(a) NCFA (b) Standard VAE

Figure 1. Comparison of graphical model

The NCFA graphical model can be constructed from the *Unconditional Dependence Graph (UDG)* which encodes the marginal dependence relations. The original NCFA framework uses independence testing to inform the UDG, but this does not guarantee a unique NCFA graphical model.



(a) UDG $\mathcal{U}$ (b) DAG $\mathcal{D}$ in UEC

Figure 2. UEC

A sufficient condition for uniqueness is that each UDG clique (complete subgraph) has a *pure measurement variable*, i.e. a variable contained in no other clique [4]. This is satisfied if there exists a Bayesian network (probabilistic DAG) over the observed variables with the encoded marginal independence relations [1].

We note that the marginal independence relations partition the space of DAGs into *Unconditional Equivalence Classes (UECs)*, each of which is represented by a UDG. Thus, if the UDG represents a non-empty UEC, the associated NCFA graphical model is identifiable (Figure 2) [1].

MCMC algorithm to estimate UEC

Our contribution is a Metropolis-Hastings (MH) algorithm targeting the posterior of DAGs $\pi(\mathcal{D} \mid X)$ over the observed variables. We integrate up to UEC and obtain the maximum a posteriori (MAP) estimate. Since the estimated UEC is non-empty, the associated NCFA graphical model is identifiable. In addition to the MAP estimate, we rank the encountered DAG models by BIC-score ($\texttt{bic}_{\mathcal{D}} := -2\ln \hat{L}(X \mid \mathcal{D}) + \lambda(k \ln N)$, $k = |E^{\mathcal{D}}|, \lambda > 0$) and return the UEC of the optimal DAG. To compute $\hat{L}(X \mid \mathcal{D})$, we assume a linear Gaussian model and use linear regression methods.

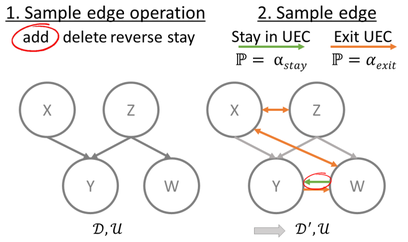
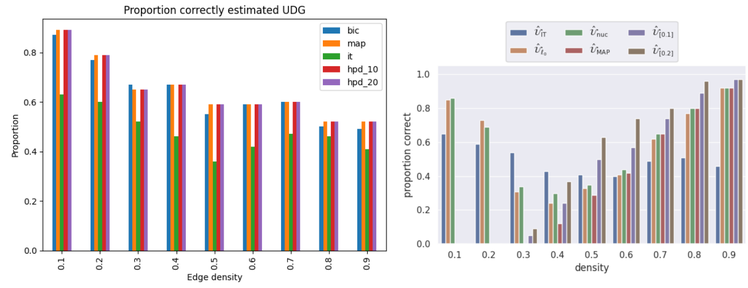


Figure 3. Proposal kernel for MH algorithm targeting $\pi(\mathcal{D} \mid X)$

To generate a proposal sample $\mathcal{D}'$ given the current state $\mathcal{D}$, we uniformly sample an edge operation and a target edge (Figure 3). The probability distribution of target edges (governed by $\alpha_{stay}, \alpha_{exit}$) can be tuned to promote or discourage UEC transitioning using the transformational characterization of unconditional equivalence given in [3]. Furthermore, we replace the likelihood $L(X \mid \mathcal{D})$, which is used in the MH acceptance probability, with $\texttt{exp}[-\texttt{bic}_{\mathcal{D}}]$ to compensate for unequal degrees of freedom.

Performance on fully observed DAG models (GrUES benchmark)

We benchmark our algorithm against GrUES [1], which is another MH algorithm estimating the UEC from data (Figure 4). We generate data from 100 fully observed DAG models on five nodes. The simulation setting is identical to Figure 11 in [1]. The BIC and MAP estimates using our algorithm are competitive or superior for DAG edge densities up to 0.7 and inferior thereafter.



(a) Our algorithm (b) GrUES [1] (Figure 11)

Figure 4. Performance on fully observed DAG models on five nodes

Tuning $\lambda, \alpha_{stay}, \alpha_{exit}$ improves performance to be competitive or superior in general (see paper).

Hyperparameters: deep dive

Performance for hyperparameter configurations (empty DAG initialization) is given in Figure 5.

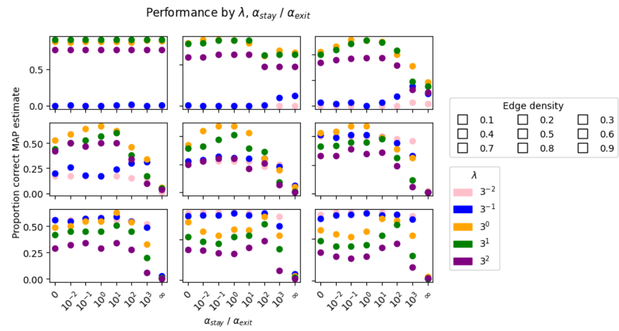
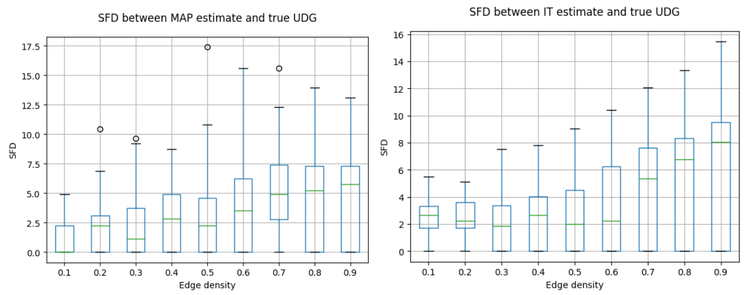


Figure 5. Performance across combinations of hyperparameters, empty DAG initialization

The default setting ($\lambda = \frac{\alpha_{stay}}{\alpha_{exit}} = 1$) works well up to dense DAGs, for which a smaller λ (less penalization on number of edges) is preferable. Conditional on λ, most performance curves morphs into a convex shape for dense DAGs. This could be explained by the large cardinality of UECs of dense UDGs: promoting UEC transitioning helps to move from the empty UDG to a dense UDG, while discouragement helps if we enter the correct UDG at a poorly scoring DAG.

Performance on partially observed NCFA models (IT benchmark)

With data generated from an NCFA model, the algorithm is competitive with independence testing (IT) in default mode (Figure 6). Tuning leads to outperformance (see paper).



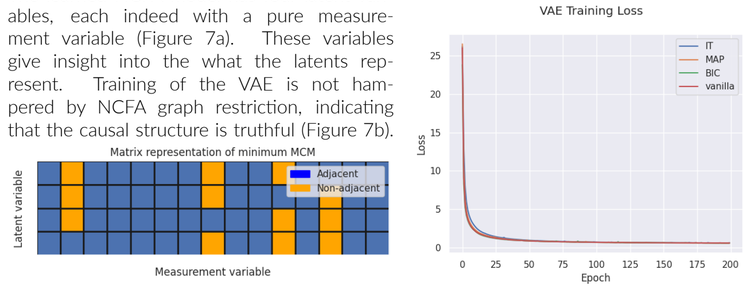
(a) MAP estimate (b) Independence testing estimate

Figure 6. Performance on partially observed NCFA models on ten measured variables (lower SFD is better)

Population clustering on the level of latent causes

We apply the NCFA framework to Spotify user behavior metrics (variable names are protected).

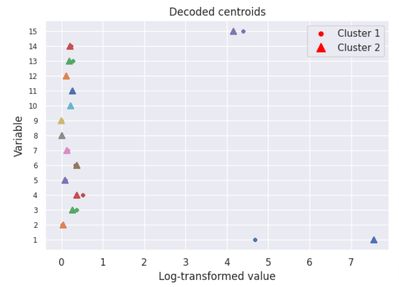
The learned NCFA model has four latent variables, each indeed with a pure measurement variable (Figure 7a). These variables give insight into the what the latents represent. Training of the VAE is not hampered by NCFA graph restriction, indicating that the causal structure is truthful (Figure 7b).



(a) Adjacency matrix of NCFA graphical model (b) VAE training loss

Figure 7. NCFA model estimation

Using the NCFA model, we encode the data and split the users into two groups using KMeans.



The center points are decoded back into the measured space. We observe distinctive differences in Variable 1, 3, 4, 13, and 15; thus, to move these metrics in experimentation, one could target the associated latents. The pure measurement variables in Figure 7a could help inform the treatment design in such an experiment.

Figure 8. Characterization of user clusters

Latent treatment effects in randomized control trial

Furthermore, we observe treatment effects in the learned latent space in a real experiment.

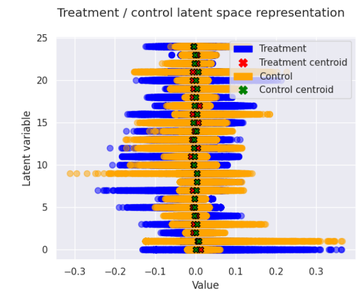


Figure 9. Treatment effects in the learned latent space

We train a separate VAE for the treatment and control group respectively, while using a shared NCFA graphical model learned on control data. The encoded data is illustrated in Figure 9. We note some differences between the groups, in particular for the upper latents. Given information about the treatment, this offers more insight into the latents. However, as shown when compared to a simulated randomized control trial with treatment applied to the latent variables (see paper), we cannot rule out that the observed treatment effects in this example are just random.

References

[1] Danai Deligeorgaki, Alex Markham, Pratik Misra, and Liam Solus. Combinatorial and algebraic perspectives on the marginal independence structure of bayesian networks, 2022.

[2] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[3] Alex Markham, Danai Deligeorgaki, Pratik Misra, and Liam Solus. A transformational characterization of unconditionally equivalent bayesian networks, 2022.

[4] Alex Markham, Mingyu Liu, Bryon Aragam, and Liam Solus. Neuro-causal factor analysis, 2023.