Causality Chain

August 2020

1 Architecture

Given a dataset of unlabeled samples $\{x \in \mathbb{R}^d\}$, we would like to order the coordinates $1..d$ using some permutation π such that if $\pi(i) < \pi(j)$ then variable (coordinate) i causes variable j.

We train a GLANN [1] architecture that will allow us to sample new data from the exact same distribution:

$$x = G(z),\ z \sim T(e),\ e \sim \mathcal{U}([0,1]^h) \tag{1}$$

such that $e \in \mathbb{R}^h, z \in \mathbb{R}^l$ and $h < l < d$. We use a regular MSE loss $l(x, \tilde{x}) = \|x - \tilde{x}\|_2^2$ instead of GLANN's [1] proposed "perceptual loss" because our data is tabular. During training we also recover a matrix C such that

$$C_{ij} = \begin{cases} 1 & \text{if } \pi(i) < \pi(j) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

2 Loss terms

We consider 3 causality classes:

- **Chain** $x_i \rightarrow x_j \rightarrow x_k$
- **Fork** $x_i \leftarrow x_j \rightarrow x_k$
- **Collider** $x_i \rightarrow x_j \leftarrow x_k$

These cases are classically used in Bayesian Network inference. The basic assumptions are that in both **Chain** and **Fork** cases - x_i, x_k are conditionally independent, given x_j. The **Collider**, however, guarantees that x_i, x_k are marginally independent.
We would like to create a constraint that allows the model learn the connections from the data, with respect to the causality networks classes.
For that we use the Squared Conditional Correlation (SCC) for all **Forks** and **Chain**, and Squared Correlation (SC) for all **Colliders**. By minimizing these term we try to ensure that each three variables are to fall into their respective causal class. [1]

However - we want to force the model to learn that either the variables are correlated **or** they are disconnected. We introduce a "xnor logic gate loss" that forces one of its terms to 1 and the other to 0:

$$\text{xnor}(a; b) = |1 - a - b - a \cdot b|$$

[1] Similarly - instead of using squared correlation we could use Mutual Information

For this function to evaluate to 0 - one of these variables has to be equal to 1 while the other one has to be equal to 0. It essentially represents a distance of two variables from being mutual exclusive. Using these prerequisites, we engineer the model loss accordingly:

- **Chain**, **Fork** both get the same loss of $L_{indep} := \text{xnor}(C_{ij}C_{jk}; Corr^2(x_i, x_k|x_j))$
- **Collider** gets a loss of marginal correlation $L_{indep} := Corr^2(x_i, x_k)$

However, a causal connection cannot be classified as both **Fork** / **Chain** and a **Collider**.In this case we also add a constraint between **Fork** / **Chain** and **Collider** in a form of "and logic gate", which is just a multiplication of both terms.
In the end, we're left with:

$$L_{independence} = \sum_{i,j,k} \mathbb{E}_x \left[\text{xnor}((C_{ij} + C_{ji})C_{jk}; Corr^2(x_i, x_k|x_j)) \cdot Corr^2(x_i, x_k)\right] \tag{3}$$

Finally, there are constraints on the structure of C. During optimization we make sure that $0 \leq C_{ij} \leq 1$ for all (i, j). Later on we can add terms to encourage the values to be either one or zero.

We then enforce transitivity via a loss term

$$L_{\text{trans}} = \sum_{i \neq j, j \neq k, i \neq k} (C_{ij}C_{jk})(1 - C_{ik})$$

3 Efficient computation of the transitivity constraint

The transitivity constraints has $O(d^3)$, requiring an efficient computation . We therefore express it in a matrix multiplication form.

Lev: it's still $O(d^3)$, at best $O(d^{2.807})$. It's just simplified.

First, we note that $1 - C_{ik}$ is just C_{ki}.

$$\begin{aligned} L_{\text{trans}} &= \sum_k \sum_{i \neq k} \sum_{\substack{j \neq i \\ j \neq k}} (C_{ij}C_{jk})(1 - C_{ik}) \\ &= \sum_k \sum_{i \neq k} (1 - C_{ik}) \left(\sum_j C_{ij}C_{jk} - C_{ii}C_{ik} - C_{ik}C_{kk} \right) \end{aligned}$$

However, from the definition of the connection matrix C, we have $C_{ii} = C_{kk} = 0$ and so the rightmost terms vanish and we are left with a simpler

$$L_{\text{trans}} = \sum_{i \neq k} (1 - C_{ik}) \sum_{j=1}^{n} C_{ij}C_{jk}$$

The rightmost sum is actually an expression for matrix multiplication:

$$\begin{aligned} L_{\text{trans}} &= \sum_{i \neq k} (1 - C_{ik}) \sum_{j=1}^{n} C_{ij}C_{jk} \\ &= \sum_{i \neq k} (1 - C_{ik})(C^2)_{ik} \\ &= \sum_{i,j} (C^2)_{ij}(1 - C_{ij}) \end{aligned}$$

where $i \neq j$ is removed since $C_{ii} = 0$.

4 Modeling the connectivity matrix

The connection matrix is constructed ”softly”, using a sigmoid on some parameter matrix C':

$$C'_{ij} \sim \mathcal{N}(0, 1)$$
$$C_{ij} = \sigma(C'_{ij}),$$

where σ is the sigmoid function. In our case - independence loss acts as an objective loss in our domain, while transitive loss acts as a regularization loss on the connection matrix C.

5 Inferring order

Recall that C_{ij} indicates if $\pi(i) < \pi(j)$. The first in order (root cause) would be smaller than anyone else. More generally, the order is determined by sorting the following causal score

$$s_i = \sum_{j \neq i} C_{ij}$$

The higher the score s_i , the earlier the i-th variable is in the causal order.

References

[1] Yedid Hoshen, Jitendra Malik. *Non-Adversarial Image Synthesis with Generative Latent Nearest Neighbors.* https://arxiv.org/abs/1812.08985