Scalable Causal Graph Learning through a Deep Neural Network

By Ethan Pellegrini

Objective

The objective of this project is to study the relationships between graphs using a specific form of causal inference, regression analysis, and low-rank approximation. The experiments described in this paper were run on a variety of real-world datasets, but for the sake of this project, the two synthetic datasets were used as benchmarks for this project.

Introduction

To understand the motivation behind this project is to understand the meaning behind all causal analysis- the effort to gain a better understanding of how the underlying mechanisms of a given system work and relate to each other. Causal graph learning in particular is an interesting problem to tackle because in most real-world systems, the causal graphs are not known or partially known to the system trying to learn them- so the relationships between nodes must oftentimes be established prior to learning. This means that there are three primary problems in traditional causal analysis:

1. The nebulous, complex, and often nonlinear relationships that exist within a system
2. Noise in a given causal graph dataset that can skew the learning algorithm
3. A lack of scalability do to the sheer number of nodes

To elaborate a bit further on these problems, these traditional causal graph learning algorithms are usually exclusively linear systems or are utilizing assumed regression analyses to determine these relationships. This completely ignores the nonlinearity in the causal relationships that can exist in some data, namely time series data.

The goal of this paper is to attempt to understand these nonlinear relationships by first understanding the relationships between the nodes in these graphs and also understand the relationships as they change over time. This tackles the problem that a nonlinear relationship can present to a traditional causal learning algorithm. We will be analyzing what this means in more detail in the rest of this report.

Algorithm Description

In the paper, the description of this algorithm is broken up into how each module was designed specifically. These include all figures and all of the math from each module, so this project will follow suit. To preface the explanation of this system, the overarching system architecture shown in this figure here:

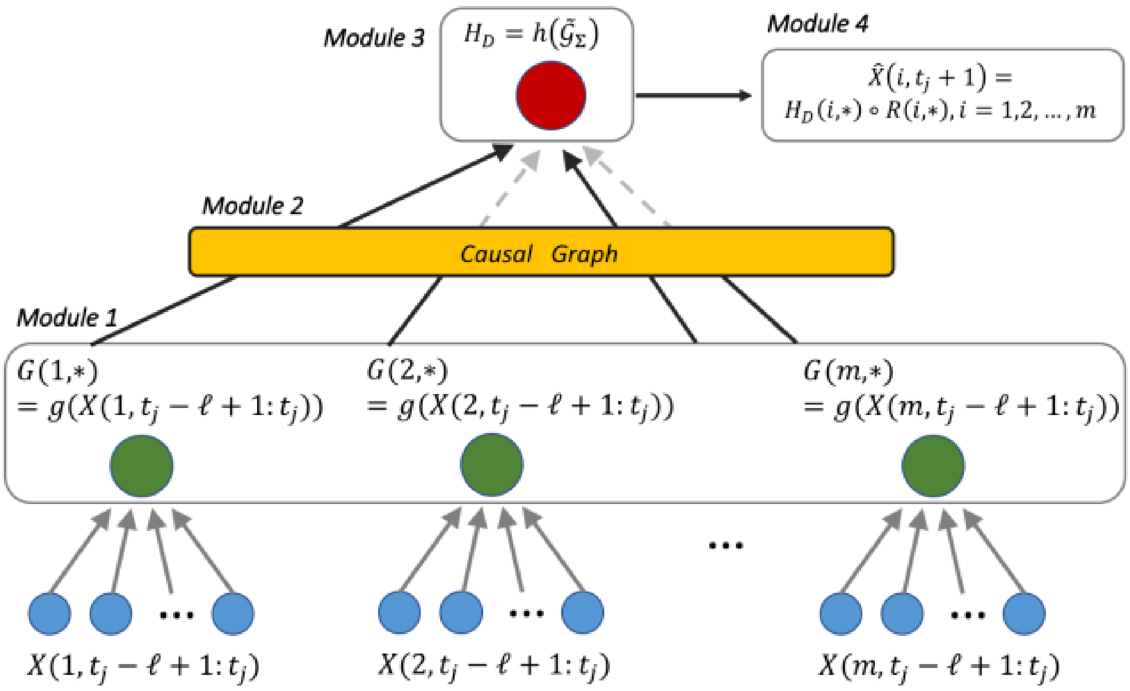

Module 3
$H_D = h(\tilde{\mathcal{G}}_\Sigma)$
Module 4
$\hat{X}(i, t_j + 1) = H_D(i,*) \circ R(i,*), i = 1,2,\dots,m$
Module 2
Causal Graph
Module 1
$G(1,*) = g(X(1, t_j - \ell + 1: t_j))$
$G(2,*) = g(X(2, t_j - \ell + 1: t_j))$
$G(m,*) = g(X(m, t_j - \ell + 1: t_j))$
$X(1, t_j - \ell + 1: t_j)$
$X(2, t_j - \ell + 1: t_j)$
$X(m, t_j - \ell + 1: t_j)$

Figure 1: Overarching System Architecture

Module 1

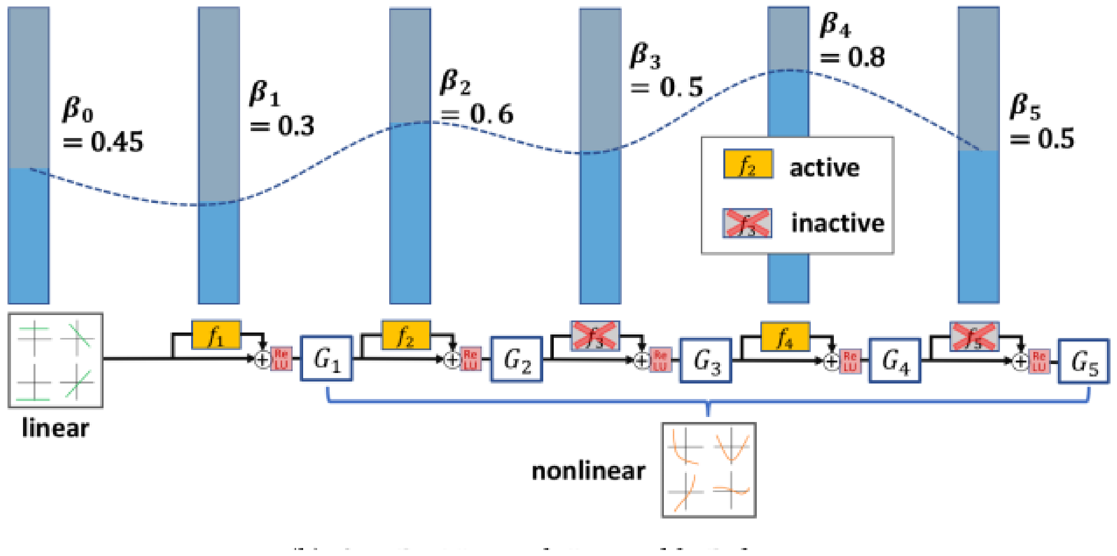


Figure 2: Architecture for Module 1 Resnet with Stochastic Depth

Before digging into the architecture, a small introduction to this module: The first module in the algorithm features something a lot of data scientists are familiar with- a ResNet implementation. This one is very slightly different, though, as it features a formula to calculate β, which is the survivability rate of a given layer. This rate determines whether or not the layer is more or less relevant to our understanding of these underlying causal graphs across different datasets.

The goal of this module is to understand the temporal nonlinearity by examining two things:

1. Learn the temporal nonlinearity at a univariate level with a deep architecture to avoid the vanishing gradients problem
2. Gain insight into the relevance of the temporal nonlinearity of the underlying causal graph

$$G_q = ReLU(G_{q-1}B_q + id(G_{q-1})), \quad q = 1, 2, \ldots, Q$$

Equation 1: ResNet Layer Output at a Given Block q

Given that at the first residual block the input will always be the regression from the previous timestamp, we can assign outputs to every residual block in the ResNet, which can be noted above in Equation 1. In an effort to move towards the architecture listed in the introduction to this module we introduce a Bernoulli Random variable to the formula, which can only be 0 or 1 depending on whether or not the block is active, which can be seen below in Equation 2.

$$G_q = ReLU(b_q(G_{q-1}B_q) + id(G_{q-1})), \quad q = 1, 2, \ldots, Q$$

Equation 2: ResNet Layer Output at Block q with Bernoulli Random Variable

Taking this formula one step further, we introduce the β term, or survivability term, which signifies the probability that the block is relevant to the causal graph. This formula can be seen in Equation 3 and the survivability formula can be found in Equation 4.

$$G_q = ReLU(\beta_q(G_{q-1}B_q) + id(G_{q-1})),$$

Equation 3: ResNet Layer Output at a Given Block q with Survivability

$$\beta_q = 1 - \frac{q}{Q}(1 - \beta_Q),$$

Equation 4: Survivability Formula

The survivability formula can be understood by looking into the following terms:

- β_Q: The survivability of the final block in the residual network
- β_q: The survivability of the current block in the residual network
- q: The number of the current residual block
- Q: The total number of residual blocks(or just the number of the final block)

This formula effectively ensures that the later residual blocks have less of an effect on the extraction of the causal variables representing the graph.

Module 2

This second model is the meat of the whole project, this is where the primary learning of the causal graph comes into play. To start, each output from Module 1 that helps to predict the regression at the next timestamp is utilized in this formula given causal graph A:

$$\tilde{\mathcal{G}}_q^T(*, i) = G_q^T A_q(*, i), \quad i = 1, 2, \ldots, m$$

Equation 5: Causal Variable Selection in a Given Timestamp

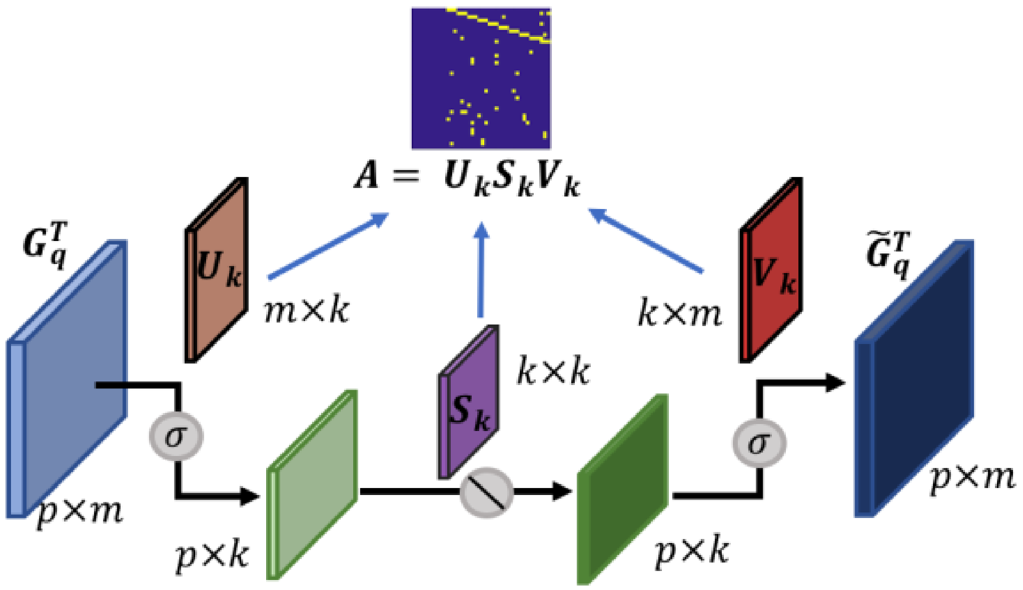


Figure 3: Low-Rank Causal Graph Learning Architecture

Figure 3 then outlines the architecture by which the causal graph learns using that low-rank approximation. Given that k<m, the architecture decomposes the residual block output matrix using k-rank approximation(a specification instead of just "low rank").

For a longer explanation, Module 2 takes each of the temporal embeddings and projects them from the m space into the k space via the weight matrix U_k. Then it is scaled with the matrix S_k to scale each low dimension before the temporal embeddings are rescaled to the m space with another weight matrix V_k. This operation is captured in Equation 6 below:

$$\tilde{\mathcal{G}}_q^T = \sigma((\sigma(G_q^T U_{q,k}) S_{q,k}) V_{q,k}),$$

Equation 6: Temporal Embedding Projection

Taking the weight matrices and the scaling matrix, we can now approximate the causal graph A using the following formula, Equation 7.

$$A_q \approx U_{q,k} S_{q,k} V_{q,k}.$$

Equation 7: Graph Approximation for a Given Block q Given the weight Matrices and Scaling Matrix

With this formula for each given block q, we can determine the approximation for the causal graph as a whole, which is represented by Equation 8. Here we see that each approximation from the residual blocks is scaled with their respective survivability terms and accumulated, thus giving us our approximation.

$$A \approx \sum_{q=1}^{Q} \beta_q A_q,$$

Equation 8: Approximation Across All Residual Blocks

Module 3

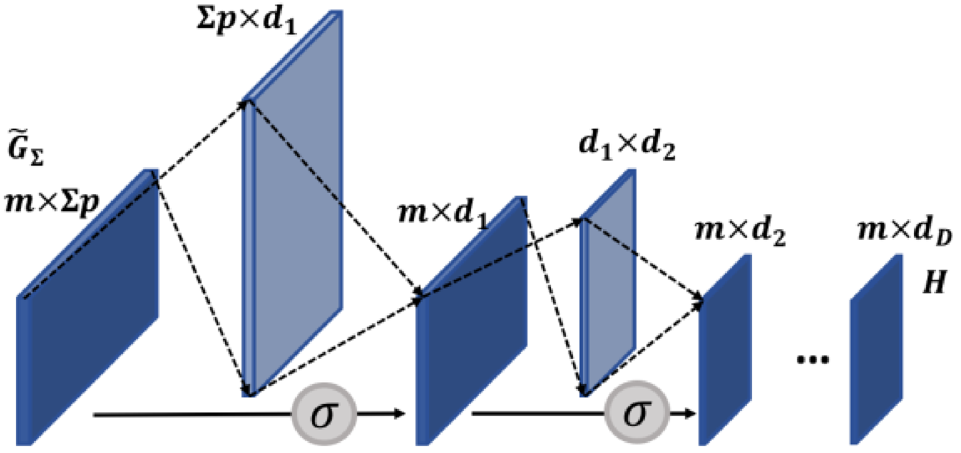


Figure 4: Intervariable Nonlinearity Architecture

The primary function of Module 3 is to learn the nonlinearity between the variables that we extracted in the previous module. We now have a series of fully-connected layers that are used to learn the intervariable nonlinearity of the temporal embedding projection represented by the following Equation:

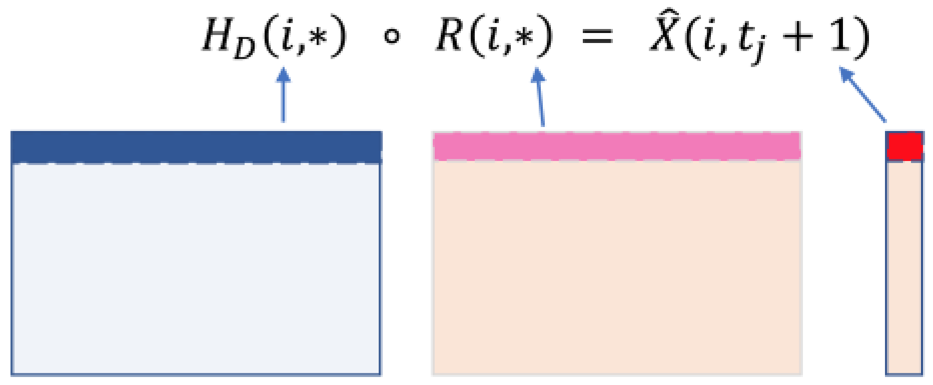
$$H_j = \sigma(H_{j-1}W_j + b_j), \quad j = 1, 2, \ldots, D$$

Equation 9: The Learned Nonlinearity from Layer 1 to D

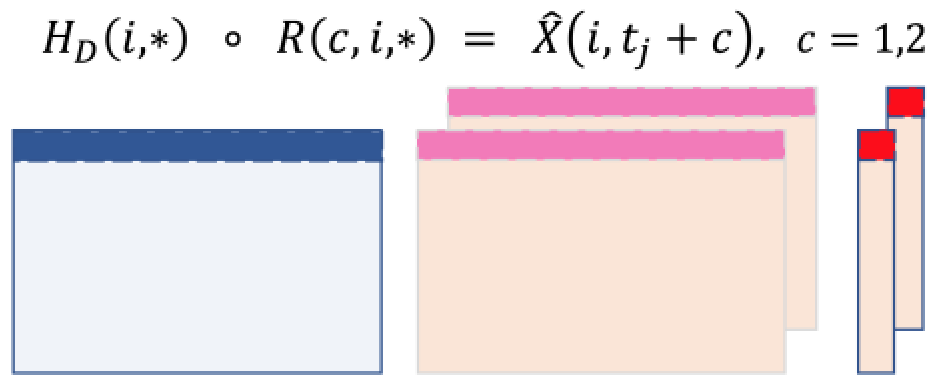
This is true given that the temporal nonlinearity embedding from the previous module is the input to Module 3.

Module 4

Module 4 is the final step in this process, where we predict the regression target using the formula outlined in Figure 4, where the regression is performed using a dot-product calculation of the rows between the output of Module 3 $H_D$and the learned weight matrix R.



(a) Module 4 with 1 time-stamp prediction


$H_D(i,*) \circ R(c,i,*) = \hat{X}\left(i, t_j + c\right), \; c = 1{,}2$

(b) Module 4 with 2 time-stamp prediction

Figure 5: Regression Target Prediction

The loss function for calculating the regression terms for these nonlinear relationships can be defined as follows in Equation 10:

$$loss = \frac{1}{mn} \sum \left(X(*, t_i + 1) - \hat{X}(*, t_i + 1) \right)^2 + \lambda_1 \sum_q \left(\left\| U_{q,k} \right\|_2 + \left\| V_{q,k} \right\|_2 \right) + \lambda_2 \sum_q \left(\left\| U_{q,k}^T U_{q,k} - I \right\|_2 + \left\| V_{q,k}^T V_{q,k} - I \right\|_2 \right)$$

Equation 10: Loss Function with Regularization Terms

The loss function for this regression is imperative to the prediction of the next timestamp and needs to be deconstructed to be fully understandable. The terms here are represented by the following:

- m: Number of variables
- n: Number of samples
- λ_1: Regularization term that is used to minimize overfitting on the causal graph
- λ_2: Regularization term used to impose orthonormality to U and V

This loss function generated the error used in backpropagation for the entire model. It is also worth noting that each of the rows of H_D contain all of the possible linear forms for the timestamp at that sample, which is then used to construct the optimal combination to approach $X(i, t_j + 1)$ (the next timestamp). Practically speaking, multiple future timestamps are calculated(see Figure 5b for an example of what that looks like).

Results

Datasets

There were two primary datasets used in this experiment and both were synthetically generated, referred to as Synthetic Dataset A and Synthetic Dataset B. Both were constructed with a similar causal structure to that of a gene regulatory network. This report won't go too far into detail regarding these as they are already pre-generated and weren't required for my understanding of this project.

My Results

In short, my results were awful. So much so that this project had to effectively be restarted from scratch. My application of the Resnet as described in the paper effectively resulted in an error of 90+% training on the synthetically generated data, which acted as my benchmark. With this much error, no relationship could remotely be predicted, so I was forced to heavily refactor my code using the code that the authors of this paper provided, and reexamine the way that I thought this was supposed to be implemented. I'll elaborate this more in the Conclusion section.

Paper Experiments

Focusing on the experiments that used the synthetic datasets, the following parameters were used in the experiment for Dataset A :

- Input Window: Set to 3
- Window(pre-win): Set to 2
- Q(Temporal Layers): Set to 5
- k(rank): Set to 30

For Dataset B, the following parameters were used:

- Input Window: Set to 5
- Window(pre-win): Set to 3
- Q(Temporal Layers): Set to 6
- k(rank): Set to 55

Two primary metrics were used in these experiments against the other commonly used methods for this type of problem: area under precision and recall curve (AUPR) and area under receiver operating characteristics curve (AUROC). To expand a bit on those commonly used methods mentioned earlier, here are the methods that SCGL is being compared against:

- VAR: Learn's causal graph using a vector autoregressive model with ridge regularization
- PCKGC: Pairwise kernel-based Granger causality analysis, only considers a pre-selected variable subset to save on the computational cost
- Copula: Transforms the distribution of variables to Gaussian domain then applies VAR learning methods.
- cLSTM: Applies LSTM to all past values of variables to regress the future values
- pTE: Calculates the phase transfer entropy by transforming the temporal variable signals to discrete phases with probability functions

All were taken, used, and compared against SCGL, the comparison of which can be seen in Figure 6, where the AUPR and AUROC scores for each method can be seen.

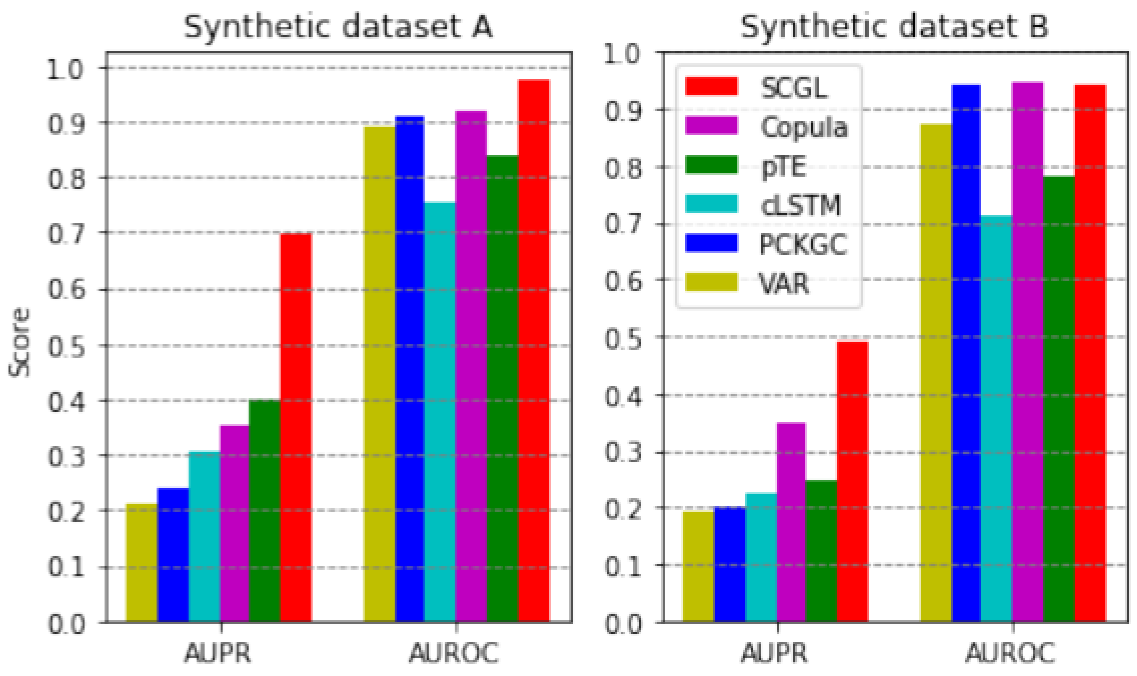


Figure 6: SCGL vs Other Methods AUPR and AUROC Scores

As can be seen, SCGL has a significantly higher AUPR score compared to the other methods for both datasets. However, looking at the AUROC scores, Dataset A is marginally better than the competition, while Dataset B is about equivalent. This means that the while not definitively better, model performances are at least comparable to the others.

Conclusion

To address it immediately, the replication of this paper was a complete failure for a number of reasons, but one primary one that I would like to address. My attempt at modeling this based on the math in this paper simply did not work. I was able to get a ResNet implementation working for Module 1, but the logic surrounding Module 2 escaped me, and after spending too much time trying to make it work, I was unable to rewrite it in time for this submission(and thus the broken code that I've submitted). With that in mind, I decided to turn this into a learning experience and focus on identifying how the math was implemented in the code that the authors wrote themselves. While it was unfortunate that I couldn't tackle this on my own, it was a lot of fun getting to see this algorithm applied and gain some insight into how I could approach a problem like this.

With this realization, there are of course things that I would have done differently. First, I would have fallen back on the code that the authors wrote about two weeks earlier than I tried to this time. With homework and other exams, the extra time could have been extremely helpful. Second, I probably would have taken a partner on this project(though nobody else was too interested in it). A second person to discuss this work with could have expedited my understanding of these algorithms.

In any case, the code I am submitting is broken, unfortunately, but I feel that I was able to still glean value out of this project. Understanding multivariate causal relationships in time series data is incredibly important and difficult to do, and I loved being able to get an introduction to it here. My hope is that I can find time in the near future to approach this project properly.