Graph Learning

Presenter: Zhe Wang

https://qdata.github.io/deep2Read

Zhe Wang

201909

Content

Gaussian Graphical Model

Strong Assumption: Observation $X \in \mathbb{R}^{N \times V}$ are sampled from multi-variate Gaussian distribution:

$$P(x) = (2\pi)^{-v/2} |\Omega|^{1/2} \exp(-\frac{1}{2} x^T \Omega x). \tag{1}$$

Ω is the precision matrix, which can be calculated via MLE:

$$\log p(x) \propto \log(|\Omega|) - Tr((x-\mu)^T \Omega (x-\mu)) \tag{2}$$

Pros and Cons

Pros:

- $\Sigma_{ij} = 0$ iff v_i, v_j are independent.
- Solid theoretical analysis.

Cons:

- Strong assumption for observation.

Content

Variational Methods

Variational inference based method: Graph VAE [1]
$X \in \mathbb{R}^{N \times D}$, where N is the number of nodes, and D is the dimension of node features. Hidden variables $Z \in \mathbb{R}^{N \times F}$.
Inference model (Approximation of posterior distribution)

$$X, A \rightarrow Z$$

$$p(Z|X,A) = \prod_{n=1}^{N} p(z_i|X,A), \quad p(z_i|X,A) = p(z_i|\mu_i, \sigma_i)$$

[1] N. Kipf and Welling, 2016

Generative model:

$$Z \rightarrow A:$$

$$P(A|Z) = \prod_{i,j} P(A_{ij}|Z_i, Z_j), \quad P(A_{ij} = 1|Z_i, Z_j) = \sigma({z_i}^T z_j)$$

Loss function (ELBO):

$$L = \int_{z \sim q(Z|X,A)} \log p(A|Z) - KL(q(Z|X,A)||p(Z))$$

Limitations:

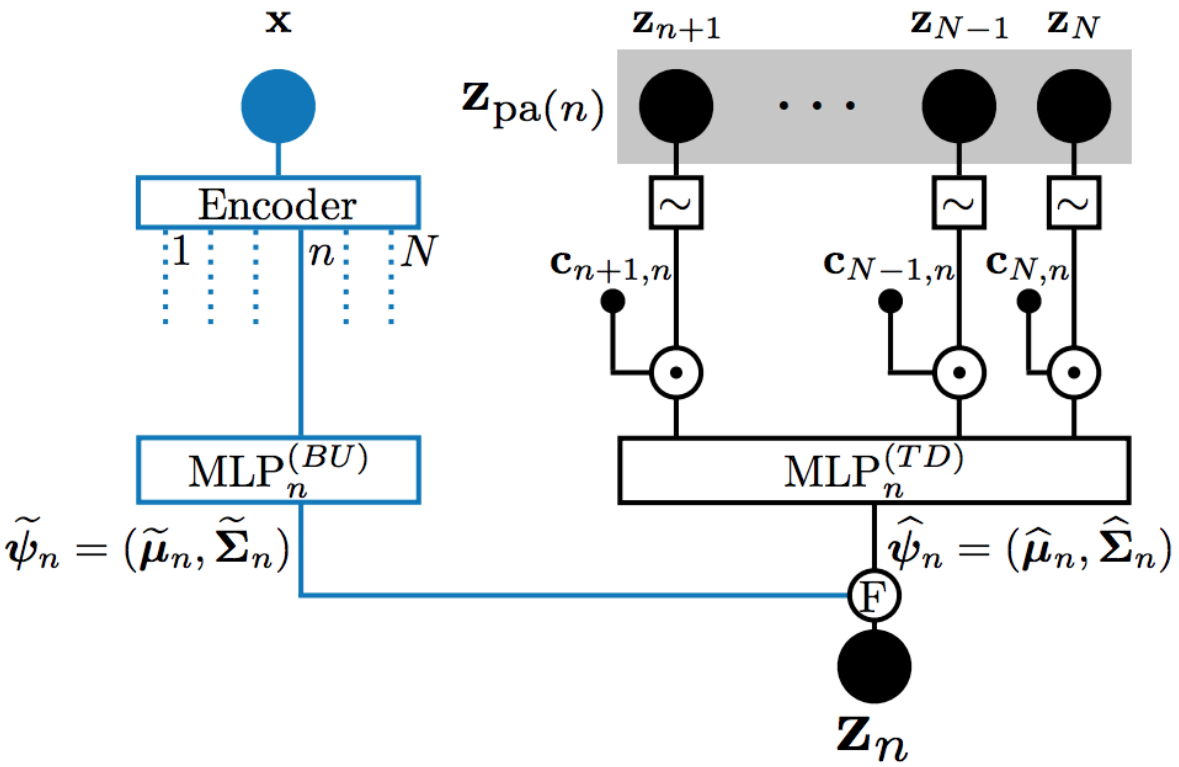
- In order for the analytic form of KL divergence, the prior and posterior are all assumed to be Gaussian distribution.
- Also all hidden variables are assumed to be independent.

Some following work add interactions for latent variables.
The parent nodes for z_i will be contained in $\{z_{i+1}, z_{i+1}, \cdots, z_N\}$, and is controlled via a binary value c_{ij}
The prior distribution is:

$$p_\theta(Z|C) = \prod_{n=1}^{N} p_\theta(z_n|z_{pa(n)}, c_{pa(n),n})$$

The approximated posterior distribution is:

$$q_\phi(Z|X, C) = \prod_{n=1}^{N} p_\phi(z_n|X, z_{pa(n)}, c_{pa(n),n})$$



F precision-weighted fusion

sampling

multiplication

bottom-up inference

top-down inference

Training algorithm

Algorithm 1 Optimizing VAEs with Latent Dependency Structure

Require: Data $\mathbf{x}$, number of latent nodes N, number of dimensions per node N'.

1: Initialize θ, ϕ, $\boldsymbol{\mu}$.
2: **repeat**
3: Sample $\mathbf{c}$ using Eq. (9) and determine $\mathbf{z}_{\text{pa}(n)}$ for each $\mathbf{z}_n$ based on the sampled structure.
4: For each node, compute $q_\phi(\mathbf{z}_n|\mathbf{x}, \mathbf{z}_{\text{pa}(n)})$ using Eq. (7).
5: Sample $\mathbf{z}$ from $q_\phi(\mathbf{z}|\mathbf{x})$ using Eq. (6) and compute $p_\theta(\mathbf{x}|\mathbf{z})$.
6: Update $\theta, \phi, \boldsymbol{\mu}$ based on the gradients derived from Eq. (8).
7: **until** Convergence.

Summary about variational inference based method:

- For the differentiation of KL divergence, both prior and posterior are assumed to be Gaussian.
- To increase model capacity, different interaction graph can be learned and embedded into the hidden variables.[He et.al ICLR2019][Sønderby et.al NeurIPS2016]

Content

DAG Learning

DAG learning:

- constraint-based approaches
- Score-based approaches
- Regression-based approaches

Constrained based algorithm:

- Use hypothesis testing to identify a set of conditional independence properties,
- Identify the network structure that best satisfies these constraints.

Algorithm 1 SGS Algorithm

1: Build a complete undirected graph H on the vertex set V .
2: For each pair of vertices i and j, if there exists a subset S of $V \setminus \{i, j\}$ such that i and j are d-separated given S, remove the edge between i and j from G.
3: Let G' be the undirected graph resulting from step 2. For each triple of vertices i, j and k such that the pair i and j and the pair j and k are each adjacent in G' (written as $i - j - k$) but the pair i and k are not adjacent in G', orient $i - j - k$ as $i \rightarrow j \leftarrow k$ if and only if there is no subset S of $\{j\} \cup V \setminus \{i, j\}$ that d-separate i and k.
4: **repeat**
5: If $i \rightarrow j$, j and k are adjacent, i and k are not adjacent, and there is no arrowhead at j, then orient $j - k$ as $j \rightarrow k$.
6: If there is a directed path from i to j, and an edge between i and j, then orient $i - j$ as $i \rightarrow j$.
7: **until** no more edges can be oriented.

Score based algorithm

Score-based algorithm

- posit a criterion by which a given Bayesian network structure can be evaluated on a given dataset
- search through the space of all possible structures and tries to identify the graph with the highest score

The main problem of the score-based algorithm is the optimization is intractable

One of the most frequently used score is Bayesian Information Criterion (BIC).

Regression based

Basic assumption: $X = XW$. In [Meinshausen and Buhlmann, 2006], the authors used the linear regression with l_1 penalty to recovery the graph from feasible region.

$$\arg\min_{W} \frac{1}{2}||X - XW||_F^2 + \lambda||W||_1$$
$$s.t. \quad G \in DAG$$

This Combinatorial optimization is NP-hard. Until in [Zheng et.al 2019], the problem is converted as a continuous optimization problem, and all state-of-the-art optimization methods can be used.
Briefly, they prove

$$G \in DAG \iff Tr(e^{W \odot W}) - d = 0$$

Learning the graph via neural network is gaining popularity. The main idea is to use the remaining nodes as the input of a mlp to approximate a particular node nonlinearly.
With that theorem, how to satisfy the constraint lies at the core of the graph learning.
In [Ke et.al 2020] and LDS [Franceschi et.al 2019], each element of the graph is sampled from a Bernulli distribution, whose parameter is r_{ij}. But in [Zheng et.al 2020] and [Lachapelle et.al 2020], each element is derived from the parameter of mlp.

For example, if the mlp contains two hidden layers.

$$C_{im} = \sum_h \sum_k |W^1_{ih}||W^2_{hk}||W^3_{km}| \tag{3}$$

the weight adjacency matrix can be defined as :

$$A_{ij} = \begin{cases} \sum_m C^j_{im}, & i \neq j, \\ 0, & i = j \end{cases} \tag{4}$$

But, in second category: $A_{ij} \sim Ber(\gamma_{ij})$

Content

Causal DAG learning

The learning of causal graph is more complicated. Nearly all of them are intervention based. [Arjovsky et.al 2020][Krueger et.al 2020][Bengio et.al 2020][Ke et.al 2020]
Given only observational setting, causal graph is only possible to be identified up to the Markov equivalence class.

- Identify if X_i and X_j are statistical associated.
- Use do-calculus to identify causal and effect. Specifically check if $p(X_i|X_j = a) = p(X_i|X_j = b)$

A more closed related work is LDS [Franceschi et.al 2019], they combine the graph learning with downstream tasks, which is node classification.

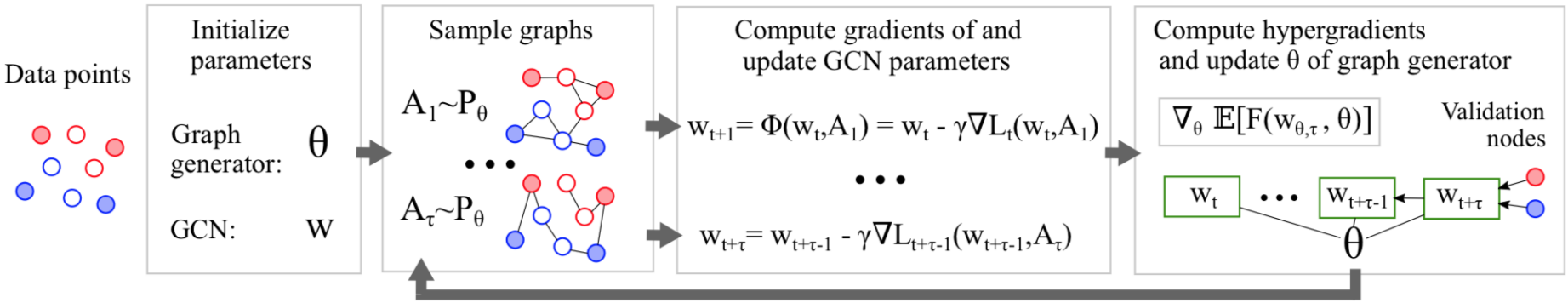


Figure 1. Schematic representation of our approach for learning discrete graph structures for GNNs.

Their model is a bi-level optimization problem, the inner level use the learned graph to update the GCN, while the outer level is in charger of discovering the task-related graph.