Sparse Causal Discovery for Scalable Gene Regulatory Network Inference

Anonymous Authors[1]

Abstract

Causal discovery is a critical problem in genomics, where understanding gene regulatory networks (GRNs) is crucial for unraveling biological processes and developing disease treatments. However, the computational complexity of existing causal discovery methods has limited their applicability to small-sized graphs, hampering the potential of advanced RNA-sequencing technologies that can measure thousands of genes. In this paper, we present Sparse Differentiable Causal Graph Inference (SDCI), a novel, scalable approach to causal discovery specifically designed for biological settings. Our key insight is that biological causal graphs are likely sparse, with genes typically turned on/off by only a few other genes rather than by any other genes in the system. SDCI employs a fast and parallelizable preselection method to identify a sparse set of candidate parent genes for each gene based on their predictive power, followed by a differentiable causal discovery algorithm optimized for sparse graphs. Our experimental results demonstrate that SDCI outperforms other methods in terms of scalability and accuracy on simulated data. This approach holds promise to extend causal discovery to larger and more complex GRNs, with applications in drug discovery and personalized medicine.

1. Introduction

Gene regulatory networks (GRNs) are crucial for understanding the fundamental mechanisms of biological systems. The advent of high-throughput and single-cell resolution RNA sequencing has led to the collection of massive amounts of data, which can be used to learn GRNs and their variability across cell types and tissue. However, traditional methods for inferring GRNs, such as SCENIC (Aibar et al., 2017) GRNBoost (Moerman et al., 2019), PIDC (Chan et al., 2017) and others (Kim, 2015; Iglesias-Martinez et al., 2021), are limited in their ability to uncover causal relationships between genes as they rely on correlation-based approaches that lack the ability to identify causal directions.

Unlike correlation-based approaches, causal discovery methods aim to infer causal relationships between variables (Glymour et al., 2019). In particular, differentiable causal discovery methods have emerged as a promising approach to cast the computationally expensive combinatorial search of a causal graph (Wang et al., 2017) into a more tractable optimization problem (Zheng et al., 2018). However, these methods are still limited by their computational complexity, with some requiring cubic time in the number of genes (Brouillard et al., 2020) and others resorting to strong approximations to scale to larger graphs (Lopez et al., 2022).

To overcome these limitations, we propose Sparse Differentiable Causal Graph Inference (SDCI), a scalable approach to causal discovery specifically designed for biological settings. SDCI leverages the insight that biological causal graphs are likely sparse. For instance, genes are usually turned on or off by a few other genes (known as regulators) rather than by any gene in the system. The sparse nature of these graphs provides an opportunity to reduce the computational complexity of causal discovery and to scale up to higher-dimension settings.

SDCI introduces a two-stage approach that combines the strengths of both correlation-based and causal discovery methods to enable a scalable and parallelizable inference. In the first stage, SDCI employs a fast and parallelizable edge pre-selection method that identifies a sparse set of candidate parent genes for each gene based on their predictive power. This process filters out unlikely causal relationships, shrinking the search space of causal graphs over a far smaller subset of candidate graphs. The second stage is a differentiable causal discovery algorithm optimized for sparse graphs. In addition to observational data, SDCI can handle labeled interventional data when available, such as from Perturb-seq experiments (Dixit et al., 2016; Replogle et al., 2022), allowing for a more accurate reconstruction of GRNs.

We theoretically show that our pre-selection stage returns a set of candidate parents that is sparse and that contains the

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by *the 2023 ICML Workshop on Computational Biology*. Do not distribute.

true parents. These two qualities ensure improved computational efficiency while establishing that the exact maximization of the SDCI objectives can identify the best recoverable causal graph based on the available data.

Our experimental results demonstrate that SDCI outperforms other methods in terms of scalability and accuracy on simulated data. This approach holds promise to extend the applicability of differentiable causal discovery to larger and more complex gene regulatory networks, with potential applications in drug discovery, personalized medicine, and other areas of biology.

2. Background

The data. We aim to infer GRNs using high-dimensional gene expression data, including observational data from single-cell (or bulk) RNA sequencing experiments and optional interventional data, such as from Perturb-seq experiments where specific genes are knocked down one at a time. The use of interventional data is recommended to improve the ability of the model to identify true causal edges but is not mandatory. Mathematically, we assume that we observe the expression of d different genes in n independent cells (or bulk samples) and denote the expression of gene g in cell i as $x_{ig} \in \mathcal{X}$. To indicate the set of genes that were perturbed in cell (or sample) i, we use I_i. If a cell was not perturbed, $I_i = \emptyset$, but if g was knocked down $I_i = \{g\}$. The set of all available interventions is $\mathcal{I} = \{I_i \mid i \in [n]\}$.

Background on causal discovery. Causal graphical models (CGMs) provide a powerful mathematical framework for reasoning about causal relationships between genes. A CGM over d genes is defined by two components:

1. A directed acyclic graph (DAG), $G^* = (V, E)$ where each node $g \in V$ is a gene, and each edge $(g, h) \in E$ indicates a direct causal relationship from g to h, i.e. the impact of gene g on the regulation of gene h.
2. A set of conditional distributions $p_g(x_g|\{x_h\}_{h\in \mathrm{pa}_g^{G^*}})$ that specify the distribution of each gene g given its causal parents $\mathrm{pa}_g^{G^*}$.

The observational joint distribution of cell i's gene expression is then the product of these conditional distributions

$$p(x_{i1}, ..., x_{id}) = \prod_{g\in V} p_g(x_{ig} \mid \{x_{ih}\}_{h\in \mathrm{pa}_g^{G^*}}). \quad (1)$$

Since CGMs model causal relationships rather than purely statistical relationships, we can reason about gene knockdown interventions' effects by modifying the intervened gene's conditional distribution. Using the notation I_i and $\tilde{p}_g$ to denote the resulting new conditional distribution of x_g when intervened upon, we derive the resulting interventional probability distribution over all genes $\tilde{p}^{I_i}(x_{i1}, ..., x_{id})$ as

$$\prod_{g\in V\setminus I_i} p_g(x_{ig}|\{x_{ih}\}_{h\in \mathrm{pa}_g^{G^*}}) \prod_{g\in I_i} \tilde{p}_g(x_j|\{x_{ih}\}_{h\in \mathrm{pa}_g^{G^*}}). \quad (2)$$

Note that $\tilde{p}^{\emptyset} = p$. The goal of causal discovery is to find the graph G^* from the observed data $\{x_{ig}, I_i\}_{i\in[n],g\in[d]}$.

Differentiable causal discovery In order to recover the causal graph G^* and the set of conditional distributions, causal discovery aims to fit a CGM to the observed data using a parameterized family of conditional densities f_ϕ, similar to that of maximum likelihood estimation. It was shown in Brouillard et al. (2020) that the following estimator returns a graph $\hat{G}$ that is a close to G^* as possible:

$$\hat{G} = \arg\max_{G\in DAG} \sup_{\phi} \sum_{I\in\mathcal{I}} \mathbb{E}_{x\sim\tilde{p}^I} \left[\log f_\phi^I(x_{1:d}; G)\right] - \lambda|G|, \quad (3)$$

where λ is a regularization term and each $f_\phi^I(.; G)$ is a density parametrized by ϕ factorizing as Eqs. (1) and (2):

$$\prod_{g\in V\setminus I} f_{\phi,g,0}(x_g; \{x_h\}_{h\in \mathrm{pa}_g^G}) \prod_{g\in I} f_{\phi,g,1}(x_j; \{x_h\}_{h\in \mathrm{pa}_g^G})$$

with $f_{\phi,g,0}$ observational and $f_{\phi,g,1}$ interventional.

To tackle this potentially intractable combinatorial problem (notice that G is over the space of all DAGs), *differentiable* causal discovery (Zheng et al., 2018; Brouillard et al., 2020) proposes to relax the constraint $G \in DAG$ by instead subtracting from Eq. (3) a *differentiable penalty of acyclicity* $h : G \mapsto h(G) \in \mathbb{R}_{\geq 0}$ which is minimized on DAGs and positive on non-DAGs. For example, if A is the adjacency matrix of G with positive weights, one possible penalty is $h(A) = \mathrm{Tr}\left[\exp(A)\right]$ where exp is the matrix exponential.

However, as shown in § 5, these methods still have difficulties scaling to many genes.

3. Sparse Differentiable Causal Graph Inference

With d reaching up to thousands of genes, the main bottlenecks for differentiable causal discovery methods are computing the log-likelihood and the acyclicity constraint gradient ∇h. The log-likelihoods $\log f_\phi^I$ require computing d conditional likelihoods $\log f(x_{ig} \mid x_{i,-g})$ which each involve all the other genes since G is not necessarily a DAG. Meanwhile, the acyclicity constraint requires taking expensive operations over a $d \times d$ matrix.

Intuition. To address these challenges, we notice that the target graph G^* is likely to be sparse. In particular, the number of parents for any given gene is limited. Therefore, we propose the following two-stage approach:

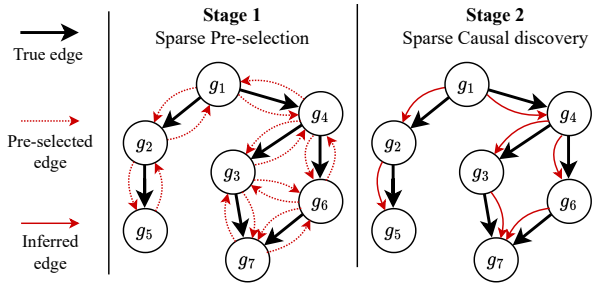

Figure 1. The SDCI pipeline. The first stage efficiently pre-selects a few candidate causal parents for each gene (red dashed arrows). The second stage identifies the true causal parent among the candidates (the solid red arrows recover the true causal black arrows).

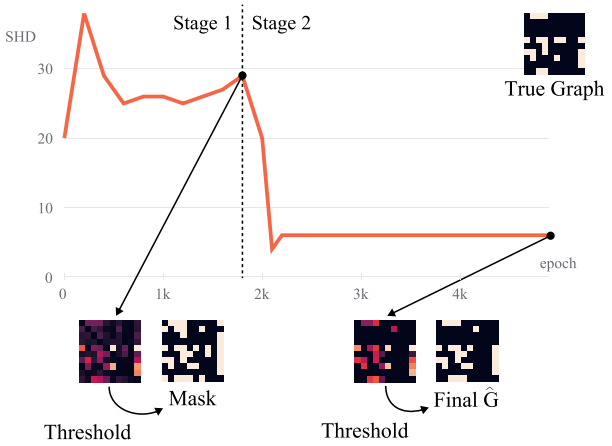

Figure 2. Illustration of the two stages of SDCI on simulated data. The first stage learns a sparse set of candidate edges (the mask), and the structural Hamming distance (SHD) decreases during training

1. First, we draw insight from existing correlation-based methods for GRNs to identify which genes are possibly parents of others and remove those which are not. We devise an efficient and parallelizable algorithm that returns a sparse set of candidate parents for each gene.

2. Next, we use differentiable causal discovery to identify the true causal parents among the candidate parents. To ensure computational efficiency, we adapt the log-likelihood and the acyclicity penalty computation to leverage the sparsity of the graphs obtained in stage 1.

Below we describe the method in detail.

The model. We assume that the gene expression data comes from a CGM as described in § 2, with observational and interventional distributions given by Eqs. (1) and (2). We aim to efficiently solve the optimization problem Eq. (3).

We use neural networks (NNs) to parametrize the response functions $f^I_{\phi,g}$. Each $f^I_{\phi,g}$ is a NN from $\mathcal{X}^{d-1}$ to $\mathcal{X}$ that conditionally models x_g given all the other x_{-g}. The NNs implicitly define the graph G, such that if $f^I_{\phi,g}$ puts zero weight on x_h, then there is no edge from h to g.

Stage 1: Gene Sparse Preselection. We first solve an unconstrained version of Eq. (3) for which G is not necessarily a DAG, but must still not contain self-loops. In this case, the optimization problem can be divided into d independent sub-problems, each one determining candidate parents of g.

$$\hat{S}_g = \arg\max_{S \subset V \setminus \{g\}} \sup_{\phi} \sum_{\substack{I \in \mathcal{I} \\ g \notin I}} \mathbb{E}_{x \sim \tilde{p}^I} \left[\log f^I_{\phi,g}(x_g \mid x_S) \right] - \lambda |S|. \tag{4}$$

Each of these problems can be solved in parallel in complexity $O(d)$. For each x_g, we are determining which of the others, $\{x_h \mid h \in [d] \setminus \{g\}\}$, are predictive of it akin to *feature selection*. We show in § 4 that $\hat{S}_g$ is a superset of the true causal parents that does not contain all genes.

Stage 2: Sparse Differentiable Causal Graph Inference We then perform sparse differentiable causal discovery. We optimize the objective Eq. (3) with the $G \in DAG$ constraint, to which we also add the constraints: $\mathrm{pa}^G_g \subset \hat{S}_g$ for each g. In other words, for each gene g, the potential causal parents are only searched among the set $\hat{S}_g$. We write $e = \sum_g |\hat{S}_g|$.

The log-likelihood in Eq. (3) is usually computed with complexity $O(d^2)$, but with the constraints $\mathrm{pa}^G_g \subset \hat{S}_g$, sparse matrix multiplications can compute it in $O(e)$. We always have $e = O(d^2)$, but if the true causal graph is sparse, we can have $e = O(d)$, as shown in § 4.

To compute the acyclicity penalty $h(A)$, we use a spectral characterization of a DAG's adjacency matrix A. Indeed, a matrix represents a DAG if and only if it is nilpotent[1], which happens if and only if all its eigenvalues are 0. We then define $h(A) = \rho(A)$ where $\rho(A)$ is the complex modulus of the leading eigenvalue of A. In appendix, we show that $\rho(A)$ is well-defined, that it is differentiable (it was not guaranteed for complex eigenvalues) and that its gradient can be computed efficiently in time $O(e)$ using a power method for estimating the leading eigenvalue.

Hence, we can efficiently search for the sparse causal structure without having to search over the space of all DAGs. In addition to the improvement in theoretical complexity, we find that in practice, this two-stage method and the use of the penalty $h(A) = \rho(A)$ converge faster than the competing methods, hence enjoying a faster running time.

[1]That is if $A^k = 0$ for some $k > 0$.

4. Theoretical guarantees

For SDCI to arrive at an accurate solution, we must ensure that the first stage does not remove any true direct parents of the genes. Additionally, to achieve computational efficiency, the first stage should only return a subset of the genes rather than all of them. We prove both of these statements to be true. In fact, the next theorem precisely characterizes which set of genes is returned by the first stage:

Theorem 1. *Under some reasonable assumptions detailed in appendix, $\hat{S}_g = \mathrm{bo}_g^{G^*}$, where $\mathrm{bo}_g^{G^*}$ contains the direct causal parents of g, the direct children of g and the parents of the children of g (i.e. the Markov boundary of g (Neapolitan et al., 2004) in the true graph G^*).*

An important assumption of the theorem is that the density functions f_ϕ can approximate the ground-truth distributions $\tilde{p}^I$. The other assumptions are mostly technical.

Hence, the first stage of SDCI preserves the parents of each variable while reducing the number of candidate parents. Suppose each node has at most k parents; then we can show that the total number of edges retained after stage 1 is at most $O(d \cdot k^2)$. When $k \ll d$, this is effectively linear in d.

5. Results

To evaluate the performance of SDCI , we compare it on simulated data against two baselines: DCDI (Brouillard et al., 2020) and DCD-FG (Lopez et al., 2022). DCDI is equivalent to using only the second stage of our method (no pre-selection) and using the matrix exponential for the acyclicity penalty. Alternatively, DCD-FG is a method that approximates DCDI with a low-rank adjacency matrix.

Simulated data For each simulated dataset, we generate the true causal graph G^* as an Erdős-Rényi DAG with d node and expected edge density p. We then randomly draw a set of fully connected neural networks to parametrize the observational conditional distributions:

$$p_g(x_g|\mathrm{pa}_g^{G^*}) \sim \mathcal{N}\left((\mu, \sigma^2)=\mathrm{MLP}^{(j)}\left(\{x_h\}_{h\in \mathrm{pa}_g^{G^*}}\right)\right).$$

For intervention distributions, $\tilde{p}_g$, we mimic gene knockout as a hard intervention on gene g and set $\tilde{p}_g(x_g|\mathrm{pa}_g^{G^*}) \sim \delta_0$. The full simulated dataset consists of $n = 50 \cdot (d+1)$ observations (i.e. cells), where each set of 50 observations is generated from a single knockout interventional distribution in addition to the observational distribution. We consider several values of d and p to simulate different scenarios.

We evaluate the performance of our algorithm by comparing it to the baselines in terms of two key metrics: structural Hamming distance (SHD), which quantifies the difference between the inferred causal graph and the ground truth, and runtime, which reflects the algorithm's efficiency.

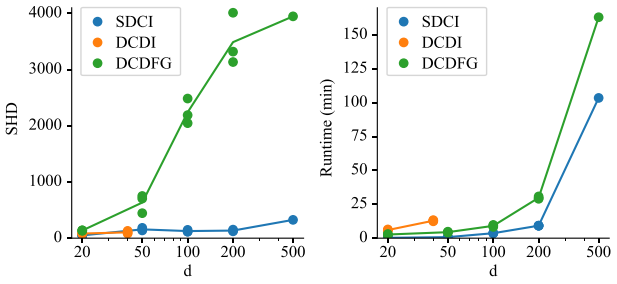

Figure 3. SHD (left) and runtime (right) comparison of SDCI to baseline methods, DCDFG, DCDI. Each marker represents one simulation generated with a unique random seed with the respective d. DCDI only has a datapoints at $d = 20, 40$ because it failed to complete for $d \geq 50$.

From the baseline comparison in Figure 3, we observe that for each number of dimensions, d, simulated, SDCI performs better than DCD-FG for every one of the simulation replicates performed. SDCI has comparable SHD values compared to DCDI for the data points available, implying that the two are indeed equivalent in reaching a common objective and that stage 1 of SDCI does not impact the overall performance. Additionally, we see that the results seem to be more robust at high dimensions compared to DCD-FG, which displays a higher variance in SHD at $d = 100$. We attribute the poor scores of DCD-FG to its low-rank assumption, which fails to adapt to sparse scenarios.

The same holds true for the runtime comparison between SDCI and the baseline methods. We ran all the methods on a 12th Gen Intel(R) Core(TM) i9-12900KF CPU. We observe that SDCI terminates in a fraction of the time of the other methods at every value of d. For example, at $d = 20$, SDCI only takes 16 seconds compared to DCD-FG taking an average of ~ 2.7 minutes and DCDI taking an average of ~ 5.7 minutes.

6. Discussion

We present SDCI, a scalable differentiable causal discovery method designed for inferring gene regulatory networks. Using a two-stage approach, SDCI exploits the sparse nature of biological causal graphs to improve the overall time complexity of the algorithm compared to state-of-the-art baseline methods without sacrificing performance. In future work, we plan to further improve the runtime of SDCI with GPU acceleration and sparse matrix multiplication. Additionally, we will run SDCI on Perturb-seq datasets to validate whether SDCI can uncover known and novel gene-gene interactions.

References

Aibar, S., González-Blas, C. B., Moerman, T., Huynh-Thu, V. A., Imrichova, H., Hulselmans, G., Rambow, F., Marine, J.-C., Geurts, P., Aerts, J., et al. Scenic: single-cell regulatory network inference and clustering. *Nature methods*, 2017.

Brouillard, P., Lachapelle, S., Lacoste, A., Lacoste-Julien, S., and Drouin, A. Differentiable causal discovery from interventional data. *Neural Information Processing Systems*, 33:21865–21877, 2020.

Chan, T. E., Stumpf, M. P., and Babtie, A. C. Gene regulatory network inference from single-cell data using multivariate information measures. *Cell systems*, 2017.

Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C. P., Jerby-Arnon, L., Marjanovic, N. D., Dionne, D., Burks, T., Raychowdhury, R., et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *cell*, 2016.

Glymour, C., Zhang, K., and Spirtes, P. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Iglesias-Martinez, L. F., De Kegel, B., and Kolch, W. Kboost: a new method to infer gene regulatory networks from gene expression data. *Scientific Reports*, 11(1): 15461, 2021.

Kim, S. ppcor: an r package for a fast calculation to semi-partial correlation coefficients. *Communications for statistical applications and methods*, 2015.

Lopez, R., Hütter, J.-C., Pritchard, J., and Regev, A. Large-scale differentiable causal discovery of factor graphs. *Neural Information Processing Systems*, 2022.

Magnus, J. R. On differentiating eigenvalues and eigenvectors. *Econometric Theory*, 1985.

Moerman, T., Aibar Santos, S., Bravo González-Blas, C., Simm, J., Moreau, Y., Aerts, J., and Aerts, S. Grnboost2 and arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics*, 2019.

Neapolitan, R. E. et al. *Learning Bayesian Networks*. Prentice Hall, 2004.

Replogle, J. M., Saunders, R. A., Pogson, A. N., Hussmann, J. A., Lenail, A., Guna, A., Mascibroda, L., Wagner, E. J., Adelman, K., Lithwick-Yanai, G., et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575, 2022.

Wang, Y., Solus, L., Yang, K., and Uhler, C. Permutation-based causal inference algorithms with interventions. *Advances in Neural Information Processing Systems*, 30, 2017.

Zheng, X., Aragam, B., Ravikumar, P. K., and Xing, E. P. DAGs with NO TEARS: Continuous optimization for structure learning. *Neural Information Processing Systems*, 2018.

A. Proofs of theorems

We prove theorem 1.

Proof. For consistency of notation with Brouillard et al. (2020), we enumerate $\mathcal{I}$ as $\{I_1, I_2, ..., I_k, ...\}$ where each I_k is a different interventions set and $I_1 = \emptyset$ refer to the observational data. We then refer to $f_\phi^{I_k}$ as $f^{(k)}(.;\phi)$.

Fix a $j \in V$ and define $\psi(S) = \sup_\phi \left(\sum_{\substack{k \\ j \notin I_k}} \mathbb{E}_{x \sim p^{(k)}} \left[\log f^{(k)}(x_j; x_S, \phi) \right] - \lambda |S| \right)$. Further define $S = \mathrm{bo}(G^*, j)$ to be the Markov boundary of node j in the true causal graph G^*. We will show that $\psi(S) > \psi(T)$ for any other $T \subset V \setminus \{j\}$.

We compute,

$$\begin{aligned}
\psi(S) - \psi(T) &= \sup_\phi \sum_{\substack{k \\ j \notin I_k}} \mathbb{E}_{x \sim p^{(k)}} \left[\log f^{(k)}(x_j; x_S, \phi) \right] - \sup_\phi \sum_{\substack{k \\ j \notin I_k}} \mathbb{E}_{x \sim p^{(k)}} \left[\log f^{(k)}(x_j; x_T, \phi) \right] && (5) \\
&\quad - \lambda |S| + \lambda |T| \\
&= - \inf_\phi \sum_{\substack{k \\ j \notin I_k}} \mathbb{E}_{p^{(k)}(x_{-j})} \left[D_{KL} \left(p^{(k)}(x_j | x_{-j}) \parallel f^{(k)}(x_j; x_S, \phi) \right) \right] && (6) \\
&\quad + \inf_\phi \sum_{\substack{k \\ j \notin I_k}} \mathbb{E}_{p^{(k)}(x_{-j})} \left[D_{KL} \left(p^{(k)}(x_j | x_{-j}) \parallel f^{(k)}(x_j; x_T, \phi) \right) \right] \\
&\quad - \lambda (|S| - |T|) \\
&= \inf_\phi \sum_{\substack{k \\ j \notin I_k}} \mathbb{E}_{p^{(k)}(x_{-j})} \left[D_{KL} \left(p^{(k)}(x_j | x_{-j}) \parallel f^{(k)}(x_j; x_T, \phi) \right) \right] && (7) \\
&\quad + \lambda (|T| - |S|).
\end{aligned}$$

Line 6 comes from $\mathbb{E}_{x \sim p^{(k)}} \left[\log f^{(k)}(x_j; x_S, \phi) \right] = -\mathbb{E}_{p^{(k)}(x_{-j})} \left[D_{KL} \left(p^{(k)}(x_j | x_{-j}) \parallel f^{(k)}(x_j; x_S, \phi) \right) \right] + \mathbb{E}_{x \sim p^{(k)}} \left[\log p^{(k)}(x_j | x_{-j}) \right]$ where we added and substracted the $\log p^{(k)}$ term. We note that we use the assumption of strictly positive density here, to define the conditional $p^{(k)}(x_j | x_{-j})$ without technical difficulties. Line 7 comes from the assumption of sufficient capacity. Indeed, we first have $p^{(k)}(x_j | x_{-j}) = p^{(k)}(x_j | x_S)$ by definition of a Markov blanket (remember that S is a Markov blanket for any $p^{(k)}$), and then the KL divergence functions $f^{(k)}$ are expressive enough to represent the ground truth distributions so the KL divergence is 0.

We further have:

$$\begin{aligned}
\psi(S) - \psi(T) &\geq \inf_\phi \mathbb{E}_{p^{(1)}(x_{-j})} \left[D_{KL} \left(p^{(1)}(x_j | x_S) \parallel f^{(1)}(x_j; x_T, \phi) \right) \right] + \lambda (|T| - |S|) && (8) \\
&= \mathbb{E}_{p^{(1)}(x_{-j})} \left[D_{KL} \left(p^{(1)}(x_j | x_S) \parallel p^{(1)}(x_j | x_T) \right) \right] + \inf_\phi \mathbb{E}_{p^{(1)}} \left[\log \frac{p^{(1)}(x_j | x_T)}{f^{(1)}(x_j; x_T, \phi)} \right] && (9) \\
&\quad + \lambda (|T| - |S|) \\
&\geq \underbrace{\mathbb{E}_{p^{(1)}(x_{-j})} \left[D_{KL} \left(p^{(1)}(x_j | x_S) \parallel p^{(1)}(x_j | x_T) \right) \right]}_{\eta(T)} + \lambda (|T| - |S|). && (10)
\end{aligned}$$

where line 10 follows from $\mathbb{E}_{p^{(1)}} \left[\log \frac{p^{(1)}(x_j | x_T)}{f^{(1)}(x_j; x_T, \phi)} \right] = \mathbb{E}_{p^{(1)}(x_T)} \left[D_{KL} \left(p^{(1)}(x_j | x_T) \| f^{(1)}(x_j; x_T, \phi) \right) \right] \geq 0$.

Let's finally fix $0 < \lambda < \min \left\{ \frac{\eta(T)}{|S| - |T|} \mid T \subset V \setminus \{j\} \text{ and } \eta(T) > 0 \text{ and } |S| > |T| \right\} \cup \{1\}$ (well defined as the minimum of a finite non-empty set with positive elements).

Let's assume now that $\psi(T) \geq \psi(S)$ for some $T \subset V \setminus \{j\}$. Then $0 \geq \eta(T) + \lambda(|T| - |S|)$ which rewrites $\lambda(|S| - |T|) \geq \eta(T)$.

Since $\eta(T) \geq 0$, we deduce that $|S| \geq |T|$ (the set T cannot be bigger than the Markov boundary S). Now further notice that if $\eta(T) > 0$ then $|S| > |T|$ and by definition of λ, $\lambda > \lambda$ which is absurd. Hence, $\eta(T) = 0$.

Finally, if $\eta(T) = 0$, then $D_{KL}\left(p^{(1)}(x_j|x_S) \parallel p^{(1)}(x_j|x_T)\right) = 0$ for all (x_{-j}) (since $p^{(1)}(x_{-j})$ has positive density. Hence, the conditional $p^{(1)}(x_j|x_S)$ and $p^{(1)}(x_j|x_T)$ are identical. Since, S was the Markov boundary of X_j, that makes T also a Markov blanket of X_j. But then by minimality of the Markov boundary in a faithful graph, we have $S \subset T$.

Remember that we had deduced $|S| \geq |T|$. So $S = T$.

This ends the proof.

□

Theorem 2. *Write* $\deg(i)$ *the in-degree of node* i *in graph* G^**. Then the number of candidate edges returned by the first stage is*

$$e = \sum_{g \in V} \deg(g) \cdot (\deg(g) + 1).$$

If $\deg(g) \leq k$ *(each gene has at most* k *parents), then* $N \leq d \cdot k(k+1) = O(d \cdot k^2)$.

Proof. We have:

$$\begin{aligned}
e &= \sum_{g \in V} |\hat{S}_g| \\
&= \sum_{g \in V} |\operatorname{bo}_g^{G^*}| \\
&= \sum_{g \in V} \left(\sum_{i \in V} \mathbb{1}\left((i,g) \in E\right) + \sum_{j \in V} \mathbb{1}\left((g,j) \in E\right) + \sum_{i,j \in V} \mathbb{1}\left(g,j\right) \in E) \, \mathbb{1}\left((i,j) \in E\right) \mathbb{1}\left(g \neq i\right) \right) \\
&= 2 \sum_{g \in V} \deg(g) + \sum_{g,i,j \in V} \mathbb{1}\left(g,j\right) \in E) \, \mathbb{1}\left((i,j) \in E\right) \mathbb{1}\left(g \neq i\right) \\
&= 2 \sum_{g \in V} \deg(g) + \sum_{j \in V} \deg(j)(\deg(j) - 1) \\
&= \sum_{g \in V} \deg(g)(\deg(g) + 1)
\end{aligned}$$

□

B. Acyclicity penalty

We recall that A is a DAG if and only if it is nilpotent. This happens if and only if all the eigenvalues of A are 0, i.e., $\operatorname{Sp}(A) = \{0\}$, where $\operatorname{Sp}(A)$ is the spectrum of A.

Therefore, a natural way to penalize a non-acyclic matrix is to penalize its eigenvalues. Yet, eigenvalues of a real matrix can be complex numbers, and the complex modulus $z \mapsto |z|$ is not complex differentiable.

Fortunately, the Perron-Froebenius theorem states that under some conditions, the eigenvalue with the largest magnitude is real-positive. We then define $h(A) = \rho(A)$ where $\rho(A) = \max\{|\lambda| \mid \lambda \in Sp(A)\}$, also called spectral radius.

Theorem 3. *If* A *is the adjacency matrix of a strongly connected graph with non-negative coefficients and with spectral radius* $\rho(A)$*, then:*

- A *has a real-positive eigenvector* v *with real-positive eigenvalue* $\rho(A)$
- *the* $\rho(A)$*-eigenspace is one-dimensional (the eigenvalue is simple)*

Finally, is it possible to compute $\nabla\rho(A)$ in that case?

Yes, we have the following result from Magnus (1985). If the eigenvalue λ_i is simple, then $A \mapsto \lambda_i(A)$ is differentiable and

$$\mathrm{d}\lambda_i(A) = \frac{v_i^\dagger \mathrm{d}A u_i}{v_i^\dagger u_i} = \frac{\operatorname{Tr}\left[dA^\top v_i u_i^\dagger\right]}{v_i^\dagger u_i} = \left\langle dA \mid \frac{v_i u_i^\dagger}{v_i^\dagger u_i} \right\rangle, \tag{11}$$

where the u_i is the right eigenvector of λ_i and v_i the left eigenvector λ_i (that is $v_i^\dagger A = \lambda_i v_i^\dagger$).

In order to compute the left and right eigenvectors associated with $\rho(A)$, we use the power method on A and $A^\top$, which requires iteration of matrix-vector multiplication. When the graph is sparse, this operation is linear in the number of edges.