HEMATOPOIETIC REGULATORY NETWORK INFERENCE

JEREMIE PERRIN, CORBIN HOPPER

1. Conditional Entropy Based Network Inference

1.1. **Approach.** One approach attempts to use conditional entropy to infer a directed graph. Although it is ultimately not robust on real data sets, it is used as a preprocessing step in the final network inference. Given that the hemapoeitic data lacks a clear temporal structure, inference about paths seem dubious. Instead, this approach considers each gene independently and appraises the most likely set of genes that regulate it. The goal is to minimize the conditional entropy of each vertex, given its input edges, relative to that of a random process. A predecessor is a vertex with a directed edge to its successor. However, the edges remain unsigned, the transfer function unspecified, and latent variables are not inferred.

Conditional entropy of a vertex given its predecessors corresponds to the amount of randomness in the vertex that is unexplained by its predecessors. The set of genes that regulates another gene should minimize the uncertainty about that gene relative to a random set of the same size. Given that biological processes are inherently stochastic, the approach should prevent certain spurious correlations. Say a network is organized such that $x \to y \to z$. Then x should not form an edge to z, since y would provide more certain information about z.

Given a graph G with vertices and edges (V, E). The goal is formally defined as follows:

$$\arg\max_{E \in G} \sum_{v \in V} I(v; e_{in}) - \mathbb{E}[I(A; B)] \tag{1}$$

Where I(Y;X) is the mutual information between the random variables Y and X. $\mathbb{E}[I(A;B)]$ refers to the expected mutual information of a subset of instances A, B that are the same size as v, e_{in},but drawn from random distributions. Over many random distributions, some small amount of distributions are expected to be correlated by chance. Equation 1 is then simplified as follows:

$$\sum_{v \in V} \arg\max_{e_{in} \in E} I(v; e_{in}) - \mathbb{E}[I(A;B)] = \sum_{v \in V} \arg\max_{e_{in} \in E} (H(v) - H(v|e_{in})) - \mathbb{E}[H(A) - H(A|B)]$$

$$= \sum_{v \in V} \arg\min_{e_{in} \in E} (-H(v) + H(v|e_{in})) - \mathbb{E}[-H(A) + H(A|B)]$$

Where I(Y;X) is the mutual information between the random variables Y and X. The entropy of v, $H(v)$, does not change over its set of input edges (and same for $H(A)$). Then equation 1 can be further rewritten as:

$$\sum_{v \in V} \arg\min_{e_{in} \in v} H(v|e_{in}) - \mathbb{E}[H(A|B)] = \sum_{v \in V} \arg\min_{e_{in} \in v} - \mathbb{E}[H(A|B)] \sum_{(i,j) \in (v, e_{in})} -p(i,j) \log_2 p(i|j) \tag{2}$$

Where H(Y|X) is the conditional entropy of the random variable Y, given X, and p(x,y) is the joint probability of events x and y occurring. One can sum over instances (say rows of a matrix)

Date: 16/11/2019.

$(v, e_{in})_0 ... (v, e_{in})_n$, instead of summing over outcomes $(i, j) \in (v, e_{in})$:

$$\arg\min_{e_{in} \in v} - \mathbb{E}[H(A|B)] \sum_{k=0}^{n} - \log p(v_k | e_{in_k}) = \arg\max_{e_{in} \in v} - \mathbb{E}[H(A|B)] \prod_{k=0}^{n} p(v_k | e_{in_k})$$

It then becomes clear that the approach maximizes the log-likelihood of the probability of observing the activity of vertices given their immediate predecessors.

It remains for $\mathbb{E}[H(A|B)]$ to be simplified. Let A be a vector of length n and B a matrix of dimensions n by m. This corresponds to the case where the output of one vertex is being evaluated, relative to that of a subset of its predecessors. If A, B are the same type of distribution, then:

$$\mathbb{E}[H(A|B_m)] = \mathbb{E}[H(A, B_m) - H(B_m)] = \mathbb{E}[H(b_1, ..., b_m, b_{m+1}) - H(b_1, ..., b_m)]$$

If there is enough data $\mathbb{E}[H(b_m)]$ should approach the entropy of n random vectors. In other words, as n approaches infinity, the probability that all m vectors are different approaches 1, and so:

$$\mathbb{E}[H(A|B)] \approx \log 2(m+1) - \log 2(m)$$

Let $m = |e_{in}|$. Then equation 2 can be rewritten as:

$$\sum_{v \in V} \arg\min_{e_{in} \in v} H(v|e_{in}) - \log_2(\frac{m+1}{m}) \tag{3}$$

1.2. **Algorithm.** An algorithm is designed for equation 3 and summarized in 1. The maximum likelihood approach has already been aggressively simplified by only considering the immediate neighbors of each node. An efficient algorithm should still avoid iterating over all possible edge combinations for each node. Since $H(Y|x_1, ..., x_m) \leq H(Y|x_1, ..., x_{m-1})$, one can safely start with all edges and iteratively remove them if:

$$H(Y|x_1, ..., x_{m-1}) - H(Y|x_1, ..., x_m) \leq \mathbb{E}[H(a|b_1, ..., b_{m-1}) - H(a|b_1, ..., b_m)]$$

$$H(Y|x_1, ..., x_{m-1}) - H(Y|x_1, ..., x_m) \leq \log_2(\frac{m}{m-1}) - \log_2(\frac{m+1}{m}) \tag{4}$$

$$H(Y|x_1, ..., x_{m-1}) - H(Y|x_1, ..., x_m) \leq \log_2(\frac{m^2}{m^2-1})$$

If none of the current edges satisfy equation 4, then the algorithm is done with that node. In contrast, one could start with an empty graph and adds edges, repeating a search through all edges only if a new edge has been added the last round. However, this approach would miss certain structures such as XOR, since $H(Y|x_1) = H(Y|x_2) = H(Y)$ would cause the algorithm to stop, despite the fact that $H(Y|x_1, x_2) > H(Y)$.

The algorithm also preprocesses nodes, such that if $H(x_i) = 0$, the node is removed. Such a gene hardly various across different cells, let alone across different anatomical stages. These genes often correspond to housekeeping genes.

Infer Graph
Begin with a fully connected graph $G(V, E)$.
For each vertex $v \in V$:
    if $H(v) = 0$: remove v
For each vertex $v \in V$:
    Infer Node (v)

Infer Node (v)
Let w be the set of all predecessors of v, such that $(w_i, v) \in G$.
For each directed edge (u,v):
    Let w\u be the set w, excluding vertex u.
    If $H(v|w\backslash u) - H(v|w) \leq \mathbb{E}[H(Y|X) - H(Y|X\backslash x)]$: 4
        Remove edge (u,v)
        Infer Node (v)

Figure 1

1.3. **Severe Limitations of The Approach.** The approach works on simple binary problems. For example, if $Y = (x_1\ NAND\ x_2\ NAND\ x_3)\ AND\ (x_4\ OR\ x_5)$, the algorithm correctly predicts that all x have a directed edge to y and there are no other edges in the graph. However, for problems such $Y = x_1\ XOR\ x_2$ it also produce edges from the output Y to both the inputs x_1, x_2. This is due to the symmetric nature of the problem, since $x_1 = Y\ XOR\ x_2$ as well.

Unfortunately, the approach does not perform well on benchmark tests. When the algorithm is tested on *Lizards*, *Coronary*, and *Asia* datasets of BNLearn, it removes all edges. It appears that the algorithm is not robust to noise. The benchmarks have few nodes, which encounter high thresholds in the algorithm and are not sufficiently correlated to remain connected. Returning to the simple binary problems, randomly flipping 10% of the bits severely impairs the inference accuracy, even when the input was repeated several times for redundancy.

1.4. **Application to Hematopoietic Regulatory Network.** Given the drawbacks of the approach, it is used as a preprocessing tool instead of inferring the hematopoietic regulatory network alone. The large amount of nodes result in lower thresholds for the expected change in conditional entropy. As such the algorithm should not be as harsh as when applied to the benchmarks. Indeed, most genes remain and the graph remains dense after running the algorithm. 4 genes were removed since their entropy alone was 0. The algorithm then continued to remove 6 more genes, reducing the total to 36. Although graph remains too dense, the algorithm trimmed some edges, resulting in an average in-degree and out-degree of 28. This reduction simplified the subsequent application of the MIIC algorithm to infer the hematopoietic regulatory network.