MAC6916 PROBABILISTIC GRAPHICAL MODELS
LECTURE 12: SCORE-BASED STRUCTURE LEARNING

DENIS D. MAUÁ

1. Introduction

Recall that a Bayesian network can be seen from two perspectives: as an efficient encoding of an independence relation, and as an efficient encoding of a high-dimensional probability distribution. The constraint-based approach to structure learning exploits the first perspective, and attempts at reconstructing a Bayesian network by analyzing the independencies on data. In the score-based approach, one invests in the second perspective, and searches for Bayesian networks that appropriately describe the data at hand. The heart of the approach consists in assigning a score value $s(G)$ to each acyclic directed graph G. The score function specifies a total order (up to equivalences) over the structures in a way that structures with better describe the data are assigned a higher value.

The score function is required to satisfy the following desiderata:

Consistency: If $(\bar{G}, p)$ is a Bayesian network, and $\mathcal{D}$ is a collection of N i.i.d. data generated from p, then

$$N \to \infty \Rightarrow \bar{G} = \arg\max_G s(G)\,.$$

Succinctness: If the log-likelihood of G and G' are equal but (G, p) has more parameters than (G', p') then $s(G) > s(G')$.

Combined, these two assumptions ensure that the generating Bayesian network (if exists) can be recovered asymptotically (up to Markov equivalences). Moreover, succinctness displays a sort of Occam's razor: when confronted with two hypothesis that are equally consistent with the facts, select the simplest one.

For computational reasons, we also usually require the following two conditions

Efficiency: The score $s(G)$ should be computed in polynomial time in the size of G and N.

Decomposability: $s(G) = \sum_{X \in \mathcal{V}} f(X, \mathrm{pa}(X))$.

The first condition enables the solution by standard procedures for combinatorial optimization. The second condition ensures that local search is efficient. We again assume complete data and categorical variables.

2. Information theory

Suppose we have a text document represented as a bag of characters, and we wish to encode it in binary form in an efficient way. A possible encoding is to use $\lceil \log_2 M \rceil$ codes to encode each possible character, where M is the total number

Date: May 18, 2016.

of distinct characters (say, 127). Then, if the document contains N occurrences of characters, the resulting encoding is $N\lceil \log M \rceil$ bits long. Very often, however, some characters are much more frequent than others. An alternative encoding is to use shorter codes for more frequent characters, and longer codes for less frequent characters. For example, if our alphabet contains only characters a, b and c, and if they occur with probabilities $p(a) = 1/2$, $p(b) = 2/6$ and $p(c) = 1/6$, then a more succinct encoding is to represent a by the code 0, b by the code 10 and c by the code 111. This encoding takes *on average* $N(1 \cdot p(a) + 2 \cdot p(b) + 3 \cdot p(c)) = 5N/3$ bits. Compare with a "flat" encoding which uses $\lceil \log_2 3 \rceil = 2$ bits per symbol and takes $2N$ bits (Note that $5/3 < 2$).

Let $p(X)$ be the distribution of occurrence of letters in a document. Then, an efficient encoding is to use a code of size $\lceil -\log_2 p(x) \rceil$ to represent each character x. The average size of such a encoding is $N \sum_x p(x) \lceil -\log_2 p(x) \rceil$ and it is minimal. Note that we could allow more symbols in our encoding (i.e., non-binary codes) and change the base of the logarithm accordingly, so that the base of the logarithm is somewhat arbitrary. The functional $-\sum_x p(x) \ln p(x)$ can be seen as a measure of the informativeness of a random variable X. The more informative is X, the more succinct is an encoding of a data set of i.i.d. realizations $X = x$ generated according to $p(X)$. This is captured by the concept of entropy:

entropy

Definition 1. The *entropy* of a set of variables $\mathcal{X}$ under the distribution $p(\mathcal{X})$ is[1]

$$\mathbb{H}_p(\mathcal{X}) = -\sum_{\mathcal{X}} p(\mathcal{X}) \ln p(\mathcal{X}) \,.$$

Entropy satisfies a number of useful properties [1]:

Proposition 1. *The following statements are true.*

(i) $0 \leq \mathbb{H}_p(\mathcal{X}) \leq \ln |dom(\mathcal{X})|$.
(ii) $\mathbb{H}_p(\mathcal{X}) = 0$ *if and only if* p *is degenerate (i.e., it assigns all mass to a single configuration).*
(iii) $\mathbb{H}_p(\mathcal{X} \cup \mathcal{Y}) = \mathbb{H}_p(\mathcal{X}) + H_p(\mathcal{Y})$ *if and only if* $\mathcal{X} \perp\!\!\!\perp \mathcal{Y}$ *(under* p*).*

The informativeness of a variable is often altered by knowledge of another variable. This is formalized by the concept of mutual information:

mutual information

Definition 2. The *mutual information* of two sets of variables $\mathcal{X}$ and $\mathcal{Y}$ under distribution $p(\mathcal{X} \cup \mathcal{Y})$ is

$$\mathbb{I}_p(\mathcal{X}, \mathcal{Y}) = \sum_{\mathcal{X},\mathcal{Y}} p(\mathcal{X} \cup \mathcal{Y}) \ln \frac{p(\mathcal{X} \cup \mathcal{Y})}{p(\mathcal{X})p(\mathcal{Y})} \,.$$

The mutual information measures the average reduction in uncertainty about the value of $\mathcal{X}$ provided by knowledge of the value of $\mathcal{Y}$ and vice-versa. We define $\mathbb{I}_p(\mathcal{X}, \emptyset) = 0$.

Proposition 2. *The following statements are true.*

(i) $\mathbb{I}_p(\mathcal{X}, \mathcal{Y}) = \mathbb{I}_p(\mathcal{Y}, \mathcal{X})$.
(ii) $\mathbb{I}_p(\mathcal{X}, \mathcal{Y}) = \mathbb{H}_p(\mathcal{X}) + \mathbb{H}_p(\mathcal{Y}) - \mathbb{H}_p(\mathcal{X} \cup \mathcal{Y})$.
(iii) $\mathbb{I}_p(\mathcal{X}, \mathcal{Y}) \geq 0$.
(iv) $\mathbb{I}_p(\mathcal{X}, \mathcal{Y}) = 0$ *if and only if* $\mathcal{X} \perp\!\!\!\perp \mathcal{Y}$ *(under* p*).*

[1] Where we assume $0 \ln 0 = 0$.

(v) $\mathbb{I}_p(\mathcal{X}, \mathcal{Y} \cup \mathcal{Z}) \geq \mathbb{I}_p(\mathcal{X}, \mathcal{Y})$ *with equality if only if* $\mathcal{X} \perp\!\!\!\perp \mathcal{Z} \mid \mathcal{Y}$.

The last property is particularly interesting. Let X be some variable and $\mathcal{Y}$ and $\mathcal{Z}$ be disjoint sets of variables. If $X \perp\!\!\!\perp \mathcal{Z} \mid \mathcal{Y}$ then $\mathbb{I}(X, \mathcal{Y} \cup \mathcal{Z}) = \mathbb{I}(X, \mathcal{Y})$. Otherwise, $\mathbb{I}(X, \mathcal{Y} \cup \mathcal{Y}) > \mathbb{I}(X, \mathcal{Y})$.

Recall that the log-likelihood of a Bayesian network (G, p) can be written as

$$LL(G, p) = \sum_{X \in \mathcal{V}} \sum_{X, \mathrm{pa}(X)} N[X, \mathrm{pa}(X)] \ln p(X|\mathrm{pa}(X)) \,.$$

Let $\tilde{p}(\mathcal{X}) = N[\mathcal{X}]/N$ denote the empirical distribution. Then,

$$LL(G, p) = N \sum_{X \in \mathcal{V}} \sum_{X, \mathrm{pa}(X)} \tilde{p}(X, \mathrm{pa}(X)) \ln p(X|\mathrm{pa}(X)) \,.$$

Moreover,

$$\begin{aligned}
LL^{\text{MLE}}(G) = LL(G, \tilde{p}) &= \arg\max_p LL(G, p) \\
&= N \sum_{X \in \mathcal{V}} \sum_{X, \mathrm{pa}(X)} \tilde{p}(X, \mathrm{pa}(X)) \ln \tilde{p}(X|\mathrm{pa}(X)) \\
&= N \sum_{X \in \mathcal{V}} \sum_{X, \mathrm{pa}(X)} \tilde{p}(X, \mathrm{pa}(X)) \ln \frac{\tilde{p}(X, \mathrm{pa}(X))}{\tilde{p}(\mathrm{pa}(X))} \frac{\tilde{p}(X)}{\tilde{p}(X)} \\
&= N \sum_{X \in \mathcal{V}} \sum_{X, \mathrm{pa}(X)} \tilde{p}(X, \mathrm{pa}(X)) \ln \frac{\tilde{p}(X, \mathrm{pa}(X))}{\tilde{p}(X)\tilde{p}(\mathrm{pa}(X))} \tilde{p}(X) \\
&= N \sum_{X \in \mathcal{V}} \mathbb{I}_{\tilde{p}}(X, \mathrm{pa}(X)) - N \sum_{X \in \mathcal{V}} \mathbb{H}_{\tilde{p}}(X) \,.
\end{aligned}$$

Note that only the first sum in the equation above depends on the graphical structure G**.** Thus, the log-likelihood of a Bayesian network with MLE parameters is (up to constants) the sum of the mutual information between every variable and its parents. The MLE log-likelihood is a consistent estimator of the structure:

Theorem 1. *Suppose the data were generated by a Bayesian network* (G, p)*. In the data size limit (i.e., when* $N \to \infty$*), we have that* $\tilde{p} \to p$*, and*

$$LL^{\text{MLE}}(G) \geq LL^{\text{MLE}}(G') \,,$$

for any structure G'*.*

However, the MLE log-likelihood is not succinct:

Theorem 2. *Let* G' *be a structure obtained by inserting arcs into* G *(and maintaining acyclicity). Then,*

$$LL^{\text{MLE}}(G') \geq LL^{\text{MLE}}(G) \,,$$

Proof. It follows directly from Proposition 2(v), using $\mathbb{I}(X, \mathcal{Y} \cup \mathcal{Z}) \geq \mathbb{I}(X, \mathcal{Y})$ with $\mathcal{Y}$ the being the parents of X in G and $\mathcal{Z}$ the additional parents in G'. □

Hence, the true graph is not the only maximizer of LL^{MLE} in the sample limit. More importantly, there is zero probability that two independent variables will remain independent under the empirical distribution with finite data size. Hence, LL^{MLE} prefers complex models over simpler ones, and will almost always lead to complete structures.

The solution is to penalize the log-likelihood by the complexity of the structure. A general score function that adopts this strategy is the *penalized log-likelihood score*:

penalized log-likelihood score

$$PLL(G) = LL^{\text{MLE}}(G) - \psi(N)\text{size}(G)\,,$$

where

$$\text{size}(G) = \sum_{X\in\mathcal{V}} (|\text{dom}(X)| - 1) \prod_{Y\in\text{pa}(X)} |\text{dom}(Y)|$$

is the number of free-parameters in a Bayesian network with structure G. The function $\psi(N)$ is a non-decreasing function that weights the penalization according to the data size. Larger data sets allow more complex models to be learned, while smaller data sets lead to simpler models. In the rest of this lecture, we will present and justify different sorts of penalized log-likelihood scores which differ by the function $\psi(N)$.

The derivation of $LL^{\text{MLE}}(G)$ in termos of mutual information and entropy, show that the (MLE) log-likelihood grows linearly with the size of the dataset. Hence, a linear penalization funcition $\psi(N) = O(N)$ creates scores that assign equal importance to data fitness and to model succinctness irrespective of the data size. This will usually lead to non-consisistent estimators. Sublinear penalization functions, the most typical being logarithmic penalization $\psi(N) = O(\log N)$, balance data fitness and model complexity differently for different data sizes, and prefer data fitness over model succinctness in the sample limit. Hence, sublinear penalization functions typically induces consistent and sunccinct estimators.

3. Minimum Description Length

Consider again our problem of encoding a text document. Previously, we have considered a fixed distribution of occurrences of characters. We can instead imagine having several distributions (models) $p_1, \ldots, p_h$ available. To encode a document we select one distribution p_i and build a code using (approximately) $-\log_2 p_i(x)$ bits for each word. Thus, the higher the likelihood of the model the shorter the encoding.

To be able to recover the original document, we need to also encode the model used. This can be made, in our simply case, by using $-\log_2 h$ bits to represent the index of the model used. When models are used with skewed probabilities, we can use shorter codes to encode more frequently used models and larger codes to encode less frequently used codes. The average length of a document encoded with model p_i is (approximately) $\mathbb{H}_{p_i}(X)$ + size(i), where the latter is the size of the encoding of the model. This is the so-called *description length* of a document (data) according to a model p_i of the distribution of the data.

description length

The Principle of Minimum Description Length (MDL) asserts that we should prefer a model with the shortest description of the data. Intuitively, a shorter description communicates the data efficiently without wasting bits for representing irrelevant or redundant information. To put it differently, the MDL states the the best model is the one that compresses the data the most. Compare with the information provided by log-likelihood, which does not distinguishes between models that assign the same probability to the data.

So consider as our class of models all Bayesian networks with MLE parameters from a data set $\mathcal{D}$ of N records. Suzuki proposes an encoding of the model that uses $\log_2 N/2$ bits of precision to encode each probability parameter in a network

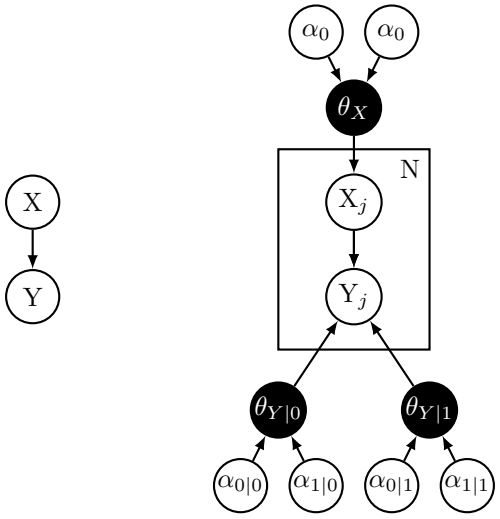

FIGURE 1. Left: Bayesian network structure. Right: Plate model of the parameter learning Bayesian network.

[2]. This allows more complex (more precise) probabilities to be learned according to the amount of data. Thus, the total description length of a data set $\mathcal{D}$ according to a Bayesian network (G, θ^{MLE}) is

$$\text{MDL}(G) = LL^{\text{MLE}}(G) - \frac{\log_2 N}{2}\text{size}(G)\,.$$

Thus, Suzuki's MDL is a penalized log-likelihood with $\psi(N) = \log_2 N/2$. It is straightforward to see that the MDL score is decomposable and efficient.

4. Bayesian Score

An alternative approach is to use a Bayesian inference scheme and score structures based on their posterior probability according to a learning model:

$$\mathbb{P}(G|\text{data}) = \frac{\mathbb{P}(G)\mathbb{P}(\text{data}|G)}{\mathbb{P}(\text{data})}\,.$$

Note that the denominator on the right-hand side is constant w.r.t. the structure G, and we can discard it without affecting the order imposed by the score, and take the logarithm:

$$\text{BAYES}(G) = \ln \mathbb{P}(G) + \ln \mathbb{P}(\text{data}|G) = \ln \mathbb{P}(G) + \ln \int_\theta \mathbb{P}(\text{data}|G,\theta)\mathbb{P}(\theta|G)d\theta\,.$$

The quantity $\mathbb{P}(\text{data}|G)$ is known as the *marginal likelihood*. Unlike the model likelihood $\mathbb{P}(\text{data}|G,\theta)$, the marginal likelihood considers all parametrizations with a given structure. This amounts to considering different pertubations to data, and prevents overfitting. It also acts as a penalization on the number of parameters.

marginal likelihood

To be able to use this approach, we need to decide on a prior distribution for the parameters and structure. We can resort to the same model used for parameter learning, which assigns Dirichlet distributions with yperparameters $\alpha_{X|\nu}$ to the parameters $\theta_{X|\nu}$. Figure 1 shows an example of the parameter model for a simple structure.

Due to the parameter independence assumption of the learning model, one can show that

$$\mathbb{P}(\text{data}|G,\theta)\mathbb{P}(\theta|G) = \prod_{X\in\mathcal{V}}\prod_{\nu\sim\text{pa}(X)} \mathbb{P}(N[X,\text{pa}(X)=\nu]|\theta_{X|\nu})\mathbb{P}(\theta_{X|\nu})\,.$$

Each local factor is the product of a Discrete Distribution by a Dirichlet Distribution, and is thus a Dirichlet distribution with parameters $N[X,\nu]+\alpha_{X|\nu}$. The expected value of Dirichlet distribution over its parameters is a well-known problem which admits the following closed-form solution

$$\int_{\theta_{X|\nu}} \mathbb{P}(N[X,\text{pa}(X)=\nu]|\theta_{X|\nu})\mathbb{P}(\theta_{X|\nu})d\theta_{X|\nu} = \frac{\Gamma(\alpha_X)}{\Gamma(N[\text{pa}(X)=\nu]+\alpha_X)}\prod_{x\sim X}\frac{\Gamma(N[X=x,\text{pa}(X)=\nu]+\alpha_{X|\nu}(x)}{\Gamma(\alpha_{X|\nu}(x))}\,,$$

where $\alpha_X = \sum_{x\sim X}\alpha_{X|\nu}(x)$, and $\Gamma(X)$ is the Gamma function, which extends the factorial function to the real line (in particular, $\Gamma(i)=i!$ if i is a positive integer). The Bayesian score is then

$$\text{BAYES}(G) = \ln\mathbb{P}(G) + \sum_{X\in\mathcal{V}}\sum_{\nu\sim\text{pa}(X)}\left(\ln\frac{\Gamma(\alpha_{X|\nu})}{\Gamma(N[\text{pa}(X)=\nu]+\alpha_{X|\nu})} + \sum_{x\sim X}\ln\frac{\Gamma(N[X=x,\text{pa}(X)=\nu]+\alpha_{X|\nu}(x))}{\Gamma(\alpha_{X|\nu}(x))}\right).$$

The Bayesian score above requires two differnt background information: the prior $\mathbb{P}(G)$ over the structures and the hyperparameters $\alpha_{X|\nu}$ over the paramters. The former is not very significant, since the loglikelihood grows linearly with the data size while $\mathbb{P}(G)$ is (by definition) constant w.r.t. N. Hence, the choice of $\mathbb{P}(G)$ will usually play a minor role in guiding the choice of a structure. A common assumption is *structure decomposability*, which forces

structure decomposability

$$\mathbb{P}(G) = \prod_X \mathbb{P}(\text{pa}(X))\,.$$

One possiblity is to assign prior probabilities based on the number of parents.

Regarding the parameter priors, we usually follow the principle of likelihood equivalence which prescribes that **two Marvkov equivalent structures should be assigned the same score**. The rationale is that two structures that represent the same independence relation, have the same number of parameters and the same log-likelihood, and should thus be indistinguishible (unless we have a particular, subjective reason to prefer one over the other). The log-likelihood and the MDL scores satisfy likelihood equivalence. It can be shown that for the case of categorical variables with Dirichlet priors and parameter independence, likelihood equivalence is satisfied by the Bayesian score if and only if (assuming structure decomposability)

$$\alpha_{x|\nu} = \alpha\cdot p(x,\nu)\,,$$

where α is a fixed parameter (over variables) known as *equivalence sample size* and $p(X,\text{pa}(X))$ is a distribution over X and $\text{pa}(X)$ (this can be specified as an

equivalence sample size

auxiliary Bayesian network). Typically, we have that

$$p(x,\nu) = \frac{1}{|\sigma_X| \prod_{Y\in \mathrm{pa}(X)} |\Omega_Y|} = \mathrm{Uniform}(X, \mathrm{pa}(X))\,.$$

The parameter α regulates overfitting; typical values for it are usually in the range $[1, 2]$.

The Bayesian score is efficient (for complete data), decomposable, consistent and succinct. Thus, the Bayesian score recovers the true structure asymptotically.

5. Bayesian Information Criterion

As data accumulates, the posterior distribution $\mathbb{P}(G|\text{data})$ becomes peaked around the maximum a posterior value $\max_G \max_\theta \mathbb{P}(G, \theta|\text{data})$ and the prior $\mathbb{P}(G, \theta)$ becomes irrelevant. Hence, the Bayesian score converges to a sort of the penalized log-likelihood known as Bayesian Information Criterion (BIC) [3]:

$$\begin{aligned}\mathrm{BIC}(G) &= LL^{\mathrm{MLE}}(G) - \frac{\ln N}{2}\mathrm{size}(G)\\ &= \lim_{N\infty} \mathrm{BAYES}(G) - O(1)\,.\end{aligned}$$

Note the resemblance to the MDL score, the only difference being the base of the logarithm. Hence, with large sample size, the Bayesian score converges to roughly the MDL score. Empirical results show that the BIC score (or MDL) and the Bayesian score achive comparable performance in recovering the true structure, and in maximizing the holdout loglikelihood (with the Bayesian score usually being slightly superior). Since the BIC score has a simpler formula and do not require any parameters, it is often adopted in practical uses (althogh the usage of the Bayesian score has been increasing lately).

The BIC score is also consistent and succinct, and will thus recover the true structure given a sufficiently large data set. It is also decomposable and efficient (when data is complete).

6. Learning Algorithms

Given a decomposable, polynomial-time computable score function, finding the optimal structure is an NP-hard problem, and practioners often resort to greedy techinques. Even though there has been recently some succes in exact (anytime) methods, they are still constrained to moderate domains (say, about 60 variables). Given the NP-hardness of the proble, we do not expect that exact methods will scale for larger domains, so that approximate algorithms continue to be the best optimum. There are two family of greedy algorithms: the ones that search directly the space of graph structures, and the ones that search the space of topological orderings.

6.1. DAG-based search.

The overall behaviour of algorithms in this class is to start with a graph G and iteratively propose a local modification to G that improves the score until no modification can be done. This strategy is usually repeated several times with different initial candidate graphs, to escape poor local optima. The typical local modifications are

- Add a non-existing arc $X \to Y$ (as long as it does not introduces a cycle).
- Remove an existing arc $X \to Y$.

- Reverse an existign arc $X \to Y$ (i.e., remove $X \to Y$ and add $Y \to X$) as long as it does not introduce a cycle.

Due to the decomposability of the score function, these operations can be performed efficiently and in any order, and allow some degree of parallelism.

6.2. **Order-based search.** One of the main difficulties when searching over the DAG space is to ensure that each local modification leads to an acyclic graph. This can be avoided by searching over the space of topological orderings.

Recall that a directed graph is acyclic if and only if it admits a topological ordering of its nodes. Consider a fixed topological ordering of the variables $X_1, \ldots, X_n$. Any way of assigning parents to variables that satisfies this ordering will thus produce an acyclic graph. That is, any procedure that searches for the parent of X_i among $X_1, \ldots, X_{i-1}$ will produce acyclic graphs (without the need of checking). Since every acyclic graph is consistent with a topological ordering, this search will eventually find the maximizing structure. In other words, we have that

$$\max_G s(G) = \max_G \sum_{i=1}^n f(X_i, \mathrm{pa}(X)) = \max_\sigma \sum_{i=1}^n \max_{\mathrm{pa}(X_i) \subseteq \{X_j : \sigma(j) < \sigma(i)\}} f(X_i, \mathrm{pa}(X_i))\,.$$

The equation above shows that given a fixed ordering, the structure learning problem decomposes into smaller and independent problems of parent assignment

$$\max_{\mathrm{pa}(X_i) \subseteq \{X_j : \sigma(j) < \sigma(i)\}} f(X_i, \mathrm{pa}(X_i))\,.$$

These problems can be solved relatively efficiently by employing a greedy search on the space of subsets of the smaller nodes with an upper bound on the number of parents. The number of topological orderings is exponentially smaller than the number of DAGs; hence searching on the ordering space is usually much more efficient than searching in the DAG space. A popular algorithm designed by Tessyer and Koller [4] is to constrain the ordering search to local moves that exchange two adjacent positions in the ordering:

$$\sigma : X_1, \ldots, X_{i-1}, X_i, X_{i+1}, \ldots, X_n \;\mapsto\; \sigma' : X_1, \ldots, X_i, X_{i-1}, X_{i+1}, \ldots, X_n\,.$$

Due to the (assumed) decomposability of the score function, every move of this type requires only the re-computation of the local scores of the associated positions $f(X_i, \mathrm{pa}(X_i))$ and $f(X_{i-1}, \mathrm{pa}(X_{i-1}))$.

7. Reading

- Tessier & Koller, Ordering-based Search: A Simple and Effective Algorithm for Learning Bayesian Networks, UAI 2005 [4].

8. Exercises

You might find the following result useful for solving these exercises.

Theorem 3 (Gibbs Inequality)**.** *$KL(p, q) \geq 0$ for any p, q and $KL(p, q) = 0$ if and only if $p = q$.*

Exercise 1. *Prove Proposition 1.*

Exercise 2. *Prove Proposition 2. Hint: note that $\mathbb{I}(\mathcal{X}, \mathcal{Y}) = KL(p(\mathcal{X} \cup \mathcal{Y}), p(\mathcal{X})p(\mathcal{Y}))$.*

Exercise 3. *Prove Theorem 1. Hint: use Gibbs Inequality with the distribution $q(\mathcal{V})$ induced by a structure G'.*

9. Assignment

No assignment this week.

References

[1] D.J.C. MacKay. Information Theory, Inference, and Learning Algorithms. Cambridge University Press, 2003.

[2] J. Suzuki. Learning Bayesian belief networks based on the minimum description length principle. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 462–470, 1996.

[3] G. Schwarz. Estimating the dimension of a model. The Annals of Statistics 6, pp. 461–464, 1978.

[4] M. Teyssier and D. Koller Ordering-based Search: A Simple and Effective Algorithm for Learning Bayesian Networks. In *Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence*, pp. 584–590.