CS262a Bayesian Networks

🅭 Tingfeng Xia @ UCLA

Winter Quarter, 2022

Preface

This document is intentionally kept short, and is not a complete summary of what is covered in this course. Excerpt from online:

> The objective of this class is to provide an in-depth exposition of knowledge representation, reasoning, and machine learning under uncertainty using the framework of Bayesian networks. Both theoretical underpinnings and practical considerations will be covered, with a special emphasis on constructing and learning graphical models, and on various exact and approximate inference algorithms. Additional topics include logical approaches to probabilistic inference, compilation techniques, sensitivity analysis, undirected graphical models, and statistical relational learning.

Instructor: Professor Adnan Darwiche

Book: Adnan Darwiche. Modeling and Reasoning with Bayesian Networks. Cambridge University Press 2009.

Contents

1 Propositional Logic

1.1 Principle Logical Forms

Inconsistent. Something that never holds; $\text{Mods}(\cdot) =$; $\Pr(\alpha) = 0$

Valid. Something that always holds; $\text{Mods}(\cdot) = \Omega$; $\Pr(\alpha) = 1$

Equivalent. $\text{Mods}(\alpha) = \text{Mods}(\beta)$

Mutual Exclusive. $\text{Mods}(\alpha) \cap \text{Mods}(\beta) = \emptyset$

Exhaustive. $\text{Mods}(\alpha) \cup \text{Mods}(\beta) = \Omega$

Entailment / Implication. $\alpha \models \beta \triangleq \text{Mods}(\alpha) \subseteq \text{Mods}(\beta)$

1.2 Equivalent Forms

- $\text{Mods}(\alpha \wedge \beta) = \text{Mods}(\alpha) \cap \text{Mods}(\beta)$
- $\text{Mods}(\alpha \vee \beta) = \text{Mods}(\alpha) \cup \text{Mods}(\beta)$
- $\text{Mods}(\neg\alpha) = \overline{\text{Mods}(\alpha)}$

1.3 Instantiation Agreement

Two instantiation, each of which can cover a subset of different varaibles, are said to be compatible with each other if they argree on all common variables. Denoted as $\mathbf{x} \sim \mathbf{y}$.

1.4 Information Theory

Entropy.

$$\text{ENT}(X) = -\sum_{x} \Pr(x) \log \Pr(x) \tag{1.1}$$

where $0 \log 0 = 0$ by convention. With a higher entropy, we say that it is more chaotic.

Conditional Entropy.

$$\text{ENT}(X|Y) = \sum_{y} \Pr(y)\text{ENT}(X|y) \quad \text{where} \quad \text{ENT}(X|y) = -\sum_{x} \Pr(x|y) \log \Pr(x|y) \tag{1.2}$$

Conditioning never increases the entropy, i.e.

$$\text{ENT}(X|Y) \leq \text{ENT}(X) \tag{1.3}$$

Mutual Information

$$\begin{aligned}\mathrm{MI}(X;Y) &= \sum_{x,y} \Pr(x,y) \log \frac{\Pr(x,y)}{\Pr(x)\Pr(y)} && (1.4)\\ &= \mathrm{ENT}(X) - \mathrm{ENT}(X|Y) && (1.5)\\ &= \mathrm{ENT}(Y) - \mathrm{ENT}(Y|X) && (1.6)\end{aligned}$$

Conditional Mutual Information

$$\begin{aligned}\mathrm{MI}(X;Y|Z) &= \sum_{x,y,z} \Pr(x,y,z) \log \frac{\Pr(x,y|z)}{\Pr(x|z)\Pr(y|z)} && (1.7)\\ &= \mathrm{ENT}(X|Z) - \mathrm{ENT}(X|Y,Z) && (1.8)\\ &= \mathrm{ENT}(Y|Z) - \mathrm{ENT}(Y|X,Z) && (1.9)\end{aligned}$$

2 Probability Calculus

2.1 Bayesian Conditioning

Bayesian Condition is specified by the formula

$$\Pr(\alpha|\beta) = \frac{\Pr(\alpha \wedge \beta)}{\Pr(\beta)} \tag{2.1}$$

In particular, not to be confused with Bayesian inference (to be added later).

2.2 Independence and Notations

Independence.

$$\begin{aligned}\alpha \perp\!\!\!\perp \beta &\iff \Pr(\alpha|\beta) = \Pr(\alpha) \vee \Pr(\beta) = 0 && (2.2)\\ &\iff \Pr(\alpha \wedge \beta) = \Pr(\alpha)\Pr(\beta) && (2.3)\end{aligned}$$

Conditional Independence.

$$\begin{aligned}(\alpha \perp\!\!\!\perp \beta)|\gamma &\iff \Pr(\alpha|\beta \wedge \gamma) = \Pr(\alpha|\gamma) \vee \Pr(\beta \wedge \gamma) = 0 && (2.4)\\ &\iff \Pr(\alpha \wedge \beta|\gamma) = \Pr(\alpha|\gamma)\Pr(\beta|\gamma) \vee \Pr(\gamma) = 1 && (2.5)\end{aligned}$$

Set Independence

$$I_{\Pr}(X,Z,Y) \iff (x \perp\!\!\!\perp y)|z, \quad \forall x,y,z \in X,Y,Z \tag{2.6}$$

3 Bayesian Networks

3.1 Soft Evidence

3.1.1 All Things Considered Method

We normalize / rescale w according to new evidence.

$$\mathrm{Pr}^{'}(w) = \begin{cases} \frac{\mathrm{Pr}^{'}(\beta)}{\mathrm{Pr}(\beta)}\mathrm{Pr}(w) & \text{if} w \models \beta \\ \frac{\mathrm{Pr}^{'}(\neg\beta)}{\mathrm{Pr}(\neg\beta)}\mathrm{Pr}(w) & \text{if} w \models \neg\beta \end{cases} \tag{3.1}$$

The closed form is called the Jefferey's Rule.

Jeffery's Rule

$$\mathrm{Pr}^{'}(\alpha) = q\mathrm{Pr}(\alpha|\beta) + (1-q)\mathrm{Pr}(\alpha|\neg\beta) \tag{3.2}$$

Jeffery's Rule - General Case

$$\mathrm{Pr}^{'}(\alpha) = \sum_{i=1}^{n} \mathrm{Pr}^{'}(\beta_i)\mathrm{Pr}(\alpha|\beta_i) \tag{3.3}$$

3.1.2 Nothing-else Considered Method

Odds.

$$O(\beta) = \frac{\mathrm{Pr}(\beta)}{\mathrm{Pr}(\neg\beta)} \tag{3.4}$$

Bayes Factor

$$k = \frac{O'(\beta)}{O(\beta)} = \frac{Pr'(\beta)/\mathrm{Pr}^{'}(\neg\beta)}{...} \tag{3.5}$$

from where we can expand and organize

$$\mathrm{Pr}^{'}(\beta) = \frac{k\mathrm{Pr}(\beta)}{k\mathrm{Pr}(\beta) + \mathrm{Pr}(\neg\beta)} \tag{3.6}$$

Closed Form Solution.

$$\mathrm{Pr}^{'}(\alpha) = \frac{k\mathrm{Pr}(\alpha \wedge \beta) + \mathrm{Pr}(\alpha \wedge \neg\beta)}{k\mathrm{Pr}(\beta) + \mathrm{Pr}(\neg\beta)} \tag{3.7}$$

3.2 Noisy Sensors

$$O'(\beta) = \underbrace{\frac{1-f_n}{f_p}}_{k^+} O(\beta) \qquad O'(\beta) = \underbrace{\frac{f_n}{1-f_p}}_{k^-} O(\beta) \tag{3.8}$$

3.3 Markov Assumptions

$$\text{Markov}(G) = \{I_{\text{Pr}}(V, \text{Parents}(V), \text{ND}(V)\}_V \tag{3.9}$$

where ND means non-descendants, and includes all nodes except for V, Parents(V) and Descendants(V) (all the way till leaf)

3.4 Graphoid Axioms

Symmetry.

$$I_{\text{Pr}}(X, Z, Y) \iff I_{\text{Pr}}(Y, Z, X) \tag{3.10}$$

Decomposition.

$$I_{\text{Pr}}(X, Z, Y \cup W) \implies I_{\text{Pr}}(X, Z, Y) \wedge I_{\text{Pr}}(X, Z, W) \tag{3.11}$$

Weak Union.

$$I_{\text{Pr}}(X, Z, Y \cup W) \implies I_{\text{Pr}}(X, Z \cup Y, W) \tag{3.12}$$

Contraction.

$$I_{\text{Pr}}(X, Z, Y) \wedge I_{\text{Pr}}(X, Z \cup Y, W) \implies I_{\text{Pr}}(X, Z, Y \cup W) \tag{3.13}$$

Triviality.

$$I_{\text{Pr}}(X, Z, \emptyset) \tag{3.14}$$

3.5 Positive Graphoid Axioms

... includes everything from Graphoid Axioms (Section 3.4) and in addition has

Intersection.

$$I_{\text{Pr}}(X, Z \cup W, Y) \wedge I_{\text{Pr}}(X, Z \cup Y, W) \implies I_{\text{Pr}}(X, Z, Y \cup W) \tag{3.15}$$

3.6 D-seperation and Graphical Rules

There are three types of valves to consider,

- A sequential valve ($\rightarrow W \rightarrow$) is closed iff variable W appears in $\mathbf{Z}$,
- A divergent valve ($\leftarrow W \rightarrow$) is closed iff variable W appears in $\mathbf{Z}$, and
- A convergent valve ($\rightarrow W \leftarrow$) is closed iff neither variable W not any of its descendants appears in $\mathbf{Z}$. [3.1]

3.7 D-seperation Linear Prune Theorem

(Theorem 4.1) Testing whether $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Y}$ in a DAG G is equivalent to testing whether $\mathbf{X}$ and $\mathbf{Y}$ are disconnected[3.2] in a new DAG G', which is obtained by pruning DAG as follows:

- We delete any leaf node W from DAG G as long as $W \notin \mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$. This process is repeated until no more nodes can be deleted.
- We delete all edges outgoing from nodes in $\mathbf{Z}$, the evidence set.

3.8 D-seperation Properties

Soundness.

$$\text{dsep}_G(X, Z, Y) \implies I_{\text{Pr}}(X, Z, Y) \tag{3.16}$$

(Weak) Completeness. There exists a parametrization Θ that for every DAG G such that

$$I_{\text{Pr}}(X, Z, Y) \iff \text{dsep}_G(X, Z, Y) \tag{3.17}$$

4 Inference by Variable Elimination

4.1 Interaction Graphs

The interaction graph is a tool to help us determine the run time complexity of variable elimination. Also it can help us choose elimination order. (Heuristics described in Section 4.2. It is defined as follows: Let $f_1, ..., f_n$ be a set of factors. The interaction graph G of these factors is an undirected graph constructed as follows. The nodes of G are the variables that appear in factors $f_1, ..., f_n$. There is an edge between two variables in G iff those variables appear in he same factor.

[3.1] This will be referred to as a collider node later, as it is one that behaves differently from the rest two.

[3.2] Disregard arrows, we only care about connectedness here.

The interaction graph is a living creature, meaning that it is continuously updated as we perform variable elimination. When we eliminate a node n, we have to add edges (if necessary) to make sure all neighbors of n are fully connected.

Order Width. The order width can be determined by keeping track of the evolving interaction graphs. Order width is defined as the maximum number of neighbors when nodes is deleted from the interaction graph. Of course, we have to add edges (if necessary) after every deletion.

4.2 Elimination Order

Choosing an optimal elimination order in VE is NP-hard. We resort to heuristics.

Minimum Degree Ordering. When choosing node to eliminate, choose the one that has the smallest number of neighbours in the interaction graph.

Minimum Fill Ordering. ..., choose the one that results in minimum number of newly added edge in the interaction graph.

4.3 Tree Width

Tree width is the smallest possible elimination order width for a network. The tree width quantifies the the extent to which a network resembles a tree structure. The more similar a network is to a tree structure, the smaller the tree width. No complete elimination order can have have an order width less than network tree width. Also, there always exists an elimination order that has order width the same as tree width. However, determining this order is NP-hard.

Poly-Tree Tree Width. Poly trees have known tree width of k where k is the maximum number of parents that any node may have.

5 Inference by Factor Elimination

5.1 Factor Elimination

The elimination of factor f_i from a set of factors S is a two step process. We first eliminate all variables V that appear only in factor f_i and then multiply the result $\sum_V f_i$ by some other factor $f_j \in S$.

Projection. Factor projection operation is defined as

$$\text{project}(f, Q) \triangleq \sum_{vars(f)-Q} f \tag{5.1}$$

which is essentially summing out all variables not in Q. In the vanilla factor elimination algorithm (`FE1`), the projection is used in the return statement. This is essentially a normalization step (onto variables specified only).

5.2 Elimination Trees

Variables. $vars(i)$ denotes the variables mentioned at node i. $vars(i, j)$ denotes all variables mentioned in nodes to the i-side of the graph (inclusive). Hence, it holds that $vars(i) \subseteq vars(i, j)$.

Separators.

$$S_{ij} \triangleq vars(i, j) \cap vars(j, i) \tag{5.2}$$

Clusters.

$$C_i \triangleq vars(i) \cup \bigcup_j S_{ij} \tag{5.3}$$

5.3 Message Passing Formulation

The Message that i sends to j is

$$M_{ij} \triangleq \text{project}\left(\phi_i \prod_{k \neq j} M_{ki}, S_{ij}\right) \tag{5.4}$$

where the product is off all the messages from edges except for the output direction and S_{ij} is the separator defined in Section 5.2.

5.4 Join Tree Algorithm

Definition. A join tree for a DAG G is a pair $(T, \mathbb{C})$ where T is a tree and $\mathbb{C}$ is a function that maps each node i in tree into a label $\mathbb{C}_i$, called a cluster. The join tree must satisfy the following properties,

- The cluster $\mathbb{C}_i$ is a set of nodes from DAG G
- Each family in the DAG G must appear in some cluster $\mathbb{C}_i$
- If a node appears in two clusters $\mathbb{C}_i$ and $\mathbb{C}_j$, then it must also appear in every cluster $\mathbb{C}_k$ in between nodes i, j. (Join-tree Property.)

Join Tree Width. The width of a join tree is equal to the size of the largest cluster minus one.

Minimal Join Tree. A join tree for a DAG G is said to be minimal if it ceases to be a join tree for G once we remove a variable from one if its clusters.

DAG Tree Width. The tree width of a DAG is defined as the width of its best join tree. [5.1]

Elimination Trees are Join Trees. The clusters of an elimination tree satisfy the three properties of a join tree states before.

6 Inference by Conditioning

6.1 Run time Comparison

6.1.1 Variable Elimination (VE)

Let w be the width of the tree, n denote the number of variables, and $|Q|$ as query variable size.

Time Complexity. $\mathcal{O}(n \exp(w))$

Space Complexity.

$$\mathcal{O}(n \exp(w) + n \exp(|Q|)) \equiv \mathcal{O}(n \exp(w)), \quad \text{if } |Q| < \infty \tag{6.1}$$

6.1.2 Massage Passing

One Specific Message. The cost to pass one specific message is

$$\mathcal{O}(\exp(|C_i|)) \tag{6.2}$$

where $|C_i|$ is the cluster size.

Every Message. Since cluster sizes are bounded above by tree width, we can say that passing of every message will have an upper-bound runtime of

$$\mathcal{O}(\exp(w)) \tag{6.3}$$

where w is the width of the elimination tree.

[5.1] In VE, we also said that the tree width of a DAG is the width of its best elimination order. These two quantities are equal indeed.

Amount of Messages. The total amount of messages is

$$O(2(m-1)) \qquad \text{where } m = |V| \tag{6.4}$$

since we have a tree structure, meaning we have exactly $(m-1)$ edges and each edge can have forward backward each once.

All Messages - All Cluster Marginals. The total time to pass all messages and compute all cluster marginals is

$$O(m \exp(w)) \qquad \text{or} \qquad O(n \exp(w)), \text{ for } O(n) \text{ edges.} \tag{6.5}$$

6.1.3 Polytree / Belief Propagation

Runtime. Define k as the max number of parents in the poly tree, then k is the same as the width of elimination tree. Let, also, n denote the number of nodes in the polytree. The algorithm has runtime

$$O(n \exp(k)) \tag{6.6}$$

6.1.4 Cut Set Conditioning

Time and Space (Total) Complexity.

$$O(n \exp(k)) \qquad \text{where } n = |N| \text{ and } k \text{ is width} \tag{6.7}$$

6.1.5 Any Space Recursive Cut Set

	no cache	all cache	Δ *no* → *all*
space	$O(wn)$	$O(n \exp(w))$	↑
time	$O(n \exp(w \log n)))$	$O(n \exp(w))$	↓

7 Compiling Bayesian Networks

7.1 Network Polynomials

The network polynomial is a summation over all instantiations of a network,

$$f \triangleq \sum_{z} \prod_{\theta_{x|u} \sim z} \theta_{x|u} \prod_{\lambda_x \sim z} \lambda_x \tag{7.1}$$

7.2 AC Properties

AC Size. of an AC is defined as the number of edges in the circuit.

AC Complexity. is the size of smallest AC that represents the network polynomial.

Decomposable. At each $\star$ node, we need

$$vars(AC_A) \cap vars(AC_B) = \emptyset \tag{7.2}$$

Deterministic. At each + node, we require at most one positive input is non-zero for all *complete instantiation*.

Smooth. At each + node, we require

$$vars(AC_A) = vars(AC_B) \tag{7.3}$$

AC for Marginals. requires decomposable and smooth. This guarantees that sub-circuits are of complete variable instantiations.

AC for Marginals and MPE. requires all three above: decomposable, deterministic, and smooth. The additional determinism guarantees a 1-to-1 mapping between sub-circuits and complete variable instantiations.

7.3 AC Derivative Probabilistic Implications

$$\frac{\partial f}{\partial \lambda_{\mathbf{x}}}(\mathbf{e}) = \Pr(\mathbf{x}, \mathbf{e} - \mathbf{X}) \tag{7.4}$$

where the notation $\mathbf{e} - \mathbf{X}$ means the instantiation that results form erasing the values of variables $\mathbf{X}$ from instantiation $\mathbf{e}$[7.1], and

$$\theta_{\mathbf{x}|\mathbf{u}} \frac{\partial f}{\partial \theta_{\mathbf{x}|\mathbf{u}}}(\mathbf{e}) = \Pr(\mathbf{x}, \mathbf{u}, \mathbf{e}) \tag{7.5}$$

7.4 Compilation via Variable Elimination

Circuit Factors. "In a circuit factor, each variable instantiation is mapped to a circuit node instead of a number."

Operations. We use $+(n_1, n_2)$ to denote an addition node that has n_1 and n_2 as its children. Similarly, $\star(n_1, n_2)$ denotes a multiplication node. An operation (multiplication or addition) of two circuit factors $f(X)$ and $f(Y)$ is a factor over variables $Z = X \cup Y$,

$$f(z) = [\star \text{ or } +](f(x), f(y)), \quad \text{where } x \sim z \quad \text{and} \quad y \sim z \tag{7.6}$$

[7.1] For example, $\mathbf{e} = ab\bar{c}$, then $\mathbf{e} - A = b\bar{c}$ and $\mathbf{e} - AC = b$

Procedure.

1. **Make nodes for each CPT.** For each family $X|U$, construct nodes $\star(\lambda_x, \theta_{x|u})$ for each instantiation xu of XU.
2. **Eliminate Everything.** We apply VE to eliminate all variables in the network to reach trivial instantiation $\top$ (corresponds to root). During the process, we construct the tree using the operations defined above: $*/+$.

8 Maximum Likelihood Learning

8.1 Empirical Distribution

A dataset D for variables $\mathbf{X}$ is a vector $\mathbf{d}_1, ..., \mathbf{d}_N$ where each $\mathbf{d}_i$ is called a case and represents a partial instantiation of variables $\mathbf{X}$. The data set is called complete if each case is a complete instantiation of variables $\mathbf{X}$; otherwise the dataset is called incomplete. The empirical distribution in the case of complete dataset is defined as

$$\Pr_D(\alpha) \triangleq \frac{D\#(\alpha)}{N} \tag{8.1}$$

where $D\#(\alpha)$ is the number of cases $\mathbf{d}_i$ in the data set D that satisfy event α, i.e. $\mathbf{d}_i \models \alpha$.

8.2 Complete Data MLE

When the dataset is complete, the definition of empirical distribution above suggests the following **maximum likelihood estimates**

$$\theta_x^{ml} \triangleq \Pr_D(x) = \frac{D\#(x)}{N} \tag{8.2}$$

$$\theta_{x|\mathbf{u}}^{ml} \triangleq \Pr_D(x|\mathbf{u}) = \frac{D\#(x, \mathbf{u})}{D\#(\mathbf{u})} \tag{8.3}$$

This formulation maximizes the likelihood, i.e.,

$$\theta^{ml} = \arg\max_\theta L(\theta|D) = \arg\max_\theta \prod_{i=1}^{N} \Pr_\theta(\mathbf{d}_i) \tag{8.4}$$

In addition, maximum likelihood estimates minimizes the KL divergence between the learnt bayesian network and the empirical distribution, i.e.,

$$\arg\max_\theta L(\theta|D) = \arg\min_\theta D_{KL}(\Pr_D(\mathbf{X}), \Pr_\theta(\mathbf{X})) \tag{8.5}$$

Distribution of Estimates. The distribution of ML estimates $\theta^{ml}_{x|\mathbf{u}}$ is asymptotically normal, converges in distribution to

$$\text{Gaussian}\left(\mu = \Pr(x|\mathbf{u}), \sigma = \frac{\Pr(x|\mathbf{u})(1 - \Pr(x|\mathbf{u}))}{N \cdot \Pr(\mathbf{u})}\right) \tag{8.6}$$

Notice that as sample size N increases, the variance of this distribution decreases.

8.3 Incomplete Data EM

EM algorithm first completes the dataset, which induces an empirical distribution, and then uses it to estimate parameters the same way we did in Section 8.2. The new set of parameters are guaranteed to have no less likelihood than the initial parameters, so this process is repeated until convergence criteria is met.

Expected Empirical Distribution 1. The expected empirical distribution of dataset D under parameters θ^k is defined as

$$\Pr_{D,\theta^k}(\alpha) \triangleq \sum_{\mathbf{d}_i,\mathbf{c}_i \models \alpha} \Pr_{\theta^k}(\mathbf{c}_i, \mathbf{d}_i) \tag{8.7}$$

where α is an event and $\mathbf{C}_i$ are the variables with missing values in case $\mathbf{d}_i$.

EM Estimates 1. The EM estimates for dataset D and parameters θ^k are defined as

$$\theta^{k+1}_{x|\mathbf{u}} \triangleq \Pr_{D,\theta^k}(x|\mathbf{u}) \tag{8.8}$$

Expected Empirical Distribution 2. The expected empirical distribution of data set D given parameters θ^k can be computed as

$$\Pr_{D,\theta^k}(\alpha) = \frac{1}{N}\sum_{i=1}^{N} \Pr_{\theta^k}(\alpha|\mathbf{d}_i) \tag{8.9}$$

EM Estimates 2. Now the EM estimates for dataset D and parameters θ^k can be computed as[8.1]

$$\theta^{k+1}_{x} = Pr_{D,\theta^k}(x) = (1/N)\sum_{i=1}^{N} Pr_{\theta^k}(x|\mathbf{d}_i) \tag{8.10}$$

and

$$\theta^{k+1}_{x|\mathbf{u}} = \frac{\sum_{i=1}^{N} \Pr_{\theta^k}(x\mathbf{u}|\mathbf{d}_i)}{\sum_{i=1}^{N} \Pr_{\theta^k}(\mathbf{u}|\mathbf{d}_i)} \tag{8.11}$$

[8.1] This is to say that the EM estimates can be computed without constructing the expected empirical distribution.

Difference. The key difference is that in **EME2** does not reference the expected empirical distribution (there is no D in the subscript of Pr; while **EME1** does reference. Instead, **EME2** computes EM estimates by performing inference on a bayesian network parameterized by the previous parameter estimates θ^k.

8.4 Graph Structure Learning

Conditional Entropy.

$$\mathrm{ENT}_D(X|\mathbf{U}) = -\sum_{x\mathbf{u}} \mathrm{Pr}_D(x\mathbf{u}) \log_2 \mathrm{Pr}_D(x|\mathbf{u}) \tag{8.12}$$

Log-Likelihood.

$$\mathrm{LL}(G|D) = -N \sum_{X\mathbf{U}} \mathrm{ENT}_D(X|\mathbf{U}) \tag{8.13}$$

where $X\mathbf{U}$ are families in the structure G and D is a complete dataset of size N.

Mutual Information.

$$\mathrm{MI}_D(X,U) = \sum_{x,u} \mathrm{Pr}_D(x,u) \log \frac{\mathrm{Pr}_D(x,u)}{\mathrm{Pr}_D(x)\mathrm{Pr}_D(u)} \tag{8.14}$$

Tree-Score Measure

$$tScore(G|D) = \sum_{U \to X} \mathrm{MI}_D(X,U) \tag{8.15}$$

Scores - General The dimension of a DAG G is defined as

$$||G|| = \sum_{i=1}^{n} ||X_i\mathbf{U}_i|| \tag{8.16}$$

where family size

$$||X_i\mathbf{U}_i|| = (X_i^{\#} - 1)\mathbf{U}_i^{\#} \tag{8.17}$$

In general, scores could be written as

$$Score(G|D) = \mathrm{LL}(G|D) - \psi(N)||G|| \tag{8.18}$$

Akaike Information Criterion (AIC). is when we choose $\psi(N)$ to be a constant independent of N.

Minimum Description Length (MDL). is when we take

$$\psi(N) = \frac{\log_2 N}{2} \tag{8.19}$$

9 Bayesian Learning

9.1 Meta Network

The meta network is a tiled version of a base bayesian network, with extra (root) variables pointing into base network nodes. Meta network of size N means it has N copies of the original network within. It satisfies the following **parameter independence** relationships: let Σ_1, Σ_2 each contain a collection of parameter sets, and $\Sigma_1 \cap \Sigma_2 = \emptyset$. Then,

- Σ_1 and Σ_2 are independent, $\mathbb{P}(\Sigma_1, \Sigma_2) = \mathbb{P}(\Sigma_1)\mathbb{P}(\Sigma_2)$, and
- Σ_1 and Σ_2 are independent given any *complete* dataset D,

$$\mathbb{P}(\Sigma_1, \Sigma_2 | D) = \mathbb{P}(\Sigma_1 | D)\mathbb{P}(\Sigma_2 | D) \tag{9.1}$$

9.2 Parameter Learning - Discrete Param Sets, Complete Data

Theorem 18.2. Given discrete parameter sets and a dataset D of size N, we have

$$\mathbb{P}(\alpha_{N+1} | D) = \sum_{\theta} \Pr_\theta(\alpha)\mathbb{P}(\theta | D) \tag{9.2}$$

where α_{N+1} is obtained from α by replacing every occurrence of variable X by its instance X_{N+1}. Notice that here we are saying $\mathbb{P}(\alpha_{N+1}|D)$ is an expectation over $\Pr_\theta(\alpha)$.

Bayesian Parameter Estimates. Let $\theta_{X|\mathbf{u}}$ b a discrete parameter set. The Bayesian estinate for parameter $\theta_{x|\mathbf{u}}$ given data set D is defined as

$$\theta^{be}_{x|\mathbf{u}} \triangleq \sum_{\theta_{X|\mathbf{u}}} \theta_{x|\mathbf{u}} \cdot \mathbb{P}(\theta_{X|\mathbf{u}} | D) \tag{9.3}$$

where (Theorem 18.4)

$$\mathbb{P}(\theta_{X|\mathbf{u}} | D) = \eta \mathbb{P}(\theta_{X|\mathbf{u}}) \prod_x \left[\theta_{x|\mathbf{u}}\right]^{D\#(x\mathbf{u})} \tag{9.4}$$

when η is a normalizing constant.

Theorem 18.3.

$$\mathbb{P}(\alpha_{N+1}|D) = \Pr_{\theta^{be}}(\alpha) \tag{9.5}$$

This is saying that $\mathbb{P}(\alpha_{N+1}|D)$ (Theorem 18.2) can be computed as an inference over the network.

9.3 Parameter Learning - Continuous Param Sets, Complete Data

9.3.1 Dirichlet Prior

Dirichlet Prior Exponents. If our network parameters are binary, then the Dirichlet prior requires to exponents,

$$\mathbb{E}[\theta_h] = \frac{\psi_h}{\psi_h + \psi_{\bar{h}}} \qquad \mathbb{E}[\theta_{\bar{h}}] = \frac{\psi_{\bar{h}}}{\psi_h + \psi_{\bar{h}}} \tag{9.6}$$

This notion generalizes similarly for variables that can take on more values.

Dirichlet Distribution. Formally, a Dirichlet prior distribution is specified by a set of exponents $\psi_{x|\mathbf{u}} \geq 1$. The constraint of the value of exponents guarantees a uni-modal Dirichlet. The **equivalent sample size** of the distribution is defined as

$$\psi_{X|\mathbf{u}} \triangleq \sum_x \phi_{x|\mathbf{u}} \tag{9.7}$$

The distribution takes the following density

$$\rho(\theta_{X|\mathbf{u}}) \triangleq \frac{\Gamma(\psi_{X|\mathbf{u}})}{\prod_x \Gamma(\psi_{x|\mathbf{u}})} \cdot \prod_x \left[\theta_{x|\mathbf{u}}\right]^{\psi_{x|\mathbf{u}}-1} \tag{9.8}$$

Dirichlet Statistics. If a network parameter has a prior specified in Dirichlet, then it has expectation

$$\mathbb{E}(\theta_{x|\mathbf{u}}) = \frac{\psi_{x|\mathbf{u}}}{\psi_{X|\mathbf{u}}} \tag{9.9}$$

and variance

$$\mathbb{V}(\theta_{x|\mathbf{u}}) = \frac{\mathbb{E}(\theta_{x|\mathbf{u}})(1 - \mathbb{E}(\theta_{x|\mathbf{u}}))}{\psi_{X|\mathbf{u}} + 1} \tag{9.10}$$

where we notice as the equivalent sample size ($\psi_{X|\mathbf{u}}$) increase, the variance decreases. The mode is

$$\mathbb{M}(\theta_{x|\mathbf{u}}) = \frac{\psi_{x|\mathbf{u}} - 1}{\psi_{X|\mathbf{u}} - |X|} \tag{9.11}$$

where $|X|$ is the number of values for variable X.

9.3.2 The Learning - Theory

Theorem 18.7.

$$\mathbb{P}(\alpha_{N+1}|D) = \int \Pr_\theta(\alpha)\rho(\theta|D)d\theta \tag{9.12}$$

Bayesian Estimates. Let $\theta_{X|\mathbf{u}}$ be a continuous parameter set. The bayesian estimate for network parameter $\theta_{x|\mathbf{u}}$ given data set D is defined as

$$\theta^{be}_{x|\mathbf{u}} \triangleq \int \theta_{x|\mathbf{u}} \cdot \rho(\theta_{X|\mathbf{u}}|D)d\theta_{X|\mathbf{u}} \tag{9.13}$$

Theorem 18.8.

$$\mathbb{P}(\alpha_{N+1}|D) = \Pr_{\theta^{be}}(\alpha) \tag{9.14}$$

9.3.3 The Learning - Practice

Theorem 18.9. Consider a meta network where each parameter set $\theta_{X|\mathbf{u}}$ has a prior Dirichlet density $\rho(\theta_{X|\mathbf{u}})$ specified by exponents $\psi_{x|\mathbf{u}}$. Let D be a complete dataset. Then the posterior density $\rho(\theta_{X|\mathbf{u}}|D)$ is also Dirichlet, with exponents

$$\psi'_{x|\mathbf{u}} = \psi_{x|\mathbf{u}} + D\#(x\mathbf{u}) \tag{9.15}$$

Posterior Bayesian Estimates. The posterior expectation of parameter $\theta_{x|\mathbf{u}}$ given complete data is given by

$$\theta^{be}_{x|\mathbf{u}} = \frac{\psi_{x|\mathbf{u}} + D\#(x\mathbf{u})}{\psi_{X|\mathbf{u}} + D\#(\mathbf{u})} \tag{9.16}$$

where $\psi_{x|\mathbf{u}}$ are the exponents in the prior Dirichlet distribution, and $psi_{X|\mathbf{u}}$ is its equivalent sample size.

Maximum-A-Posteriori Estimates. The MAP estimate given complete data is

$$\theta^{ma}_{x|\mathbf{u}} = \frac{\psi_{x|\mathbf{u}} + D\#(x\mathbf{u}) - 1}{\psi_{X|\mathbf{u}} + D\#(\mathbf{u}) - |X|} \tag{9.17}$$

where $|X|$ is the number of values for variable X.

9.3.4 Non-informative Prior

Non-informative prior refers to the case (in Dirichlet) where all exponents are equal to one: $\psi_{x|\mathbf{u}} = 1$. Thus the expectation is $1/|X|$ for all classes. Under this prior,

$$\theta^{be}_{x|\mathbf{u}} = \frac{1 + D\#(x\mathbf{u})}{|X| + D\#(\mathbf{u})} \tag{9.18}$$

and

$$\theta^{ma}_{x|\mathbf{u}} = \frac{D\#(x\mathbf{u})}{D\#(\mathbf{u})} \tag{9.19}$$

which coincides with the MLE. However, *this only works when the prior exponents are all equal to one.*

10 Causality - Interventions

10.1 Notations

Causal Effect (CE). of $X = x$ on $Y = y$ can be written as

$$\Pr(Y = y|do(X = x)) \equiv \Pr(y|do(x)) \equiv \Pr(y_x) \tag{10.1}$$

Interventional Distribution. For $\Pr(X, Y, Z)$, the interventional distribution for $do(X = x)$ is denoted as

$$\Pr_{X=x}(Y, Z) \tag{10.2}$$

10.2 Types of Causal Graphs

[10.1]

10.2.1 Markovian Model

Each hidden variable in a Markovian Model has at most one child. It has an alternative name of "no hidden confounders". In this case, causal effects are always identifiable.

10.2.2 Semi-Markovian Model

Some hidden variable has more than one child. In this case, causal effects are not always identifiable.

10.3 Identifiability Criterion

10.3.1 Causal Effect Rule

The Causal Effect Rule links together association and intervention. It states the following: if $\mathbf{Z}$ are the parents of X, then

$$\Pr(y|do(x)) = \Pr(y_x) = \sum_{\mathbf{z}} \Pr(y|x, \mathbf{z})\Pr(\mathbf{z}) \tag{10.3}$$

[10.1] Hidden variables are roots.

The catch to this formulation is one have to know the parents - meaning that we need to have a correct causal structure prior to using this formula. This is a strong assumption. Often, the structure is exactly what we are after.[10.2]

10.3.2 Backdoor Criteria

A path between X and Y is *blocked* by Z iff

- some collider is not in Z, or
- some non-collider is in Z.

where a collider node is simply a convergent valve defined earlier ($\rightarrow W \leftarrow$). Here we distinguish only between colliders and non-colliders. The Backdoor Criteria states the following: Consider a causal graph G and causal effect $\Pr(y_x)$. A set of variables $\mathbf{Z}$ satisfis the backdoor criteria iff

- no node in $\mathbf{Z}$ is a descendant of X,
- $\mathbf{Z}$ *blocks* every path between X and Y that contains an arrow into X.

Then, if $\mathbf{Z}$ is a backdoor, then

$$\Pr(y_x) = \sum_{\mathbf{z}} \Pr(y|x, \mathbf{z}) \Pr(\mathbf{z}) \tag{10.4}$$

Incompleteness. The backdoor criteria is incomplete. When it identifies that a causal effect has no backdoor, the causal effect can be either identifiable or not identifiable (inconclusive).

10.3.3 Frontdoor Criteria

Consider a causal graph G and causal effect $\Pr(y_x)$. A set of variables $\mathbf{Z}$ satisfies the frontdoor criteria iff .... Then if $\mathbf{Z}$ is a frontdoor, then,

$$\Pr(y_x) = \sum_{\mathbf{z}} \Pr(\mathbf{z}|x) \sum_{x'} \Pr(y|x', \mathbf{z}) \Pr(x') \tag{10.5}$$

10.3.4 Exogenous X

For the query $\Pr(y_x)$, when X is an exogenous variable, meaning that it has no parents,

$$\Pr(y_x) = \Pr(y|x) \tag{10.6}$$

[10.2] Recall that different causal structures can generate the same distribution, and data alone is not enough.

10.4 The Do-Calculus

The key idea is to apply a series of rules until we get a formula that is comprised of solely associational quantities. There are three re-write rules in total. $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W}$ are disjoint sets of variables,

Rule 1. Ignoring Observations.

$$\Pr(\mathbf{y}|do(\mathbf{x}), \mathbf{z}, \mathbf{w}) = \Pr(\mathbf{y}|do(\mathbf{x}), \mathbf{w}) \quad \text{if} \quad \text{dsep}_{G_{\bar{\mathbf{X}}}}(\mathbf{Y}, \mathbf{XW}, \mathbf{Z}) \tag{10.7}$$

where we perform dsep test on an altered graph $G_{\bar{\mathbf{X}}}$, rather than the original causal graph G. (Detailed in Section 10.4.1).

Rule 2. Action / Observation Exchange.

$$\Pr(\mathbf{y}|do(\mathbf{x}), do(\mathbf{z}), \mathbf{w}) = \Pr(\mathbf{y}|do(\mathbf{x}), \mathbf{z}, \mathbf{w}) \quad \text{if} \quad \text{dsep}_{G_{\bar{\mathbf{X}}\underline{\mathbf{Z}}}}(\mathbf{Y}, \mathbf{XW}, \mathbf{Z}) \tag{10.8}$$

There are several weakened versions of this rule. First, we consider the case of $\mathbf{X} = \emptyset$, then

$$\Pr(\mathbf{y}|do(\mathbf{z}), \mathbf{w}) = \Pr(\mathbf{y}|\mathbf{z}, \mathbf{w}) \quad \text{if} \quad \text{dsep}_{G_{\underline{\mathbf{Z}}}}(\mathbf{Y}, \mathbf{W}, \mathbf{Z}) \tag{10.9}$$

Simplifying even further, we can remove the common "kept" part $\mathbf{W}$, and get

$$\Pr(\mathbf{y}|do(\mathbf{z})) = \Pr(\mathbf{y}|\mathbf{z}) \quad \text{if} \quad \text{dsep}_{G_{\underline{\mathbf{Z}}}}(\mathbf{Y}, \emptyset, \mathbf{Z}) \tag{10.10}$$

notice that this Exogenous X rule (also stated in Section 10.3.4) follows directly as a corollary of Rule 2.

Rule 3. Ignoring Actions.

$$\Pr(\mathbf{y}|do(\mathbf{x}), do(\mathbf{z}), \mathbf{w}) = \Pr(\mathbf{y}|do(\mathbf{x}), \mathbf{w}) \quad \text{if} \quad \text{dsep}_{G_{\bar{\mathbf{X}}\overline{Z(W)}}}(\mathbf{Y}, \mathbf{XW}, \mathbf{Z}) \tag{10.11}$$

where encounter a special notation $\overline{Z(W)}$ that means "not all variables in $\mathbf{Z}$, but only those variables in $\mathbf{Z}$ that do not have ancestor in $\mathbf{W}$".

10.4.1 Graph Alterations

The calculus rules we specified earlier performs dsep tests on altered graphs, where

- $G_{\bar{\mathbf{X}}}$ is obtained via removing edges pointing into variables $\mathbf{X}$ from G.
- $G_{\underline{\mathbf{X}}}$ is obtained via removing edges pointing away from variables $\mathbf{X}$ from G.

11 Counterfactual Reasoning

Causal effects are on the second level: interventions. In counterfactual reasoning, we step up onto level three: what-if's.

11.1 The Information Hierarchy

To add to what we have said above,

- Associational reasoning: Bayesian Network
- Interventional reasoning: Causal Bayesian network (causal graph)
- Counterfactual reasoning: Functional Bayesian network (functional dependencies)

11.2 Counterfactual Queries

11.2.1 Probability of Necessity (PN)

Probability that y would not have occurred in the absence of x ($do(\bar{x})$), given that x and y did in fact occur, i.e,

$$PN = \Pr(\bar{y}_{\bar{x}}|x, y) \tag{11.1}$$

11.2.2 Probability of Sufficiency (PS)

Probability that setting x would produce y in a situation where both x and y are absent, i.e.,

$$PS = \Pr(y_x|\bar{y}, \bar{x}) \tag{11.2}$$

11.2.3 Probability of Necessity and Sufficiency (PNS)

Probability that y responds to x both ways (measures the necessity and sufficiency for x to produce y):

$$PNS = \Pr(y_x, \bar{y}_{\bar{x}}) = \Pr(x, y)PN + \Pr(\bar{x}, \bar{y})PS \tag{11.3}$$

11.2.4 Probability of Disablement

Probability that y would have been prevented if it were not for x,

$$PD = \Pr(\bar{y}_{\bar{x}}|y) \tag{11.4}$$

11.2.5 Probability of Enablement

Probability that y would have been realized if it were not for absence of x,

$$PE = \Pr(y_x|\bar{y}) \tag{11.5}$$

11.3 Structural Causal Models (SCM)

World. In SCM, a world is defined slightly different from what we had in associational graphs. A world is an instantiation of exogenous variables. This is because fixing exogenous variables fully specifies the entire network.

11.4 Evenets

11.4.1 Observational Event

- Form: $\mathbf{x}$ where $\mathbf{X}$ is a set of endogenous variables
- Meaning: variables $\mathbf{X}$ took the value $\mathbf{x}$
- Examples: $x, \quad \bar{y}, z$

11.4.2 Interventional Event

- Form: $\mathbf{y}_{\mathbf{x}}$ where $\mathbf{X}$ and $\mathbf{Y}$ are sets of endogenous variables
- Meaning: variables $\mathbf{Y}$ took the value $\mathbf{y}$ after setting $\mathbf{X} = \mathbf{x}$
- Examples: $y_x, \quad y_{\bar{x}z}, \quad (x\bar{y})_{zw}$

11.4.3 Counterfactual Event

- Form: $\eta_1, ..., \eta_n$ where η_i is an observational or interventional event
- Meaning: $\bigwedge_i \eta_i$
- Examples: mix of above two event examples

11.5 Satisfaction

Recalling what we had earlier, we denote $\mathbf{u}$ satisfies event η in SCM $\mathcal{M}$ as

$$\mathbf{u} \models_{\mathcal{M}} \eta \tag{11.6}$$

11.5.1 Observational Event

We write

$$\mathbf{u} \models_{\mathcal{M}} \mathbf{x} \tag{11.7}$$

iff world $\mathbf{u}$ fixes (endogenous) variables $\mathbf{X}$ to $\mathbf{x}$ in SCM $\mathcal{M}$.

11.5.2 Interventional Event

$$\mathbf{u} \models_{\mathcal{M}} \mathbf{y_x} \iff \mathbf{u} \models_{\mathcal{M}_\mathbf{x}} \mathbf{y} \tag{11.8}$$

11.5.3 Counterfactual Event

$$\mathbf{u} \models_{\mathcal{M}} \eta_1, ..., \eta_n \iff \bigwedge_i \mathbf{u} \models_{\mathcal{M}} \eta_i \tag{11.9}$$

11.6 Queries

The computation of observational events η in SCM is easy. First we compute the table with world probabilities. The table should have a row for each instantiation of the exogenous variables $\mathbf{u} = [U_1, ..., U_n]$. Endogenous variables, e.g. X, Y, Z will be computed using the factors, for example $f_X(...) = ...$ Then, to compute probability of observational event $\Pr(\eta)$, we simply select rows that satisfies η and add up their respective $\Pr(\mathbf{u})$ values. To compute an interventional event η, we have to first transform the table. This is a two step process,

1. Force the endogenous variable(s) inside $do(\cdot)$ to take value of 1. For example if we want y_x, then we have $do(x)$ and thus we set all the values of X in the table to x.[11.1]

2. Now we need to fix the table by adjusting values y. We do so by simplifying the relationship, for example

$$f_Y(X, U_y, U_x) = xu_r + \bar{x}u_y u_r + \bar{x}\bar{u}_y\bar{u}_r \tag{11.10}$$

simplifies to

$$f_Y(X = x, U_x, U_r) = u_r \tag{11.11}$$

Then, the values of y should be computed using this simplified formula (on the altered table). In the new table, this interventional probability can be calculated as if it is an associational query, i.e. we pick rows that satisfies and add up probabilities.

11.7 Exogeneity and Monotonicity

Exogeneity. X is exogenous relative to Y iff events y_x and x are independent and so are $y_{\bar{x}}$ and x,

$$\Pr(y_x|x) = \Pr(y_x) \quad \text{and} \quad \Pr(y_{\bar{x}}|x) = \Pr(y_{\bar{x}}) \tag{11.12}$$

Monotonicity. Y is monotonic relative to X iff the event $\bar{y}_x, y_{\bar{x}}$ is unsatisfiable. This notion is general, x and y does not have to be inside the same family. A more general notation is to condition on probability of event $\bar{y}_x, y_{\bar{x}}$ being zero.

[11.1] In binary case, this means 1 if we have $do(x)$ and 0 if we have $do(\bar{x})$

12 Sensitivity Analysis

12.1 Global to Local Belief Change

For each network parameter $\theta_{x|\mathbf{u}}$, what is the minimal amount of change that can enforce query constraints such as

$$\Pr(y|e) \geq \varepsilon \quad \Pr(y|e) - \Pr(z|e) \geq \varepsilon \quad \Pr(y|e)/\Pr(z|e) \geq \varepsilon \tag{12.1}$$

To do so, we need to first calculate partial derivatives where we define τ's such that

$$\theta_{x|\mathbf{u}} = \tau_{x|\mathbf{u}} \qquad \theta_{\bar{x}|\mathbf{u}} = 1 - \tau_{x|\mathbf{u}} \tag{12.2}$$

Then,

$$\alpha_e = \frac{\partial \Pr(e)}{\partial \tau_{x|u}} = \frac{\Pr(e,x,u)}{\theta_{x|u}} - \frac{\Pr(e,\bar{x},u)}{\theta_{\bar{x}|u}} \tag{12.3}$$

This results in the following formulation. To ensure constraint $\Pr'(y|e) \geq \varepsilon$, we need to change the meta parameter $\tau_{x|u}$ by δ such that

$$\Pr(y,e) - \varepsilon \Pr(e) \geq \delta[-\alpha_{y,e} + \varepsilon \alpha_e] \tag{12.4}$$

where δ is the only unknown and shall result in $\delta \geq q$ or $\delta \leq q$.

Complexity. You get this for free. Running inference yields us derivatives and from there we can solve q. Hence time complexity is $\mathcal{O}(n2^w)$. n is the number of variables and w is the tree width.

12.2 Local to Global Belief Change

Given a change on a parameter, what bounds can we provide on the changes on queries?

Bounding the Partial Derivative The bound is in terms of the current value of query $\Pr(\mathbf{y}|\mathbf{e})$ and the parameter value $\Pr(x|\mathbf{u})$

$$\left|\frac{\partial \pi(\mathbf{y}|\mathbf{e})}{\partial \tau_{x|\mathbf{u}}}\right| \leq \frac{\Pr(\mathbf{y}|\mathbf{e})(1 - \Pr(\mathbf{y}|\mathbf{e}))}{\Pr(x|\mathbf{u})(1 - \Pr(x|\mathbf{u}))} \tag{12.5}$$

Extreme queries tend to be robust when changing non-extreme parameters, yet non-extreme queries may change considerably when changing extreme parameters. [12.1]

[12.1] A query that is close to 0 or 1 tends to stick, and a query with value around 1/2 tend to oscillate more. The other way around for parameters. Tinkering parameter that is around 1/2 does not incur big changes, but changing extreme parameters will result in rapid changes.

Arbitrary Parameter Changes First, we recall definition of Odds,

$$O(x|\mathbf{u}) = \Pr(x|\mathbf{u})/(1 - \Pr(x|\mathbf{u})) \tag{12.6}$$

and

$$O(\mathbf{y}|\mathbf{e}) = \Pr(\mathbf{y}|\mathbf{e})/(1 - \Pr(\mathbf{y}|\mathbf{e})) \tag{12.7}$$

and $O'(x|\mathbf{u}), O'(\mathbf{y}|\mathbf{e})$ denotes the odds after having applied an arbitrary change to the meta parameter $\tau_{x|\mathbf{u}}$. The bounds can be specified as

$$|\ln(O'(\mathbf{y}|\mathbf{e})) - \ln(O(\mathbf{y}|\mathbf{e}))| \leq |\ln(O'(x|\mathbf{u})) - \ln(O(x|\mathbf{u}))| \tag{12.8}$$

12.3 Chan-Darwiche Measure

$$D_{CD}(\Pr, \Pr') \triangleq \ln \max_w \frac{\Pr'(w)}{\Pr(w)} - \ln \min_w \frac{\Pr'(w)}{\Pr(w)} \tag{12.9}$$

which is a true distance metric.[12.2] Then, we can provide the bound as follows. Let $\Pr, \Pr'$ be two distributions, α, β be any two events,

$$e^{-D(\Pr,\Pr')} \leq \frac{O'(\alpha|\beta)}{O(\alpha|\beta)} \leq e^{D(\Pr,\Pr')} \tag{12.10}$$

As an alternative formulation, given $\Pr, \Pr'$ and define $p = \Pr(\alpha|\beta), d = D_{CD}(\Pr, \Pr')$, the bounds are

$$\frac{pe^{-d}}{pe^{-d} - p + 1} \leq \Pr'(\alpha|\beta) \leq \frac{pe^{d}}{pe^{d} - p + 1} \tag{12.11}$$

Distance Between Networks. Consider the following case

- Bayesian network N' is obtained from N by changing the CPT of X from $\Theta_{X|u}$ to $\Theta'_{X|u}$.
- N and N' induce distributions $\Pr$ and $\Pr'$.

Then, the D_{CD} between the two networks is the same as the distance between the two changed CPTs, i.e.

$$D_{CD}(\Pr, \Pr') = D_{CD}(\Theta_{X|u}, \Theta'_{X|u}) \tag{12.12}$$

This works because the changed CPT is indeed itself a distribution.

♡

[12.2] Original paper: https://www.aaai.org/Papers/AAAI/2002/AAAI02-081.pdf