

Bayesuvius,
a small visual dictionary of Bayesian Networks

Robert R. Tucci
www.ar-tiste.xyz

August 5, 2020



Figure 1: View of Mount Vesuvius from Pompeii



Figure 2: Mount Vesuvius and Bay of Naples

Contents

0.1	Foreword	4
0.2	Notational Conventions	5
1	Back Propagation (Auto Differentiation): COMING SOON	8
2	Basic Curve Fitting Using Gradient Descent	9
3	Bell and Clauser-Horne Inequalities in Quantum Mechanics	11
4	Binary Decision Diagrams	12
5	Decision Trees	15
6	Do-Calculus: COMING SOON	18
7	D-Separation: COMING SOON	19
8	Expectation Maximization: COMING SOON	20
9	Generative Adversarial Networks (GANs)	21
10	Graph Structure Learning for bnets: COMING SOON	26
11	Hidden Markov Model	27
12	Influence Diagrams & Utility Nodes	31
13	Kalman Filter	33
14	Linear and Logistic Regression	36
15	Markov Blankets	40
16	Markov Chain Monte Carlo (MCMC): COMING SOON	42
17	Message Passing (Belief Propagation)	43

18 Monty Hall Problem	44
19 Naive Bayes	46
20 Neural Networks	47
21 Non-negative Matrix Factorization	54
22 Recurrent Neural Networks	56
23 Reinforcement Learning (RL)	65
24 Restricted Boltzmann Machines	74
25 Simpson's Paradox	76
26 Turbo Codes	77
Bibliography	83

0.2 Notational Conventions

bnet=Bayesian Network

Define $\mathbb{Z}, \mathbb{R}, \mathbb{C}$ to be the integers, real numbers and complex numbers, respectively.

For $a < b$, define \mathbb{Z}_I to be the integers in the interval I , where $I = [a, b], [a, b), (a, b], (a, b)$ (i.e., I can be closed or open on either side).

$A_{>0} = \{k \in A : k > 0\}$ for $A = \mathbb{Z}, \mathbb{R}$.

Random Variables will be indicated by underlined letters and their values by non-underlined letters. Each node of a bnet will be labelled by a random variable. Thus, $\underline{x} = x$ means that node \underline{x} is in state x .

$P_{\underline{x}}(x) = P(\underline{x} = x) = P(x)$ is the probability that random variable \underline{x} equals $x \in S_{\underline{x}}$. $S_{\underline{x}}$ is the set of states (i.e., values) that \underline{x} can assume and $n_{\underline{x}} = |S_{\underline{x}}|$ is the size (aka cardinality) of that set. Hence,

$$\sum_{x \in S_{\underline{x}}} P_{\underline{x}}(x) = 1 \quad (1)$$

$$P_{\underline{x}, \underline{y}}(x, y) = P(\underline{x} = x, \underline{y} = y) = P(x, y) \quad (2)$$

$$P_{\underline{x}|\underline{y}}(x|y) = P(\underline{x} = x | \underline{y} = y) = P(x|y) = \frac{P(x, y)}{P(y)} \quad (3)$$

Kronecker delta function: For x, y in discrete set S ,

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases} \quad (4)$$

Dirac delta function: For $x, y \in \mathbb{R}$,

$$\int_{-\infty}^{+\infty} dx \delta(x - y) f(x) = f(y) \quad (5)$$

Transition probability matrix of a node of a bnet can be either a discrete or a continuous probability distribution. To go from continuous to discrete, one replaces integrals over states of node by sums over new states, and Dirac delta functions by Kronecker delta functions. More precisely, consider a function $f : S \rightarrow \mathbb{R}$. Let $S_{\underline{x}} \subset S$ and $S \rightarrow S_{\underline{x}}$ upon discretization (binning). Then

$$\int_S dx P_{\underline{x}}(x) f(x) \rightarrow \frac{1}{n_{\underline{x}}} \sum_{x \in S_{\underline{x}}} f(x) . \quad (6)$$

Both sides of last equation are 1 when $f(x) = 1$. Furthermore, if $y \in S_{\underline{x}}$, then

$$\int_S dx \delta(x - y) f(x) = f(y) \rightarrow \sum_{x \in S_{\underline{x}}} \delta(x, y) f(x) = f(y) . \quad (7)$$

Indicator function (aka Truth function):

$$\mathbb{1}(\mathcal{S}) = \begin{cases} 1 & \text{if } \mathcal{S} \text{ is true} \\ 0 & \text{if } \mathcal{S} \text{ is false} \end{cases} \quad (8)$$

For example, $\delta(x, y) = \mathbb{1}(x = y)$.

$$\vec{x} = (x[0], x[1], x[2] \dots, x[nsam(\vec{x}) - 1]) = x[:] \quad (9)$$

$nsam(\vec{x})$ is the number of samples of \vec{x} . $x[i]$ are i.i.d. (independent identically distributed) samples with

$$x[i] \sim P_{\underline{x}} \text{ (i.e. } P_{x[i]} = P_{\underline{x}}) \quad (10)$$

$$P(\underline{x} = x) = \frac{1}{nsam(\vec{x})} \sum_i \mathbb{1}(x[i] = x) \quad (11)$$

If we use two sampled variables, say \vec{x} and \vec{y} , in a given bnnet, their number of samples $nsam(\vec{x})$ and $nsam(\vec{y})$ need not be equal.

$$P(\vec{x}) = \prod_i P(x[i]) \quad (12)$$

$$\sum_{\vec{x}} = \prod_i \sum_{x[i]} \quad (13)$$

$$\partial_{\vec{x}} = [\partial_{x[0]}, \partial_{x[1]}, \partial_{x[2]}, \dots, \partial_{x[nsam(\vec{x})-1]}] \quad (14)$$

$$P(\vec{x}) \approx \left[\prod_x P(x)^{P(x)} \right]^{nsam(\vec{x})} \quad (15)$$

$$= e^{nsam(\vec{x}) \sum_x P(x) \log P(x)} \quad (16)$$

$$= e^{-nsam(\vec{x}) H(P_{\underline{x}})} \quad (17)$$

$$f^{[1, \partial_x, \partial_y]}(x, y) = [f, \partial_x f, \partial_y f] \quad (18)$$

$$f^+ = f^{[1, \partial_x, \partial_y]} \quad (19)$$

For probabilty distributions $p(x), q(x)$ of $x \in S_{\underline{x}}$

- Entropy:

$$H(p) = - \sum_x p(x) \log p(x) \geq 0 \quad (20)$$

- Kullback-Liebler divergence:

$$D_{KL}(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \geq 0 \quad (21)$$

- Cross entropy:

$$CE(p \rightarrow q) = - \sum_x p(x) \log q(x) \quad (22)$$

$$= H(p) + D_{KL}(p \parallel q) \quad (23)$$

Normal Distribution: $x, \mu, \sigma \in \mathbb{R}, \sigma > 0$

$$\mathcal{N}(\mu, \sigma^2)(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (24)$$

Uniform Distribution: $a < b, x \in [a, b]$

$$\mathcal{U}(a, b)(x) = \frac{1}{b-a} \quad (25)$$

Expected Value

Given a random variable \underline{x} with states $S_{\underline{x}}$ and a function $f : S_{\underline{x}} \rightarrow \mathbb{R}$, define

$$E_{\underline{x}}[f(\underline{x})] = E_{x \sim P(x)}[f(x)] = \sum_x P(x)f(x) \quad (26)$$

Conditional Expected Value

Given a random variable \underline{x} with states $S_{\underline{x}}$, a random variable \underline{y} with states $S_{\underline{y}}$, and a function $f : S_{\underline{x}} \times S_{\underline{y}} \rightarrow \mathbb{R}$, define

$$E_{\underline{x}|\underline{y}}[f(\underline{x}, \underline{y})] = \sum_x P(x|\underline{y})f(x, \underline{y}) , \quad (27)$$

$$E_{\underline{x}|\underline{y}=\underline{y}}[f(\underline{x}, \underline{y})] = E_{\underline{x}|\underline{y}}[f(\underline{x}, \underline{y})] = \sum_x P(x|\underline{y})f(x, \underline{y}) . \quad (28)$$

Note that

$$E_{\underline{y}}[E_{\underline{x}|\underline{y}}[f(\underline{x}, \underline{y})]] = \sum_{x,y} P(x|\underline{y})P(\underline{y})f(x, \underline{y}) \quad (29)$$

$$= \sum_{x,y} P(x, \underline{y})f(x, \underline{y}) \quad (30)$$

$$= E_{\underline{x}, \underline{y}}[f(\underline{x}, \underline{y})] . \quad (31)$$

Sigmoid function: For $x \in \mathbb{R}$,

$$\text{sig}(x) = \frac{1}{1 + e^{-x}} \quad (32)$$

$\mathcal{N}(!a)$ will denote a normalization constant that does not depend on a . For example, $P(x) = \mathcal{N}(!x)e^{-x}$ where $\int_0^\infty dx P(x) = 1$.

A **one hot** vector of zeros and ones is a vector with all entries zero with the exception of a single entry which is one. A **one cold** vector has all entries equal to one with the exception of a single entry which is zero. For example, if $x^n = (x_0, x_1, \dots, x_{n-1})$ and $x_i = \delta(i, 0)$ then x^n is one hot.

Chapter 4

Binary Decision Diagrams



Figure 4.1: Binary decision tree and truth table for the function $f(x_1, x_2, x_3) = \bar{x}_1\bar{x}_2\bar{x}_3 + x_1x_2 + x_2x_3$

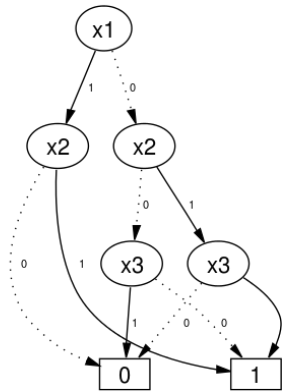


Figure 4.2: BDD for the function f of Fig.4.1.

This chapter is based on Wikipedia article Ref.[2].

Binary Decision Diagrams (BDDs) can be understood as a special case of Decision Trees (dtrees). We will assume that the reader has read Chapter 5 on dtrees before tackling this chapter.

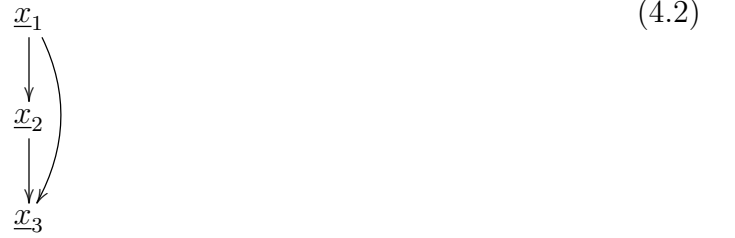
Both Figs.4.1 and 4.2 were taken from the aforementioned Wikipedia article. They give a simple example of a function $f : \{0, 1\}^3 \rightarrow \{0, 1\}$ represented in Fig.4.1 as a **binary decision tree** and in Fig.4.2 as a **binary decision diagram (BDD)**. The goal of this chapter is to represent both figures as bnets with almost the same graph structures.

We begin by noting that the function $f : \{0, 1\}^3 \rightarrow \{0, 1\}$ is a special case of a probability distribution $P : \{0, 1\}^3 \rightarrow [0, 1]$. In fact, if we restrict P to be deterministic, then $P_{det} : \{0, 1\}^3 \rightarrow \{0, 1\}$ has the same domain and range as f . Henceforth, we will refer to $f(x_1, x_2, x_3)$ as $P(x_1, x_2, x_3)$, keeping in mind that we are restricting our attention to deterministic probability distributions.

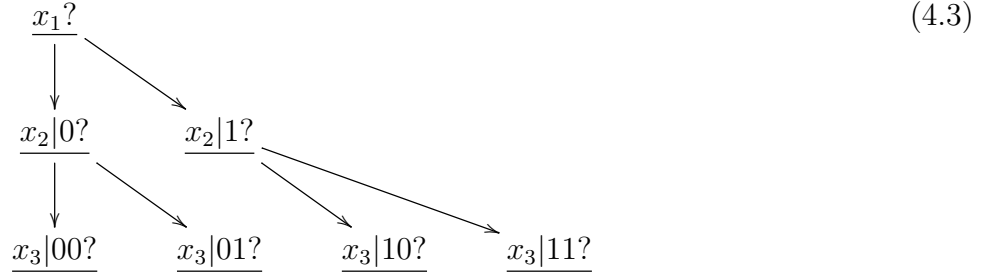
If we apply the chain rule for conditional probabilities to $P(x_1, x_2, x_3)$, we get

$$P(x_1, x_2, x_3) = P(x_3|x_1, x_2)P(x_2|x_1)P(x_1), \quad (4.1)$$

which can be represented by the bnet:



But in Chapter 5, we learned how to represent the bnet of Eq.(4.2) as the bnet tree Eq.(4.3). In that tree, the nodes pose questions with 3 possible answers 0, 1, *null*.



The node transition probability matrices, printed in blue, for the bnet of Eq.(4.3) are as follows. If $x_1, x_2, x_3 \in \{0, 1, null\}$ and $a, b \in \{0, 1\}$, then

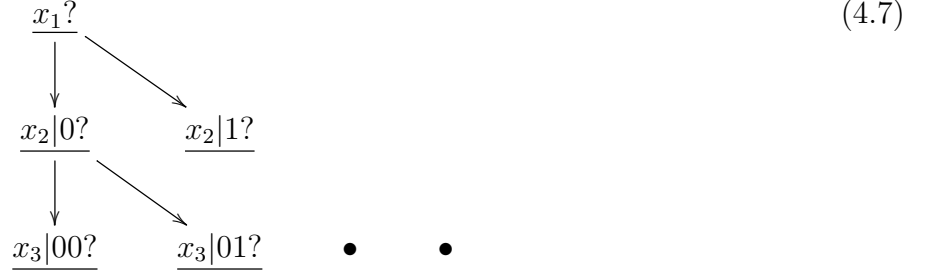
$$P(\underline{x_1?} = x_1) = \begin{cases} P_{x_1}(x_1) & \text{if } x_1 \in \{0, 1\} \\ 0 & \text{if } x_1 = null \end{cases} \quad (4.4)$$

$$P(\underline{x_2|a?} = x_2 \mid \underline{x_1?} = x_1) = \begin{cases} P_{x_2|\underline{x_1}}(x_2|a) & \text{if } x_1 = a \\ \mathbb{1}(x_2 = null) & \text{otherwise} \end{cases} \quad (4.5)$$

$$P(\underline{x_3|a, b? = x_3} \mid \underline{x_2|b? = x_2}) = \begin{cases} P_{\underline{x_3|\underline{x_1}, \underline{x_2}}}(x_3|a, b) & \text{if } (x_1, x_2) = (a, b) \\ \mathbb{1}(x_3 = null) & \text{otherwise} \end{cases} \quad (4.6)$$

The bnet shown in Eq.(4.3) contains the same info and has the same graph structure as the binary decision tree Fig.4.1.

The bnet shown in Fig.4.2 appears to correspond to the bnet of Eq.(4.7).



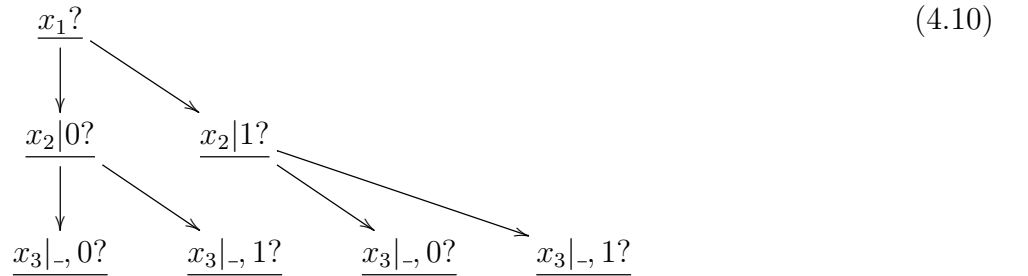
What happens if we find out that $P(x_3|x_1, x_2) = P(x_3|x_2)$ so that one of the arcs of the fully connected bnet Eq.(4.2) is unnecessary? In that case,

$$P(x_1, x_2, x_3) = P(x_3|x_2)P(x_2|x_1)P(x_1), \quad (4.8)$$

which can be represented by the Markov chain bnet:



Following the prescriptions of Chapter 5, we can represent the bnet of Eq.(4.9) as the bnet tree Eq.(4.10). In that tree, the nodes pose questions with 3 possible answers 0, 1, *null*.



Bibliography

- [1] Robert R. Tucci. Bell's inequalities for Bayesian statisticians. blog post in blog Quantum Bayesian Networks, <https://qbnets.wordpress.com/2008/09/19/bells-inequities-for-bayesian-statistician/>.
- [2] Wikipedia. Binary decision diagram. https://en.wikipedia.org/wiki/Binary_decision_diagram.
- [3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, David Warde-Farley Bing Xu, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. <https://arxiv.org/abs/1406.2661>.
- [4] Wikipedia. Hidden Markov model. https://en.wikipedia.org/wiki/Hidden_Markov_model.
- [5] Gregory Nuel. Tutorial on exact belief propagation in Bayesian networks: from messages to algorithms. <https://arxiv.org/abs/1201.4724>.
- [6] Wikipedia. Kalman filter. https://en.wikipedia.org/wiki/Kalman_filter.
- [7] Wikipedia. Markov blanket. https://en.wikipedia.org/wiki/Markov_blanket.
- [8] Judea Pearl. *Probabilistic Inference in Intelligent Systems*. Morgan Kaufmann, 1988.
- [9] Nitish Srivastava, G E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. <http://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf>.
- [10] Wikipedia. Non-negative matrix factorization. https://en.wikipedia.org/wiki/Non-negative_matrix_factorization.
- [11] Andrew Ng. Lecture at deeplearning.ai on recurrent neural networks. <http://www.ar-tiste.com/ng-lec-rnn.pdf>.
- [12] Wikipedia. Long short term memory. https://en.wikipedia.org/wiki/Long_short-term_memory.
- [13] Wikipedia. Gated recurrent unit. https://en.wikipedia.org/wiki/Gated_recurrent_unit.

- [14] Charles Fox, Neil Girdhar, and Kevin Gurney. A causal bayesian network view of reinforcement learning. <https://www.aaai.org/Papers/FLAIRS/2008/FLAIRS08-030.pdf>".
- [15] Sergey Levine. Course CS 285 at UC Berkeley, Deep reinforcement learning. <http://rail.eecs.berkeley.edu/deeprlcourse/>.
- [16] Robert R. Tucci. Simpson's paradox, the bane of clinical trials. blog post in blog Quantum Bayesian Networks <https://qbnets.wordpress.com/2020/07/09/simpsons-paradox-the-bane-of-clinical-trials/>.
- [17] Robert J. McEliece, David J. C. MacKay, and Jung-Fu Cheng. Turbo decoding as an instance of Pearls belief propagation algorithm. <http://authors.library.caltech.edu/6938/1/MCEieeejstc98.pdf>.