

**Bayesuvius,**  
a small visual dictionary of Bayesian Networks

Robert R. Tucci  
[www.ar-tiste.xyz](http://www.ar-tiste.xyz)

August 9, 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	16
6	Do-Calculus: COMING SOON	19
7	D-Separation: COMING SOON	20
8	Expectation Maximization	21
9	Generative Adversarial Networks (GANs)	25
10	Graph Structure Learning for bnets: COMING SOON	30
11	Hidden Markov Model	31
12	Influence Diagrams & Utility Nodes	35
13	Kalman Filter	37
14	Linear and Logistic Regression	40
15	Markov Blankets	44
16	Markov Chains	46
17	Markov Chain Monte Carlo (MCMC): COMING SOON	47

18 Message Passing (Belief Propagation): COMING SOON	48
19 Monty Hall Problem	49
20 Naive Bayes	51
21 Neural Networks	52
22 Non-negative Matrix Factorization	59
23 Program evaluation and review technique (PERT)	61
24 Recurrent Neural Networks	62
25 Reinforcement Learning (RL)	71
26 Reliability Models: COMING SOON	80
27 Restricted Boltzmann Machines	81
28 Simpson's Paradox	83
29 Turbo Codes	84
Bibliography	90

## 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.

---


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

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

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

---


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

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

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

---


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

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

---

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

- Entropy:

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

- Kullback-Liebler divergence:

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

- Cross entropy:

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

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

---

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 23

## Program evaluation and review technique (PERT)

This chapter is based on Refs.[13] and [14].

PERT diagrams are used for scheduling a project consisting of a series of interdependent activities and estimating how long it will take to finish the project. PERT diagrams were invented by the NAVY in 1958 to manage a submarine project. Nowadays they are taught in many business and management courses.

A **PERT diagram** is a Directed Acyclic Graph (DAG) with the following properties. (See Fig.23.2 for an example of a PERT diagram). The nodes  $\underline{E}_i$  for  $i = 1, 2, \dots, ne$  of a PERT diagram are called **events**. The edges  $i \rightarrow j$  of a PERT diagram are called **activities**. An event represents the starting (kickoff) date of one or more activities. A PERT diagram has single root node ( $i = 1$ , start event) and single leaf node ( $i = ne$ , end event).

The PERT diagram user must initially provide a **Duration Times (DT) table** which gives  $(DO_{i \rightarrow j}, DP_{i \rightarrow j}, DM_{i \rightarrow j})$  for each activity  $i \rightarrow j$ , where

$DO_{i \rightarrow j}$  = optimistic duration time of activity  $i \rightarrow j$

$DP_{i \rightarrow j}$  = pessimistic duration time of activity  $i \rightarrow j$

$DM_{i \rightarrow j}$  = median duration time of activity  $i \rightarrow j$

From the DT table, one calculates:

Duration time of activity  $i \rightarrow j$

$$D_{i \rightarrow j} = \frac{1}{6}(DO_{i \rightarrow j} + DP_{i \rightarrow j} + 4DM_{i \rightarrow j}) \quad (23.1)$$

Duration Variance of activity  $i \rightarrow j$

$$V_{i \rightarrow j} = \left( \frac{DO_{i \rightarrow j} - DP_{i \rightarrow j}}{DM_{i \rightarrow j}} \right)^2 \quad (23.2)$$

Often, it is convenient to define “dummy” edges with  $D_{i \rightarrow j} = 0$ . That is perfectly fine.

Define:

$TES_i$  = Earliest start time for event  $i$

$TLS_i$  = Latest start time for event  $i$



$slack_i = TLS_i - TES_i = \text{slack for event } i$

$TEF_{i \rightarrow j} = TES_i + D_{i \rightarrow j} = \text{Earliest finish time for activity } i \rightarrow j.$

$TLF_{i \rightarrow j} = TLS_j - D_{i \rightarrow j} = \text{Latest finish time for activity } i \rightarrow j. \text{ See footnote below. }^1$

A **critical path** is a directed path (i.e., a chain of connected arrows, all pointing in the same direction) going from the start to the end node, such that slack equals zero at every node visited. A critical path must also have all other nodes as neighbors; i.e, the union of the neighbors of every node in the path plus the nodes in the path itself, equals all nodes in the graph. In a DAG, the neighbors of a node is the union of its parent and children nodes.

---

**GOAL of PERT analysis:** The main goal of PERT analysis is to find, based on the data of the DT table, the interval  $[TES_i, TLS_i]$  giving a lower and an upper bound to the starting time of each node  $i$ . Another thing that we will find is a critical path for the PERT diagram (which represents an entire project). By adding the  $D_{i \rightarrow j}$  of each edge of the critical path, we will get the mean value of the total duration of the entire project, and by adding the variances of each edge along the critical path, we will get an estimate of the total variance of the total duration. Knowing the mean and variance of the total duration and assuming a normal distribution, we will be able to predict probabilities that the actual duration will deviate by a certain amount from its mean.

---

To calculate the interval  $[TES_i, TLS_i]$ , one follows the following two steps.

1. Assume  $TES_1 = 0$  and solve

$$TES_i = \max_{a \in pa(i)} \underbrace{(TES_a + D_{a \rightarrow i})}_{TEF_{a \rightarrow i}} \quad (23.3)$$

for  $i \in [2, ne]$ . This recursive equation is solved by what is called “forward propagation”, wherein one moves up the list of nodes  $i$  in order of increasing  $i$  starting at  $i = 1$  and  $TES_1 = 0$ .

2. Assume  $TLS_{ne} = TES_{ne}$  and solve

$$TLS_i = \min_{b \in ch(i)} \underbrace{(TLS_b - D_{i \rightarrow b})}_{TLF_{i \rightarrow b}} \quad (23.4)$$

for  $i \in [1, ne - 1]$ . This recursive equation is solved by what is called “backward propagation”, wherein one moves down the list of nodes  $i$  in order of decreasing  $i$  starting at  $i = ne$  and  $TLS_{ne} = TES_{ne}$ .  $TES_{ne}$  is known from step 1.

Eqs.(23.3) and (23.4) are illustrated in Fig.23.1.

---

<sup>1</sup> In the popular educational literature, the edge variables  $TEF_{i \rightarrow j}$  and  $TLF_{i \rightarrow j}$  are sometimes associated with the nodes, but they are clearly edge variables. This makes things confusing. The reason this is done is that some software draws PERT diagrams as trees whereas other software draws them as DAGs. For trees, storing  $TEF_{i \rightarrow j}$  and  $TLF_{i \rightarrow j}$  in a node makes some sense but not for DAGs. You will notice that giving specific names to the variables  $TEF_{i \rightarrow j}$  and  $TLF_{i \rightarrow j}$  is unnecessary. It is possible to delete all mention of their names from this chapter without losing any details. I only declare their names in this chapter so as tell the reader what they are in case he/she hears them mentioned and wonders what they are equal to in our notation.

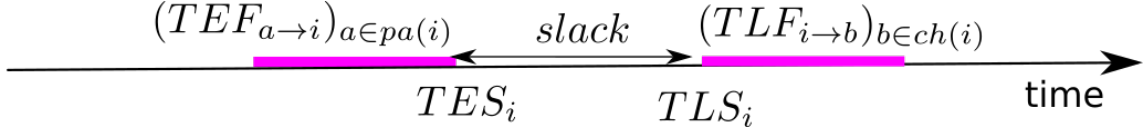


Figure 23.1:  $TES_i$  defined from info received from parents of  $i$  and  $TLS_i$  defined from info received from children of  $i$ .

## Example

To illustrate PERT analysis, we end with an example. We present the example in the form of an exercise question and then provide the answer. This example comes from Ref.[13], except for part (e) about bnets, which is our own.

**Question:** For the PERT diagram of Fig.23.2, calculate the following:

- (a) Interval  $[TES_i, TLS_i]$  for all  $i$ .
- (b) A critical path for this PERT diagram.
- (c) The mean and variance of the total duration of the critical path.
- (d) The probability that the total duration will be 225 days or less.
- (e) A bnet interpretation of this problem.

**Answer to (a)**  $[TES_i, TLS_i]$  are given by Fig.23.3.

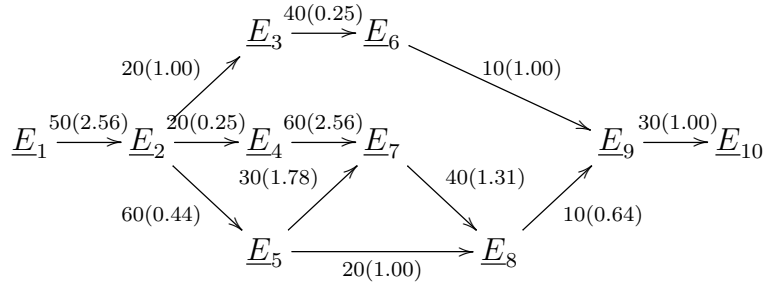


Figure 23.2: Example of a PERT diagram. The numbers attached to the arrows are the duration times  $D_{i \rightarrow j}$  in days followed by, enclosed in parentheses, the variance  $V_{i \rightarrow j}$  of that duration. The info given in this PERT diagram was derived from a DT table in Ref.[13]. The info in this PERT diagram is sufficient for calculating  $TES_i$  and  $TLS_i$  for each node  $i$ . The results of that calculation are given in Fig.23.3.

**Answer to (b)** The critical path is given in red in Fig.23.3. Note that this path does indeed have zero slack at each node it visits and the union of its neighborhood and the path itself encompasses all nodes.

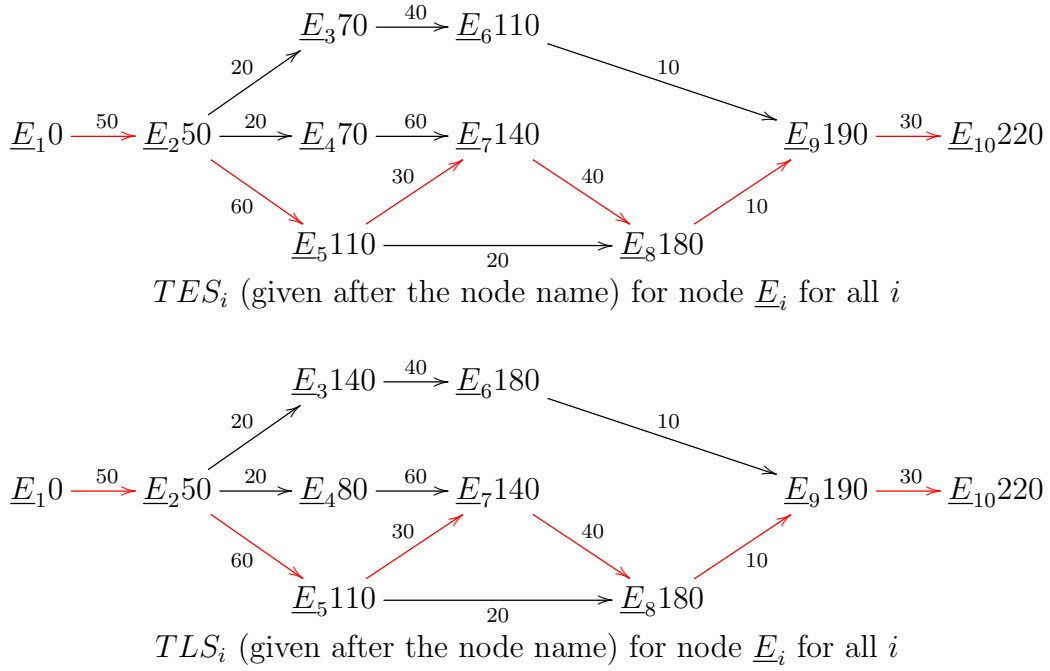


Figure 23.3: Results of calculating  $TES_i$  for all  $i$  via a forward pass, followed by calculating  $TLS_i$  for all  $i$  via a backward pass. Critical path indicated in red.

**Answer to (c)** The mean and variance of the total duration are given by Table 23.1.

**Answer to (d)**

$$P(x < 225) = p \left[ \frac{x - \mu}{\sigma} \leq \frac{225 - 220}{\sqrt{7.73}} \right] \quad (23.5)$$

$$= P[z \leq 1.80] \quad (23.6)$$

$$= 0.9641 \quad (23.7)$$

**Answer to (e)** Define 2 bnets.

1. The first PERT bnet is for calculating  $TES_i$  for all  $i$  and is given by Fig.23.4.

The node transition prob matrices, printed in blue, for the bnet Fig.23.4 are given by (this equation is to be evaluated recursively by a forward pass through the bnet):

$$P(TES_i | (TES_a)_{a \in pa(i)}) = \delta(TES_i, \max_{a \in pa(i)} (TES_a + D_{a \rightarrow i})) \quad (23.8)$$

2. The second PERT bnet is for calculating  $TLS_i$  for all  $i$  and is given by Fig.23.5. Note that the directions of all the arrows in the PERT diagram Fig.23.2 have been reversed so Fig.23.5 is a time reversed graph.

edge $i \rightarrow j$	duration $D_{i \rightarrow j}$	variance $V_{i \rightarrow j}$
A (1 $\rightarrow$ 2)	50	2.56
D (2 $\rightarrow$ 5)	60	0.44
G (5 $\rightarrow$ 7)	30	1.78
J (7 $\rightarrow$ 8)	40	1.31
K (8 $\rightarrow$ 9)	10	0.64
L (9 $\rightarrow$ 10)	30	1.00
Total	220	7.73

Table 23.1: Calculation of mean and variance of total duration along critical path.

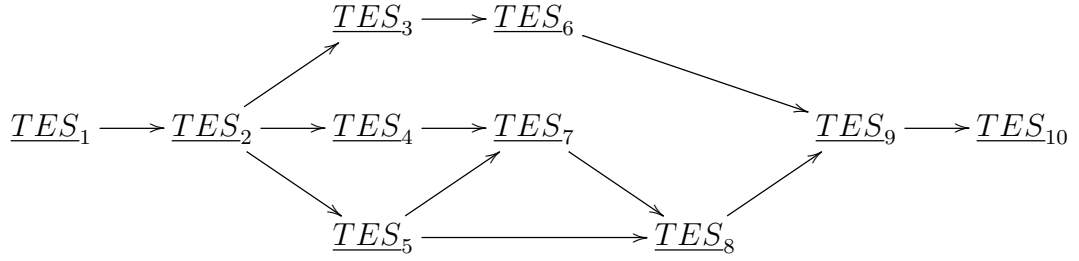


Figure 23.4: bnet for  $TES_i$  calculation.

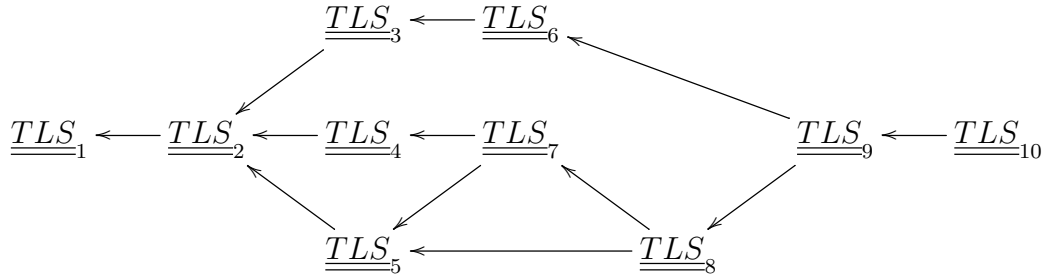


Figure 23.5: bnet for  $TLS_i$  calculation.

The node transition prob matrices, printed in blue, for the bnet Fig.23.5 are given by (this equation is to be evaluate recursively by a backward pass through the bnet):

$$P(TLS_i | (TLS_b)_{b \in pa(i)}) = \delta(TLS_i, \min_{b \in pa(i)} (TLS_b - D_{b \rightarrow i}^T)) , \quad (23.9)$$

where  $D_{i \rightarrow j}^T = D_{j \rightarrow i}$ .

# 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-inequaties-for-bayesian-statistician/>.
- [2] Wikipedia. Binary decision diagram. [https://en.wikipedia.org/wiki/Binary\\_decision\\_diagram](https://en.wikipedia.org/wiki/Binary_decision_diagram).
- [3] Wikipedia. Expectation maximization. [https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization\\_algorithm](https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm).
- [4] Wikipedia. k-means clustering. [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering).
- [5] 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>.
- [6] Wikipedia. Hidden Markov model. [https://en.wikipedia.org/wiki/Hidden\\_Markov\\_model](https://en.wikipedia.org/wiki/Hidden_Markov_model).
- [7] Gregory Nuel. Tutorial on exact belief propagation in Bayesian networks: from messages to algorithms. <https://arxiv.org/abs/1201.4724>.
- [8] Wikipedia. Kalman filter. [https://en.wikipedia.org/wiki/Kalman\\_filter](https://en.wikipedia.org/wiki/Kalman_filter).
- [9] Wikipedia. Markov blanket. [https://en.wikipedia.org/wiki/Markov\\_blanket](https://en.wikipedia.org/wiki/Markov_blanket).
- [10] Judea Pearl. *Probabilistic Inference in Intelligent Systems*. Morgan Kaufmann, 1988.
- [11] 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>.
- [12] Wikipedia. Non-negative matrix factorization. [https://en.wikipedia.org/wiki/Non-negative\\_matrix\\_factorization](https://en.wikipedia.org/wiki/Non-negative_matrix_factorization).
- [13] theinvestorsbook.com. Pert analysis. <https://theinvestorsbook.com/pert-analysis.html>.

- [14] Wikipedia. Program evaluation and review technique. [https://en.wikipedia.org/wiki/Program\\_evaluation\\_and\\_review\\_technique](https://en.wikipedia.org/wiki/Program_evaluation_and_review_technique).
- [15] Andrew Ng. Lecture at deeplearning.ai on recurrent neural networks. <http://www.ar-tiste.com/ng-lec-rnn.pdf>.
- [16] Wikipedia. Long short term memory. [https://en.wikipedia.org/wiki/Long\\_short-term\\_memory](https://en.wikipedia.org/wiki/Long_short-term_memory).
- [17] Wikipedia. Gated recurrent unit. [https://en.wikipedia.org/wiki/Gated\\_recurrent\\_unit](https://en.wikipedia.org/wiki/Gated_recurrent_unit).
- [18] 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>.
- [19] Sergey Levine. Course CS 285 at UC Berkeley, Deep reinforcement learning. <http://rail.eecs.berkeley.edu/deeprlcourse/>.
- [20] 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/>.
- [21] 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>.