Bayesuvius, a small visual dictionary of Bayesian Networks

Robert R. Tucci
www.ar-tiste.xyz

August 10, 2020

Figure 1: View of Mount Vesuvius from Pompeii

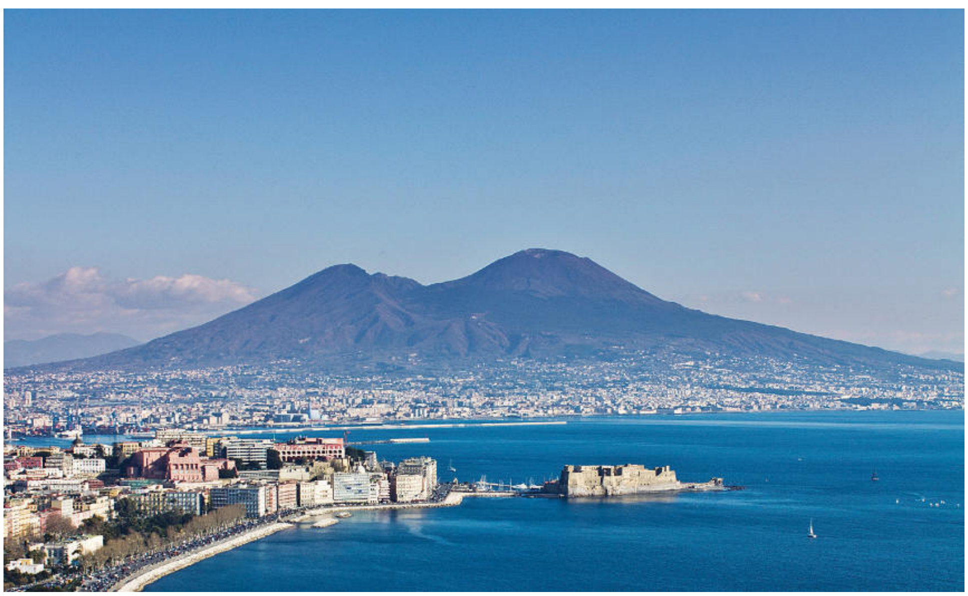


Figure 2: Mount Vesuvius and Bay of Naples

Contents

0.2 Notational Conventions

bnet=B net=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,

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

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

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

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

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

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

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

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 $f : S \to \mathbb{R}$. Let $S_{\underline{x}} \subset S$ and $S \to S_{\underline{x}}$ upon discretization (binning). Then

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

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

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

Indicator function (aka Truth function):

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

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

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

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

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

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

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

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

$$\sum_{\vec{x}} = \prod_i \sum_{x[i]} \tag{13}$$

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

$$\begin{aligned} P(\vec{x}) &\approx [\prod_x P(x)^{P(x)}]^{nsam(\vec{x})} && (15)\\ &= e^{nsam(\vec{x})\sum_x P(x)\ln P(x)} && (16)\\ &= e^{-nsam(\vec{x})H(P_{\underline{x}})} && (17)\end{aligned}$$

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

$$f^+ = f^{[1,\partial_x,\partial_y]} \tag{19}$$

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

- Entropy:

$$H(p) = -\sum_x p(x)\ln p(x) \geq 0 \tag{20}$$

- Kullback-Liebler divergence:

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

- Cross entropy:

$$\begin{aligned} CE(p \to q) &= -\sum_x p(x)\ln q(x) && (22)\\ &= H(p) + D_{KL}(p \parallel q) && (23)\end{aligned}$$

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

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

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

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

Expected Value

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

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

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 $f : S_{\underline{x}} \times S_{\underline{y}} \to \mathbb{R}$, define

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

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

Note that

$$\begin{aligned} E_{\underline{y}}[E_{\underline{x}|\underline{y}}[f(\underline{x}, \underline{y})]] &= \sum_{x,y} P(x|y)P(y)f(x,y) && (29) \\ &= \sum_{x,y} P(x,y)f(x,y) && (30) \\ &= E_{\underline{x},\underline{y}}[f(\underline{x}, \underline{y})] \ . && (31) \end{aligned}$$

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

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

$\mathcal{N}(!a)$ 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 $x^n = (x_0, x_1, \ldots, x_{n-1})$ and $x_i = \delta(i, 0)$ then x^n is one hot.

Chapter 6

Digital Circuits

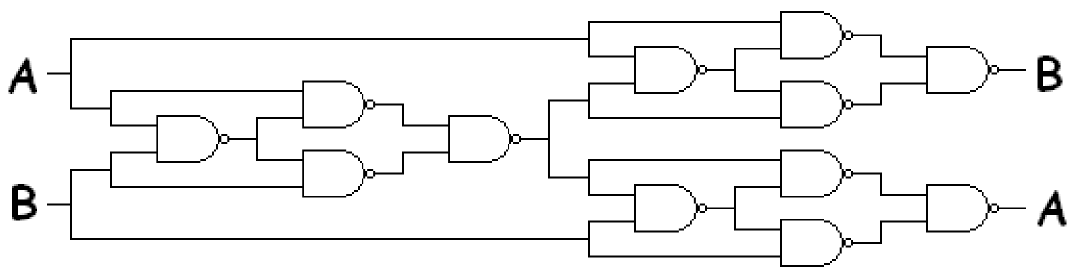


Figure 6.1: Typical digital circuit of NAND gates.

Digital (logic) gate: node with na input ports and nx output ports which represents a function

$$f : \{0,1\}^{na} \rightarrow \{0,1\}^{nx} . \tag{6.1}$$

Suppose
$a^{na} = (a_i)_{i=0,1,\ldots,na-1}$ where $a_i \in \{0,1\}$,
$x^{nx} = (x_i)_{i=0,1,\ldots,nx-1}$ where $x_i \in \{0,1\}$.
f maps a^{na} into x^{nx}.
Digital circuit (dcircuit) = circuit of digital gates.

Mapping any dcircuit to a bnet (Option A- See Fig.6.2)):

1. Replace every dcircuit gate described by Eq.(6.1) by nx bnet nodes $\underline{x}_i$ for $i = 0, 1, \ldots, nx - 1$ such that

$$P(x_i|a^{na}) = \delta(x_i, f_i(a^{na})) \tag{6.2}$$

2. Replace all connectors of the dcircuit by arrows pointing in the direction of the bit flow.

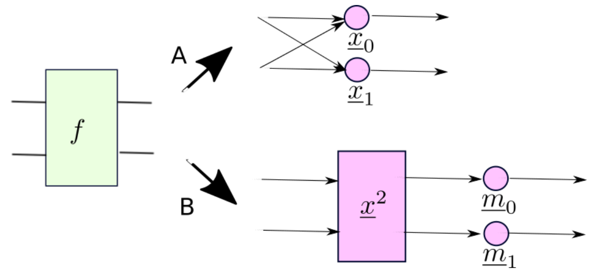


Figure 6.2: 2 options for mapping dcircuit node with multiple output ports into bnet.

Mapping any dcircuit to a bnet (Option B- See Fig.6.2)):

1. Replace every dcircuit gate described by Eq.(6.1) with one bnet node called $\underline{x}^{nx}$ and, if $nx > 0$, nx "marginalizer nodes" $\underline{m}_i$ for $i = 0, 1, \ldots, nx - 1$, such that

$$P(x^{nx}|a^{na}) = \delta(x^{nx}, f(a^{na})) \; , \tag{6.3}$$

and

$$P(m_i|x^{nx}) = \delta(m_i, x_i)) \; . \tag{6.4}$$

2. Replace all connectors of the dcircuit by arrows pointing in the direction of the bit flow.

Options A and B don't work for digital circuits with feedback loops such as flip-flops.

Chapter 24

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.24.2 for an example of a PERT diagram). The nodes $\underline{E}_i$ for $i = 1, 2, \ldots, ne$ 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 a single root node ($i = 1$, start event) and a 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 $i \rightarrow j$

$$D_{i\rightarrow j} = \frac{1}{6}(DO_{i\rightarrow j} + DP_{i\rightarrow j} + 4DM_{i\rightarrow j}) \tag{24.1}$$

Duration Variance of activity $i \rightarrow j$

$$V_{i\rightarrow j} = \left(\frac{DO_{i\rightarrow j} - DP_{i\rightarrow j}}{DM_{i\rightarrow j}}\right)^2 \tag{24.2}$$

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

$slack_i = TLS_i - TES_i$ = slack for event i

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

$TLF_{i\to j} = TLS_j - D_{i\to j}$ = Latest finish time for activity $i \to j$. 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. In a DAG, the neighbors of a node is the union of its parent and children nodes. 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.

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 goal is to find a critical path for the PERT diagram (which represents an entire project). By adding the $D_{i\to j}$ of each edge of the critical path, one can get the mean value of the total duration of the entire project, and by adding the variances of each edge along the critical path, one can 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, one can predict the probability 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

$$TES_i = \max_{a\in pa(i)} (\underbrace{TES_a + D_{a\to i}}_{TEF_{a\to i}}) \tag{24.3}$$

   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 $i = 1$ with $TES_1 = 0$.

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

$$TLS_i = \min_{b\in ch(i)} (\underbrace{TLS_b - D_{i\to b}}_{TLF_{i\to b}}) \tag{24.4}$$

   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 $i = ne$ with $TLS_{ne} = TES_{ne}$. TES_{ne} is known from step 1.

Eqs.(24.3) and (24.4) are illustrated in Fig.24.1.

[1] In the popular educational literature, the edge variables $TEF_{i\to j}$ and $TLF_{i\to j}$ 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 $TEF_{i\to j}$ and $TLF_{i\to j}$ in a node makes some sense but not for DAGs. You will notice that giving specific names to the variables $TEF_{i\to j}$ and $TLF_{i\to j}$ 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.

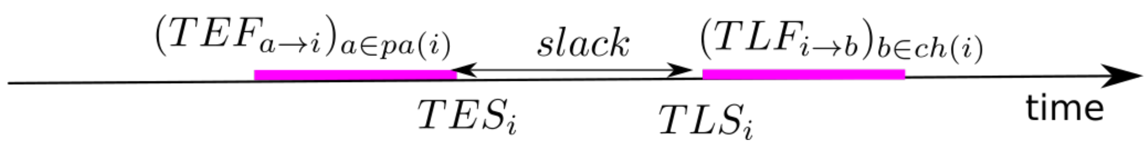


Figure 24.1: TES_i defined from info received from parents of i and TLS_i defined from info received from children of i.

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.24.2, 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.24.3.

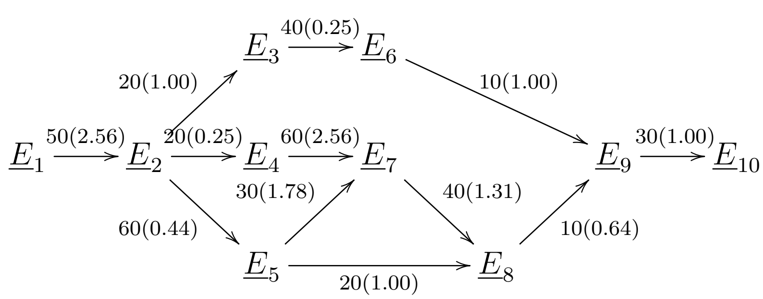
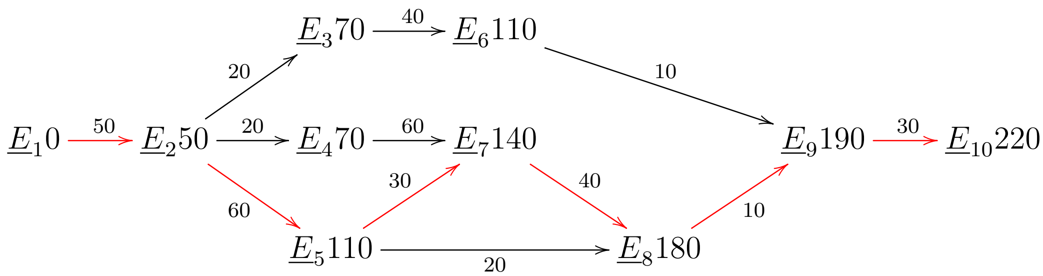
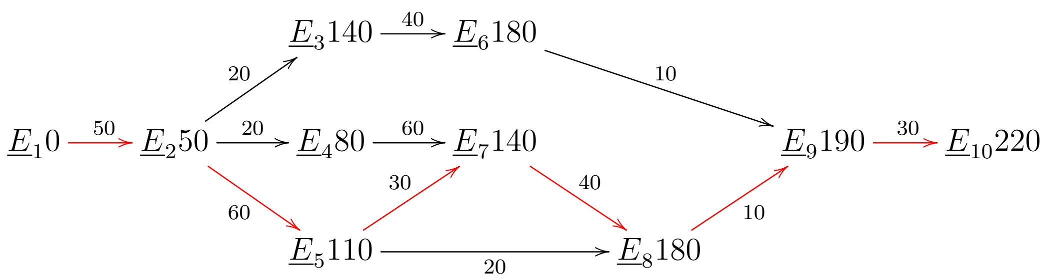


Figure 24.2: Example of a PERT diagram. The numbers attached to the arrows are the duration times $D_{i\to j}$ in days followed by, enclosed in parentheses, the variance $V_{i\to j}$ of that duration. The info given in this PERT diagram was derived from a DT table in Ref.[13]. The info in this PERT diagram is sufficient for calculating TES_i and TLS_i for each node i. The results of that calculation are given in Fig.24.3.

Answer to (b) The critical path is given in red in Fig.24.3. 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.



TES_i (given after the node name) for node $\underline{E}_i$ for all i



TLS_i (given after the node name) for node $\underline{E}_i$ for all i

Figure 24.3: 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 calculated in Table 24.1.

Answer to (d)

$$P(x < 225) = p\left[\frac{x-\mu}{\sigma} \leq \frac{225-220}{\sqrt{7.73}}\right] \tag{24.5}$$
$$= P[z \leq 1.80] \tag{24.6}$$
$$= 0.9641 \tag{24.7}$$

Answer to (e) Define 2 bnets.

1. The first PERT bnet is for calculating TES_i for all i and is given by Fig.24.4.
   The node transition prob matrices, printed in blue, for the bnet Fig.24.4 are given by (this equation is to be evaluated recursively by a forward pass through the bnet):

$$P(TES_i|(TES_a)_{a\in pa(i)}) = \delta(TES_i, \max_{a\in pa(i)}(TES_a + D_{a\to i})) \tag{24.8}$$

2. The second PERT bnet is for calculating TLS_i for all i and is given by Fig.24.5. Note that the directions of all the arrows in the PERT diagram Fig.24.2 have been reversed so Fig.24.5 is a time reversed graph.

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

Table 24.1: Calculation of mean and variance of total duration along critical path.

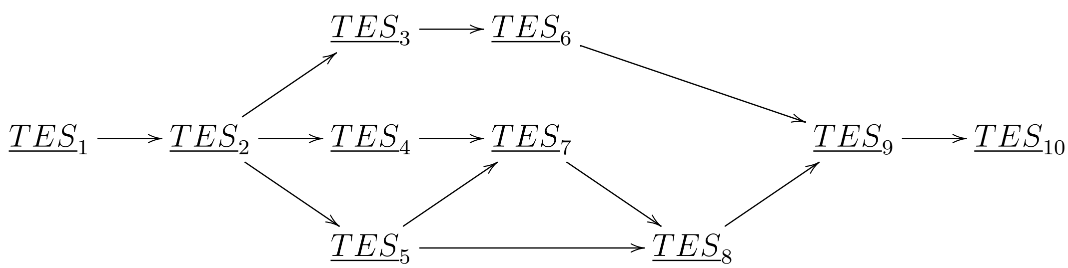


Figure 24.4: bnet for TES_i calculation.

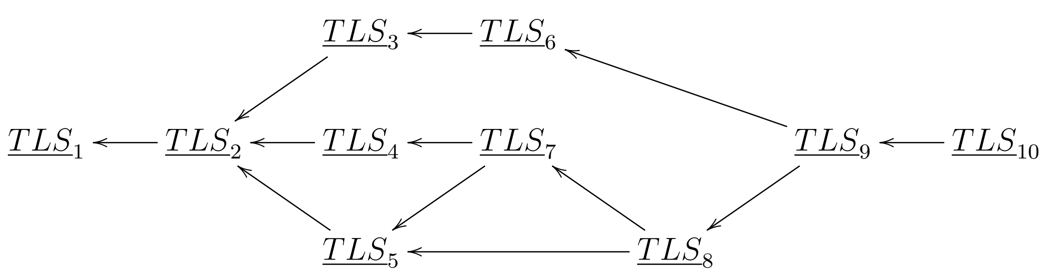


Figure 24.5: bnet for TLS_i calculation.

The node transition prob matrices, printed in blue, for the bnet Fig.24.5 are given by (this equation is to be evaluate recursively by a backward pass through the bnet):

$$P(\underline{TLS}_i|(\underline{TLS}_b)_{b\in pa(i)}) = \delta(\underline{TLS}_i, \min_{b\in pa(i)}(\underline{TLS}_b - D^T_{b\to i})) \;, \tag{24.9}$$

where $D^T_{i\to j} = D_{j\to i}$.

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.

[3] Wikipedia. Expectation maximization. https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm.

[4] Wikipedia. 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.

[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.

[9] Wikipedia. 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.

[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.

[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.

[17] Wikipedia. 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.