Bayesuvius,
a small visual dictionary of Bayesian Networks

Robert R. Tucci
www.ar-tiste.xyz

October 17, 2020

Figure 1: View of Mount Vesuvius from Pompeii

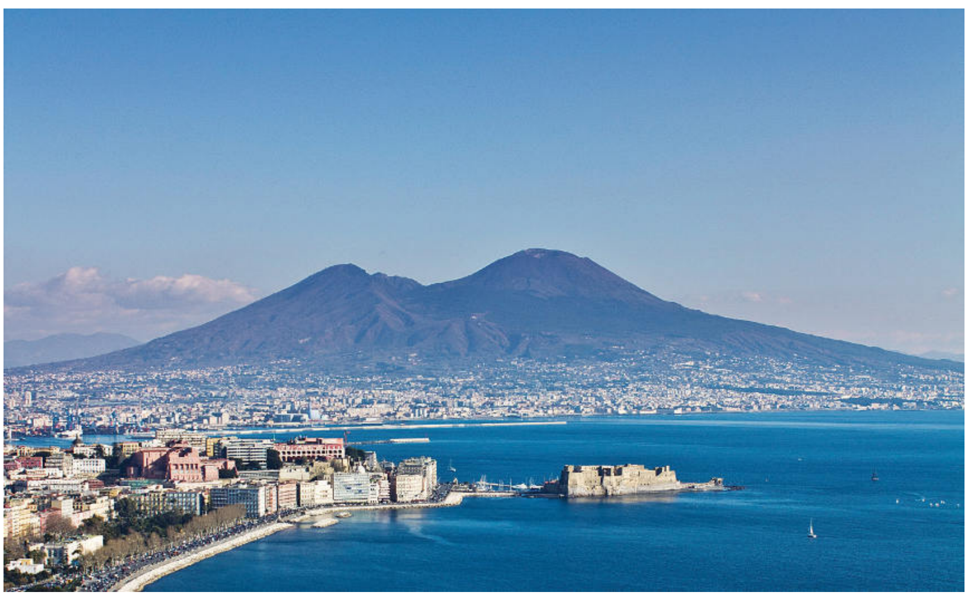

Figure 2: Mount Vesuvius and Bay of Naples

Contents

Chapter 16

Gaussian Nodes with Linear Dependence on Parents

Bnet nodes that have a Gaussian TPM with a linear dependence on their parent nodes (GLP) are a very popular way of modeling continuous nodes of bnets. A convenient aspect of them is that their parent nodes can be either continuous or discrete. Also, they can be learned easily from the data because their parameters can be expressed as two node covariances. For these reasons, they are commonly used when doing structure learning of bnets with continuous nodes (see Chapter 40).

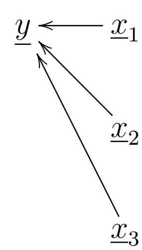

Figure 16.1: GLP node $\underline{y}$ with 3 parent nodes $\underline{x}^3 = (\underline{x}_1, \underline{x}_2, \underline{x}_3)$.

Recall our notation for a Gaussian distribution:

$$\mathcal{N}(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \ , \tag{16.1}$$

where $x, \mu \in \mathbb{R}$ and $\sigma > 0$.

A GLP node $\underline{y}$ with n parents $\underline{x}^n = (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n)$ has the following TPM:

$$P(y|x^n) = \mathcal{N}(y; \beta_0 + \beta^{nT} x^n, \sigma^2) \tag{16.2}$$

where $\underline{y}, \beta_0, \in \mathbb{R}$ and $\sigma^2 > 0$, and where $\underline{x}^n, \beta^n \in \mathbb{R}^n$ are **column vectors**. The T in β^{nT} stands for transpose. Any $\underline{x}_i$ can have a discrete set of states as long as they are real valued and ordinal (ordered by size). Fig.16.1 shows a diagrammatic representation of a GPL node with 3 parents.

An equivalent way of defining a GLP node $\underline{y}$ is in terms of a random variable equation expressing $\underline{y}$ as a hyperplane function of the parents $\underline{x}^n$ plus a Gaussian noise variable. Define an estimator $\underline{\hat{y}}$ of $\underline{y}$ by

$$\underline{\hat{y}} = \beta_0 + \beta^{nT} \underline{x}^n \tag{16.3a}$$

and

$$\underline{y} = \underline{\hat{y}} + \underline{\epsilon} \tag{16.3b}$$

where the residual $\underline{\epsilon}$ satisfies

$$P(\epsilon) = \mathcal{N}(\epsilon; 0, \sigma^2) \tag{16.3c}$$

and

$$\langle \underline{x}^n, \underline{\epsilon} \rangle = 0 \ . \tag{16.3d}$$

The notation $\langle \underline{x}, \underline{y} \rangle$ for the covariance of random variables $\underline{x}$ and $\underline{y}$ is explained in Chapter 0.3.

Claim 13 *The parameters of a GLP node can be expressed as 2-node covariances. Specifically,*

$$\beta^n = \langle \underline{x}^n, \underline{x}^{nT} \rangle^{-1} \langle \underline{y}, \underline{x}^n \rangle \tag{16.4}$$

$$\beta_0 = \langle \underline{y} \rangle - \beta^{nT} \langle \underline{x}^n \rangle \tag{16.5}$$

$$\sigma^2 = \langle \underline{y}, \underline{y} \rangle - \beta^{nT} \langle \underline{x}^n, \underline{y} \rangle \tag{16.6}$$

proof:

Note that $\langle \underline{x}^n, \underline{x}^{nT} \rangle^T = \langle \underline{x}^n, \underline{x}^{nT} \rangle$ and $\langle \underline{y}, \underline{x}^{nT} \rangle^T = \langle \underline{y}, \underline{x}^n \rangle$.

$$\langle \underline{y}, \underline{x}^{nT} \rangle = \beta^{nT} \langle \underline{x}^n, \underline{x}^{nT} \rangle \tag{16.7}$$

$$\langle \underline{y}, \underline{x}^n \rangle = \langle \underline{x}^n, \underline{x}^{nT} \rangle \beta^n \tag{16.8}$$

$$\beta^n = \langle \underline{x}^n, \underline{x}^{nT} \rangle^{-1} \langle \underline{y}, \underline{x}^n \rangle \tag{16.9}$$

$$\langle \underline{y} \rangle = \beta_0 + \beta^{nT} \langle \underline{x}^n \rangle \tag{16.10}$$

$$\begin{aligned} \langle \underline{y}, \underline{y} \rangle &= \langle \beta_0 + \beta^{nT} \underline{x}^n + \underline{\epsilon}, \underline{y} \rangle && (16.11) \\ &= \beta^{nT} \langle \underline{x}^n, \underline{y} \rangle + \sigma^2 && (16.12) \end{aligned}$$

QED

Chapter 40

Structure and Parameter Learning for bnets: COMING SOON

Ref.[46]
[47]

Overview

- Parameter Learning (PL)-post SL
  - missing data
- Structure Learning (SL)-pre PL
  - tree-like structures given a priori
    - Naive Bayes
    - Chow-Liu tree
    - Tree Augmented Naive Bayes (TAN)
    - ARACNE
  - score based
    - algorithms
      - hill climbing (HC)
      - HC with random restarts
      - Tabu search (Tabu)
      - simulated annealing
      - genetic algorithms
    - scoring functions
      - log-likelihood (LL)
      - predictive log-likelihood (PLL)
      - Akaike Information Criterion (AIC)
      - Bayesian Information Criterion (BIC)
      - Minimum Description Length (MDL) (same as BIC)
      - Bayesian Dirichlet (BD) family
        - K2 score

- score equivalent Dirichlet posterior density (BDe)
- sparse Dirichlet posterior density (BDs)
- Dirichlet posterior density based on Jeffrey's prior (BDJ)
- modified Bayesian Dirichlet for mixtures of interventional and observational data
- locally averaged BDe score (BDla)

- constraint based
  - algorithms
    - Inductive Causation (IC)
    - Parents & Children (PC) family
      - PC (the stable version)
      - Max-Min Parents & Children (MMPC)
      - Semi-Interleaved Hiton-PC (SI-HITON-PC)
      - Hybrid Parents & Children (HPC)
    - Grow-Shrink (GS)
    - IAMB family
      - Incremental Association Markov Blanket (IAMB)
      - Fast Incremental Association (Fast-IAMB)
      - Interleaved Incremental Association (Inter-IAMB)
      - Incremental Association with FDR Correction (IAMB-FDR)
  - conditional independence tests
    - mutual information (parametric, semiparametric and permutation tests)
    - shrinkage-estimator for the mutual information
    - Pearson's X2 (parametric, semiparametric and permutation tests)
    - Jonckheere-Terpstra (parametric and permutation tests)
    - linear correlation (parametric, semiparametric and permutation tests)
    - Fisher's Z (parametric, semiparametric and permutation tests)
- hybrid
  - Max-Min Hill Climbing (MMHC)
  - Hybrid HPC (H2PC)
  - General 2-Phase Restricted Maximization (RSMAX2)

- parallel programming structure learning
- node types
  - discrete
    - categorical (unordered) (multinomial distribution)
    - ordinal (ordered)
  - continuous (multivariate normal distribution)
  - mixed (conditional Gaussian distribution)

Tidbits

linear regression

$$\underline{y} = \beta_0 + \sum_{j=1}^{N} \beta_j \underline{x}_j \tag{40.1}$$

For $k = 1, \ldots, N$,

$$\langle \underline{x}_k, \underline{y} \rangle = \sum_{j=1}^{N} \beta_j \langle \underline{x}_k, \underline{x}_j \rangle \tag{40.2}$$

$\underline{x} = (\underline{x}_1, \ldots, \underline{x}_N)^T$

$$\langle \underline{x}, \underline{y} \rangle = \langle \underline{x}, \underline{x}^T \rangle \beta \tag{40.3}$$

$$\beta = \langle \underline{x}, \underline{x}^T \rangle^{-1} \langle \underline{x}, \underline{y} \rangle \tag{40.4}$$

Categorical and Dirichlet Distributions

Ref.[48] Ref.[49]
$q_+ = \sum_i q_i$, $q. = (q_0, q_1, \ldots, q_{nq-1})$

$$cat(x; \pi.) = \pi_x = \prod_k \pi_k^{\mathbb{1}(k=x)} \tag{40.5}$$

$$Dir(\pi.; \alpha.) = \mathbb{1}(\pi_+ = 1)\Gamma(\alpha_+) \prod_k \frac{\pi_k^{\alpha_k - 1}}{\Gamma(\alpha_k)} \tag{40.6}$$

$$cat(x; \pi.)Dir(\pi.; \alpha.) = \mathcal{N}(!\pi.)Dir(\pi.; \alpha.') \tag{40.7}$$

$$\alpha'_k = \alpha_k + \mathbb{1}(x = k) \tag{40.8}$$

$$P(x|\pi.) = cat(x; \pi.) \tag{40.9}$$

$$P(\pi.) = Dir(\pi.; \alpha.) \tag{40.10}$$

$$P(x|\pi.)P(\pi.) = \mathcal{N}(!\pi.)P(\pi.|x) \tag{40.11}$$

$$\begin{aligned} P(\pi.|x) &= \mathcal{N}(!\pi.)cat(x; \pi.)Dir(\pi.; \alpha.) && (40.12) \\ &= Dir(\pi.; \alpha'.) && (40.13) \end{aligned}$$

Most popular prob distributions used to model PTMs

D=Discrete C=Continuous

$$\underline{x}_i \longleftarrow \underline{\Theta}^i \tag{40.14}$$

- (D—D)

$$cat(k; \pi^i_{.|j}) = \pi^i_{k|j} \tag{40.15}$$

$$[\Theta^i]_{k,j} = \pi^i_{k|j} \tag{40.16}$$

$$P(\underline{x}_i = k | pa(\underline{x}_i) = j, \Theta^i) = cat(k; \pi^i_{.|j}) \tag{40.17}$$

$$P(\Theta^i) = \prod_j Dir(\pi^i_{.|j}; \alpha^i_{.|j}) \tag{40.18}$$

$$P(\underline{x}_i = k | pa(\underline{x}_i) = j, \Theta^i) P(\Theta^i) = \mathcal{N}(!\Theta^i) P(\Theta^i | \underline{x}_i = k, pa(\underline{x}_i) = j) \tag{40.19}$$

$$\begin{aligned} P(\Theta^i | \underline{x}_i = k, pa(\underline{x}_i) = j) &= \mathcal{N}(!\Theta^i) cat(k; \pi^i_{.|j}) \prod_{j'} Dir(\pi^i_{.|j'}; \alpha^i_{.|j'}) &(40.20) \\ &= \prod_{j'} Dir(\pi^i_{.|j'}; \beta^i_{.|j'}) &(40.21) \end{aligned}$$

$$\beta^i_{k'|j'} = \alpha^i_{k'|j'} + \mathbb{1}(k = k', j = j') \tag{40.22}$$

- (C—C)
- (C—D)
- (D—C)

Bibliography

[1] Wikipedia. Boolean algebra. https://en.wikipedia.org/wiki/Boolean_algebra.

[2] Judea Pearl. *Probabilistic Inference in Intelligent Systems*. Morgan Kaufmann, 1988.

[3] Judea Pearl. *Causality: Models, Reasoning, and Inference, Second Edition*. Cambridge University Press, 2013.

[4] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[5] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic Books, 2018.

[6] Christina Heinze-Deml. Causality, spring semester 2019 at ETH Zurich. https://stat.ethz.ch/lectures/ss19/causality.php#course_materials.

[7] 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/.

[8] Wikipedia. Binary decision diagram. https://en.wikipedia.org/wiki/Binary_decision_diagram.

[9] Wikipedia. Chow-Liu tree. https://en.wikipedia.org/wiki/Chow%E2%80%93Liu_tree.

[10] Wikipedia. Minimum spanning tree. https://en.wikipedia.org/wiki/Minimum_spanning_tree.

[11] Judea Pearl. Causal and counterfactual inference. *The Handbook of Rationality*, pages 1–41, 2019. https://ftp.cs.ucla.edu/pub/stat_ser/r485.pdf.

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

[13] Wikipedia. k-means clustering. https://en.wikipedia.org/wiki/K-means_clustering.

[14] 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.

[15] Wikipedia. Hidden Markov model. https://en.wikipedia.org/wiki/Hidden_Markov_model.

[16] Gregory Nuel. Tutorial on exact belief propagation in Bayesian networks: from messages to algorithms. https://arxiv.org/abs/1201.4724.

[17] Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–194, 1988. http://www.eecis.udel.edu/~shatkay/Course/papers/Lauritzen1988.pdf.

[18] Wikipedia. Junction tree algorithm. https://en.wikipedia.org/wiki/Junction_tree_algorithm.

[19] Cecil Huang and Adnan Darwiche. Inference in belief networks: A procedural guide. *International journal of approximate reasoning*, 15(3):225–263, 1996. http://www.ar-tiste.com/Huang-Darwiche1996.pdf.

[20] Robert R. Tucci. Quantum Fog. https://github.com/artiste-qb-net/quantum-fog.

[21] Wikipedia. Kalman filter. https://en.wikipedia.org/wiki/Kalman_filter.

[22] Wikipedia. Markov blanket. https://en.wikipedia.org/wiki/Markov_blanket.

[23] Wikipedia. Monte Carlo methods. https://en.wikipedia.org/wiki/Category:Monte_Carlo_methods.

[24] Wikipedia. Inverse transform sampling. https://en.wikipedia.org/wiki/Inverse_transform_sampling.

[25] Wikipedia. Rejection sampling. https://en.wikipedia.org/wiki/Rejection_sampling.

[26] Dan Bendel. Metropolis-Hastings: A comprehensive overview and proof. https://similarweb.engineering/mcmc/.

[27] Wikipedia. Metropolis-Hastings method. https://en.wikipedia.org/wiki/Metropolis%E2%80%93Hastings_algorithm.

[28] Wikipedia. Gibbs sampling. https://en.wikipedia.org/wiki/Gibbs_sampling.

[29] Wikipedia. Importance sampling. https://en.wikipedia.org/wiki/Importance_sampling.

[30] Judea Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. https://www.aaai.org/Papers/AAAI/1982/AAAI82-032.pdf, 1982.

[31] Wikipedia. Belief propagation. https://en.wikipedia.org/wiki/Belief_propagation.

[32] Richard E Neapolitan. *Learning bayesian networks*. Pearson Prentice Hall, 2004.

[33] 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.

[34] Wikipedia. Non-negative matrix factorization. https://en.wikipedia.org/wiki/Non-negative_matrix_factorization.

[35] Bruno Gonçalves. Model testing and causal search. blog post https://medium.com/data-for-science/causal-inference-part-vii-model-testing-and-causal-search-536b796f

[36] theinvestorsbook.com. Pert analysis. https://theinvestorsbook.com/pert-analysis.html.

[37] Wikipedia. Program evaluation and review technique. https://en.wikipedia.org/wiki/Program_evaluation_and_review_technique.

[38] Andrew Ng. Lecture at deeplearning.ai on recurrent neural networks. http://www.ar-tiste.com/ng-lec-rnn.pdf.

[39] Wikipedia. Long short term memory. https://en.wikipedia.org/wiki/Long_short-term_memory.

[40] Wikipedia. Gated recurrent unit. https://en.wikipedia.org/wiki/Gated_recurrent_unit.

[41] 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".

[42] Sergey Levine. Course CS 285 at UC Berkeley, Deep reinforcement learning. http://rail.eecs.berkeley.edu/deeprlcourse/.

[43] ReliaSoft. System analysis reference. http://reliawiki.org/index.php/System_Analysis_Reference.

[44] W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl. Fault tree handbook nureg-0492. https://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0492/.

[45] Wikipedia. Simpson's paradox. https://en.wikipedia.org/wiki/Simpson's_paradox.

[46] Marco Scutari. bnlearn. https://www.bnlearn.com/.

[47] Marco Scutari, Catharina Elisabeth Graafland, and José Manuel Gutiérrez. Who learns better bayesian network structures: Accuracy and speed of structure learning algorithms. https://arxiv.org/abs/1805.11908.

[48] Wikipedia. Categorical distribution. https://en.wikipedia.org/wiki/Categorical_distribution.

[49] Wikipedia. Dirichlet distribution. https://en.wikipedia.org/wiki/Dirichlet_distribution.

[50] 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.

[51] Wikipedia. Variational bayesian methods. https://en.wikipedia.org/wiki/Variational_Bayesian_methods.