COMS30035, Machine learning: Probabilistic Graphical Models 2

James Cussens
james.cussens@bristol.ac.uk

Department of Computer Science, SCEEM
University of Bristol

October 4, 2022

Agenda

- Various examples of ML models represented by Bayesian networks

Using BNs to represent ML models

- Machine learning research papers frequently use Bayesian networks to graphically represent machine learning models.
- They represent *the data-generating process*.
- Here's an example from NeurIPS 2019 [BS19].

Differentially private Bayesian linear regression

3.1 Privacy mechanism

Using the Laplace mechanism to release the noisy sufficient statistics z results in the model shown in Figure 1. This is the same model used in non-private linear regression except for the introduction of z, which requires the exact sufficient statistics s to have finite sensitivity. A standard assumption in literature [Awan and Slavkovic, 2018, Sheffet, 2017, Wang, 2018, Zhang et al., 2012] is to assume x and y have known a priori lower and upper bounds, $(a_{\mathbf{x}}, b_{\mathbf{x}})$ and (a_y, b_y), with bound widths $w_{\mathbf{x}} = b_{\mathbf{x}} - a_{\mathbf{x}}$ (assuming, for simplicity, equal bounds for all covariate dimensions) and $w_y = b_y - a_y$, respectively. We can then reason about the worst case influence of an individual on each component of $\mathrm{s} = [X^T X, X^T \mathrm{y}, \mathrm{y}^T \mathrm{y}]$, recalling that $\mathrm{s} = \sum_i t(\mathrm{x}^{(i)}, y^{(i)})$, so that $[\Delta_{(X^T X)_{jk}}, \Delta_{(X y)_j}, \Delta_{y^2}] = [w_{\mathbf{x}}^2, w_{\mathbf{x}} w_y, w_y^2]$. The number of unique elements[2] in s is $[d(d+1)/2,\ d,\ 1]$, so $\Delta_{\mathbf{s}} = w_{\mathbf{x}}^2 d(d+1)/2 + w_{\mathbf{x}} w_y d + w_y^2$. The noisy sufficient statistics fit for public release are

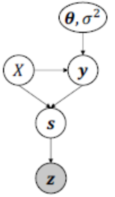


Figure 1: Private regression model.

James Cussens
james.cussens@bristol.ac.uk

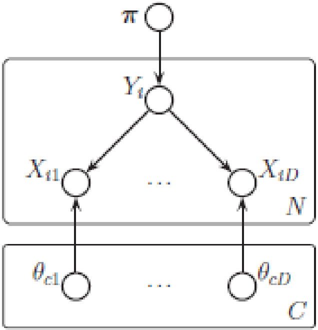
Naive Bayes

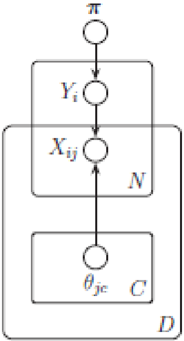
- Kevin Murphy's book makes extensive use of graphical models for machine learning.
- In a naive Bayes model for classification [Bis06, p. 380] the observed variables $\mathbf{x} = (x_1, \ldots x_D)$ are assumed independent conditional on the class variable $\mathbf{z}$:

$$P(\mathbf{x}, \mathbf{z}) = P(\mathbf{z})P(\mathbf{x}|\mathbf{z}) = P(\mathbf{z}) \prod_{i=1}^{D} P(x_i|\mathbf{z}) \tag{1}$$

- Let's have a look at a naive Bayes model. [Mur12, p. 322].
- And a latent variable model [Mur12, p. 345].

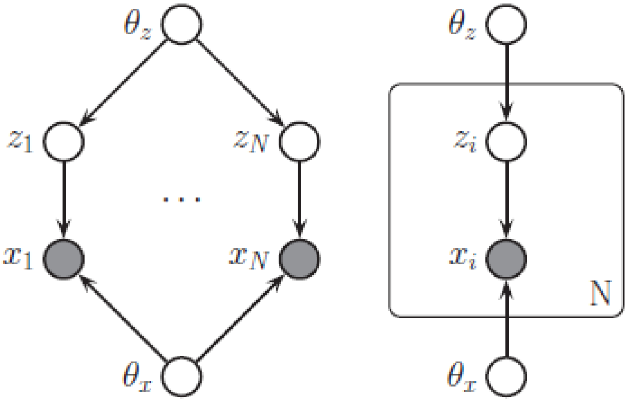
Naive Bayes





$$P(\pi, Y, X, \theta) = P(\pi) \left[\prod_{j=1}^{D} P(\theta_{cj}) \right] \left\{ \prod_{i=1}^{N} \left[P(Y_i|\pi) \prod_{j=1}^{D} P(X_{ij}|Y_i, \theta_{cj}) \right] \right\}$$

A model with latent variables

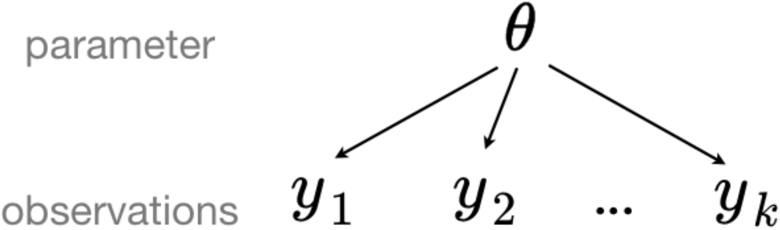


$$P(\theta_x, x, z, \theta_z) = P(\theta_x)P(\theta_z)\prod_{i=1}^{N} P(z_i|\theta_z)P(x_i|z_i, \theta_x)$$

Hierarchical Linear Regression

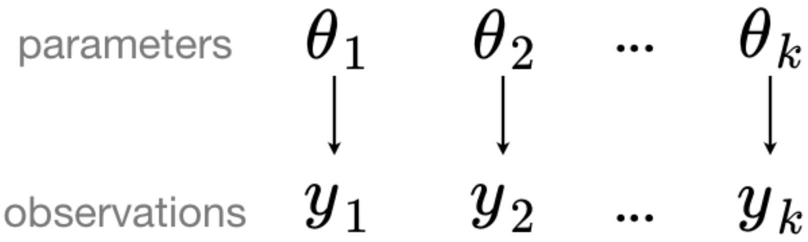
Here's a nice example of using Bayesian networks to represent different approaches to a linear regression problem where there is extra 'structure'.

Standard regression (abbreviated)



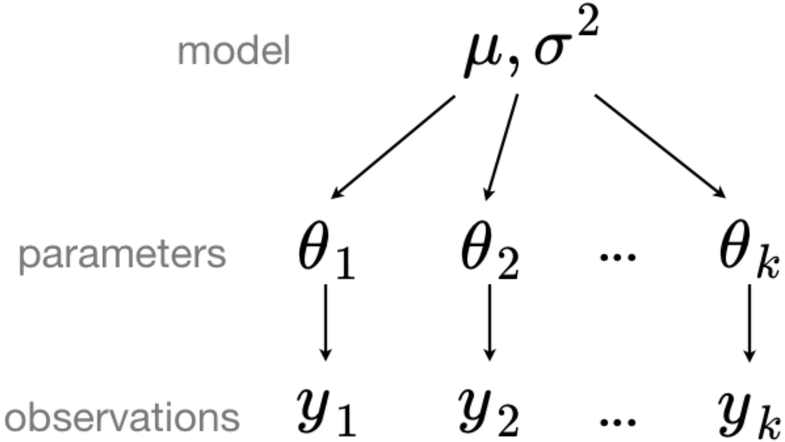
$$P(\theta, y) = P(\theta) \prod_{i=1}^{k} P(y_i | \theta)$$

Separate regressions (abbreviated)



$$P(\theta, y) = \prod_{i=1}^{k} P(y_i|\theta_i)P(\theta_i)$$

Hierarchical regression (abbreviated)



$$P(\theta, y, \mu, \sigma^2) = P(\mu, \sigma^2) \prod_{i=1}^{k} P(y_i|\theta_i) P(\theta_i|\mu, \sigma^2)$$

Christopher M. Bishop.
Pattern Recognition and Machine Learning.
Springer, 2006.

Garrett Bernstein and Daniel R Sheldon.
Differentially private Bayesian linear regression.
In *Advances in Neural Information Processing Systems*, pages 525–535, 2019.

Kevin Murphy.
Machine learning: A probabilistic perspective.
MIT Press, 2012.