This document gives a cursory overview of Probabilistic Graphical Networks. The material has been gleaned from different sources. I make no claim to original authorship of this material.

Bayesian Graphical Models are not really models, but are in fact a tool for describing conditional independencies between random variables in a joint probability distribution, and a tool for visualizing inference algorithms for a particular probability distribution. Examples of these are dynamic programming, Markov Chain Monte Carlo (MCMC).

Exploiting the properties of the graph allow for efficient inference. This is the main reason why PGMs are useful. One can go from a specified acyclic graph to a joint probability distribution, or from a known distribution to a directed acyclic graph.

I begin with a few definitions.

Theorem 0.1 (General Multiplication Rule ("multi-mult")) *For any n event A_1,A_2,...,A_n on probability space Ω,*

$$P(\bigcap_{i=1}^{n} A_i) = P(A_1 \cap A_2 \cap ... \cap A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2)...P(A_n|A_1 \cap A_2... \cap A_{n-1})$$

That is, the probability of a series of multiple events can be computed by calculating the product of a series of conditional probabilities.

Definition 0.2 (Probabilistic Graphical Network (PGN)) *A PGN (sometimes called a Bayesian Network, or Belief Network, or PGM—probabilistic graphical model)* $\mathbf{B}$ *is an annotated directed acyclic graph (DAG) that represents a joint probability distribution for a set of random variables* $\mathbf{V}$*.*

The network is defined by a graph and a parameter set:

$$\mathbf{B} = \langle G, \Theta \rangle$$

where G is the DAG with nodes X_1,X_2, ... , X_n, which represent random variables, and where the edges (lines linking nodes) represent the dependencies between variables.

The graph G encodes independence assumptions. Each variable X_i is independent of its nondescendants given its parents in graph G. The second component of parameters is Θ. This set contains the parameter

$$\theta_{x_i|\pi_i} = P_{\mathbf{B}}(x_i|\pi_i)$$

for every realization x_i of X_i conditioned on π_i, which is the set of parents of X_i in graph G.
B *defines a unique joint probability distribution (JPDF) over* **V** *such that:*

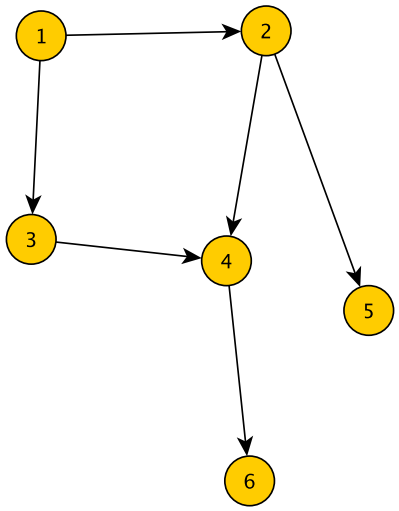
$$P_{\mathbf{B}}(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P_{\mathbf{B}}(X_i|\pi_i) = \prod_{i=1}^{n} \theta_{X_i|\pi_i}$$

A Bayes Network is so-called not because it necessarily uses Bayesian statistics (although it can), but because it uses Bayes Theorem to calculate conditional probabilities (and thus requires summation or integration).

In general, the full summation (integration) over discrete (continuous) variables is called *exact inference* and known to be an NP-hard problem.
An example of a PGM is given by the following probability distribution and graph:

$$\mathrm{P}(X_1, ..., X_6) = \prod_{i=1}^{n} \mathrm{P}_{\mathbf{B}}(X_i|\pi_i) = p(X_1)p(X_2|X_1)p(X_3|X_1)p(X_4|X_2, X_3)p(X_5|X_2)p(X_6|X_4)$$

Figure 1: An example DAG



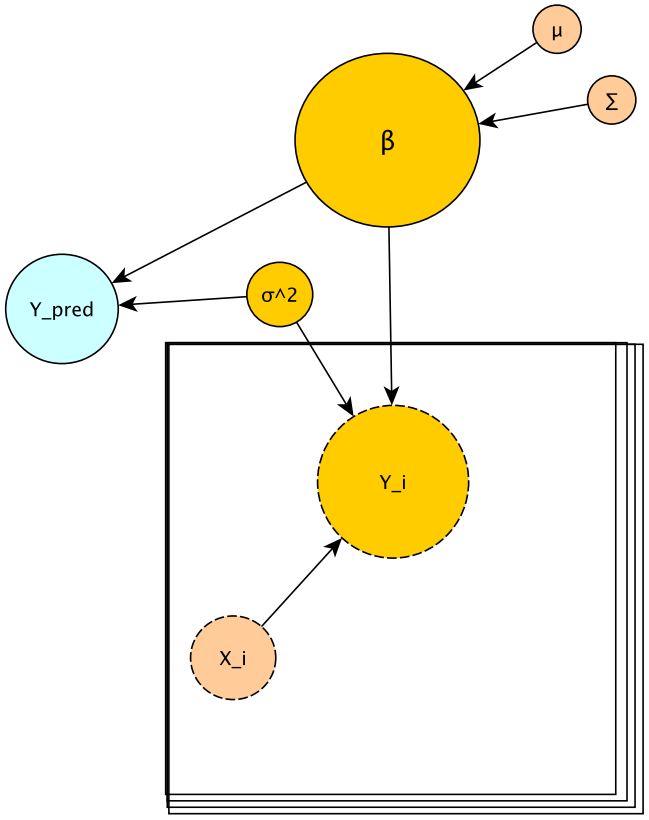
*Note that the arrow structure does *not* necessarily imply dependence. Only *independence* can be read off a graph. Thus, there are often multiple ways to graph the same probability distribution.

Approximate inference to find the distribution of various probabilities is accomplished via stochastic sampling methods. A variety of Markov chain Monte Carlo (MCMC) techniques, including *Gibbs sampling* and the *Metropolis-Hastings algorithm*, are commonly used for approximate inference. There are other methods for approximate inference such as *loopy belief propagation* and *variational methods*.

Types of Graphical Connections:

- Direct Connection
- Serial Connection
- Diverging Connection (common cause)
- Converging Connection (common effect)

Figure 2: An example of a PGN for Bayesian Linear Regression



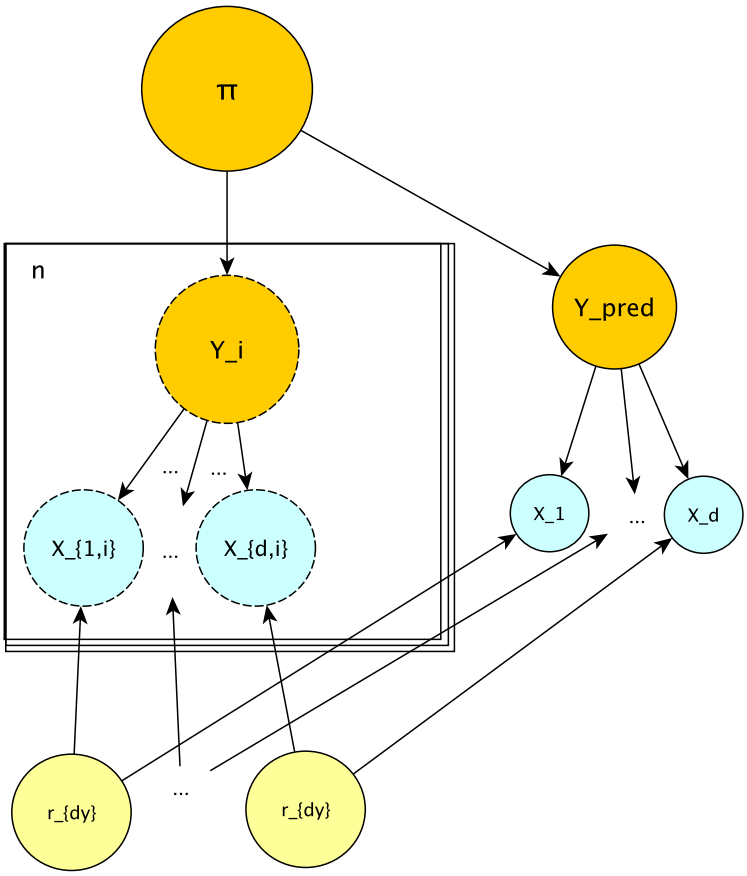
Another example of PGN use is in *Bayesian* Naïve Bayes Classification.
A quick digression: a standard Naïve Bayes Classifier is formed from observed categorical data Y_i in some finite set, with each observation having corresponding d-dimensional "feature" vector $X^{(i)} = \{X_1^{(i)}, X_2^{(i)}, \ldots, X_d^{(i)}\}$. This is a basic machine learning technique for classification. The reason it is "naïve" is that the model assumes that each component of the feature variable, $X_j^{(i)}$, is independent of the other components, conditional on Y_i. This means that for a single observation tuple (Y,X) with fixed observation index i,

$$P_\theta(Y_i, X^{(i)}) = \text{ (the "Bayes" part) } \frac{P_\theta(Y_i)P_\theta(X^{(i)}|Y_i)}{P_\theta(X^{(I)})} \propto \text{ ("naïvely") } P_\theta(Y_i)\prod_{j=1}^{d} P_\theta(X_j^{(i)}|Y_i)$$

The classification occurs by estimating θ from the data, choosing a probability model for Y and each term of X, and finding the maximum *a posteriori* (MAP) estimate of Y: that is, the value of Y that maximizes the above expression. Turning the subscript θ into a conditioning term, we can put a prior on θ and get a posterior distribution. For a categorical scenario, denote $\mathrm{P}(\mathrm{y}|\theta) = \pi(y)$ $= (\pi(1), ..., \pi(m))$, and $\mathrm{P}(\mathrm{x}_j|\mathrm{Y}{=}\mathrm{y},\theta) = r_{jy}(x_j)$, with $\theta = (\pi, \{r_{yj}\}_{j=1}^{d})$.

The next step is to choose priors. Starting with π, we have $\mathrm{P}(\pi) \sim \mathrm{Dirichlet}(\pi|\boldsymbol{\alpha})$, a multinomial generalization of the Beta distribution. Then $\mathrm{P}(r_{y=k,j}) = \mathrm{Dirichlet}(r_{y=k,j}|\boldsymbol{\beta})$. Assuming marginal independence among components of θ, we have $\mathrm{P}(\theta) = \mathrm{P}(\pi)\prod_{j=1,k=1}^{d,m} P(r_{k=k,j})$. We would like to know the appropriate Y for X. So for new data we want the predictive distribution of the probability of Y given X and the data D. The PGN for this model can be written:

Figure 3: An example of a PGN for Bayesian Naïve Bayes Classification



Definition 0.3 (Ancestral Set) *Let* $\mathbf{X}$ *be a set of nodes in network* N*. Then the* **ancestral set** *an*($\mathbf{X}$) *of* $\mathbf{X}$ *consists of all nodes in* $\mathbf{X}$ *and all the ancestors of their ancestors.*

If an($\mathbf{X}$) $= \mathbf{X}$ *then we say* $\mathbf{X}$ *is "ancestral."*

Definition 0.4 (Leaf Node) *A leaf node is a node without any children nodes.*

Definition 0.5 (Complete DAG) *A Complete graph has directed edges between each of the vertices.*

Definition 0.6 (Blocked Path) *A path between nodes* X *and* Y *is blocked by set* $\mathbf{Z}$ *if*

1. *The path contains a node* Z *that is in* $\mathbf{Z}$ *and the connection at* Z *is either serial or diverging.*
2. *Or, the path contains a node* W *such that* W *and its descendants are not in* $\mathbf{Z}$ *and the connection at* W *is a converging connection (collision node).*

Definition 0.7 (d-separation) *Two nodes* X *and* Y *are* $\mathbf{d-separated}$ *by set* $\mathbf{Z}$ *if all paths between* X *and* Y *are blocked by* $\mathbf{Z}$*.*

If this is so, then $X \perp Y \mid Z$*.*

*note: the proof of the fact that X ⊥Y | Z (given the usual set of assumptions—acyclicity, etc.) is shown in the N.L. Zhang ppt., Introduction to Bayesian Networks: Lecture 3, pp. 25-26. This is the so-called **Global Markov Property** of Bayes Networks.

For a given graph G we want to know if X_i is independent of X_j given node C. In the example below, we are looking at whether any two given nodes are independent given X_3, the shaded node.

Figure 4: An example of a PGN conditioned on node 3 (red dots are annotated blockages)

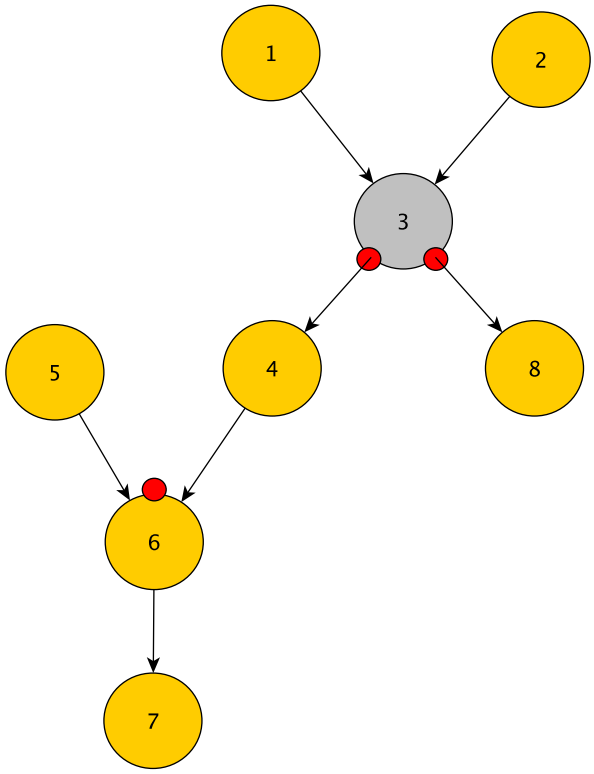


Table 1: Are nodes i and j conditionally independent given node 3?

i	j	d-sep?
1	4	yes
1	2	no
4	5	yes
4	7	no
4	8	yes
4	6	no
2	7	yes
2	5	yes
5	8	yes

Definition 0.8 (Local Markov Property) *For a PGM, a variable X is independent of all its non-descendants, given its parents.*

*note: when constructing a causal network in the fashion of a PGM, we are implicitly assuming that causality implies the local Markov property.

Definition 0.9 (Markov Blanket) *A Markov Blanket (MB) of a node X is the set the parents of X, the children of X, and the parents of the children of X.*

A corollary to this definition is that in a PGM, a variable X is conditionally independent of all other remaining variables, given the known values of its Markov Blanket (as the MB d-separates X from all other nodes in the network).