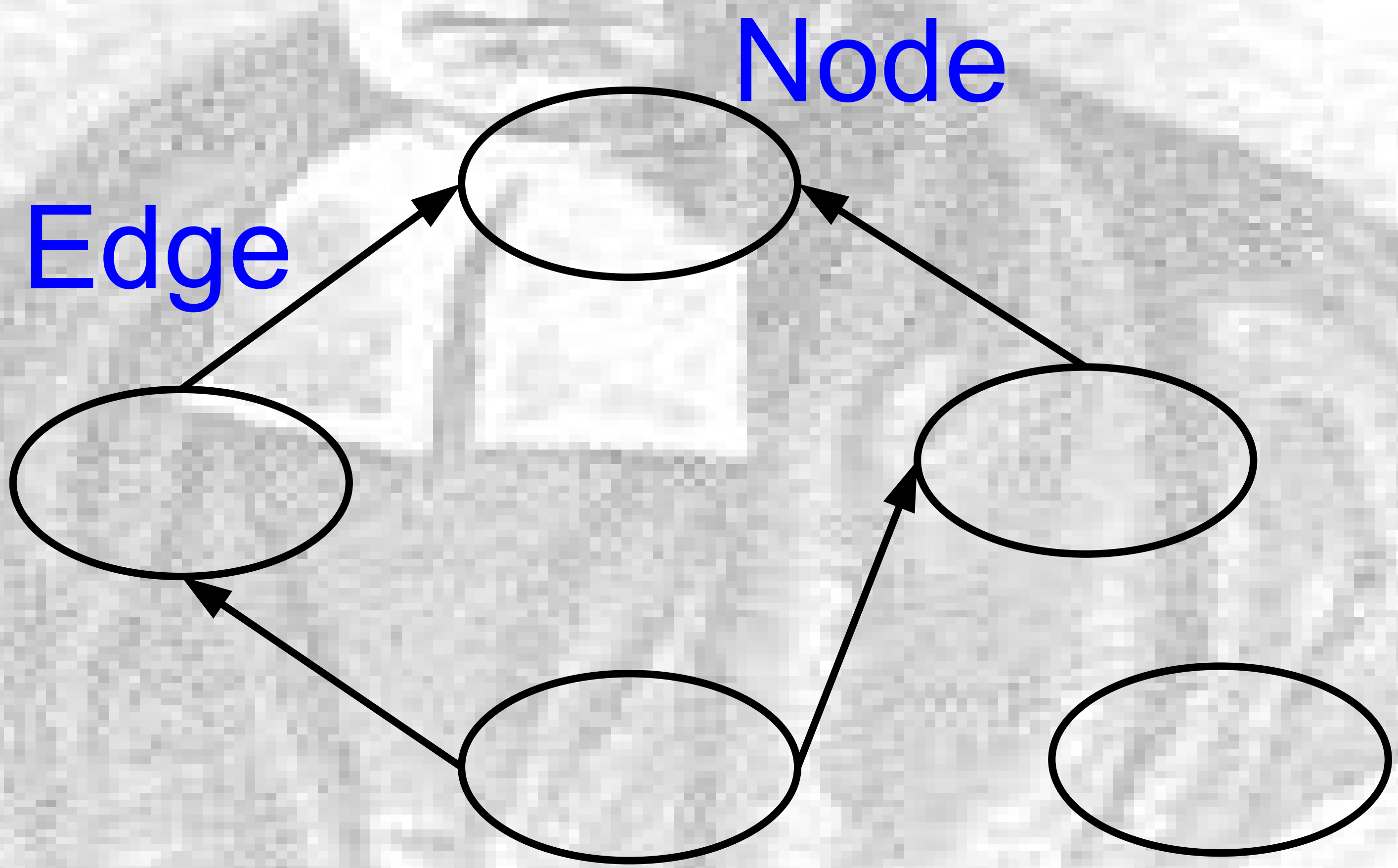
Graphical Models

Writing Models

- Some statistical models can be complex
- The Bayesian approach makes it easier to fit the models
  - as we will see later
- We can represent our models as pictures
  - graphs
- Also helps the model fitting

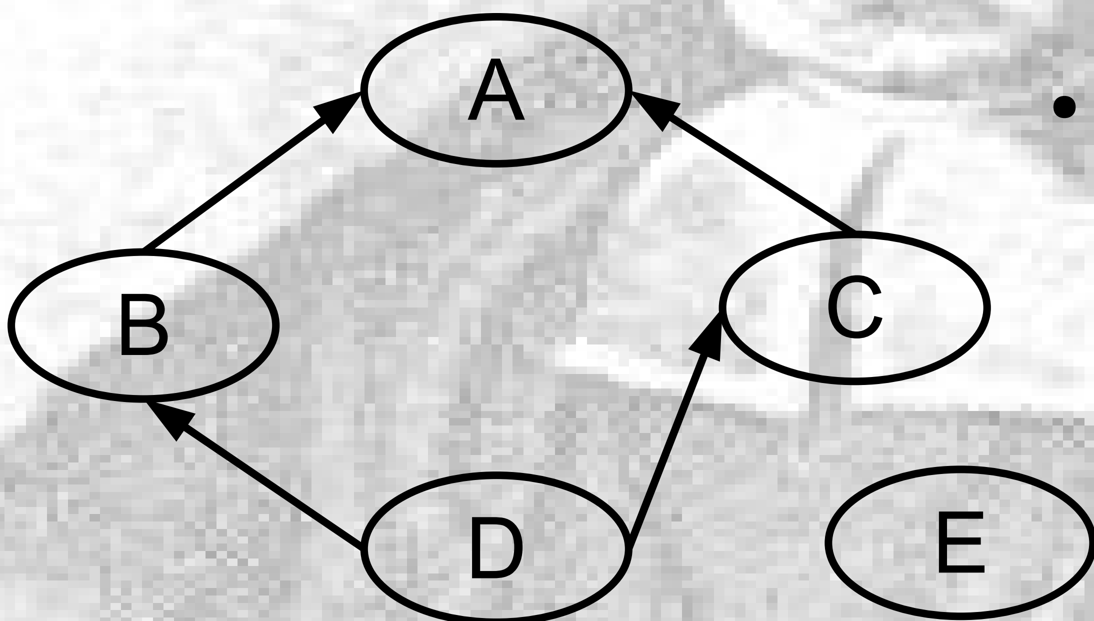
Example: Pictorial Representation of a Graph



Attributing Graphs

- The structure (topology) of a graph can be used to represent a wide range of systems.
- Attaching attributes both to the nodes and the edges of a graph allows complex systems to be modelled.

Nodes (ovals) and Quantities

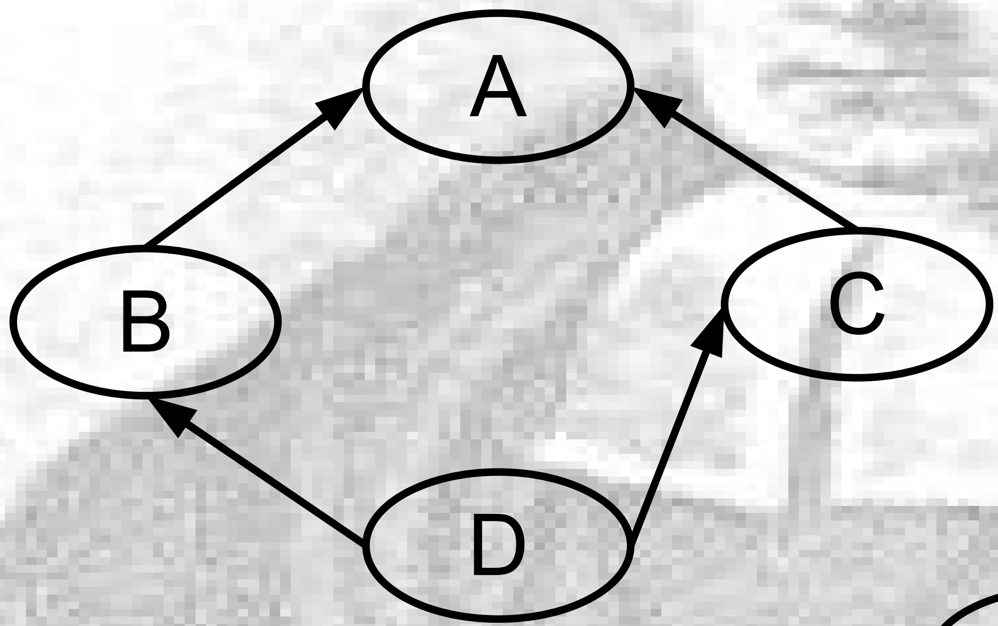


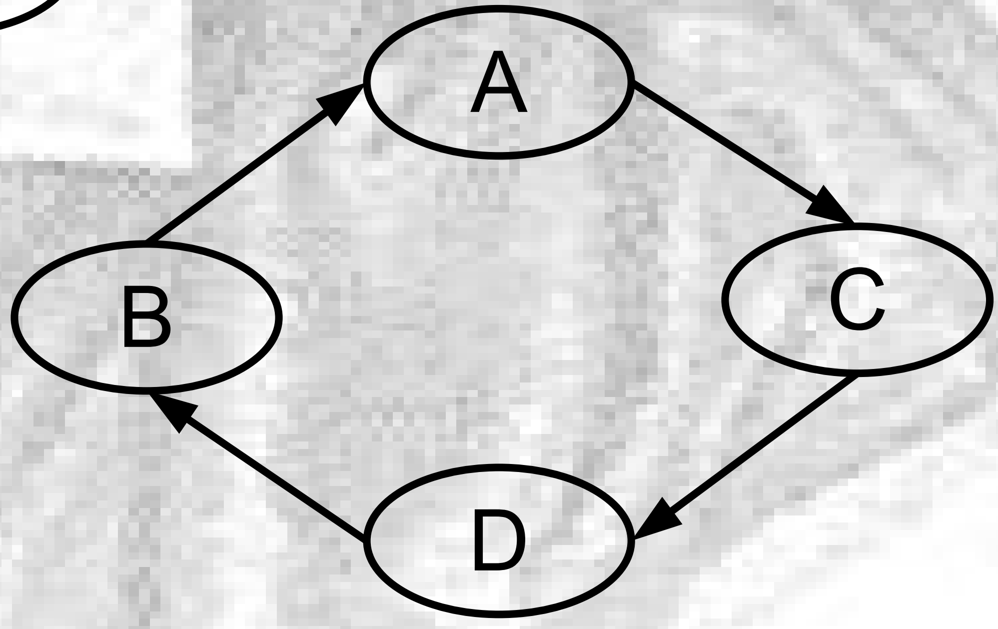
- Nodes in graph have labels.
- Quantities in statistical model have names.

Edges (arrows) and Relations

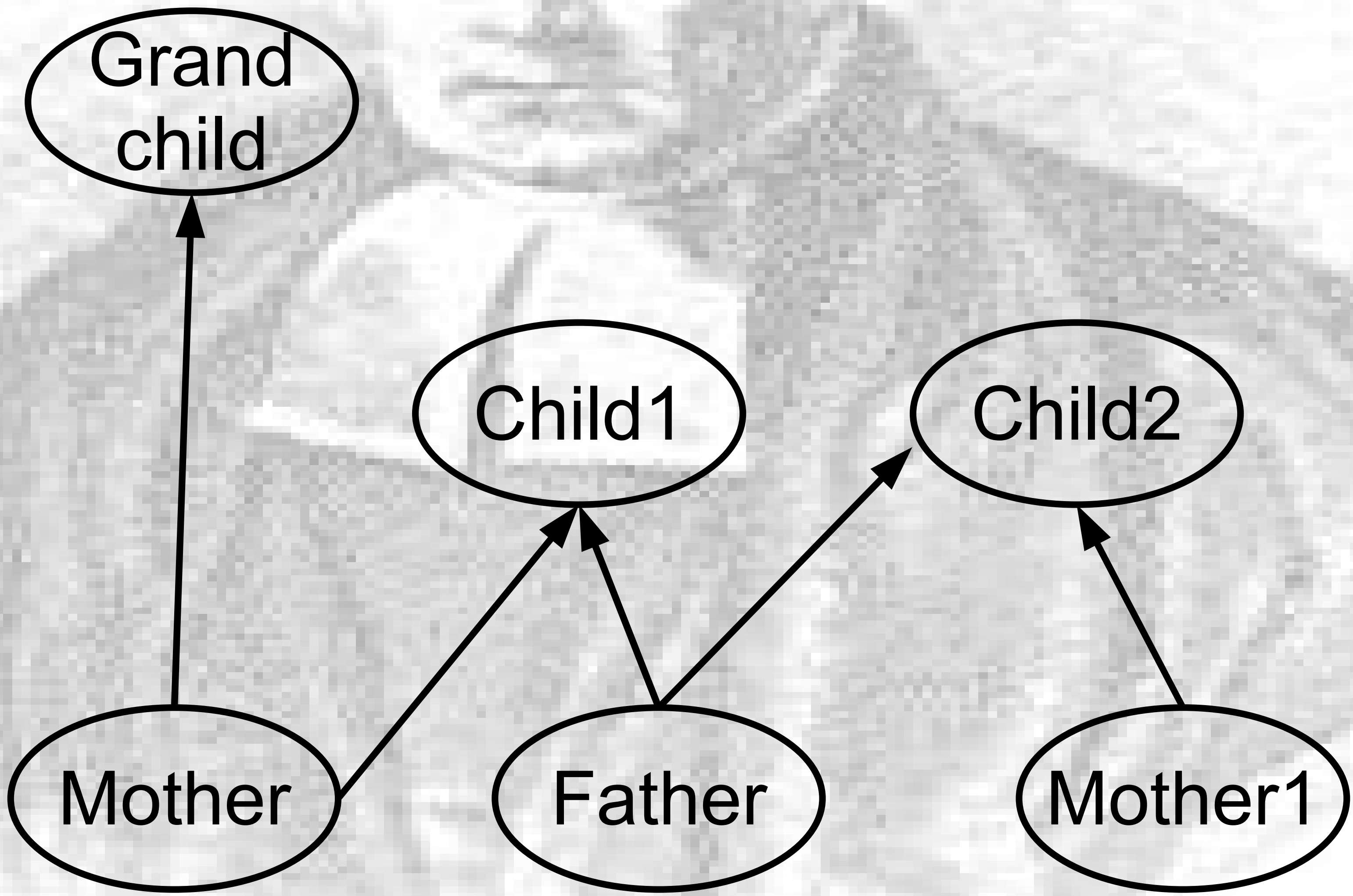
- Edges in a graph can have attributes, different types of edge can be used to describe different types of relations.
- A very important edge attribute is direction, represented by an arrow head.
- Edges can have no arrow head (undirected), a single arrow head (directed) or two opposed arrowheads (bi-directional).

Directed Graphs and DAGs

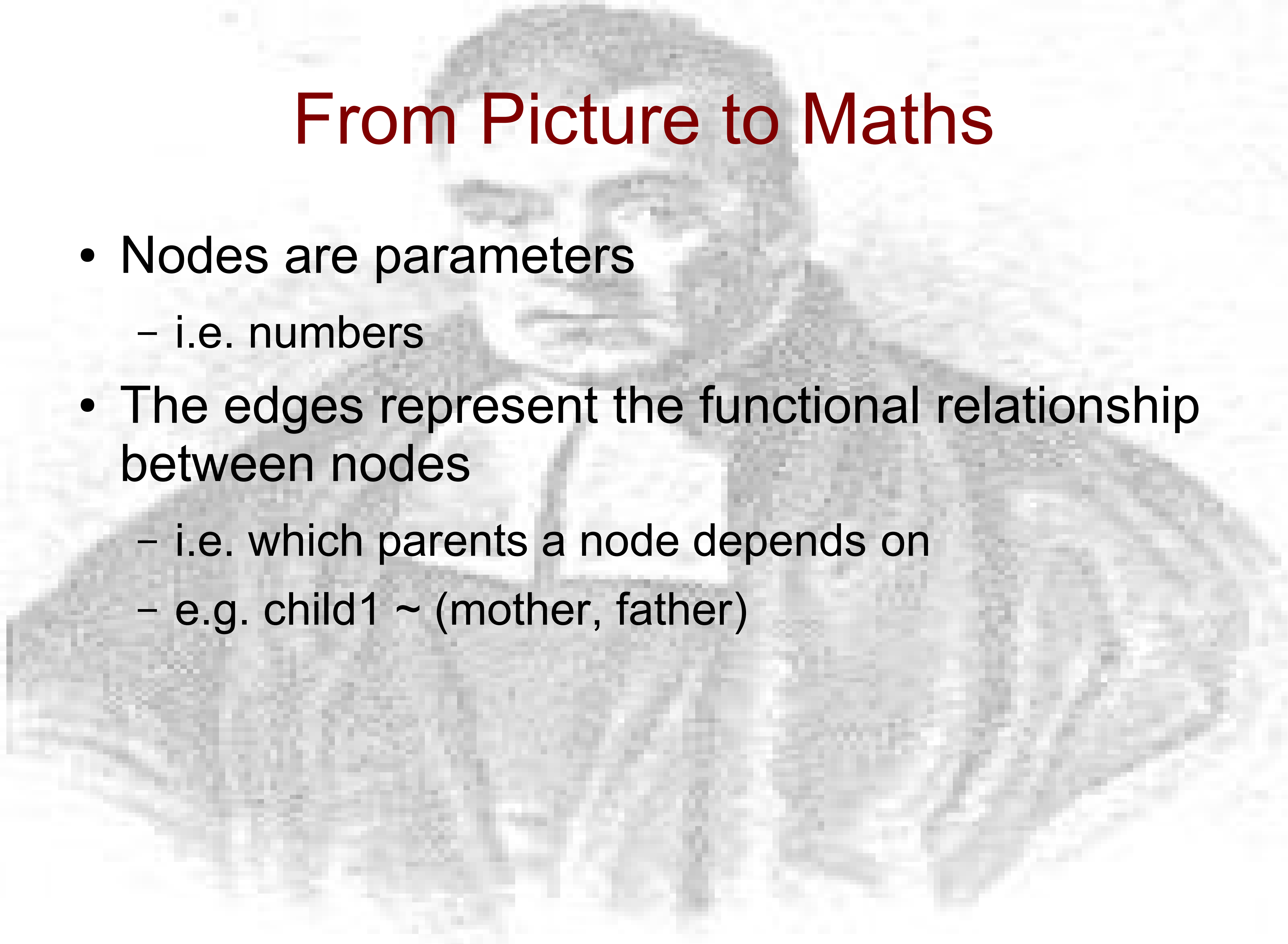




Parents and Children



From Picture to Maths



- Nodes are parameters
  - i.e. numbers
- The edges represent the functional relationship between nodes
  - i.e. which parents a node depends on
  - e.g. child1 ~ (mother, father)

Probability: a node attribute

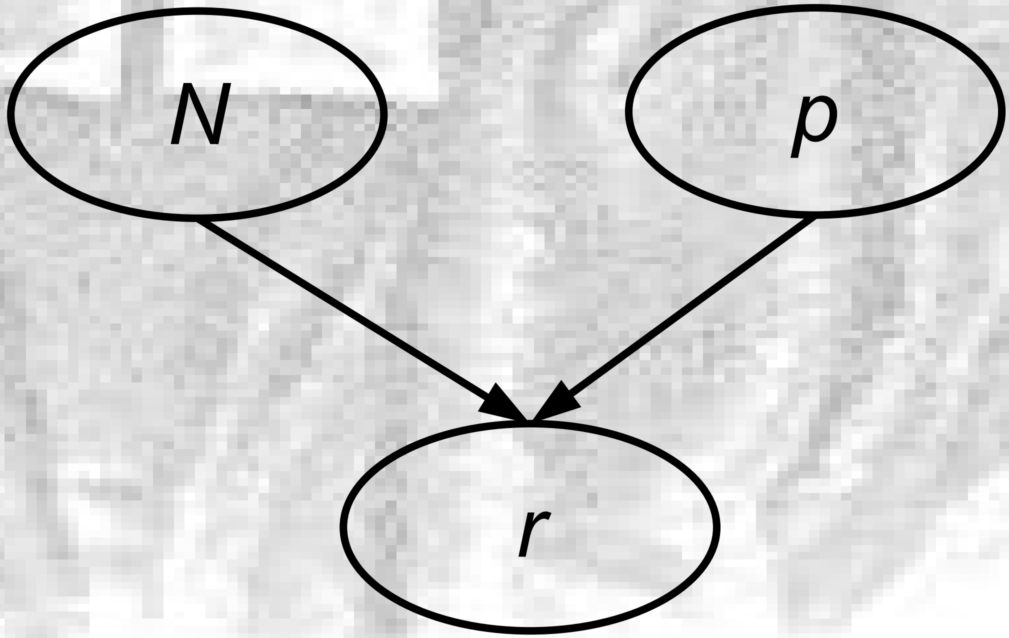
- Can associate a probability distribution with each node in the graph
- The node variable is distributed as the probability distribution.
- The parameters of the distribution are the parents of the node.
  - P(child1|mother, father)

Example: Binomial Distribution

- Probability of r successes out of N trials
  - r ~ Bin(N, p)

$$Pr(N=n|p)=\frac{N!}{N!(N-n)!}p^n(1-p)^{N-n}$$

- Or, in pictures



Graphs to Models

- As well as describing a model, graphs can be used to calculate the posterior probability of a model
  - product of the probabilities for each node
- Probability of each node: likelihood
- Parents of a node: node's prior

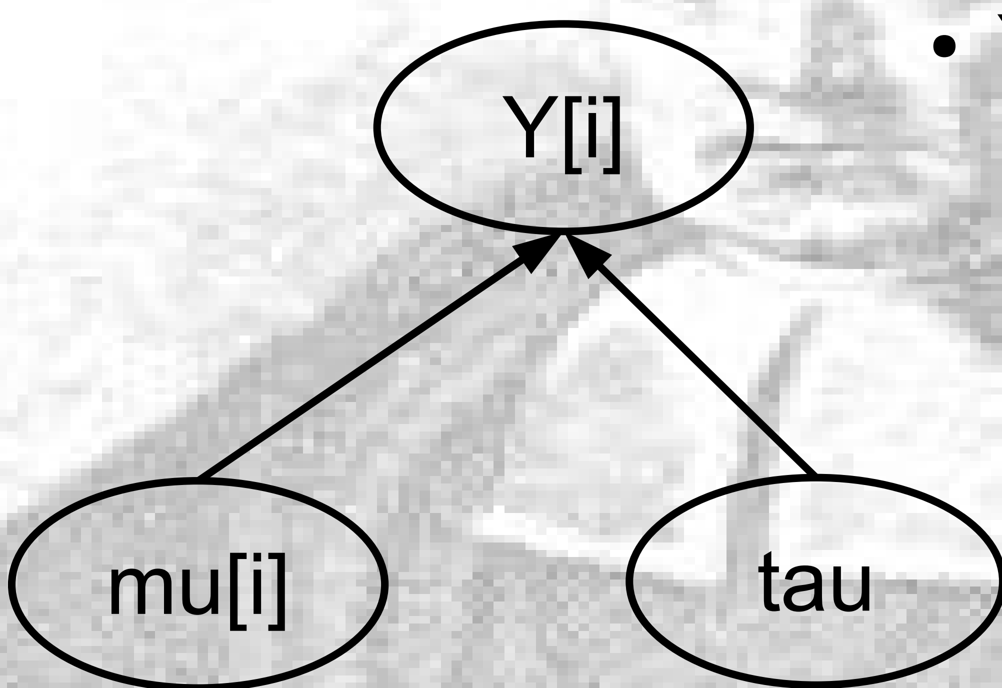
Some Connections

- Joint probability function has a product form: represent as a graph
- BUGS language: a description of a graph
- Both DoodleBUGS and the BUGS language were developed to describe probabilistic models.

Doodle and the BUGS Language

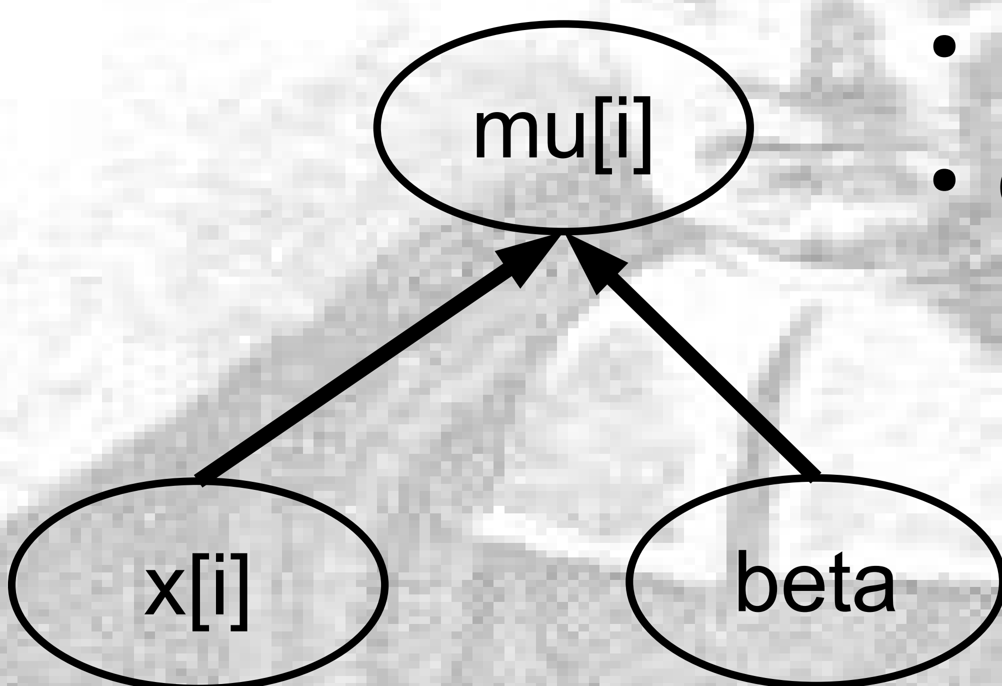
- Three elements:
  - stochastic relations
  - logical relations
  - repetition

Stochastic Relations



- Y[i] ~ P(mu[i], tau)

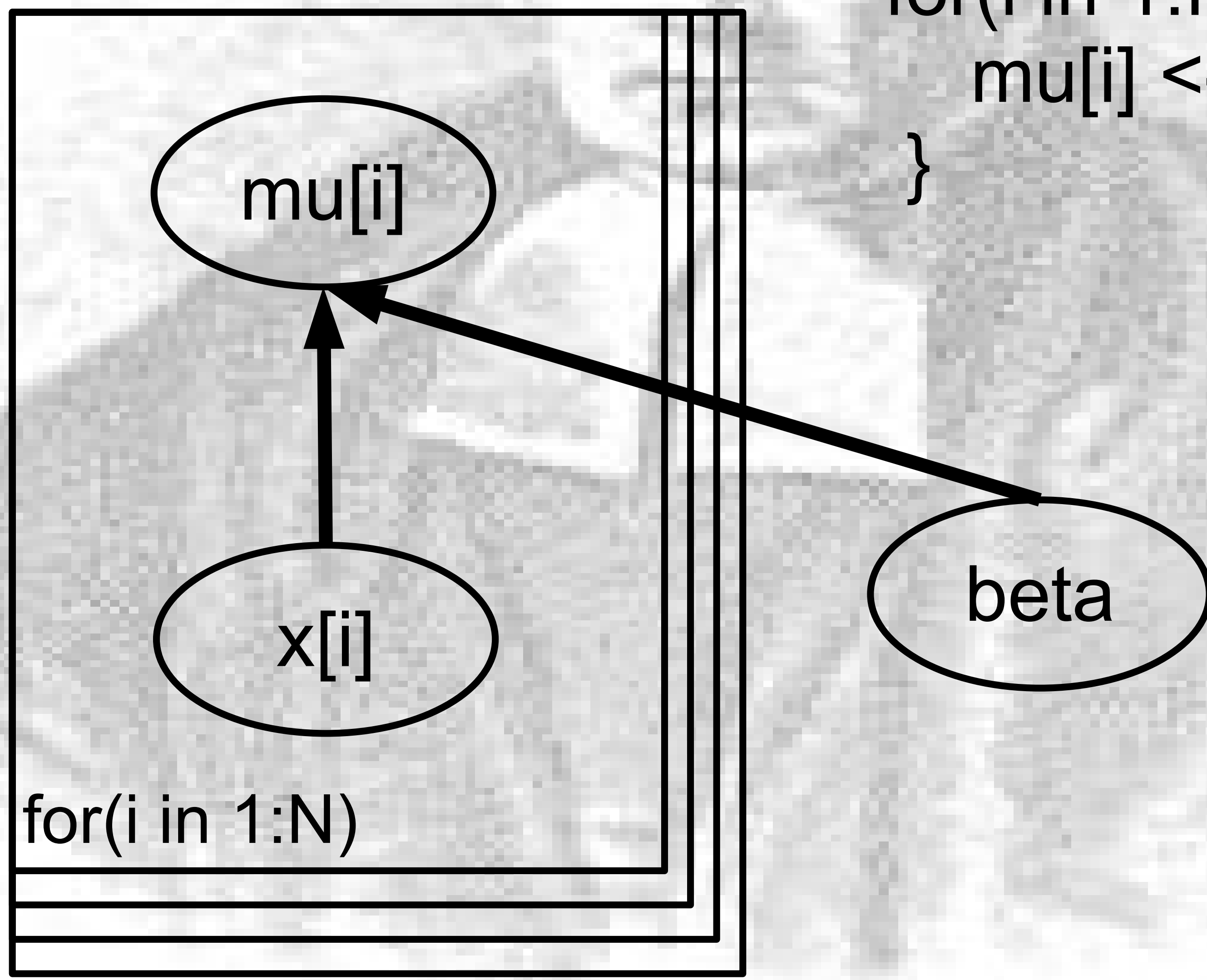
Logical Relations



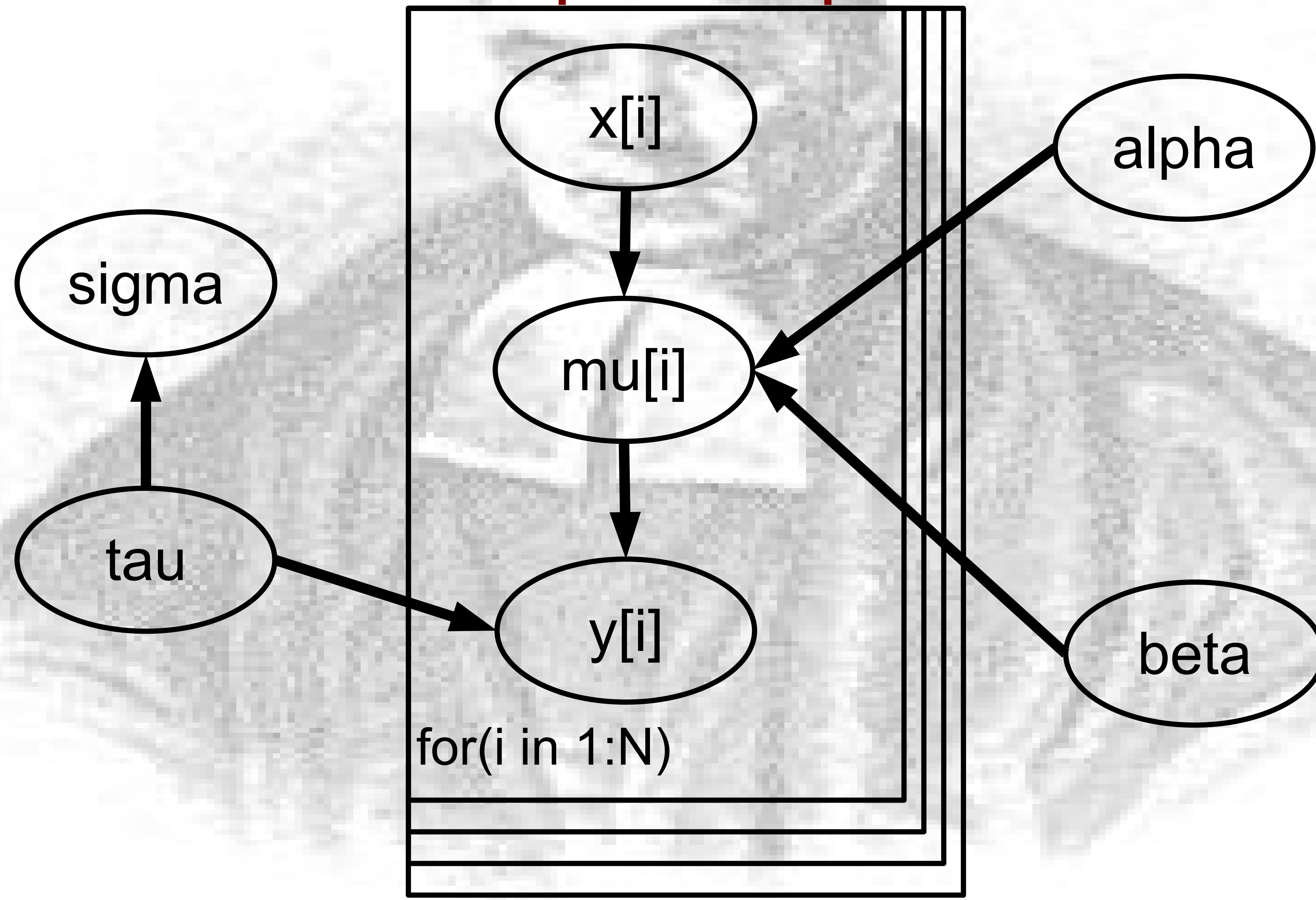
- mu[i] <- f(x[i], beta)
- e.g. mu[i] = x[i]*beta

Repitition

- for(i in 1:N) {
      mu[i] <- f(x[i], beta)
  }



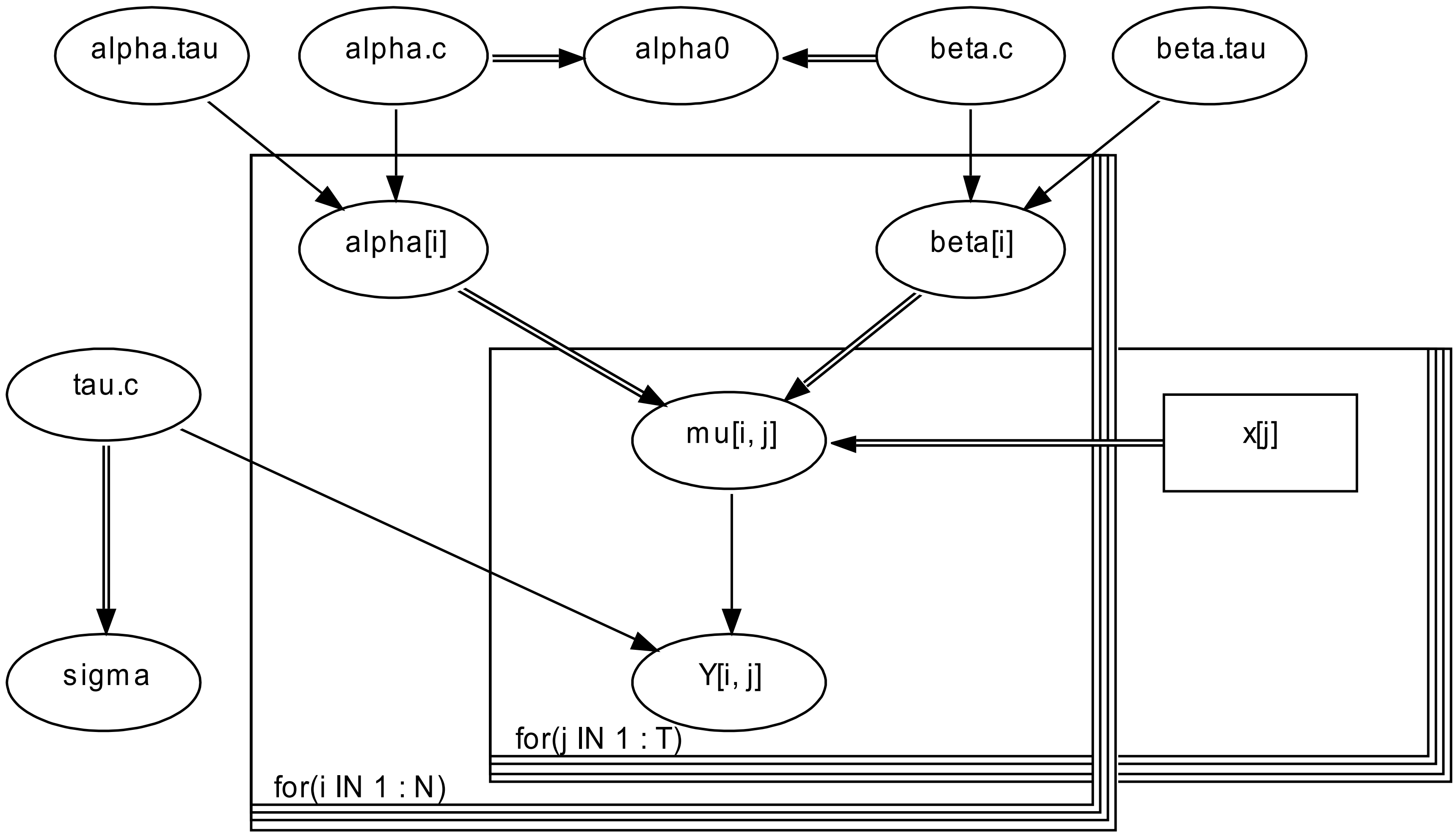
Line: a Simple Graphical Model



Line described in BUGS language

```
model{
    for( i in 1 : N ) {
        Y[i] ~ dnorm(mu[i],tau)
        mu[i] <- alpha + beta * (x[i] - xbar)
    }
    tau ~ dgamma(0.001,0.001)
    sigma <-1 / sqrt(tau)
    alpha ~ dnorm(0.0,1.0E-6)
    beta ~ dnorm(0.0,1.0E-6)
}
```

Doodle for Rats Model



BUGS language for Rats model

```
model{
     for( i in 1 :N) {
      for( j in 1 : T ) {
        Y[i , j] ~ dnorm(mu[i , j],tau.c)
        mu[i,j] <- alpha[i] + beta[i] * (x[j] - xbar)
      }
      alpha[i] ~ dnorm(alpha.c, alpha.tau)
      beta[i] ~ dnorm(beta.c, beta.tau)
      }
     tau.c ~ dgamma(0.001,0.001)
     sigma <-1 / sqrt(tau.c)
     alpha.c ~ dnorm(0.0,1.0E-6)
     alpha.tau ~ dgamma(0.001,0.001)
     beta.c ~ dnorm(0.0,1.0E-6)
     beta.tau ~ dgamma(0.001,0.001)
     alpha0 <-alpha.c -xbar * beta.c
}
```