Bayes Rule

\[ p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)} \]

\[ p(parameters \mid data) = \frac{p(data \mid parameter)p(parameters)}{p(data)} \]

\[ p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{p(D)} \]

\[ p(\theta \mid D) = \frac{\int p(D \mid \theta)p(\theta) \, d\theta}{\int p(D \mid \theta)p(\theta) \, d\theta} \]
What do we want to know?

1) best-fitting parameters of a model

\[ p(\theta \mid D, M) = \frac{p(D \mid \theta, M)p(\theta, M)}{p(D, M)} \]

2) which model best accounts for the data

\[ \frac{p(M_{(1)} \mid D)}{p(M_{(2)} \mid D)} = \frac{p(D \mid M_{(1)})}{p(D \mid M_{(2)})} \frac{p(M_{(1)})}{p(M_{(2)})} \]
What do we want to know?

1) best-fitting parameters of a model

\[ p(\theta | D, M) = \frac{p(D | \theta, M)p(\theta, M)}{p(D, M)} \]

traditional method: nonlinear optimization to maximize likelihood

2) which model best accounts for the data

\[ \frac{p(M_{(1)} | D)}{p(M_{(2)} | D)} = \frac{p(D | M_{(1)}) p(M_{(1)})}{p(D | M_{(2)}) p(M_{(2)})} \]

traditional method: nested and nonnested model comparison
What do we want to know?

1) best-fitting parameters of a model

\[ p(\theta \mid D, M) = \frac{p(D \mid \theta, M)p(\theta, M)}{\int p(D \mid \theta, M)p(\theta, M) d\theta} \]

2) which model best accounts for the data

\[ \frac{p(M_{(1)} \mid D)}{p(M_{(2)} \mid D)} = \frac{\int p(D \mid \theta, M_{(1)})p(\theta, M_{(1)}) d\theta}{\int p(D \mid \theta, M_{(2)})p(\theta, M_{(2)}) d\theta} \frac{p(M_{(1)})}{p(M_{(2)})} \]
one challenge of doing Bayesian Statistical Analysis is that the solutions require solving complex integrals

\[ p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{\int p(D \mid \theta)p(\theta)} \]

\[ p(\theta \mid D) = \int p(\theta, \phi \mid D) d\phi \]

\[ p(\theta \mid D) = \int p(\theta \mid \phi, D)p(\phi) d\phi \]

\[ E[f(\theta)] = \int f(\theta)p(\theta \mid D) d\theta \]
Consider this integral:

\[ \int \theta p(\theta \mid D) d\theta \]

\[ E[\theta \mid D] = \int \theta p(\theta \mid D) d\theta \]
consider this integral:

$$ \int \theta p(\theta \mid D) d\theta $$

$$ E[\theta \mid D] = \int \theta p(\theta \mid D) d\theta $$

how can we evaluate this?
- analytically – hard and often impossible
- numerical integration techniques – inefficient
- Monte Carlo methods – often preferred or only method
Simple Monte Carlo Integration

\[ E[\theta | D] = \int \theta p(\theta | D) d\theta \]

Monte Carlo simulation of an integral ...
Simple Monte Carlo Integration

\[ E[g(\theta) \mid D] = \int g(\theta)p(\theta \mid D)d\theta \]

\[ p(\theta \mid D) \]

\[ \theta^{(1)} \quad \theta^{(2)} \quad \theta^{(3)} \quad \theta^{(4)} \quad \theta^{(5)} \quad \ldots \]

\[ E[g(\theta) \mid D] = \int g(\theta)p(\theta \mid D)d\theta \approx \sum_{j} g(\theta^{(j)}) \frac{1}{N} \]

Monte Carlo simulation of an integral ...
Simple Monte Carlo Integration

\[ E[\theta \mid D] = \int \theta p(\theta \mid D) d\theta \]

Of course, this assumes you can create an “engine” that spits out independent samples from a distribution ...

We’ve talked about some such “engines”, line rand() or randn() or other matlab random number routines
Simple Monte Carlo Integration

\[ E[\theta \mid D] = \int \theta p(\theta \mid D) d\theta \]

there are random number generators for “simple” univariate distributions: uniform, normal, t, F, exponential, weibull, gamma

some for multivariate: normal, t
Simple Monte Carlo Integration

$$E[\theta \mid D] = \int \theta p(\theta \mid D) d\theta$$

But independent sampling from a posterior density $p(\theta \mid D)$ is usually not feasible (and usually impossible) ...

WHY? Keep in mind that in the most general case $p(\theta \mid D)$ can be arbitrarily complex and have many many parameters
Simple Monte Carlo Integration

\[ E[\theta \mid D] = \int \theta p(\theta \mid D) d\theta \]

But independent sampling from a posterior density \( p(\theta \mid D) \)

is usually not feasible (or simply impossible) ...

but we can do dependent or autocorrelated sampling ... Monte Carlo Markov Chains
Independent Sampling
\[ P(\theta^{(t)} | \theta^{(1)} ... \theta^{(t-1)}) = P(\theta^{(t)}) \]

because the samples are independent, smaller sample sizes are needed to approximate distributions or integrals

Sampling from a Markov Chain Process
\[ P(\theta^{(t)} | \theta^{(1)} ... \theta^{(t-1)}) = P(\theta^{(t)} | \theta^{(t-1)}) \]

because the samples are dependent, far larger sample sizes are needed to approximate distributions or integrals
Independent Sampling

\[ P(\theta(t) \mid \theta(1) \ldots \theta(t-1)) = P(\theta(t)) \]

First, let’s look at independent sampling ...

see week14.m
Independent Sampling
Sampling from a Markov Chain Process:

$$P(\theta(t) \mid \theta(1) \ldots \theta(t-1)) = P(\theta(t) \mid \theta(t-1))$$

What is a Monte Carlo Markov Chain ... first, let’s see it in action

see week14.m
Monte Carlo Markov Chain Sampling
Independent

Remember:
this is what we
are trying to derive ...

MCMC

this is what we’re deriving it from ...
Remember: this is what we are trying to derive ...

this comes from gamrnd()
Remember: this is what we are trying to derive...

where does this come from?
steps

\( \theta \)

this is my starting point …
I need a starting point because this is a Markov process.

this is my starting point …
I’m going to propose a new point …
I’ll tell you how later …
I’m going to move to that new point probabilistically …
I’m going to move to that new point probabilistically …

If $P(\theta_{\text{proposed}}) > P(\theta_{\text{current}})$
then move to the proposed point with probability 1
I’m going to move to that new point probabilistically …

If $P(\theta_{\text{proposed}}) > P(\theta_{\text{current}})$
then move to the proposed point with probability 1

this could be $P(\theta|D)$ (posterior)
I’m going to move to that new point probabilistically …

If \( P(\theta_{\text{proposed}}) \leq P(\theta_{\text{current}}) \)
then move to the proposed point with probability
\[
P(\theta_{\text{proposed}})/P(\theta_{\text{current}})
\]
otherwise stay where you are
I’m going to move to that new point probabilistically …

\[ p_{move} = \min\left( \frac{P(\theta_{\text{proposed}})}{P(\theta_{\text{current}})}, 1 \right) \]

*Metropolis Algorithm (Metropolis et al., 1953)*
I’m going to move to that new point probabilistically …

\[ p_{move} = \min \left( \frac{P(\theta_{proposed})}{P(\theta_{current})}, 1 \right) \]

Metropolis Algorithm (Metropolis et al., 1953)
I’m going to move to that new point probabilistically …

\[ p_{move} = \min \left( \frac{P(\theta_{\text{proposed}})}{P(\theta_{\text{current}})}, 1 \right) \]

Metropolis Algorithm (Metropolis et al., 1953)
I’m going to move to that new point probabilistically …

\[ p_{move} = \min \left( \frac{P(\theta_{proposed})}{P(\theta_{current})}, 1 \right) \]

Metropolis Algorithm (Metropolis et al., 1953)
I’m going to move to that new point probabilistically …

\[ p_{\text{move}} = \min \left( \frac{P(\theta_{\text{proposed}})}{P(\theta_{\text{current}})}, 1 \right) \]

*Metropolis Algorithm* (Metropolis et al., 1953)
I’m going to move to that new point probabilistically …

$$p_{move} = \min \left( \frac{P(\theta_{proposed})}{P(\theta_{current})}, 1 \right)$$

Metropolis Algorithm (Metropolis et al., 1953)
I’m going to move to that new point probabilistically …

\[ p_{move} = \min\left( \frac{P(\theta_{\text{proposed}})}{P(\theta_{\text{current}})}, 1 \right) \]

Metropolis Algorithm (Metropolis et al., 1953)

How is a new point proposed?
I’m going to move to that new point probabilistically …

\[ p_{\text{move}} = \min\left(\frac{P(\theta_{\text{proposed}})}{P(\theta_{\text{current}})}, 1\right) \]

**Metropolis Algorithm** *(Metropolis et al., 1953)*

How is a new point proposed? e.g., normal, uniform
I’m going to move to that new point probabilistically …

\[ p_{\text{move}} = \min \left( \frac{P(\theta_{\text{proposed}})Q(\theta_{\text{current}} \mid \theta_{\text{proposed}})}{P(\theta_{\text{current}})Q(\theta_{\text{proposed}} \mid \theta_{\text{current}})}, 1 \right) \]

Metropolis-Hastings allows asymmetric proposal distributions

How is a new point proposed?

asymmetric distribution
I’m going to move to that new point probabilistically …

\[
p_{\text{move}} = \min \left( \frac{P(\theta_{\text{proposed}})Q(\theta_{\text{current}} | \theta_{\text{proposed}})}{P(\theta_{\text{current}})Q(\theta_{\text{proposed}} | \theta_{\text{current}})} , 1 \right)
\]

*Metropolis-Hastings allows asymmetric proposal distributions*

Choice of proposal distributions matters a lot to convergence …

*Goldilocks at work*
I’m going to move to that new point probabilistically …

\[
p_{\text{move}} = \min \left( \frac{P(\theta_{\text{proposed}})Q(\theta_{\text{current}} | \theta_{\text{proposed}})}{P(\theta_{\text{current}})Q(\theta_{\text{proposed}} | \theta_{\text{current}})}, 1 \right)
\]

Metropolis-Hastings allows asymmetric proposal distributions

Choice of proposal distributions matters a lot to convergence …

Goldilocks at work

MCMC packages do a lot of the work
Often need to do 10’s or even 100’s of thousands of steps …

And allow a “burn-in” period that’s discarded (the initial walk)
Often need to do 10’s or even 100’s of thousands of steps …

And allow a “burn-in” period that’s discarded (the initial walk)
sometimes, multiple parallel chains are used (each with different starting points)
multiple chains can be used to judge convergence …
e.g., if 4 chains with 4 different starting points all converge
to the same underlying distribution (after deleting the burn-in)
What does this do for us in evaluating Bayes rule?

\[ p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{\int p(D \mid \theta)p(\theta) \, d\theta} \]
What does this do for us in evaluating Bayes rule?

\[
p(\theta | D) = \frac{p(D | \theta)p(\theta)}{\int p(D | \theta)p(\theta) \, d\theta}
\]

Consider the \( p_{\text{move}} \) in the Metropolis algorithm, evaluating \( \theta_{\text{current}} \) and \( \theta_{\text{proposed}} \)

\[
p_{\text{move}} = \min \left( \frac{P(\theta_{\text{proposed}} | D)}{P(\theta_{\text{current}} | D)}, 1 \right)
\]
What does this do for us in evaluating Bayes rule?

\[ p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{\int p(D \mid \theta)p(\theta)\,d\theta} \]

Consider the \( p_{\text{move}} \) in the Metropolis algorithm, evaluating \( \theta_{\text{current}} \) and \( \theta_{\text{proposed}} \)

\[ p_{\text{move}} = \min \left( \frac{p(D \mid \theta_{\text{proposed}})p(\theta_{\text{proposed}})}{\int p(D \mid \theta)p(\theta)\,d\theta}, 1 \right) \]
What does this do for us in evaluating Bayes rule?

\[ p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{\int p(D \mid \theta)p(\theta)\,d\theta} \]

Consider the \( p_{\text{move}} \) in the Metropolis algorithm, evaluating \( \theta_{\text{current}} \) and \( \theta_{\text{proposed}} \)

\[ p_{\text{move}} = \min \left( \frac{p(D \mid \theta_{\text{proposed}})p(\theta_{\text{proposed}})}{\int p(D \mid \theta)p(\theta)\,d\theta}, 1 \right) \]

*these are the really hard parts of the calculation*
What does this do for us in evaluating Bayes rule?

\[
p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{\int p(D \mid \theta)p(\theta) \, d\theta}
\]

Consider the \( p_{\text{move}} \) in the Metropolis algorithm, evaluating \( \theta_{\text{current}} \) and \( \theta_{\text{proposed}} \)

\[
p_{\text{move}} = \min \left( \frac{p(D \mid \theta_{\text{proposed}})p(\theta_{\text{proposed}})}{\int p(D \mid \theta)p(\theta) \, d\theta}, 1 \right)
\]

\[\text{these are the same}\]
What does this do for us in evaluating Bayes rule?

\[
p(\theta | D) = \frac{p(D | \theta) p(\theta)}{\int p(D | \theta) p(\theta) \, d\theta}
\]

Consider the \( p_{\text{move}} \) in the Metropolis algorithm, evaluating \( \theta_{\text{current}} \) and \( \theta_{\text{proposed}} \)

\[
p_{\text{move}} = \min \left( \frac{\int p(D | \theta_{\text{proposed}}) p(\theta_{\text{proposed}}) \, d\theta}{\int p(D | \theta_{\text{current}}) p(\theta_{\text{current}}) \, d\theta}, 1 \right)
\]

why are they the same?
What does this do for us in evaluating Bayes rule?

\[
p(\theta | D) = \frac{p(D | \theta)p(\theta)\int p(D | \theta)p(\theta)\,d\theta}{\int p(D | \theta)p(\theta)\,d\theta}
\]

Consider the \( p_{\text{move}} \) in the Metropolis algorithm, evaluating \( \theta_{\text{current}} \) and \( \theta_{\text{proposed}} \)

\[
p_{\text{move}} = \min\left(\frac{p(D | \theta_{\text{proposed}})p(\theta_{\text{proposed}})}{p(D | \theta_{\text{current}})p(\theta_{\text{current}})}, 1\right)
\]
What does this do for us in evaluating Bayes rule?

$$p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{\int p(D \mid \theta)p(\theta)\,d\theta}$$

Consider the $p_{\text{move}}$ in the Metropolis algorithm, evaluating $\theta_{\text{current}}$ and $\theta_{\text{proposed}}$

$$p_{\text{move}} = \min \left( \frac{p(D \mid \theta_{\text{proposed}})p(\theta_{\text{proposed}})}{p(D \mid \theta_{\text{current}})p(\theta_{\text{current}})}, 1 \right)$$

we’ve made a really really hard problem really really easy (albeit at the cost of doing thousands of simulated moves)
e.g., find the posterior using the Metropolis algorithm

\[ P(p \mid x) = \frac{\binom{N}{x} p^x (1 - p)^{N-x} \frac{p^{a-1}(1 - p)^{b-1}}{Be(a, b)}}{P(x)} \]
e.g., find the posterior using the Metropolis algorithm

\[
P(p \mid x) = \frac{\binom{N}{x} p^x (1 - p)^{N-x} \frac{p^{a-1} (1 - p)^{b-1}}{Be(a,b)}}{\int \binom{N}{x} p^x (1 - p)^{N-x} \frac{p^{a-1} (1 - p)^{b-1}}{Be(a,b)} dp}
\]
e.g., find the posterior using the Metropolis algorithm

\[
P(p | x) = \frac{\binom{N}{x} p^x (1 - p)^{N-x} \frac{p^{a-1}(1 - p)^{b-1}}{Be(a,b)}}{\int \binom{N}{x} p^x (1 - p)^{N-x} \frac{p^{a-1}(1 - p)^{b-1}}{Be(a,b)} dp}
\]

\[
p_{\text{move}} = \min \left( \frac{\binom{N}{x} p^{x_{\text{proposed}}}(1 - p^{x_{\text{proposed}}})^{N-x} \frac{p^{a-1}_{\text{proposed}}(1 - p^{x_{\text{proposed}}})^{b-1}}{Be(a,b)}}{\int \binom{N}{x} p^x (1 - p)^{N-x} \frac{p^{a-1}(1 - p)^{b-1}}{Be(a,b)} dp} , 1 \right)
\]
e.g., find the posterior using the Metropolis algorithm

\[ P(p | x) = \frac{\binom{N}{x} p^x (1-p)^{N-x} \frac{p^{a-1}(1-p)^{b-1}}{\text{Be}(a,b)}}{\int \binom{N}{x} p^x (1-p)^{N-x} \frac{p^{a-1}(1-p)^{b-1}}{\text{Be}(a,b)} dp} \]

\[ p_{move} = \min \left( \frac{\binom{N}{x} p_{\text{proposed}}^x (1-p_{\text{proposed}})^{N-x} \frac{p_{\text{proposed}}^{a-1}(1-p_{\text{proposed}})^{b-1}}{\text{Be}(a,b)}}{\binom{N}{x} p_{\text{current}}^x (1-p_{\text{current}})^{N-x} \frac{p_{\text{current}}^{a-1}(1-p_{\text{current}})^{b-1}}{\text{Be}(a,b)}}, 1 \right) \]
e.g., find the posterior using the Metropolis algorithm

\[
P(p \mid x) = \frac{\binom{N}{x} p^x (1-p)^{N-x} \frac{p^{a-1}(1-p)^{b-1}}{\text{Be}(a,b)}}{\int \binom{N}{x} p^x (1-p)^{N-x} \frac{p^{a-1}(1-p)^{b-1}}{\text{Be}(a,b)} \, dp}
\]

\[
p_{\text{move}} = \min\left(\frac{p_{\text{proposed}}^x (1-p_{\text{proposed}})^{N-x} p_{\text{proposed}}^{a-1} (1-p_{\text{proposed}})^{b-1}}{p_{\text{current}}^x (1-p_{\text{current}})^{N-x} p_{\text{current}}^{a-1} (1-p_{\text{current}})^{b-1}}, 1\right)
\]

taken a problem that involves a difficult integral and turned it into something simpler via simulation
e.g., find the posterior using the Metropolis-Hastings algorithm

\[
P(p | x) = \frac{\binom{N}{x} p^x (1 - p)^{N-x} \frac{p^{a-1}(1 - p)^{b-1}}{\text{Be}(a,b)} \right)}{\int \binom{N}{x} p^x (1 - p)^{N-x} \frac{p^{a-1}(1 - p)^{b-1}}{\text{Be}(a,b)} \, dp}
\]

\[
p_{\text{move}} = \min \left( \frac{p_{\text{proposed}}^x (1 - p_{\text{proposed}})^{N-x} p_{\text{proposed}}^{a-1} (1 - p_{\text{proposed}})^{b-1} Q(p_{\text{current}} | p_{\text{proposed}})}{p_{\text{current}}^x (1 - p_{\text{current}})^{N-x} p_{\text{current}}^{a-1} (1 - p_{\text{current}})^{b-1} Q(p_{\text{proposed}} | p_{\text{current}})}, 1 \right)
\]

may need to use an asymmetric proposal distribution – otherwise, proposed p’s might be less than zero or greater than one
e.g., find the posterior using the Metropolis-Hastings algorithm

\[ P(p \mid x) = \frac{\binom{N}{x} p^x (1 - p)^{N-x} \frac{p^{a-1}(1 - p)^{b-1}}{Be(a,b)}}{\int \binom{N}{x} p^x (1 - p)^{N-x} \frac{p^{a-1}(1 - p)^{b-1}}{Be(a,b)} dp} \]

\[ p_{\text{move}} = \min \left( \frac{p_{\text{proposed}}^x (1 - p_{\text{proposed}})^{N-x} p_{\text{proposed}}^{a-1} (1 - p_{\text{proposed}})^{b-1}}{p_{\text{current}}^x (1 - p_{\text{current}})^{N-x} p_{\text{current}}^{a-1} (1 - p_{\text{current}})^{b-1}}, 1 \right) \]

can also solve the problem of bounded values by means of the prior distribution (i.e., that \( P(p) = 0 \) for \( p < 0 \) and \( p > 1 \)) using a symmetric (normal) proposal distribution
e.g., find the posterior using the Metropolis-Hastings algorithm

\[
P(p \mid x) = \frac{\binom{N}{x} p^x (1 - p)^{N-x} \frac{p^{a-1}(1 - p)^{b-1}}{\text{Be}(a,b)}}{\int \binom{N}{x} p^x (1 - p)^{N-x} \frac{p^{a-1}(1 - p)^{b-1}}{\text{Be}(a,b)} dp}
\]

\[
p_{\text{move}} = \min \left( \frac{p_{\text{proposed}}^x (1 - p_{\text{proposed}})^{N-x} p_{\text{proposed}}^{a-1}(1 - p_{\text{proposed}})^{b-1}}{p_{\text{current}}^x (1 - p_{\text{current}})^{N-x} p_{\text{current}}^{a-1}(1 - p_{\text{current}})^{b-1}}, 1 \right)
\]

can also do parameter transformations to take something that’s in a 0-1 range and make it have an infinite range
e.g., z-transform or logit
see week14.m
what about models with more than one parameter?
what about models with more than one parameter?

\[ P(X \mid \mu, \sigma^2) = \frac{1}{(\sqrt{2\pi\sigma^2})^N} \exp\left( -\frac{\sum (x_j - \mu)^2}{2\sigma^2} \right) \]

\[ P(\mu, \sigma^2 \mid X) = \frac{P(X \mid \mu, \sigma^2) P(\mu, \sigma^2)}{P(X)} \]
what about models with more than one parameter?

\[ P(X \mid \mu, \sigma^2) = \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} \exp \left( \frac{-\sum (x_j - \mu)^2}{2\sigma^2} \right) \]

\[ P(\mu, \sigma^2 \mid X) = \frac{P(X \mid \mu, \sigma^2)P(\mu)P(\sigma^2)}{P(X)} \]

assumes independence of priors

\[ P(\mu, \sigma^2) = P(\mu)P(\sigma^2) \]
what about models with more than one parameter?

\[
P(X \mid \mu, \sigma^2) = \frac{1}{(\sqrt{2\pi\sigma^2})^N} \exp\left(-\sum_j \frac{(x_j - \mu)^2}{2\sigma^2}\right)
\]

\[
P(\mu, \sigma^2 \mid X) = \frac{P(X \mid \mu, \sigma^2)P(\mu)P(\sigma^2)}{P(X)}
\]

NOTE: with Bayesian Graphical Models priors are often dependent by definition
imagine normally distributed data \( X = (x_1, x_2, x_3, \ldots, x_N) \)
what is the **Bayesian** estimate of \( \mu \) and \( \sigma^2 \)?

\[
P(X \mid \mu, \sigma^2) = \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} \exp \left( -\frac{\sum_j (x_j - \mu)^2}{2\sigma^2} \right)
\]

\[
P(\mu, \sigma^2 \mid X) = \frac{P(X \mid \mu, \sigma^2)P(\mu)P(\sigma^2)}{P(X)}
\]

\[
P(\mu, \sigma^2 \mid X) = \frac{1}{P(X)} \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} \exp \left( -\frac{\sum_j (x_j - \mu)^2}{2\sigma^2} \right) \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp \left( -\frac{(\mu - \mu_0)^2}{2\sigma_0^2} \right) \frac{b^a}{\Gamma(a)(\sigma^2)^{a+1}} \exp \left( -\frac{b}{\sigma^2} \right)
\]
imagine normally distributed data $X=(x_1, x_2, x_3, \ldots, x_N)$
what is the **Bayesian** estimate of $\mu$ and $\sigma^2$?

$$P(X \mid \mu, \sigma^2) = \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} \exp\left(\frac{\sum_j (x_j - \mu)^2}{2\sigma^2}\right)$$

$$P(\mu, \sigma^2 \mid X) = \frac{P(X \mid \mu, \sigma^2)P(\mu)P(\sigma^2)}{P(X)}$$

$$P(\mu, \sigma^2 \mid X) = \frac{1}{P(X)} \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} \exp\left(\frac{\sum_j (x_j - \mu)^2}{2\sigma^2}\right) \frac{1}{\left(\sqrt{2\pi\sigma_0^2}\right)^{\frac{1}{2}}} \exp\left(\frac{-(\mu - \mu_0)^2}{2\sigma_0^2}\right) \frac{b^a}{\Gamma(a)(\sigma^2)^{a+1}} \exp\left(-\frac{b}{\sigma^2}\right)$$
imagine normally distributed data $X=(x_1, x_2, x_3, \ldots, x_N)$

what is the **Bayesian** estimate of $\mu$ and $\sigma^2$?

$$P(X \mid \mu, \sigma^2) = \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} \exp\left(-\frac{\sum_j (x_j - \mu)^2}{2\sigma^2}\right)$$

$$P(\mu, \sigma^2 \mid X) = \frac{P(X \mid \mu, \sigma^2) P(\mu) P(\sigma^2)}{P(X)}$$

$$P(\mu, \sigma^2 \mid X) = \frac{1}{P(X)} \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} \exp\left(-\frac{\sum_j (x_j - \mu)^2}{2\sigma^2}\right) \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{(\mu - \mu_0)^2}{2\sigma_0^2}\right) \frac{b^a}{\Gamma(a)\sigma^2_0^{a+1}} \exp\left(-\frac{b}{\sigma^2}\right)$$
imagine normally distributed data \( X=(x_1, x_2, x_3, \ldots, x_N) \)

what is the **Bayesian** estimate of \( \mu \) and \( \sigma^2 \)?

\[
P(X | \mu, \sigma^2) = \frac{1}{\left(\sqrt{2\pi \sigma^2}\right)^N} \exp \left( -\frac{\sum (x_j - \mu)^2}{2\sigma^2} \right)
\]

\[
P(\mu, \sigma^2 | X) = \frac{P(X | \mu, \sigma^2)P(\mu)P(\sigma^2)}{P(X)}
\]

\[
P(\mu, \sigma^2 | X) = \frac{1}{P(X)} \frac{1}{\left(\sqrt{2\pi \sigma^2}\right)^N} \exp \left( -\frac{\sum (x_j - \mu)^2}{2\sigma^2} \right) \exp \left( -\frac{(\mu - \mu_0)^2}{2\sigma_0^2} \right) \frac{b^a}{\Gamma(a)(\sigma^2)^{a+1}} \exp \left( -\frac{b}{\sigma^2} \right)
\]
imagine normally distributed data $X = (x_1, x_2, x_3, \ldots, x_N)$
what is the **Bayesian** estimate of $\mu$ and $\sigma^2$?

$$P(X | \mu, \sigma^2) = \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} \exp\left(-\sum_j (x_j - \mu)^2 \right) \frac{1}{2\sigma^2}$$

$$P(\mu, \sigma^2 | X) = \frac{P(X | \mu, \sigma^2)P(\mu)P(\sigma^2)}{P(X)}$$

$$P(\mu, \sigma^2 | X) \propto \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} \exp\left(-\sum_j (x_j - \mu)^2 \right) \exp\left(-\frac{(\mu - \mu_0)^2}{2\sigma_0^2}\right) \frac{1}{\sigma^2} \exp\left(-\frac{b}{\sigma^2}\right)$$

$$p_{\text{move}} = \min\left(\frac{P(\mu_{\text{proposed}}, \sigma^2_{\text{proposed}} | X)Q(\mu_{\text{current}}, \sigma^2_{\text{current}} | \mu_{\text{proposed}}, \sigma^2_{\text{proposed}})}{P(\mu_{\text{current}}, \sigma^2_{\text{current}} | X)Q(\mu_{\text{proposed}}, \sigma^2_{\text{proposed}} | \mu_{\text{current}}, \sigma^2_{\text{current}})}, 1\right)$$
imagine normally distributed data $X=(x_1, x_2, x_3, \ldots, x_N)$
what is the **Bayesian** estimate of $\mu$ and $\sigma^2$?

$$P(\mu, \sigma^2 | X) \propto \frac{1}{(\sqrt{2\pi\sigma^2})^N} \exp\left(\frac{-\sum_j (x_j - \mu)^2}{2\sigma^2}\right) \exp\left(\frac{-(\mu - \mu_0)^2}{2\sigma_0^2}\right) \frac{1}{(\sigma^2)^{a+1}} \exp\left(\frac{-b}{\sigma^2}\right)$$

$$p_{\text{move}} = \min\left(\frac{P(\mu_{\text{proposed}}, \sigma^2_{\text{proposed}} | X)Q(\mu_{\text{current}}, \sigma^2_{\text{current}} | \mu_{\text{proposed}}, \sigma^2_{\text{proposed}})}{P(\mu_{\text{current}}, \sigma^2_{\text{current}} | X)Q(\mu_{\text{proposed}}, \sigma^2_{\text{proposed}} | \mu_{\text{current}}, \sigma^2_{\text{current}})}, 1\right)$$

**Problems with this approach:**

1) This will require a multidimensional proposal distribution $Q(\theta_{\text{proposed}} | \theta_{\text{current}})$. In this case, what’s tricky is that here we can’t just use a normal distribution. We need a distribution that has a range $(-\infty, +\infty)$ for $\mu$, but $(0, +\infty)$ for $\sigma^2$. And it needs to be a distribution we can draw random samples from ... I know of no such distribution.
imagine normally distributed data \(X=(x_1, x_2, x_3,...,x_N)\)
what is the **Bayesian** estimate of \(\mu\) and \(\sigma^2\)?

\[
P(\mu, \sigma^2 \mid X) \propto \frac{1}{(\sqrt{2\pi}\sigma^2)^N} \exp\left(\frac{-\sum_j (x_j - \mu)^2}{2\sigma^2}\right) \exp\left(\frac{-(\mu - \mu_0)^2}{2\sigma_0^2}\right) \frac{1}{(\sigma^2)^{a+1}} \exp\left(\frac{-b}{\sigma^2}\right)
\]

\[
p_{\text{move}} = \min\left(\frac{P(\mu_{\text{proposed}}, \sigma_{\text{proposed}}^2 \mid X)Q(\mu_{\text{current}}, \sigma_{\text{current}}^2 \mid \mu_{\text{proposed}}, \sigma_{\text{proposed}}^2)}{P(\mu_{\text{current}}, \sigma_{\text{current}}^2 \mid X)Q(\mu_{\text{proposed}}, \sigma_{\text{proposed}}^2 \mid \mu_{\text{current}}, \sigma_{\text{current}}^2)}, 1\right)
\]

**Problems with this approach:**

2) In some cases (e.g., Bayesian Graphical Models) the priors have dependencies, making it even less likely that any mere mortal could develop a proposal distribution (and a means to take random samples from that distribution) that captures this dependency
imagine normally distributed data $X = (x_1, x_2, x_3, \ldots, x_N)$
what is the **Bayesian** estimate of $\mu$ and $\sigma^2$?

\[
P(\mu, \sigma^2 \mid X) \propto \frac{1}{(\sqrt{2\pi}\sigma^2)^N} \exp \left( \frac{-\sum (x_j - \mu)^2}{2\sigma^2} \right) \exp \left( \frac{-(\mu - \mu_0)^2}{2\sigma_0^2} \right) \frac{1}{(\sigma^2)^{a+1}} \exp \left( \frac{-b}{\sigma^2} \right)
\]

\[
p_{\text{move}} = \min \left( \frac{P(\mu_\text{proposed}, \sigma^2_\text{proposed} \mid X)Q(\mu_\text{current}, \sigma^2_\text{current} \mid \mu_\text{proposed}, \sigma^2_\text{proposed})}{P(\mu_\text{current}, \sigma^2_\text{current} \mid X)Q(\mu_\text{proposed}, \sigma^2_\text{proposed} \mid \mu_\text{current}, \sigma^2_\text{current})}, 1 \right)
\]

**Problems with this approach:**

3) Even if you could have a multidimensional proposal distribution (e.g., all parameters could be normal) doing it this way, with all parameters proposed at once, produces serious convergence problems. The multidimensional proposal distribution must be exquisitely tuned.
Alternative: Do component-wise Metropolis-Hastings.

\[
P(\theta_h | D, \theta_1, \ldots, \theta_{h-1}, \theta_{h+1}, \ldots, \theta_N) = \frac{P(D | \theta_1, \ldots, \theta_{h-1}, \theta_h, \theta_{h+1}, \ldots, \theta_N)P(\theta_h | \theta_1, \ldots, \theta_{h-1}, \theta_{h+1}, \ldots, \theta_N)}{P(D)}
\]

\[
P(\theta_h | D, \theta_{[h]}) = \frac{P(D | \theta_h, \theta_{[h]})P(\theta_h | \theta_{[h]})}{P(D)}
\]

conditional posterior distributions
(remember we talked about those earlier)

basically, you treat all the other parameters as constants

(the trick is that they’re constant when updated one parameter on one step, but then they become the parameter that’s updated on the next step)
Alternative: Do component-wise Metropolis-Hastings.

\[
P(\theta_h \mid D, \theta_1, \ldots, \theta_{h-1}, \theta_{h+1}, \ldots, \theta_N) = \frac{P(D \mid \theta_1, \ldots, \theta_{h-1}, \theta_h, \theta_{h+1}, \ldots, \theta_N) P(\theta_h \mid \theta_1, \ldots, \theta_{h-1}, \theta_{h+1}, \ldots, \theta_N)}{P(D)}
\]

\[
P(\theta_h \mid D, \theta_{[h]}) = \frac{P(D \mid \theta_h, \theta_{[h]}) P(\theta_h \mid \theta_{[h]})}{P(D)}
\]
Alternative: Do component-wise Metropolis-Hastings.

\[
P(\theta_h \mid D, \theta_1, \ldots, \theta_{h-1}, \theta_{h+1}, \ldots, \theta_N) = \frac{P(D \mid \theta_1, \ldots, \theta_{h-1}, \theta_h, \theta_{h+1}, \ldots, \theta_N)P(\theta_h \mid \theta_1, \ldots, \theta_{h-1}, \theta_{h+1}, \ldots, \theta_N)}{P(D)}
\]

\[
P(\theta_h \mid D, \theta_{[h]}) = \frac{P(D \mid \theta_h, \theta_{[h]})P(\theta_h \mid \theta_{[h]})}{P(D)}
\]

\[
p_{\text{move}} = \min\left(\frac{P(\theta_{h,\text{proposed}} \mid D, \theta_{[h]})Q(\theta_{h,\text{current}} \mid \theta_{h,\text{proposed}})}{P(\theta_{h,\text{current}} \mid D, \theta_{[h]})Q(\theta_{h,\text{proposed}} \mid \theta_{h,\text{current}})}, 1\right)
\]

do Metropolis-Hastings on each single parameter in turn ...
- can do it in a consistent order (works well)
- can permute the order (works well)
- do not want to do it randomly (poor convergence)

sometimes do Metropolis-Hastings on blocks of parameters ...
- e.g., might have similar individual differences parameters that are all proposed at once in one step
imagine normally distributed data \( X=(x_1, x_2, x_3, \ldots, x_N) \)
what is the **Bayesian** estimate of \( \mu \) and \( \sigma^2 \)?

\[
P(\mu \mid X, \sigma^2) = \frac{1}{P(X \mid \sigma^2)} \left( \frac{1}{\sqrt{2\pi\sigma^2}} \right)^N \exp \left( -\sum_j (x_j - \mu)^2 \right) \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp \left( -\frac{(\mu - \mu_0)^2}{2\sigma_0^2} \right)
\]

\[
P(\sigma^2 \mid X, \mu) = \frac{1}{P(X \mid \mu)} \left( \frac{1}{\sqrt{2\pi\sigma^2}} \right)^N \exp \left( -\sum_j (x_j - \mu)^2 \right) \frac{b^a}{\Gamma(a)(\sigma^2)^{a+1}} \exp \left( \frac{-b}{\sigma^2} \right)
\]
imagine normally distributed data $X= (x_1, x_2, x_3, \ldots, x_N)$ 
what is the **Bayesian** estimate of $\mu$ and $\sigma^2$?

$$P(\mu \mid X, \sigma^2) \propto \exp \left( \frac{-\sum (x_j - \mu)^2}{2\sigma^2} \right) \exp \left( \frac{-(\mu - \mu_0)^2}{2\sigma_0^2} \right)$$

$$P(\sigma^2 \mid X, \mu) \propto \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^N} \exp \left( \frac{-\sum (x_j - \mu)^2}{2\sigma^2} \right) \frac{1}{(\sigma^2)^{a+1}} \exp \left( \frac{-b}{\sigma^2} \right)$$
see week14.m
imagine normally distributed data $X=\left(x_1, x_2, x_3, \ldots, x_N\right)$
what is the **Bayesian** estimate of $\mu$ and $\sigma^2$?

\[
P(\mu \mid X, \sigma^2) \sim N(\mu', \sigma'^2)
\]

\[
\mu' = \left(\frac{1}{\sigma^2} \sum_j x_j + \frac{1}{\sigma_0^2} \mu_0\right) \sigma^2,
\]

\[
\sigma'^2 = \left(N \frac{1}{\sigma^2} + \frac{1}{\sigma_0^2}\right)^{-1}
\]

\[
P(\sigma^2 \mid X, \mu) \sim \text{InverseGamma}(a', b')
\]

\[
a' = N / 2 + a
\]

\[
b' = \left[\sum_j (x_j - \mu)^2\right] / 2 + b
\]

Remember that we were able to solve for the conditional distributions exactly …

while it’s usually not possible to solve for the full joint distribution $P(\theta_1, \theta_2, \theta_3 \ldots \mid X)$, it is common to be able to solve for the conditionals $P(\theta_1 \mid X, \theta_2, \theta_3 \ldots)$, $P(\theta_2 \mid X, \theta_1, \theta_3 \ldots)$, etc.
imagine normally distributed data $X=(x_1, x_2, x_3, \ldots, x_N)$
what is the **Bayesian** estimate of $\mu$ and $\sigma^2$?

$$P(\mu \mid X, \sigma^2) \sim N(\mu', \sigma^2')$$
$$\mu' = \left( \frac{1}{\sigma^2} \sum_j x_j + \frac{1}{\sigma_0^2} \mu_0 \right) \sigma^2,$$
$$\sigma^2' = \left( N \frac{1}{\sigma^2} + \frac{1}{\sigma_0^2} \right)^{-1}$$

$$P(\sigma^2 \mid X, \mu) \sim \text{InverseGamma}(a', b')$$
$$a' = N / 2 + a$$
$$b' = \left[ \sum_j (x_j - \mu)^2 \right] / 2 + b$$

We can generate random samples from Normal directly
We can generate random samples from InvGam directly

Utilize this in **Gibbs Sampling**
where it can be shown that $p_{\text{move}} = 1$

has better convergence than standard M-H
see week14.m
Now that you understand some the guts of Bayesian statistical modeling, you rarely ever have to worry about them ever again.

WinBUGS and OpenBUGS

Because packages like WinBUGS take care of everything:
http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml

MATBUGS interfaces Matlab with WinBUGS:
http://code.google.com/p/matbugs/

R2WinBUGS interfaces R with WinBUGS:
http://cran.r-project.org/web/packages/R2WinBUGS/index.html

Other interfaces to Python, SAS, and even Excel:
http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/remote14.shtml
Simple example: *Binomial distribution (again)*

**Bayesian model**

```r
model { 
  theta ~ dbeta(1,1)  # Prior on Rate
  k ~ dbin(theta,n)   # likelihood
}
```

**Data**

```r
list(
  k=5,       # number of successes
  n=10       # number of trials
)
```

**Starting values for MCMC chains**

```r
list(
  theta=0.1 # initial for chain 1
)
```

```r
list(
  theta=0.9 # initial for chain 2
)
```
model is syntactically correct
data loaded
model compiled
chain initialized but other chain(s) contain uninitialized variables
model is initialized

Dynamic trace

theta chains 2:1

19850 19900 19950
iteration

Node statistics

<table>
<thead>
<tr>
<th>node</th>
<th>mean</th>
<th>sd</th>
<th>MC error</th>
<th>2.5%</th>
<th>median</th>
<th>97.5%</th>
<th>start</th>
<th>sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>theta</td>
<td>0.5004</td>
<td>0.1386</td>
<td>6.329E-4</td>
<td>0.2342</td>
<td>0.5006</td>
<td>0.765</td>
<td>1</td>
<td>40000</td>
</tr>
</tbody>
</table>

Kernel density

theta chains 1:2 sample: 40000

Figure 2.5: Example of an output log file.
Simple example: *Normal distribution (again)*

**Bayesian model**

# Inferring The Mean And Standard Deviation Of A Gaussian

```r
model{
    # likelihood
    for (i in 1:n){
        x[i] ~ dnorm(mu,lambda)
    }

    # Priors
    mu ~ dnorm(0,.001)
    lambda <- 1/pow(sigma,2)
}
```
WinBUGS/OpenBUGS figures out whether to do standard Metropolis-Hastings or Gibbs Sampling for each parameter, has machine learning / expert system engines to help optimize the simulation of the posterior distribution, has built-in checks on convergence, does all of the necessary computations.

Michael Lee and E.J. Wagenmakers have written
A Practical Course in Bayesian Graphical Modeling

Highly recommended to everyone:
Bayesian Graphical Modeling
Consider the binomial example again. But now let’s imagine that there are 10 questions (N=10) and each participant get $x_i$ of the questions right.

We want to figure out the probability of success $p_i$ for each participant in the experiment.

(Right now, we’re just figuring out $p_i$, but you could imagine having an underlying model that determines $p_i$, like we did when we developing the approach to maximum likelihood fitting with the SCM and GCM earlier in the semester)
Consider the binomial example again. But now let’s imagine that there are 10 questions (N=10) and each participant get $x_i$ of the questions right.

**APPROACH #0**

*Just average all the subjects and fit the average data.*

*Remember our discussion of the potential problems associated with averaging ... if we can avoid it, we usually should.*
Consider the binomial example again. But now let’s imagine that there are 10 questions (N=10) and each participant gets $x_i$ of the questions right.

We want to figure out the probability of success $p_i$ for each participant in the experiment.

**APPROACH #1**
Just fit each participant separately, figuring the Bayesian posterior distribution of $P(p_i \mid x_i)$

**APPROACH #2**
Fit all participants simultaneously, figuring the Bayesian posterior distribution of $P(p_1, \ldots, p_M \mid x_1 \ldots x_M)$

**APPROACH #3**
Structured individual differences, $p_i \sim \text{beta}(A,B)$ in a hierarchical model
APPROACH #3
Structured individual differences, \( p_i \sim \text{beta}(A,B) \)
in a hierarchical model
APPROACH #3
Structured individual differences, \( p_i \sim \text{beta}(A,B) \) in a hierarchical model

\[
\begin{align*}
\mu &\sim \text{Beta}(A_\mu, B_\mu) \\
\kappa &\sim \text{Gamma}(S_\kappa, R_\kappa) \\
A &= \mu \kappa \\
B &= (1-\mu) \kappa \\
p_i &\sim \text{Beta}(A, B) \\
x_i &\sim \text{Binomial}(p_i, N)
\end{align*}
\]
APPROACH #3
Structured individual differences, $p_i \sim \text{beta}(A,B)$ in a hierarchical model

\[ \mu \sim \text{Beta}(A_\mu, B_\mu) \]
\[ \kappa \sim \text{Gamma}(S_\kappa, R_\kappa) \]

\[ A = \mu\kappa \]
\[ B = (1-\mu)\kappa \]

\[ p_i \sim \text{Beta}(A, B) \]
\[ x_i \sim \text{Binomial}(p_i, N) \]
**APPROACH #3**

Structured individual differences, $p_i \sim \text{beta}(A,B)$ in a hierarchical model

\[ \begin{align*}
\mu & \sim \text{Beta}(A_{\mu}, B_{\mu}) \\
\kappa & \sim \text{Gamma}(S_\kappa, R_\kappa) \\
A &= \mu \kappa \\
B &= (1-\mu)\kappa \\
\mu & \sim \text{Gamma}(1,10) \\
p_i & \sim \text{Beta}(A, B) \\
x_i & \sim \text{Binomial}(p_i, N)
\end{align*} \]
**APPROACH #3**

*Structured individual differences, $p_i \sim \text{beta}(A,B)$ in a hierarchical model*

\[
\begin{align*}
\mu & \sim \text{Beta}(A_\mu,B_\mu) \\
\kappa & \sim \text{Gamma}(S_\kappa,R_\kappa) \\
A & = \mu \kappa \\
B & = (1-\mu)\kappa \\
\mu & \sim \text{Gamma}(1,10) \\
p_i & \sim \text{Beta}(A,B) \quad \text{(mean} = \mu) \\
x_i & \sim \text{Binomial}(p_i,N) \\
\end{align*}
\]
WinBUGS calculates: $P(p_1, p_2, \ldots, p_M, \mu, \kappa \mid x_1 \ldots x_M)$

we can calculate things like marginals, e.g., $P(p_1 \mid x_1 \ldots x_M)$

\[ \begin{align*}
\mu &\sim \text{Beta}(A_\mu, B_\mu) \\
\kappa &\sim \text{Gamma}(S_\kappa, R_\kappa) \\
A &= \mu \kappa \\
B &= (1-\mu)\kappa \\
\end{align*} \]

\[ \begin{align*}
p_i &\sim \text{Beta}(A, B) \\
x_i &\sim \text{Binomial}(p_i, N) \\
\end{align*} \]
WinBUGS calculates: $P(p_1,p_2,\ldots,p_M,\mu,\kappa \mid x_1\ldots x_M)$

essentially, the graphical model defines conditional probabilities - recall that Gibbs Sampling is based on conditional probabilities

$$\mu \sim \text{Beta}(A_\mu,B_\mu)$$

$$\kappa \sim \text{Gamma}(S_\kappa,R_\kappa)$$

$$A = \mu \kappa$$

$$B = (1-\mu)\kappa$$

$$p_i \sim \text{Beta}(A,B)$$

$$x_i \sim \text{Binomial}(p_i,N)$$
EXAMPLE: simple model of forgetting over time

\[ \theta_t = \exp(-\alpha t) + \beta \]

\( \theta_t \) is the probability an item is recalled after delay \( t \)

Table 1
Fictitious memory retention data, giving the number out of 18 items correctly recalled for three participants over nine time intervals and including an extra retention interval of 200 sec and an extra participant as missing data

<table>
<thead>
<tr>
<th>Participant</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>7</th>
<th>12</th>
<th>21</th>
<th>35</th>
<th>59</th>
<th>99</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18</td>
<td>18</td>
<td>16</td>
<td>13</td>
<td>9</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>17</td>
<td>13</td>
<td>9</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>10</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
No individual differences model

\[ \alpha \sim \text{Uniform}(0, 1) \]
\[ \beta \sim \text{Uniform}(0, 1) \]
\[ \theta_j = \exp(-\alpha t_j) + \beta \quad 0 < \theta_j < 1 \]
\[ k_{ij} \sim \text{Binomial}(\theta_j, n_{ij}) \]
Full individual differences model

\[ \alpha_i \sim \text{Uniform}(0, 1) \]
\[ \beta_i \sim \text{Uniform}(0, 1) \]
\[ \theta_{ij} = \exp(-\alpha_i t_j) + \beta_i \quad 0 < \theta_j < 1 \]
\[ k_{ij} \sim \text{Binomial}(\theta_{ij}, n_{ij}) \]
Structured individual differences model

\[ \lambda_{\alpha}, \mu_{\alpha}, \mu_{\beta}, \lambda_{\beta} \]

\[ \alpha_{i}, \beta_{i} \]

\[ \theta_{ij} \]

\[ t_{j} \]

\[ k_{ij} \]

\[ n_{ij} \]

\[ i = 1, \ldots, N \]

\[ j = 1, \ldots, T \]

\[ \mu_{\alpha} \sim \text{Uniform}(0, 1) \]

\[ \lambda_{\alpha} \sim \text{Gamma}(0.001, 0.001) \]

\[ \mu_{\beta} \sim \text{Uniform}(0, 1) \]

\[ \lambda_{\beta} \sim \text{Gamma}(0.001, 0.001) \]

\[ \alpha_{i} \sim \text{Gaussian}(\mu_{\alpha}, \lambda_{\alpha}) \quad 0 < \alpha_{i} < 1 \]

\[ \beta_{i} \sim \text{Gaussian}(\mu_{\beta}, \lambda_{\beta}) \quad 0 < \beta_{i} < 1 \]

\[ \theta_{ij} = \exp(-\alpha_{i}t_{j}) + \beta_{i} \quad 0 < \theta_{ij} < 1 \]

\[ k_{ij} \sim \text{Binomial}(\theta_{ij}, n_{ij}) \]