Bayesian Inference for Dynamic Systems: Background and Concepts

Mike Dowd
Dalhousie University

January 2020

Outline

- The Bayesian Approach
- Process Model
- Data Model
- Prior Information
- MCMC

What is Bayes Theorem?

GOAL: Given observations, y, we want to determine the parameters, θ, taking into account our prior knowledge.

Bayes' theorem states:

$$[\theta|y] = \frac{[y|\theta] \cdot [\theta]}{[y]} \propto [y|\theta] \cdot [\theta]$$

where:

- $[\theta|y]$: posterior pdf of parameters given observations (the target quantity we want to estimate)
- $[y|\theta]$: likelihood of data given a set of parameters
- $[\theta]$: prior pdf of the parameters
- $[y]$: (unconditional) observation pdf (normalizing constant, not needed)

What is Bayes Theorem?

GOAL: Given observations, y, we want to determine the parameters, θ, taking into account our prior knowledge.

Bayes' theorem states:

$$[\theta|y] = \frac{[y|\theta] \cdot [\theta]}{[y]} \propto [y|\theta] \cdot [\theta]$$

where:

- $[\theta|y]$: posterior pdf of parameters given observations (the target quantity we want to estimate)
- $[y|\theta]$: likelihood of data given a set of parameters
- $[\theta]$: prior pdf of the parameters
- $[y]$: (unconditional) observation pdf (normalizing constant, not needed)

What is Bayes Theorem?

GOAL: Given observations, y, we want to determine the parameters, θ, taking into account our prior knowledge.

Bayes' theorem states:

$$[\theta|y] = \frac{[y|\theta] \cdot [\theta]}{[y]} \propto [y|\theta] \cdot [\theta]$$

where:

- $[\theta|y]$: posterior pdf of parameters given observations (the target quantity we want to estimate)
- $[y|\theta]$: likelihood of data given a set of parameters
- $[\theta]$: prior pdf of the parameters
- $[y]$: (unconditional) observation pdf (normalizing constant, not needed)

What is Bayes Theorem?

GOAL: Given observations, y, we want to determine the parameters, θ, taking into account our prior knowledge.

Bayes' theorem states:

$$[\theta|y] = \frac{[y|\theta] \cdot [\theta]}{[y]} \propto [y|\theta] \cdot [\theta]$$

where:

- $[\theta|y]$: posterior pdf of parameters given observations (the target quantity we want to estimate)
- $[y|\theta]$: likelihood of data given a set of parameters
- $[\theta]$: prior pdf of the parameters
- $[y]$: (unconditional) observation pdf (normalizing constant, not needed)

What is Bayes Theorem?

GOAL: Given observations, y, we want to determine the parameters, θ, taking into account our prior knowledge.

Bayes' theorem states:

$$[\theta|y] = \frac{[y|\theta] \cdot [\theta]}{[y]} \propto [y|\theta] \cdot [\theta]$$

where:

- $[\theta|y]$: posterior pdf of parameters given observations (the target quantity we want to estimate)
- $[y|\theta]$: likelihood of data given a set of parameters
- $[\theta]$: prior pdf of the parameters
- $[y]$: (unconditional) observation pdf (normalizing constant, not needed)

What is Bayes Theorem?

GOAL: Given observations, y, we want to determine the parameters, θ, taking into account our prior knowledge.

Bayes' theorem states:

$$[\theta|y] = \frac{[y|\theta] \cdot [\theta]}{[y]} \propto [y|\theta] \cdot [\theta]$$

where:

- $[\theta|y]$: posterior pdf of parameters given observations (the target quantity we want to estimate)
- $[y|\theta]$: likelihood of data given a set of parameters
- $[\theta]$: prior pdf of the parameters
- $[y]$: (unconditional) observation pdf (normalizing constant, not needed)

Dynamic Systems: the Process Model

A dynamic system takes the form

$$dx/dt = f(x, \theta, w(t))$$

where

- $x(t)$: state of the system over time (univariate or multivariate)
- θ: parameters
- f: dynamical operator (nonlinear function)
- $w(t)$: forcing (deterministic or stochastic)

Note: the state x can be compared directly to observations y, but the parameters θ are only indirectly related (via the state).

Dynamic Systems: the Process Model

A dynamic system takes the form

$$dx/dt = f(x, \theta, w(t))$$

where

- $x(t)$: state of the system over time (univariate or multivariate)
- θ: parameters
- f: dynamical operator (nonlinear function)
- $w(t)$: forcing (deterministic or stochastic)

Note: the state x can be compared directly to observations y, but the parameters θ are only indirectly related (via the state).

Dynamic Systems: the Process Model

A dynamic system takes the form

$$dx/dt = f(x, \theta, w(t))$$

where

- $x(t)$: state of the system over time (univariate or multivariate)
- θ: parameters
- f: dynamical operator (nonlinear function)
- $w(t)$: forcing (deterministic or stochastic)

Note: the state x can be compared directly to observations y, but the parameters θ are only indirectly related (via the state).

Dynamic Systems: the Process Model

A dynamic system takes the form

$$dx/dt = f(x, \theta, w(t))$$

where

- $x(t)$: state of the system over time (univariate or multivariate)
- θ: parameters
- f: dynamical operator (nonlinear function)
- $w(t)$: forcing (deterministic or stochastic)

Note: the state x can be compared directly to observations y, but the parameters θ are only indirectly related (via the state).

Dynamic Systems: the Process Model

A dynamic system takes the form

$$dx/dt = f(x, \theta, w(t))$$

where

- $x(t)$: state of the system over time (univariate or multivariate)
- θ: parameters
- f: dynamical operator (nonlinear function)
- $w(t)$: forcing (deterministic or stochastic)

Note: the state x can be compared directly to observations y, but the parameters θ are only indirectly related (via the state).

Dynamic Systems: the Process Model

A dynamic system takes the form

$$dx/dt = f(x, \theta, w(t))$$

where

- $x(t)$: state of the system over time (univariate or multivariate)
- θ: parameters
- f: dynamical operator (nonlinear function)
- $w(t)$: forcing (deterministic or stochastic)

Note: the state x can be compared directly to observations y, but the parameters θ are only indirectly related (via the state).

Bayesian Hierarchical Model

Bayes theorem can expanded hierarchically

$$[process, parameter|data] \propto$$
$$[data|process, parameter] \cdot [process|parameter] \cdot [parameter]$$

or

$$[x, \theta|y] \propto [y|x, \theta] \cdot [x|\theta] \cdot [\theta]$$

The target posterior, $[x, \theta|y]$, is now expressed as a product of:

1. data model: $[y|x, \theta]$
2. process model: $[x|\theta]$
3. parameter prior: $[\theta]$

Advantage: each of these can be developed in isolation, and then recombined.

Bayesian Hierarchical Model

Bayes theorem can expanded hierarchically

$$[process, parameter|data] \propto$$
$$[data|process, parameter] \cdot [process|parameter] \cdot [parameter]$$

or

$$[x, \theta|y] \propto [y|x, \theta] \cdot [x|\theta] \cdot [\theta]$$

The target posterior, $[x, \theta|y]$, is now expressed as a product of:

1. data model: $[y|x, \theta]$
2. process model: $[x|\theta]$
3. parameter prior: $[\theta]$

Advantage: each of these can be developed in isolation, and then recombined.

Bayesian Hierarchical Model

Bayes theorem can expanded hierarchically

$$[process, parameter|data] \propto$$
$$[data|process, parameter] \cdot [process|parameter] \cdot [parameter]$$

or

$$[x, \theta|y] \propto [y|x, \theta] \cdot [x|\theta] \cdot [\theta]$$

The target posterior, $[x, \theta|y]$, is now expressed as a product of:

1. data model: $[y|x, \theta]$
2. process model: $[x|\theta]$
3. parameter prior: $[\theta]$

Advantage: each of these can be developed in isolation, and then recombined.

Bayesian Hierarchical Model

Bayes theorem can expanded hierarchically

$$[process, parameter|data] \propto$$
$$[data|process, parameter] \cdot [process|parameter] \cdot [parameter]$$

or

$$[x, \theta|y] \propto [y|x, \theta] \cdot [x|\theta] \cdot [\theta]$$

The target posterior, $[x, \theta|y]$, is now expressed as a product of:

1. data model: $[y|x, \theta]$
2. process model: $[x|\theta]$
3. parameter prior: $[\theta]$

Advantage: each of these can be developed in isolation, and then recombined.

Bayesian Hierarchical Model

Bayes theorem can expanded hierarchically

$$[process, parameter|data] \propto$$
$$[data|process, parameter] \cdot [process|parameter] \cdot [parameter]$$

or

$$[x, \theta|y] \propto [y|x, \theta] \cdot [x|\theta] \cdot [\theta]$$

The target posterior, $[x, \theta|y]$, is now expressed as a product of:

1. data model: $[y|x, \theta]$
2. process model: $[x|\theta]$
3. parameter prior: $[\theta]$

Advantage: each of these can be developed in isolation, and then recombined.

Bayesian Hierarchical Model

Bayes theorem can expanded hierarchically

$$[process, parameter|data] \propto$$
$$[data|process, parameter] \cdot [process|parameter] \cdot [parameter]$$

or

$$[x, \theta|y] \propto [y|x, \theta] \cdot [x|\theta] \cdot [\theta]$$

The target posterior, $[x, \theta|y]$, is now expressed as a product of:

1. data model: $[y|x, \theta]$
2. process model: $[x|\theta]$
3. parameter prior: $[\theta]$

Advantage: each of these can be developed in isolation, and then recombined.

Bayesian Hierarchical Model

Bayes theorem can expanded hierarchically

$$[process, parameter|data] \propto$$
$$[data|process, parameter] \cdot [process|parameter] \cdot [parameter]$$

or

$$[x, \theta|y] \propto [y|x, \theta] \cdot [x|\theta] \cdot [\theta]$$

The target posterior, $[x, \theta|y]$, is now expressed as a product of:

1. data model: $[y|x, \theta]$
2. process model: $[x|\theta]$
3. parameter prior: $[\theta]$

Advantage: each of these can be developed in isolation, and then recombined.

Bayesian Computation

- The target posterior probability distributions can only analytically solved for (i.e. the probability distributions mathematically manipulated) in the simplest of cases (e.g. linear and Gaussian problems, exponential family distributions)
- **Idea**: probability distributions can be represented by samples
- Computational Bayesian methods are based on rules for manipulating / transforming samples, instead of transforming probability distributions directly

DEMO: DISTRIBUTIONS AS SAMPLES

Bayesian Computation

- The target posterior probability distributions can only analytically solved for (i.e. the probability distributions mathematically manipulated) in the simplest of cases (e.g. linear and Gaussian problems, exponential family distributions)
- **Idea**: probability distributions can be represented by samples
- Computational Bayesian methods are based on rules for manipulating / transforming samples, instead of transforming probability distributions directly

DEMO: DISTRIBUTIONS AS SAMPLES

Bayesian Computation

- The target posterior probability distributions can only analytically solved for (i.e. the probability distributions mathematically manipulated) in the simplest of cases (e.g. linear and Gaussian problems, exponential family distributions)
- **Idea**: probability distributions can be represented by samples
- Computational Bayesian methods are based on rules for manipulating / transforming samples, instead of transforming probability distributions directly

DEMO: DISTRIBUTIONS AS SAMPLES

Bayesian Computation

- The target posterior probability distributions can only analytically solved for (i.e. the probability distributions mathematically manipulated) in the simplest of cases (e.g. linear and Gaussian problems, exponential family distributions)
- **Idea**: probability distributions can be represented by samples
- Computational Bayesian methods are based on rules for manipulating / transforming samples, instead of transforming probability distributions directly

DEMO: DISTRIBUTIONS AS SAMPLES

P Growth Toy Model

A simple one compartment phytoplankton growth model is

$$\frac{dP}{dt} = \gamma(1 + sin(\omega t))P - \lambda P^2$$

where

- P: phytoplankton biomass/concentration,
- γ: growth rate
- λ: mortality/loss term.

Features: (i) nonlinear (quadratic loss), (ii) annual modulation of growth

P Growth Toy Model

A simple one compartment phytoplankton growth model is

$$\frac{dP}{dt} = \gamma(1 + sin(\omega t))P - \lambda P^2$$

where

- P: phytoplankton biomass/concentration,
- γ: growth rate
- λ: mortality/loss term.

Features: (i) nonlinear (quadratic loss), (ii) annual modulation of growth

Numerical Solution

The first step is the discretize the model (simplest Euler Method)

$$P_{t+\Delta} = P_t + \Delta \left(\gamma(1 + sin(\omega t))P_t - \lambda P_t^2\right)$$

where Δ is the time step.

DEMO: DISCRETIZING A MODEL

Remarks:

- this discretization is not unique (built-in ODE solvers typically use Runge-Kutta)
- the size of the time step Δ matters (smaller is accurate, but also slow). Affects numerical stability

DEMO: NUMERICAL SIMULATION

Numerical Solution

The first step is the discretize the model (simplest Euler Method)

$$P_{t+\Delta} = P_t + \Delta\left(\gamma(1 + sin(\omega t))P_t - \lambda P_t^2\right)$$

where Δ is the time step.

DEMO: DISCRETIZING A MODEL

Remarks:

- this discretization is not unique (built-in ODE solvers typically use Runge-Kutta)
- the size of the time step Δ matters (smaller is accurate, but also slow). Affects numerical stability

DEMO: NUMERICAL SIMULATION

Numerical Solution

The first step is the discretize the model (simplest Euler Method)

$$P_{t+\Delta} = P_t + \Delta \left(\gamma(1 + sin(\omega t))P_t - \lambda P_t^2 \right)$$

where Δ is the time step.

DEMO: DISCRETIZING A MODEL

Remarks:

- this discretization is not unique (built-in ODE solvers typically use Runge-Kutta)
- the size of the time step Δ matters (smaller is accurate, but also slow). Affects numerical stability

DEMO: NUMERICAL SIMULATION

Numerical Solution

The first step is the discretize the model (simplest Euler Method)

$$P_{t+\Delta} = P_t + \Delta \left(\gamma(1 + sin(\omega t))P_t - \lambda P_t^2\right)$$

where Δ is the time step.

DEMO: DISCRETIZING A MODEL

Remarks:

- this discretization is not unique (built-in ODE solvers typically use Runge-Kutta)
- the size of the time step Δ matters (smaller is accurate, but also slow). Affects numerical stability

DEMO: NUMERICAL SIMULATION

Numerical Solution

The first step is the discretize the model (simplest Euler Method)

$$P_{t+\Delta} = P_t + \Delta\left(\gamma(1 + sin(\omega t))P_t - \lambda P_t^2\right)$$

where Δ is the time step.

DEMO: DISCRETIZING A MODEL

Remarks:

- this discretization is not unique (built-in ODE solvers typically use Runge-Kutta)
- the size of the time step Δ matters (smaller is accurate, but also slow). Affects numerical stability

DEMO: NUMERICAL SIMULATION

Numerical Solution

The first step is the discretize the model (simplest Euler Method)

$$P_{t+\Delta} = P_t + \Delta \left(\gamma(1 + sin(\omega t))P_t - \lambda P_t^2 \right)$$

where Δ is the time step.

DEMO: DISCRETIZING A MODEL

Remarks:

- this discretization is not unique (built-in ODE solvers typically use Runge-Kutta)
- the size of the time step Δ matters (smaller is accurate, but also slow). Affects numerical stability

DEMO: NUMERICAL SIMULATION

Stochastic Dynamics

Bayesian models often rely on stochastic dynamics. Randomness can be incorporated as:

1. Additive Noise on the State
2. Stochastic Parameters
3. Stochastic Dynamic Parameters

Concepts:

- *Realizations*: One run of a stochastic model (a possible outcome)
- *Ensembles*: A set of realizations from which statistical properties can be derived

DEMO: REALIZATIONS AND ENSEMBLES

Stochastic Dynamics

Bayesian models often rely on stochastic dynamics. Randomness can be incorporated as:

1. Additive Noise on the State
2. Stochastic Parameters
3. Stochastic Dynamic Parameters

Concepts:

- *Realizations*: One run of a stochastic model (a possible outcome)
- *Ensembles*: A set of realizations from which statistical properties can be derived

DEMO: REALIZATIONS AND ENSEMBLES

Stochastic Dynamics

Bayesian models often rely on stochastic dynamics. Randomness can be incorporated as:

1. Additive Noise on the State
2. Stochastic Parameters
3. Stochastic Dynamic Parameters

Concepts:

- *Realizations*: One run of a stochastic model (a possible outcome)
- *Ensembles*: A set of realizations from which statistical properties can be derived

DEMO: REALIZATIONS AND ENSEMBLES

Stochastic Dynamics

Bayesian models often rely on stochastic dynamics. Randomness can be incorporated as:

1. Additive Noise on the State
2. Stochastic Parameters
3. Stochastic Dynamic Parameters

Concepts:

- *Realizations*: One run of a stochastic model (a possible outcome)
- *Ensembles*: A set of realizations from which statistical properties can be derived

DEMO: REALIZATIONS AND ENSEMBLES

Stochastic Dynamics

Bayesian models often rely on stochastic dynamics. Randomness can be incorporated as:

1. Additive Noise on the State
2. Stochastic Parameters
3. Stochastic Dynamic Parameters

Concepts:

- *Realizations*: One run of a stochastic model (a possible outcome)
- *Ensembles*: A set of realizations from which statistical properties can be derived

DEMO: REALIZATIONS AND ENSEMBLES

Stochastic Dynamics

Bayesian models often rely on stochastic dynamics. Randomness can be incorporated as:

1. Additive Noise on the State
2. Stochastic Parameters
3. Stochastic Dynamic Parameters

Concepts:

- *Realizations*: One run of a stochastic model (a possible outcome)
- *Ensembles*: A set of realizations from which statistical properties can be derived

DEMO: REALIZATIONS AND ENSEMBLES

Stochastic Dynamics

Bayesian models often rely on stochastic dynamics. Randomness can be incorporated as:

1. Additive Noise on the State
2. Stochastic Parameters
3. Stochastic Dynamic Parameters

Concepts:

- *Realizations*: One run of a stochastic model (a possible outcome)
- *Ensembles*: A set of realizations from which statistical properties can be derived

DEMO: REALIZATIONS AND ENSEMBLES

Stochastic Dynamics

Bayesian models often rely on stochastic dynamics. Randomness can be incorporated as:

1. Additive Noise on the State
2. Stochastic Parameters
3. Stochastic Dynamic Parameters

Concepts:

- *Realizations*: One run of a stochastic model (a possible outcome)
- *Ensembles*: A set of realizations from which statistical properties can be derived

DEMO: REALIZATIONS AND ENSEMBLES

The Data Model

- The data model is the probability distribution of the observations $[y|\theta]$. It can expressed as a *likelihood* $L(\theta|y)$.
- It measures how consistent (or likely) the model parameters are with the observations.
- The likelihood in intimately linked to cost function used in data assimilation (the form of cost function is dictated by $[y|\theta]$). Optimization is used to estimate parameters

The Data Model

- The data model is the probability distribution of the observations $[y|\theta]$. It can expressed as a *likelihood* $L(\theta|y)$.
- It measures how consistent (or likely) the model parameters are with the observations.
- The likelihood in intimately linked to cost function used in data assimilation (the form of cost function is dictated by $[y|\theta]$). Optimization is used to estimate parameters

The Data Model

- The data model is the probability distribution of the observations $[y|\theta]$. It can expressed as a *likelihood* $L(\theta|y)$.
- It measures how consistent (or likely) the model parameters are with the observations.
- The likelihood in intimately linked to cost function used in data assimilation (the form of cost function is dictated by $[y|\theta]$). Optimization is used to estimate parameters

The Data Model

- The data model is the probability distribution of the observations $[y|\theta]$. It can expressed as a *likelihood* $L(\theta|y)$.
- It measures how consistent (or likely) the model parameters are with the observations.
- The likelihood in intimately linked to cost function used in data assimilation (the form of cost function is dictated by $[y|\theta]$). Optimization is used to estimate parameters

Remarks of Observations of Lower Trophic Level Biology

- Bias is often an important or perhaps dominant form of error. It, however, is usually treated as an external calibration exercise or as part of the Bayesian model (i.e. estimating offsets)
- Variability is more than just instrument or laboratory errors. It includes unresolved environmental variability. These are errors of representativeness or change of support (e. g. data conforms to a point sample, while you are modelling a spatial and/or temporal average).
- Systems of interest are usually *partially observed*. Not all prognostic variables are measured, and sampling through time may not occur at regularly spaced intervals.

Remarks of Observations of Lower Trophic Level Biology

- Bias is often an important or perhaps dominant form of error. It, however, is usually treated as an external calibration exercise or as part of the Bayesian model (i.e. estimating offsets)
- Variability is more than just instrument or laboratory errors. It includes unresolved environmental variability. These are errors of representativeness or change of support (e. g. data conforms to a point sample, while you are modelling a spatial and/or temporal average).
- Systems of interest are usually *partially observed*. Not all prognostic variables are measured, and sampling through time may not occur at regularly spaced intervals.

Remarks of Observations of Lower Trophic Level Biology

- Bias is often an important or perhaps dominant form of error. It, however, is usually treated as an external calibration exercise or as part of the Bayesian model (i.e. estimating offsets)
- Variability is more than just instrument or laboratory errors. It includes unresolved environmental variability. These are errors of representativeness or change of support (e. g. data conforms to a point sample, while you are modelling a spatial and/or temporal average).
- Systems of interest are usually *partially observed*. Not all prognostic variables are measured, and sampling through time may not occur at regularly spaced intervals.

Remarks of Observations of Lower Trophic Level Biology

- Bias is often an important or perhaps dominant form of error. It, however, is usually treated as an external calibration exercise or as part of the Bayesian model (i.e. estimating offsets)
- Variability is more than just instrument or laboratory errors. It includes unresolved environmental variability. These are errors of representativeness or change of support (e. g. data conforms to a point sample, while you are modelling a spatial and/or temporal average).
- Systems of interest are usually *partially observed*. Not all prognostic variables are measured, and sampling through time may not occur at regularly spaced intervals.

Estimating Parameters via the Likelihood

Assume (for simplicity) that the state, x, is a deterministic function of the parameters, i.e. $x = g(\theta)$.

Steps:

1. **Define the data model.** Example: For a normal distribution $[y|\theta] = L(\theta|y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{\frac{1}{2\sigma^2}(y - g(\theta))^2\}$
2. **Find the θ value that maximizes $L(\theta|y)$.** Example: With σ^2 constant, we maximize $J(\theta) = -(y - g(\theta))^2$ with respect to θ. Same as nonlinear least squares.

DEMO: PROFILE LIKELIHOOD, LIKELIHOOD SURFACE, OPTIMIZERS

Estimating Parameters via the Likelihood

Assume (for simplicity) that the state, x, is a deterministic function of the parameters, i.e. $x = g(\theta)$.
Steps:

1. **Define the data model.** Example: For a normal distribution $[y|\theta] = L(\theta|y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{\frac{1}{2\sigma^2}(y - g(\theta))^2\}$
2. **Find the θ value that maximizes $L(\theta|y)$.** Example: With σ^2 constant, we maximize $J(\theta) = -(y - g(\theta))^2$ with respect to θ. Same as nonlinear least squares.

DEMO: PROFILE LIKELIHOOD, LIKELIHOOD SURFACE, OPTIMIZERS

Estimating Parameters via the Likelihood

Assume (for simplicity) that the state, x, is a deterministic function of the parameters, i.e. $x = g(\theta)$.
Steps:

1. **Define the data model.** Example: For a normal distribution $[y|\theta] = L(\theta|y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{\frac{1}{2\sigma^2} (y - g(\theta))^2\}$
2. **Find the θ value that maximizes $L(\theta|y)$.** Example: With σ^2 constant, we maximize $J(\theta) = -(y - g(\theta))^2$ with respect to θ. Same as nonlinear least squares.

DEMO: PROFILE LIKELIHOOD, LIKELIHOOD SURFACE, OPTIMIZERS

Estimating Parameters via the Likelihood

Assume (for simplicity) that the state, x, is a deterministic function of the parameters, i.e. $x = g(\theta)$.
Steps:

1. **Define the data model.** Example: For a normal distribution $[y|\theta] = L(\theta|y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{\frac{1}{2\sigma^2}(y - g(\theta))^2\}$
2. **Find the θ value that maximizes $L(\theta|y)$.** Example: With σ^2 constant, we maximize $J(\theta) = -(y - g(\theta))^2$ with respect to θ. Same as nonlinear least squares.

DEMO: PROFILE LIKELIHOOD, LIKELIHOOD SURFACE, OPTIMIZERS

Estimating Parameters via the Likelihood

Assume (for simplicity) that the state, x, is a deterministic function of the parameters, i.e. $x = g(\theta)$.
Steps:

1. **Define the data model.** Example: For a normal distribution $[y|\theta] = L(\theta|y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{\frac{1}{2\sigma^2}(y - g(\theta))^2\}$
2. **Find the θ value that maximizes $L(\theta|y)$.** Example: With σ^2 constant, we maximize $J(\theta) = -(y - g(\theta))^2$ with respect to θ. Same as nonlinear least squares.

DEMO: PROFILE LIKELIHOOD, LIKELIHOOD SURFACE, OPTIMIZERS

Prior Information

The use of prior information for estimation is the salient (and unique) feature of Bayesian inference

- Probability distributions $[\theta]$ are specified for all of the parameters. They act as constraints on plausible parameter values (e.g. same way optimization uses bounding and ranges)
- They are based on expert knowledge. In ecology they are derived from lab and field experiments (i.e. the literature).
- Priors are classified as informative or non-informative (vague). You generally put a prior on everything.

DEMO: PRIORS, AND SAMPLING FROM PRIORS

Prior Information

The use of prior information for estimation is the salient (and unique) feature of Bayesian inference

- Probability distributions $[\theta]$ are specified for all of the parameters. They act as constraints on plausible parameter values (e.g. same way optimization uses bounding and ranges)
- They are based on expert knowledge. In ecology they are derived from lab and field experiments (i.e. the literature).
- Priors are classified as informative or non-informative (vague). You generally put a prior on everything.

DEMO: PRIORS, AND SAMPLING FROM PRIORS

Prior Information

The use of prior information for estimation is the salient (and unique) feature of Bayesian inference

- Probability distributions $[\theta]$ are specified for all of the parameters. They act as constraints on plausible parameter values (e.g. same way optimization uses bounding and ranges)
- They are based on expert knowledge. In ecology they are derived from lab and field experiments (i.e. the literature).
- Priors are classified as informative or non-informative (vague). You generally put a prior on everything.

DEMO: PRIORS, AND SAMPLING FROM PRIORS

Prior Information

The use of prior information for estimation is the salient (and unique) feature of Bayesian inference

- Probability distributions $[\theta]$ are specified for all of the parameters. They act as constraints on plausible parameter values (e.g. same way optimization uses bounding and ranges)
- They are based on expert knowledge. In ecology they are derived from lab and field experiments (i.e. the literature).
- Priors are classified as informative or non-informative (vague). You generally put a prior on everything.

DEMO: PRIORS, AND SAMPLING FROM PRIORS

Prior Information

The use of prior information for estimation is the salient (and unique) feature of Bayesian inference

- Probability distributions $[\theta]$ are specified for all of the parameters. They act as constraints on plausible parameter values (e.g. same way optimization uses bounding and ranges)
- They are based on expert knowledge. In ecology they are derived from lab and field experiments (i.e. the literature).
- Priors are classified as informative or non-informative (vague). You generally put a prior on everything.

DEMO: PRIORS, AND SAMPLING FROM PRIORS

Bayesian Computation

- Computational Bayesian approaches are concerned with solving the following equations (the BHM model):

$$[x, \theta|y] \propto [y|x, \theta] \cdot [x|\theta] \cdot [\theta]$$

  That is, determining the *posterior* using the *data model* , the *process model*, and the *prior distributions*

- Markov Chain Monte Carlo (MCMC) algorithms are used. These provide for sampling based solutions (they generate a samples that has has the property of being a draw from the target posterior). Statistics (e.g. mean, variance) can then be derived from the samples.

Bayesian Computation

- Computational Bayesian approaches are concerned with solving the following equations (the BHM model):

$$[x, \theta|y] \propto [y|x, \theta] \cdot [x|\theta] \cdot [\theta]$$

  That is, determining the *posterior* using the *data model* , the *process model*, and the *prior distributions*

- Markov Chain Monte Carlo (MCMC) algorithms are used. These provide for sampling based solutions (they generate a samples that has has the property of being a draw from the target posterior). Statistics (e.g. mean, variance) can then be derived from the samples.

Bayesian Computation

- Computational Bayesian approaches are concerned with solving the following equations (the BHM model):

  $$[x, \theta | y] \propto [y | x, \theta] \cdot [x | \theta] \cdot [\theta]$$

  That is, determining the *posterior* using the *data model* , the *process model*, and the *prior distributions*
- Markov Chain Monte Carlo (MCMC) algorithms are used. These provide for sampling based solutions (they generate a samples that has has the property of being a draw from the target posterior). Statistics (e.g. mean, variance) can then be derived from the samples.

MCMC: The Metropolis-Hasting Algorithm

- The Metropolis-Hasting algorithm is a popular, effective, and understandable MCMC algorithm for sample-based inference.
- It comprises a set of rules for generating a samples $\{x^{(i)}, \theta^{(i)}\}_{i=1}^{n}$ from the target posterior $[x, \theta|y]$ (i.e. which is the answer to the problem you are solving).
- The basic idea is to (intelligently) propose answers, then evaluate how probable they each are (relative to previous proposals).

Next , for simplicity we'll consider M-H MCMC using a deterministic system where $x = g(\theta)$, so the posterior $[x, \theta|y] = [\theta|y]$

MCMC: The Metropolis-Hasting Algorithm

- The Metropolis-Hasting algorithm is a popular, effective, and understandable MCMC algorithm for sample-based inference.
- It comprises a set of rules for generating a samples $\{x^{(i)}, \theta^{(i)}\}_{i=1}^{n}$ from the target posterior $[x, \theta|y]$ (i.e. which is the answer to the problem you are solving).
- The basic idea is to (intelligently) propose answers, then evaluate how probable they each are (relative to previous proposals).

Next , for simplicity we'll consider M-H MCMC using a deterministic system where $x = g(\theta)$, so the posterior $[x, \theta|y] = [\theta|y]$

MCMC: The Metropolis-Hasting Algorithm

- The Metropolis-Hasting algorithm is a popular, effective, and understandable MCMC algorithm for sample-based inference.
- It comprises a set of rules for generating a samples $\{x^{(i)}, \theta^{(i)}\}_{i=1}^{n}$ from the target posterior $[x, \theta|y]$ (i.e. which is the answer to the problem you are solving).
- The basic idea is to (intelligently) propose answers, then evaluate how probable they each are (relative to previous proposals).

Next , for simplicity we'll consider M-H MCMC using a deterministic system where $x = g(\theta)$, so the posterior $[x, \theta|y] = [\theta|y]$

MCMC: The Metropolis-Hasting Algorithm

- The Metropolis-Hasting algorithm is a popular, effective, and understandable MCMC algorithm for sample-based inference.
- It comprises a set of rules for generating a samples $\{x^{(i)}, \theta^{(i)}\}_{i=1}^{n}$ from the target posterior $[x, \theta|y]$ (i.e. which is the answer to the problem you are solving).
- The basic idea is to (intelligently) propose answers, then evaluate how probable they each are (relative to previous proposals).

Next , for simplicity we'll consider M-H MCMC using a deterministic system where $x = g(\theta)$, so the posterior $[x, \theta|y] = [\theta|y]$

MCMC: The Metropolis-Hasting Algorithm

- The Metropolis-Hasting algorithm is a popular, effective, and understandable MCMC algorithm for sample-based inference.
- It comprises a set of rules for generating a samples $\{x^{(i)}, \theta^{(i)}\}_{i=1}^{n}$ from the target posterior $[x, \theta|y]$ (i.e. which is the answer to the problem you are solving).
- The basic idea is to (intelligently) propose answers, then evaluate how probable they each are (relative to previous proposals).

Next , for simplicity we'll consider M-H MCMC using a deterministic system where $x = g(\theta)$, so the posterior $[x, \theta|y] = [\theta|y]$

Metropolis-Hastings Algorithm: Prior as Proposal, Independence sampler

Goal: estimate a sample from the posterior $[\theta|y]$ using the (i) observations y, (ii) the process model $x = g(\theta)$, and (ii) the prior $[\theta]$

Start with an initial sample member $\theta^{(0)}$

- For $i = 1, 2, \ldots, n$
  1. Draw a candidate θ^c from the prior $[\theta]$.
  2. Compute the acceptance probability $\alpha = \frac{[y|\theta^c]}{[y|\theta^{(i-1)}]} = \frac{L(\theta^c|y)}{L(\theta^{(i-1)}|y)}$.
  3. Accept $\theta^{(i)} = \theta^c$ with probability $\alpha^* = min\{\alpha, 1\}$, otherwise $\theta^{(i)} = \theta^{(i-1)}$
- Yields the sample $\{\theta^{(i)}\}_{i=1}^n$

This is perhaps the simplest M-H algorithm, but not the best. Why? Prior is not ideal proposal, no memory effects in generating sample

DEMO: M-H INDEPENDENCE SAMPLER

Metropolis-Hastings Algorithm: Prior as Proposal, Independence sampler

Goal: estimate a sample from the posterior $[\theta|y]$ using the (i) observations y, (ii) the process model $x = g(\theta)$, and (ii) the prior $[\theta]$
Start with an initial sample member $\theta^{(0)}$

- For $i = 1, 2, \ldots, n$
  1. Draw a candidate θ^c from the prior $[\theta]$.
  2. Compute the acceptance probability $\alpha = \frac{[y|\theta^c]}{[y|\theta^{(i-1)}]} = \frac{L(\theta^c|y)}{L(\theta^{(i-1)}|y)}$.
  3. Accept $\theta^{(i)} = \theta^c$ with probability $\alpha^* = min\{\alpha, 1\}$, otherwise $\theta^{(i)} = \theta^{(i-1)}$
- Yields the sample $\{\theta^{(i)}\}_{i=1}^n$

This is perhaps the simplest M-H algorithm, but not the best. Why?
Prior is not ideal proposal, no memory effects in generating sample

DEMO: M-H INDEPENDENCE SAMPLER

Metropolis-Hastings Algorithm: Prior as Proposal, Independence sampler

Goal: estimate a sample from the posterior $[\theta|y]$ using the (i) observations y, (ii) the process model $x = g(\theta)$, and (ii) the prior $[\theta]$
Start with an initial sample member $\theta^{(0)}$

- For $i = 1, 2, \ldots, n$
  1. Draw a candidate θ^c from the prior $[\theta]$.
  2. Compute the acceptance probability $\alpha = \frac{[y|\theta^c]}{[y|\theta^{(i-1)}]} = \frac{L(\theta^c|y)}{L(\theta^{(i-1)}|y)}$.
  3. Accept $\theta^{(i)} = \theta^c$ with probability $\alpha^* = min\{\alpha, 1\}$, otherwise $\theta^{(i)} = \theta^{(i-1)}$
- Yields the sample $\{\theta^{(i)}\}_{i=1}^n$

This is perhaps the simplest M-H algorithm, but not the best. Why?
Prior is not ideal proposal, no memory effects in generating sample

DEMO: M-H INDEPENDENCE SAMPLER

Metropolis-Hastings Algorithm: Prior as Proposal, Independence sampler

Goal: estimate a sample from the posterior $[\theta|y]$ using the (i) observations y, (ii) the process model $x = g(\theta)$, and (ii) the prior $[\theta]$
Start with an initial sample member $\theta^{(0)}$

- For $i = 1, 2, \ldots, n$
  1. Draw a candidate θ^c from the prior $[\theta]$.
  2. Compute the acceptance probability $\alpha = \frac{[y|\theta^c]}{[y|\theta^{(i-1)}]} = \frac{L(\theta^c|y)}{L(\theta^{(i-1)}|y)}$.
  3. Accept $\theta^{(i)} = \theta^c$ with probability $\alpha^* = min\{\alpha, 1\}$, otherwise $\theta^{(i)} = \theta^{(i-1)}$
- Yields the sample $\{\theta^{(i)}\}_{i=1}^n$

This is perhaps the simplest M-H algorithm, but not the best. Why?
Prior is not ideal proposal, no memory effects in generating sample

DEMO: M-H INDEPENDENCE SAMPLER

Metropolis-Hastings Algorithm: Prior as Proposal, Independence sampler

Goal: estimate a sample from the posterior $[\theta|y]$ using the (i) observations y, (ii) the process model $x = g(\theta)$, and (ii) the prior $[\theta]$
Start with an initial sample member $\theta^{(0)}$

- For $i = 1, 2, \ldots, n$
  1. Draw a candidate θ^c from the prior $[\theta]$.
  2. Compute the acceptance probability $\alpha = \frac{[y|\theta^c]}{[y|\theta^{(i-1)}]} = \frac{L(\theta^c|y)}{L(\theta^{(i-1)}|y)}$.
  3. Accept $\theta^{(i)} = \theta^c$ with probability $\alpha^* = min\{\alpha, 1\}$, otherwise $\theta^{(i)} = \theta^{(i-1)}$
- Yields the sample $\{\theta^{(i)}\}_{i=1}^n$

This is perhaps the simplest M-H algorithm, but not the best. Why?
Prior is not ideal proposal, no memory effects in generating sample

DEMO: M-H INDEPENDENCE SAMPLER

Metropolis-Hastings Algorithm: Prior as Proposal, Independence sampler

Goal: estimate a sample from the posterior $[\theta|y]$ using the (i) observations y, (ii) the process model $x = g(\theta)$, and (ii) the prior $[\theta]$
Start with an initial sample member $\theta^{(0)}$

- For $i = 1, 2, \ldots, n$
  1. Draw a candidate θ^c from the prior $[\theta]$.
  2. Compute the acceptance probability $\alpha = \frac{[y|\theta^c]}{[y|\theta^{(i-1)}]} = \frac{L(\theta^c|y)}{L(\theta^{(i-1)}|y)}$.
  3. Accept $\theta^{(i)} = \theta^c$ with probability $\alpha^* = min\{\alpha, 1\}$, otherwise $\theta^{(i)} = \theta^{(i-1)}$
- Yields the sample $\{\theta^{(i)}\}_{i=1}^{n}$

This is perhaps the simplest M-H algorithm, but not the best. Why? Prior is not ideal proposal, no memory effects in generating sample

DEMO: M-H INDEPENDENCE SAMPLER

Metropolis-Hastings Algorithm: Random Walk Sampler

Goal: estimate a sample from the posterior $[\theta|y]$ using the (i) observations y, (ii) the process model $x = g(\theta)$, and (ii) the prior $[\theta]$

Start with an initial sample member $\left\{\theta^{(0)}\right\}$

- For $i = 1, 2, \ldots, n$
  1. Draw a candidate $\theta^c = \theta^{(i-1)} + \epsilon$ where $\epsilon \sim N(0, \sigma_\epsilon^2)$
  2. Compute the acceptance probability $\alpha = \frac{[y|\theta^c][\theta=\theta^c]}{[y|\theta^{(i-1)}][\theta=\theta^{(i-1)}]}$.
  3. Accept $\theta^{(i)} = \theta^c$ with probability $\alpha^* = min\{\alpha, 1\}$, otherwise $\theta^{(i)} = \theta^{(i-1)}$
- Yields the sample $\{\theta^{(i)}\}_{i=1}^n$

The parameter random walk allow effective exploration for the posterior. Key quantity is the random walk variance ϵ

DEMO: M-H RANDOM WALK SAMPLER

Metropolis-Hastings Algorithm: Random Walk Sampler

Goal: estimate a sample from the posterior $[\theta|y]$ using the (i) observations y, (ii) the process model $x = g(\theta)$, and (ii) the prior $[\theta]$
Start with an initial sample member $\left\{\theta^{(0)}\right\}$

- For $i = 1, 2, \ldots, n$
  1. Draw a candidate $\theta^c = \theta^{(i-1)} + \epsilon$ where $\epsilon \sim N(0, \sigma_\epsilon^2)$
  2. Compute the acceptance probability $\alpha = \frac{[y|\theta^c][\theta=\theta^c]}{[y|\theta^{(i-1)}][\theta=\theta^{(i-1)}]}$.
  3. Accept $\theta^{(i)} = \theta^c$ with probability $\alpha^* = min\{\alpha, 1\}$, otherwise $\theta^{(i)} = \theta^{(i-1)}$
- Yields the sample $\{\theta^{(i)}\}_{i=1}^n$

The parameter random walk allow effective exploration for the posterior. Key quantity is the random walk variance ϵ

DEMO: M-H RANDOM WALK SAMPLER

Metropolis-Hastings Algorithm: Random Walk Sampler

Goal: estimate a sample from the posterior $[\theta|y]$ using the (i) observations y, (ii) the process model $x = g(\theta)$, and (ii) the prior $[\theta]$
Start with an initial sample member $\left\{\theta^{(0)}\right\}$

- For $i = 1, 2, \ldots, n$
  1. Draw a candidate $\theta^c = \theta^{(i-1)} + \epsilon$ where $\epsilon \sim N(0, \sigma_\epsilon^2)$
  2. Compute the acceptance probability $\alpha = \frac{[y|\theta^c][\theta=\theta^c]}{[y|\theta^{(i-1)}][\theta=\theta^{(i-1)}]}$.
  3. Accept $\theta^{(i)} = \theta^c$ with probability $\alpha^* = min\{\alpha, 1\}$, otherwise $\theta^{(i)} = \theta^{(i-1)}$
- Yields the sample $\{\theta^{(i)}\}_{i=1}^n$

The parameter random walk allow effective exploration for the posterior.
Key quantity is the random walk variance ϵ

DEMO: M-H RANDOM WALK SAMPLER

Metropolis-Hastings Algorithm: Random Walk Sampler

Goal: estimate a sample from the posterior $[\theta|y]$ using the (i) observations y, (ii) the process model $x = g(\theta)$, and (ii) the prior $[\theta]$
Start with an initial sample member $\left\{\theta^{(0)}\right\}$

- For $i = 1, 2, \ldots, n$
  1. Draw a candidate $\theta^c = \theta^{(i-1)} + \epsilon$ where $\epsilon \sim N(0, \sigma_\epsilon^2)$
  2. Compute the acceptance probability $\alpha = \frac{[y|\theta^c][\theta=\theta^c]}{[y|\theta^{(i-1)}][\theta=\theta^{(i-1)}]}$.
  3. Accept $\theta^{(i)} = \theta^c$ with probability $\alpha^* = min\{\alpha, 1\}$, otherwise $\theta^{(i)} = \theta^{(i-1)}$
- Yields the sample $\{\theta^{(i)}\}_{i=1}^n$

The parameter random walk allow effective exploration for the posterior. Key quantity is the random walk variance ϵ

DEMO: M-H RANDOM WALK SAMPLER

Metropolis-Hastings Algorithm: Random Walk Sampler

Goal: estimate a sample from the posterior $[\theta|y]$ using the (i) observations y, (ii) the process model $x = g(\theta)$, and (ii) the prior $[\theta]$
Start with an initial sample member $\left\{\theta^{(0)}\right\}$

- For $i = 1, 2, \ldots, n$
  1. Draw a candidate $\theta^c = \theta^{(i-1)} + \epsilon$ where $\epsilon \sim N(0, \sigma_\epsilon^2)$
  2. Compute the acceptance probability $\alpha = \frac{[y|\theta^c][\theta=\theta^c]}{[y|\theta^{(i-1)}][\theta=\theta^{(i-1)}]}$.
  3. Accept $\theta^{(i)} = \theta^c$ with probability $\alpha^* = min\{\alpha, 1\}$, otherwise $\theta^{(i)} = \theta^{(i-1)}$
- Yields the sample $\{\theta^{(i)}\}_{i=1}^n$

The parameter random walk allow effective exploration for the posterior.
Key quantity is the random walk variance ϵ

DEMO: M-H RANDOM WALK SAMPLER

Some Practical issues MCMC

Choice of Proposal: the proposal sets how efficiently and effectively the chains are able to sample from the target posterior.

Key Performance Diagnostics:

- **Burn-in**: The chain is not sampling from the posterior until the effect of the initial conditions is forgotten. *Discard first part of sample*
- **Convergence**: The chain must be in a statistical steady state to be sampling from the posterior *Assess stationarity, stability of the statistical moments*
- **Mixing**: the chain must effectively and fully explore the region of parameter space where the posterior density in non-negligible. *Time series properties of chain such as autocorrelation.*

Some Practical issues MCMC

Choice of Proposal: the proposal sets how efficiently and effectively the chains are able to sample from the target posterior.

Key Performance Diagnostics:

- **Burn-in**: The chain is not sampling from the posterior until the effect of the initial conditions is forgotten. *Discard first part of sample*
- **Convergence**: The chain must be in a statistical steady state to be sampling from the posterior *Assess stationarity, stability of the statistical moments*
- **Mixing**: the chain must effectively and fully explore the region of parameter space where the posterior density in non-negligible. *Time series properties of chain such as autocorrelation.*

Some Practical issues MCMC

Choice of Proposal: the proposal sets how efficiently and effectively the chains are able to sample from the target posterior.

Key Performance Diagnostics:

- **Burn-in**: The chain is not sampling from the posterior until the effect of the initial conditions is forgotten. *Discard first part of sample*
- **Convergence**: The chain must be in a statistical steady state to be sampling from the posterior *Assess stationarity, stability of the statistical moments*
- **Mixing**: the chain must effectively and fully explore the region of parameter space where the posterior density in non-negligible. *Time series properties of chain such as autocorrelation.*

Some Practical issues MCMC

Choice of Proposal: the proposal sets how efficiently and effectively the chains are able to sample from the target posterior.

Key Performance Diagnostics:

- **Burn-in**: The chain is not sampling from the posterior until the effect of the initial conditions is forgotten. *Discard first part of sample*
- **Convergence**: The chain must be in a statistical steady state to be sampling from the posterior *Assess stationarity, stability of the statistical moments*
- **Mixing**: the chain must effectively and fully explore the region of parameter space where the posterior density in non-negligible. *Time series properties of chain such as autocorrelation.*

Some Practical issues MCMC

Choice of Proposal: the proposal sets how efficiently and effectively the chains are able to sample from the target posterior.

Key Performance Diagnostics:

- **Burn-in**: The chain is not sampling from the posterior until the effect of the initial conditions is forgotten. *Discard first part of sample*
- **Convergence**: The chain must be in a statistical steady state to be sampling from the posterior *Assess stationarity, stability of the statistical moments*
- **Mixing**: the chain must effectively and fully explore the region of parameter space where the posterior density in non-negligible. *Time series properties of chain such as autocorrelation.*

Some Practical issues MCMC

Choice of Proposal: the proposal sets how efficiently and effectively the chains are able to sample from the target posterior.

Key Performance Diagnostics:

- **Burn-in**: The chain is not sampling from the posterior until the effect of the initial conditions is forgotten. *Discard first part of sample*
- **Convergence**: The chain must be in a statistical steady state to be sampling from the posterior *Assess stationarity, stability of the statistical moments*
- **Mixing**: the chain must effectively and fully explore the region of parameter space where the posterior density in non-negligible. *Time series properties of chain such as autocorrelation*.