ASTR 3890 - Selected Topics: Data Science for Large Astronomical Surveys (Spring 2022)

Bayesian Statistical Inference: II

Dr. Nina Hernitschek
March 14, 2022

Bayesian and Frequentist Model Comparison

Model comparison and hypothesis testing in Bayesian inference are enormously different from classical/frequentist statistics.

In **frequentist inference**, we used measures such as χ^2 per degree of freedom (χ^2_{dof}).
Frequentists compare the model likelihood, $P(D|M_1)$ and $P(D|M_2)$.

In **Bayesian inference**, we probabilistically **rank models based on their ability to explain the data under our prior knowledge**.
Bayesians compare the model posterior, $P(M_1|D)$ and $P(M_2|D)$.

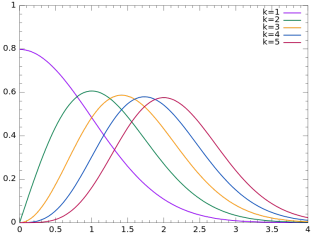
Frequentist Model Comparison

Use χ^2 per degree of freedom to determine which fit is "best":

$$\chi^2_{\mathrm{dof}} = \frac{1}{N-k} \sum_i^N \left(\frac{y - y_{\mathrm{fit}}}{\sigma_y} \right)^2 ,$$

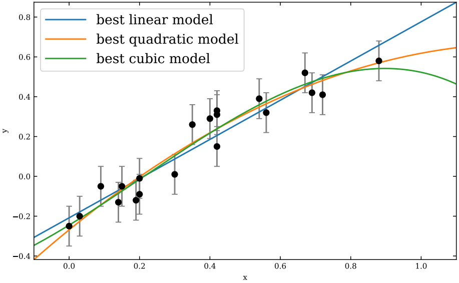
where N is the number of data points and k is the number of free model parameters.

For $(N-k) > 10$, the distribution of χ^2 per degre of freedom is approximately Gaussian with a width of $\sigma = \sqrt{2/(N-k)}$.



Frequentist Model Comparison

An example:

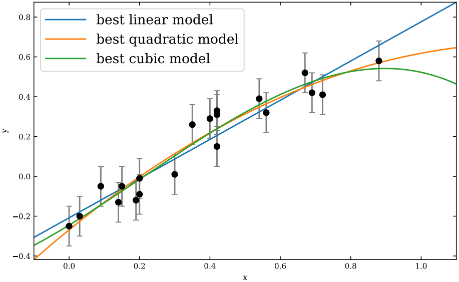


The cubic model has the lowest $\chi2$, but has 4 free parameters while the linear model has only 2 free parameters.

How do we trade χ^2 vs. increasing model complexity?

Frequentist Model Comparison

An example:



The cubic model has the lowest $\chi 2$, but has 4 free parameters while the linear model has only 2 free parameters.
How do we trade χ^2 vs. increasing model complexity?

Occam's razor: All else being equal (i.e., each model fits the data equally well), the less complex model is favored.

Bayesian Model Comparison

For the **Bayesian** approach to model comparison, we start with Bayes' Theorem,

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

$$p(M, \theta \mid D, I) = \frac{p(D \mid M, \theta, I) \times p(M, \theta \mid I)}{p(D \mid I)},$$

Bayesian Model Comparison

For the **Bayesian** approach to model comparison, we start with Bayes' Theorem,

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

$$p(M, \theta \,|\, D, I) = \frac{p(D \,|\, M, \theta, I) \; \times \; p(M, \theta \,|\, I)}{p(D \,|\, I)},$$

and marginalize over model parameter space θ to obtain the **probability of model** M given data D and prior information I:

$$\begin{aligned} p(M \,|\, D, I) &\equiv \int p(M, \theta \,|\, D, I) \, d\theta \\ &= \int \frac{p(D \,|\, M, \theta, I) \, p(M, \theta \,|\, I)}{p(D \,|\, I)} \, d\theta \\ &= \frac{p(M \,|\, I)}{p(D \,|\, I)} \int p(D \,|\, M, \theta, I) \, p(\theta \,|\, M, I) \, d\theta \end{aligned}$$

Bayesian Model Comparison

To then determine which of two models is better we compute the ratio of the posterior probabilities or the **odds ratio** as

$$O_{21} \equiv \frac{p(M_2|D,I)}{p(M_1|D,I)}.$$

The posterior probability that the model M is correct given data D (a number between 0 and 1) is

$$p(M|D,I) = \frac{p(D|M,I)p(M|I)}{p(D|I)}.$$

We finally get for the odds ratio:

$$O_{21} = \boxed{\frac{p(D \mid M_2, I)}{p(D \mid M_1, I)}} \frac{p(M_2 \mid I)}{p(M_1 \mid I)} \equiv B_{21} \frac{p(M_2 \mid I)}{p(M_1 \mid I)},$$

where B_{21} is called the **Bayes factor**.

Bayesian Model Comparison

We finally get for the odds ratio:

$$O_{21} = \frac{p(D \mid M_2, I)}{p(D \mid M_1, I)} \frac{p(M_2 \mid I)}{p(M_1 \mid I)} \equiv B_{21} \frac{p(M_2 \mid I)}{p(M_1 \mid I)},$$

where B_{21} is called the **Bayes factor**.

The Bayes factor compares how well the models fit the data. It is a ratio of data likelihoods averaged over all allowed values of the model parameters. For models fitting the data equally well, decision is made based on the priors.

Bayesian Model Comparison

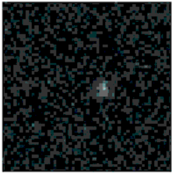
We finally get for the odds ratio:

$$O_{21} = \frac{p(D \mid M_2, I)\, p(M_2 \mid I)}{p(D \mid M_1, I)\, p(M_1 \mid I)} \equiv B_{21}\, \frac{p(M_2 \mid I)}{p(M_1 \mid I)},$$

where B_{21} is called the **Bayes factor**.

The Bayes factor compares how well the models fit the data. It is a ratio of data likelihoods averaged over all allowed values of the model parameters. For models fitting the data equally well, decision is made based on the priors.

example: Consider a noisy image of a source which is equally likely to be a star or a galaxy. The posterior probability that the source is a star will greatly depend on whether we are looking at the Galactic plane or not.



Bayesian Model Comparison

We can compute

$$E(M) \equiv p(D \mid M, I) = \int p(D \mid M, \theta, I)\, p(\theta \mid M, I)\, d\theta,$$

where $E(M)$ is called the **marginal likelihood for model** M (or **evidence** or textitfully marginalized likelihood). It quantifies the probability that the data D would be observed if the model M were the correct model.

Bayesian Model Comparison

We can compute

$$E(M) \equiv p(D \mid M, I) = \int p(D \mid M, \theta, I)\, p(\theta \mid M, I)\, d\theta,$$

where $E(M)$ is called the **marginal likelihood for model** M (or **evidence** or textitfully marginalized likelihood). It quantifies the probability that the data D would be observed if the model M were the correct model.

The evidence is a weighted average of the likelihood function, where the prior for model parameters acts as the weighting function.

Bayesian Model Comparison

How do we **interpret** the values of the odds ratio in practice?

Bayesian Model Comparison

How do we **interpret** the values of the odds ratio in practice?

Jeffreys (1936, 1961) proposed a scale for interpreting the odds ratio, where $O_{21} > 10$ represents strong evidence in favor of M_2 (M_2 is ten times more probable than M_1), and $O_{21} > 100$ is decisive evidence (M_2 is one hundred times more probable than M_1). When $O_{21} < 3$, the evidence is not worth more than a bare mention.

Bayesian Model Comparison

How do we **interpret** the values of the odds ratio in practice?

Jeffreys (1936, 1961) proposed a scale for interpreting the odds ratio, where $O_{21} > 10$ represents strong evidence in favor of M_2 (M_2 is ten times more probable than M_1), and $O_{21} > 100$ is decisive evidence (M_2 is one hundred times more probable than M_1). When $O_{21} < 3$, the evidence is not worth more than a bare mention.

caution:

- These are just definitions of conventions, i.e., a way to give a quantitative meaning to qualitative phrases.
- The odds ratio compares the models, it doesn't tell us about the absolute goodness of fit.
  Model A can be 100× better than model B, but still don't fit the data well.

Hypothesis Testing

In **hypothesis testing** we are essentially comparing a model, M_1, to its complement, that is $p(M_1) + p(M_2) = 1$.
If we take M_1 to be the **null hypothesis** (which is generally that, for example, a correlation does not exist), then we are asking whether or not the **data reject the null hypothesis**.

Hypothesis Testing

In **hypothesis testing** we are essentially comparing a model, M_1, to its complement, that is $p(M_1) + p(M_2) = 1$.
If we take M_1 to be the **null hypothesis** (which is generally that, for example, a correlation does not exist), then we are asking whether or not the **data reject the null hypothesis**.

In classical hypothesis testing we can ask whether or not a **single model** provides a good description of the data.
In Bayesian hypothesis testing, we need to have an **alternative model** to compare to.

Hypothesis Testing

example: Coin Flip Bayesian Model Comparison

Let's assume we have N draws and k are heads.

Hypothesis Testing

example: Coin Flip Bayesian Model Comparison

Let's assume we have N draws and k are heads.

We compare two hypotheses:

- M1: the coin has a known heads probability b_* (say, a fair coin with $b_* = 0.5$), with a prior given by a delta function, $\delta(b - b_*)$,
- M2: the heads probability b is unknown, with a uniform prior in the range 0-1.

Hypothesis Testing

example: Coin Flip Bayesian Model Comparison

Let's assume we have N draws and k are heads.

We compare two hypotheses:

- M1: the coin has a known heads probability b_* (say, a fair coin with $b_* = 0.5$), with a prior given by a delta function, $\delta(b - b_*)$,
- M2: the heads probability b is unknown, with a uniform prior in the range 0-1.

The model is the binomial distribution, parametrized by the probability of success b, with k successes:

$$p(k \,|\, b, N) = \frac{N!}{k!\,(N-k)!}\, b^k\,(1-b)^{N-k}$$

For model M2, the prior for b is flat in the range 0-1 and the product of the data likelihood and prior is same as above.

Hypothesis Testing

example: Coin Flip Bayesian Model Comparison
The model is the binomial distribution:

$$p(k \,|\, b, N) = \frac{N!}{k!\,(N-k)!}\, b^k\, (1-b)^{N-k}$$

prior for M2: flat in the range 0-1 and the product of the data likelihood and prior is $p(k \,|\, b, N)$.
prior for M1: delta function $\delta(b - b_*) \Rightarrow$ product of the data likelihood and prior (which picks out $b = b_*$) is

$$p(k \,|\, b_*, N, M1)\, p(b|M1, I) = \frac{N!}{k!\,(N-k)!}\, b_*^k\, (1-b_*)^{N-k}.$$

Consequently, the **odds ratio** is given by

$$O_{21} = \int_0^1 \left(\frac{b}{b_*}\right)^k \left(\frac{1-b}{1-b_*}\right)^{N-k} db$$

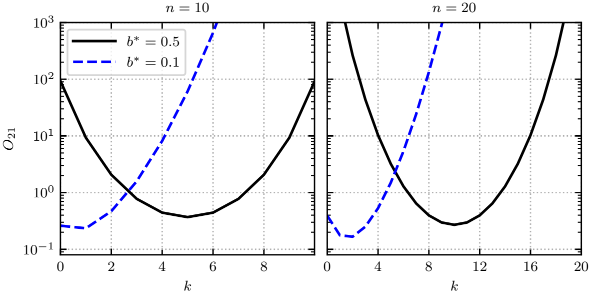
Hypothesis Testing

example: Coin Flip Bayesian Model Comparison

Consequently, the **odds ratio** is given by

$$O_{21} = \int_0^1 \left(\frac{b}{b_*}\right)^k \left(\frac{1-b}{1-b_*}\right)^{N-k} db,$$

as illustrated in the following figure.



Approximate Bayesian Model Comparison

The full odds ratio can be costly to compute ⇒ **approximate methods** that balance between *goodness of fit* and *model complexity*.

Approximate Bayesian Model Comparison

Akaike information criterion (AIC)

$$\mathrm{AIC} \equiv -2\ln[L_0(M)] + 2k + \frac{2k(k+1)}{N-k-1}.$$

with

k: number of model parameters

$L_0(M)$: maximum value of the likelihood function

The term $\frac{2k(k+1)}{N-k-1}$ is sometimes ignored.

The **preferred model** is the one with the minimum AIC value. AIC rewards goodness of fit (as assessed by the likelihood function), but also includes a penalty for an increasing number of parameters to discourage overfitting.

Approximate Bayesian Model Comparison

Bayesian information criterion (BIC)

The BIC can be derived from the Bayesian odds ratio **by assuming that the likelihood is Gaussian**.
$\Rightarrow$ easier to compute than the odds ratio as it is based on the maximum value of the likelihood, $L_0(M)$, rather than on the integration of the likelihood over the full parameter space (i.e. evidence $E(M)$).

The BIC is for N data points and a model with k parameters:

$$\mathrm{BIC} \equiv -2 \ln[L_0(M)] + k \ln N.$$

The 1^{st} term is equal to the model's χ^2 (under the assumption of normality; note that this is not χ^2 per degree of freedom!). The 2^{nd} term on the right hand side penalizes complex models relative to simple ones.

Approximate Bayesian Model Comparison

When two models are compared, the model with the smaller BIC/AIC value wins.
If the models are equally successful in describing the data (i.e., they have the same value of $L_0(M)$), then the model with fewer free parameters wins (Occam's razor).

Approximate Bayesian Model Comparison

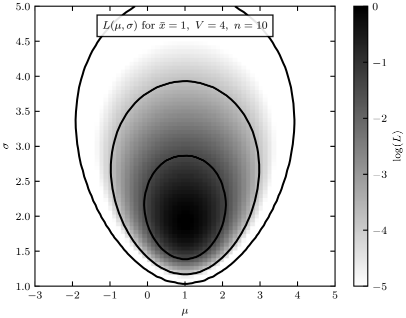
When two models are compared, the model with the smaller BIC/AIC value wins.
If the models are equally successful in describing the data (i.e., they have the same value of $L_0(M)$), then the model with fewer free parameters wins (Occam's razor).

caution:
Both BIC and AIC are **approximations** and might not be valid if the underlying assumptions (e.g.: Gaussian likelihood) are not met.
$\Rightarrow$ If computationally feasible, compute the odds ratio.

Monte Carlo Methods & Markov Chains

motivation:
lecture 5: estimating location and scale parameters for homoscedastic data drawn from a Gaussian distribution - two-dimensional posterior pdf for μ and σ

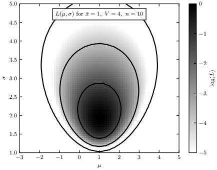
solution: numerically integrate the posterior pdf (easy in this case) and find its maximum by using a brute-force grid search



Monte Carlo Methods & Markov Chains

motivation:
lecture 5: estimating location and scale parameters for homoscedastic data drawn from a Gaussian distribution - two-dimensional posterior pdf for μ and σ

solution: numerically integrate the posterior pdf (easy in this case) and find its maximum by using a brute-force grid search

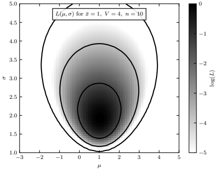


Feasible in this case: With 100 grid points per coordinate it was only 100 × 100 values.

Monte Carlo Methods & Markov Chains

motivation:
lecture 5: estimating location and scale parameters for homoscedastic data drawn from a Gaussian distribution - two-dimensional posterior pdf for μ and σ

solution: numerically integrate the posterior pdf (easy in this case) and find its maximum by using a brute-force grid search



Feasible in this case: With 100 grid points per coordinate it was only 100 × 100 values.

But what about high-dimensional parameter spaces?

Monte Carlo Methods & Markov Chains

motivation:

tricky situations:

- high-dimensional parameter space

computing time scales exponentially with number of parameters k

- spiky distributions

for odd shaped posterior distributions, brute-force grid methods can either totally miss the maximum L, or the grid needs to be very fine

Monte Carlo Methods & Markov Chains

motivation:

tricky situations:

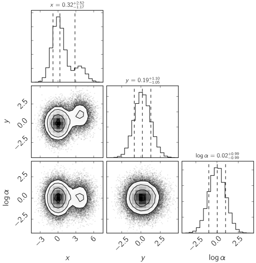
- high-dimensional parameter space

computing time scales exponentially with number of parameters k

- spiky distributions

for odd shaped posterior distributions, brute-force grid methods can either totally miss the maximum L, or the grid needs to be very fine

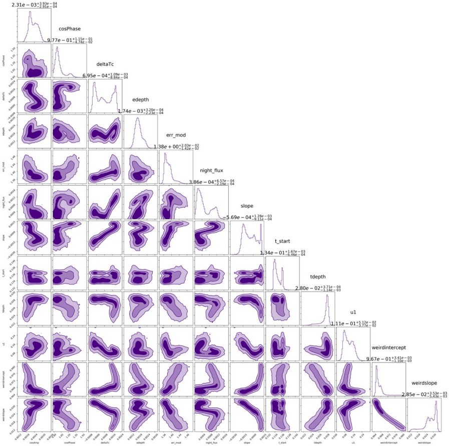
example realistic posterior distribution (I):



Monte Carlo Methods & Markov Chains

motivation:

example realistic posterior distribution (II):



Monte Carlo Methods & Markov Chains

motivation:

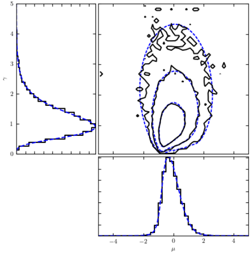
a better way to solve this is an algorithm that **samples the full multi-dimensional parameter space**, in a way that builds up the most sample density in regions that are closest to the maximum probability

Monte Carlo Methods & Markov Chains

motivation:

a better way to solve this is an algorithm that **samples the full multi-dimensional parameter space**, in a way that builds up the most sample density in regions that are closest to the maximum probability

exactly this is the **Markov Chain Monte Carlo (MCMC)** algorithm

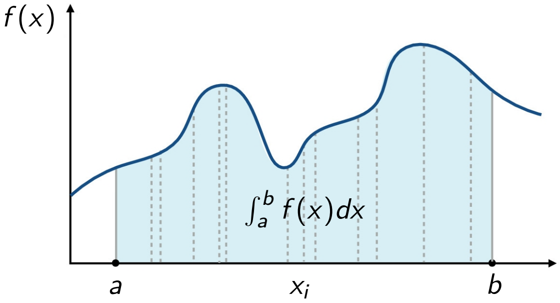


The dashed lines are the known (analytic) solution. The solid lines are from the MCMC estimate with 10,000 sample points.

Monte Carlo Methods

Algorithms from the family of **Monte Carlo methods** use **random sampling** to obtain a numerical result where there is no analytic result or it is difficult to compute.

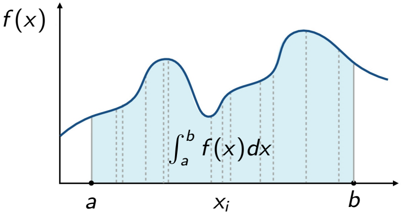
Monte Carlo integration:
simple idea: estimate the integral of a function by averaging random samples of the function's value



Monte Carlo Methods

Monte Carlo integration:

simple idea: estimate the integral of a function by averaging random samples of the function's value



define integral:

$\int_a^b f(x)dx$

random variable: $X_i \sim p(x)$

note: $p(x)$ must be nonzero for all x where $f(x)$ is nonzero

Monte Carlo estimator

$$F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}$$

Monte Carlo Methods

Monte Carlo integration:
simple idea: estimate the integral of a function by averaging random samples of the function's value

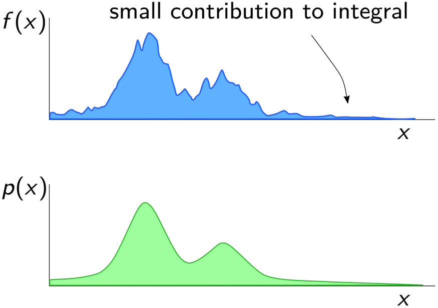
improvement: sample the integrand according to how much we expect it to contribute to the integral

⇨ **Importance Sampling**

Monte Carlo Methods

Importance Sampling:

improvement: sample the integrand according to how much we expect it to contribute to the integral



inefficient to sample here due to small contribution to integral

Basic Monte Carlo

$$F_N = \frac{b-a}{N} \sum_{i=1}^{N} f(X_i)$$

where x_i are sampled uniformly

Importance-Sampled Monte Carlo

$$F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

where x_i are sampled proportional to p

Markov-Chain Monte Carlo Methods

A **Markov Chain** is defined as

a sequence of random variables where a parameter depends only on the preceding value. Such processes are **memoryless**.

Markov-Chain Monte Carlo Methods

A **Markov Chain** is defined as

a sequence of random variables where a parameter depends only on the preceding value. Such processes are **memoryless**.

We thus have

$$p(\theta_{i+1}|\theta_i, \theta_{i-1}, \theta_{i-2}, \cdots) = p(\theta_{i+1}|\theta_i).$$

For equilibrium, or a stationary distribution of positions, it is necessary that the transition probability is symmetric:

$$p(\theta_{i+1}|\,\theta_i) = p(\theta_i|\,\theta_{i+1}).$$

This is called the **principle of detailed balance** or reversibility condition (i.e. the probability of a jump between two points does not depend on the direction of the jump).

Markov-Chain Monte Carlo Methods

The use of Markov chains to perform Monte Carlo integration is called **Markov Chain Monte Carlo (MCMC)**.

Given such a Markov chain of length N that corresponds to draws of $p(\theta)$, integrals can be estimated as

$$\int g(\theta)\, p(\theta)\, d\theta \approx \frac{1}{N} \sum_{i=1}^{N} g(\theta_i).$$

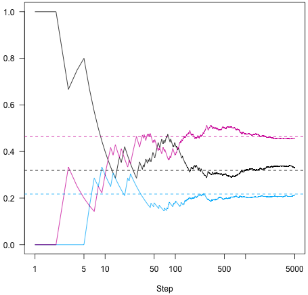
To estimate the expectation value for θ_1 (i.e., $g(\theta) = \theta_1$), we simply take the mean value of all θ_1 in the chain.

To visualize the posterior pdf for parameter θ_1, marginalized over all other parameters, $\theta_2, \ldots, \theta_k$, we can construct a histogram of all θ_1 values in the chain, and normalize its integral to 1.

To get an estimate for θ_1, we find the maximum of this marginalized pdf.

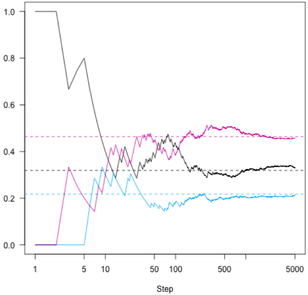
Markov-Chain Monte Carlo Methods

trace plot of a Markov Chain:



Markov-Chain Monte Carlo Methods

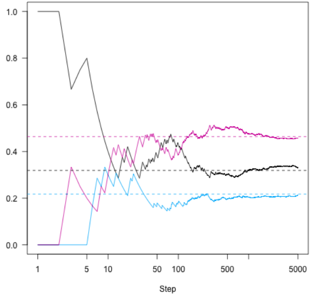
trace plot of a Markov Chain:



In MCMC the process must be **stationary**: the chain statistics look the same in each part of the chain.
Obviously that isn't going to be the case in the early steps of the chain (see plot).

Markov-Chain Monte Carlo Methods

trace plot of a Markov Chain:



In MCMC the process must be **stationary**: the chain statistics look the same in each part of the chain.
Obviously that isn't going to be the case in the early steps of the chain (see plot).

solution: discard the first few steps, the ***burn-in phase***.
How many steps to discard? Make a trace plot.

The Metropolis-Hastings Algorithm

The Metropolis-Hastings Algorithm is the most commonly used algorithm for MCMC sampling.
It adopts the following **acceptance probability** for newly proposed points θ^* to step to:

$$p_{\mathrm{acc}}(\theta_i, \theta^*) = \frac{p(\theta^*)}{p(\theta_i)},$$

where the proposed point θ^* is drawn from an arbitrary symmetric density distribution $q(\theta^* \mid \theta_i)$. Since it is symmetric, the **ratio of transition probabilities cancels out and detailed balance is ensured**. A Gaussian distribution centered on the current point θ_i is often used for $q(\theta^*|\theta_i)$.

This algorithm guarantees that the chain will reach an **equilibrium, or stationary, distribution**, and it will approximate a sample drawn from $p(\theta)$.

The Metropolis-Hastings Algorithm

Algorithm 1: Metropolis-Hastings MCMC

```
Input: $p(\theta)$: probability distribution
       $q(\theta)$: proposal distribution
       $N_{\rm iter}$: number of sample iterations
Output: $\{\theta_i\}$: chain of samples
begin
    initialization $\theta_0$
    for $i = 1, ..., N_{\rm iter}$ do
        sample $\theta^* \in q(\theta^*|\theta_i)$
        $p_{\rm acc} = \min\left(1, \frac{p(\theta^*)q(\theta_i|\theta^*)}{p(\theta_i)q(\theta^*|\theta_i)}\right)$
        sample $u \in U[0, 1]$
        if $u < p_{\rm acc}$ then
            $\theta_{i+1} = \theta^*$
        else
            $\theta_{i+1} = \theta_i$
```

The Metropolis-Hastings Algorithm

Algorithm 2: Metropolis-Hastings MCMC

```
Input: $p(\theta)$: probability distribution
       $q(\theta)$: proposal distribution
       $N_{\mathrm{iter}}$: number of sample iterations
Output: $\{\theta_i\}$: chain of samples
begin
    initialization $\theta_0$
    for $i = 1, \ldots, N_{\mathrm{iter}}$ do
        sample $\theta^* \in q(\theta^*|\theta_i)$
        $p_{\mathrm{acc}} = \min\left(1, \frac{p(\theta^*)q(\theta_i|\theta^*)}{p(\theta_i)q(\theta^*|\theta_i)}\right)$
        sample $u \in U[0,1]$
        if $u < p_{\mathrm{acc}}$ then
            $\theta_{i+1} = \theta^*$
        else
            $\theta_{i+1} = \theta_i$
```

Given θ_i and $q(\theta_{i+1}|\theta_i)$, draw θ^* as proposed value for θ_{i+1}.

The Metropolis-Hastings Algorithm

Algorithm 3: Metropolis-Hastings MCMC

```
Input: $p(\theta)$: probability distribution
       $q(\theta)$: proposal distribution
       $N_{\mathrm{iter}}$: number of sample iterations
Output: $\{\theta_i\}$: chain of samples
begin
    initialization $\theta_0$
    for $i = 1, \ldots, N_{\mathrm{iter}}$ do
        sample $\theta^* \in q(\theta^*|\theta_i)$
        $p_{\mathrm{acc}} = \min\left(1, \frac{p(\theta^*)q(\theta_i|\theta^*)}{p(\theta_i)q(\theta^*|\theta_i)}\right)$
        sample $u \in U[0,1]$
        if $u < p_{\mathrm{acc}}$ then
            $\theta_{i+1} = \theta^*$
        else
            $\theta_{i+1} = \theta_i$
```

Compute acceptance probability $p_{\mathrm{acc}}(\theta_i, \theta^*)$.

The Metropolis-Hastings Algorithm

Algorithm 4: Metropolis-Hastings MCMC

```
Input: $p(\theta)$: probability distribution
       $q(\theta)$: proposal distribution
       $N_{\mathrm{iter}}$: number of sample iterations
Output: $\{\theta_i\}$: chain of samples
begin
    initialization $\theta_0$
    for $i = 1, ..., N_{\mathrm{iter}}$ do
        sample $\theta^* \in q(\theta^*|\theta_i)$
        $p_{\mathrm{acc}} = \min\left(1, \frac{p(\theta^*)q(\theta_i|\theta^*)}{p(\theta_i)q(\theta^*|\theta_i)}\right)$
        sample $u \in U[0,1]$
        if $u < p_{\mathrm{acc}}$ then
            $\theta_{i+1} = \theta^*$
        else
            $\theta_{i+1} = \theta_i$
```

Draw a random number between 0 and 1 from a uniform distribution; if it is smaller than $p_{\mathrm{acc}}(\theta_i, \theta^*)$, then accept θ^* as θ_{i+1}.

Purpose: If we only accepted points of higher probability we would find the maximum of the pdf, but not map out the full pdf.

The Metropolis-Hastings Algorithm

Algorithm 5: Metropolis-Hastings MCMC

```
Input: $p(\theta)$: probability distribution
       $q(\theta)$: proposal distribution
       $N_{\text{iter}}$: number of sample iterations
Output: $\{\theta_i\}$: chain of samples
begin
    initialization $\theta_0$
    for $i = 1, ..., N_{\text{iter}}$ do
        sample $\theta^* \in q(\theta^*|\theta_i)$
        $p_{\text{acc}} = \min\left(1, \frac{p(\theta^*)q(\theta_i|\theta^*)}{p(\theta_i)q(\theta^*|\theta_i)}\right)$
        sample $u \in U[0, 1]$
        if $u < p_{\text{acc}}$ then
            $\theta_{i+1} = \theta^*$
        else
            $\theta_{i+1} = \theta_i$
```

If θ^* is accepted, added it to the chain. If not, add θ_i to the chain.

Practical MCMC Chain Checks

check the acceptance rate:

Some MCMC samplers give an updating estimate of the current acceptance rate of new samples. Ideally for a sampler using some form of Metropolis-Hastings, this should be somewhere between $\sim 20-50\%$ depending on the type of problem you're trying to solve.

- A high acceptance rate indicates that the chain is moving but might not be exploring the pdf well. This gives high acceptance rate but poor global exploration of the posterior distribution function.
- A low acceptance rate indicates that the chain is hardly moving, i.e. it's stuck in a rut or trying to jump to new points that are too far away.

Practical MCMC Chain Checks

check trace plots:

Ideally, our tracplot in each parameter would be mixing well (moving across parameter space without getting stuck). This will tell you whether your chain is getting stuck or encountering inefficiencies.

Practical MCMC Chain Checks

check trace plots:

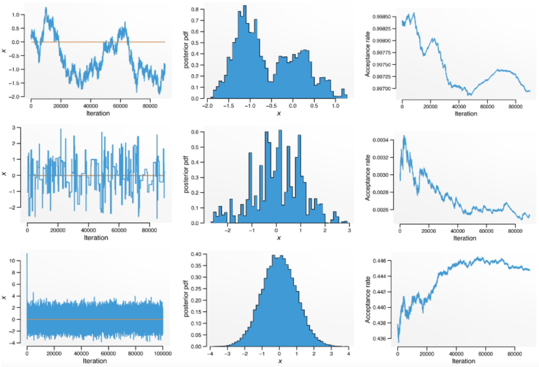
Ideally, our tracplot in each parameter would be mixing well (moving across parameter space without getting stuck). This will tell you whether your chain is getting stuck or encountering inefficiencies.

check autocorrelation length:

The MCMC chain with Metropolis-Hastings will not give fully-independent random samples. The next point is influenced by where the previous point was. We need to check how much to down-sample the chain so that the points lack memory and influence from others. This is given by the autocorrelation length.

Practical MCMC Chain Checks

first column: trace plot, second column: histogram of the chain, third column: acceptance rate of newly proposed θ^*



In the top row, the proposal width was too small. In the middle row, the proposal width was too big. Only the bottom row shows reasonable sampling.

Break & Questions

afterwards we continue with `lecture_7.ipynb` from the `github` repository