MCMC Pratical Session

- What can you do?
- How can you do it?
- What is important in practice?

https://github.com/aponteeduardo/cpp_lecture

samples/metropolis_hastings.m

aponteeduardo@gmail.com

Write down your model

$$\underbrace{p(\theta|y,u,m)}_{posterior} = \frac{\overbrace{p(y|u,\theta,m)}^{likelihood}\,\overbrace{p(\theta|m)}^{prior}}{\underbrace{p(y|m)}_{model\ evidence}}$$

y: $Experimental\ data$

u: $Experimental\ design$

m: $Model$

θ: $Parameters$

Write down your model

Meaning	**Variable**
Experimental data	y
Experimental design	u
Model parameters	theta
Priors	ptheta
MCMC parameters	htheta
Log likelihood	llh
Log prior probability	lpp

Inference

- Goal samples: Draw samples $\theta_1, \ldots, \theta_n$ from the distribution $p(\theta|y, u, m)$.
- Typically you are interested in:

$$E[\theta|y, u, m] \approx \frac{1}{n}\sum_{i=1}^{n}\theta_i$$

$$\mathrm{Var}[\theta|y, u, m] \approx \frac{1}{n-1}\sum_{i=1}^{n}\left(\theta_i - \frac{1}{n}\sum_{j=1}^{n}\theta_j\right)^2$$

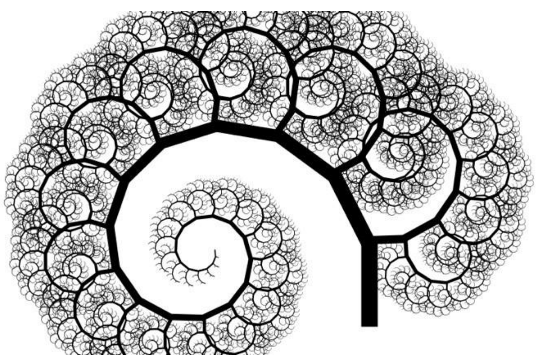
$$p(y^*|y, u, m) = \int p(y^*|u, \theta, m)p(\theta|y, u, m)d\theta$$

$$\approx \frac{1}{n}\sum_{i}^{n} p(y^*|u, \theta_i, m)$$

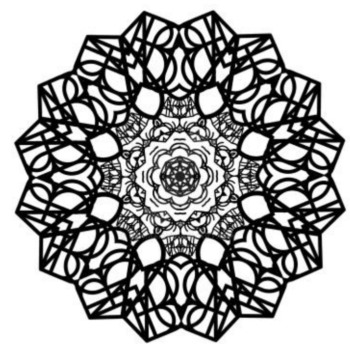
Model specification

- A simple learning task.

A



B



Decision: $y = [A, B, A, A, B,...]$

Monetary reward: $u = [2.1, 3.4, 0.5, ...]$

Likelihood function

- Participants learn the expected reward of each option over trials:

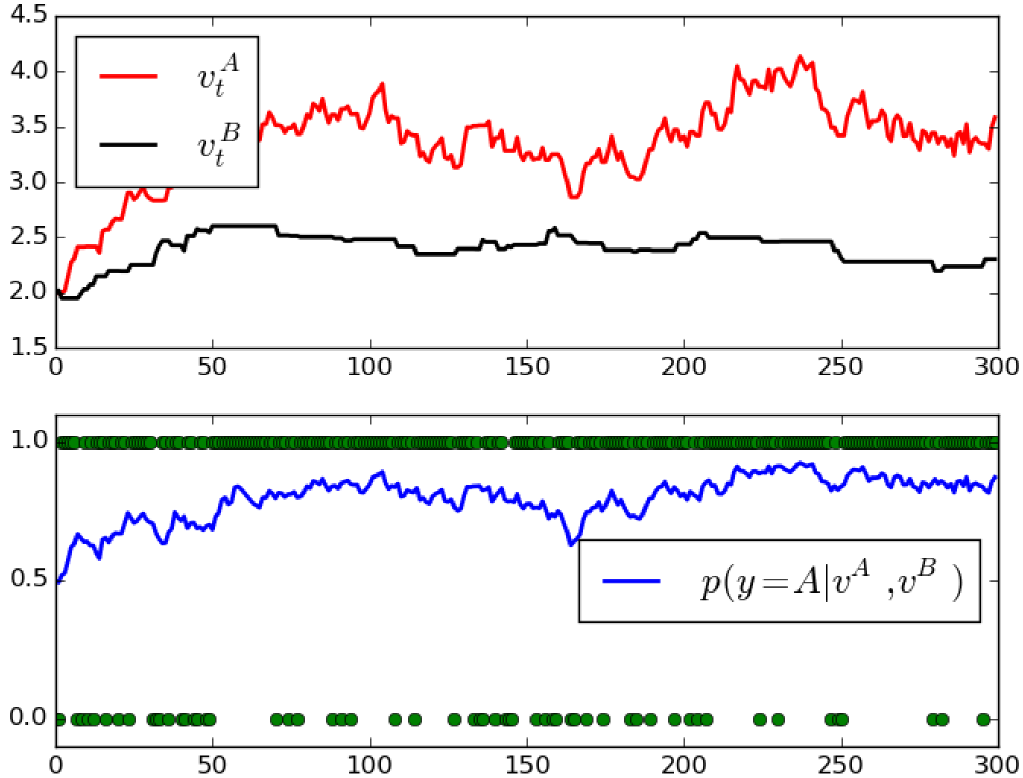
$$v_{t+1}^{A} = v_t^{A} + \alpha(u_t - v_t^{A})$$

$$v_{t+1}^{B} = v_t^{B} + \alpha(u_t - v_t^{B})$$

- They prefer the option with the highest expected reward:

$$p(y = A | v^A, v^B) = \frac{\exp \beta v^A}{\exp \beta v^A + \exp \beta v^B}$$

Likelihood function



Inference

- The model has two free parameters and a few priors.
  - Parameters: α, β
  - Priors (among others): v_0^A, v_0^B.
- The algorithm has the following parameters:
  - Number of iterations.
  - Proposal distribution.

Inference

- Assumption α and β are subject specific traits.
- Given experimental data y and u, obtain samples of the posterior distribution of α and β.

Inferece

- Strategy: Draw samples from a Markov chain whose equilibrium distribution is the posterior $p(\theta|y,u), \theta = [\alpha, \beta]$.

$$\cdots \quad \boxed{\theta_{t-1}} \xrightarrow[K(\theta_t|\theta_{t-1})]{} \boxed{\theta_t} \xrightarrow[K(\theta_{t+1}|\theta_t)]{} \boxed{\theta_{t+1}} \quad \cdots$$

Inference

- Sampling from $K(\cdot\,|\theta)$ is done in two steps:
  - Obtain a proposed sample from a distribution $q(\cdot\,|\theta)$.
  - Accept or reject the proposed sample with certain probability. You need to draw a sample from the uniform distribution in the interval [0, 1].

- $K(\cdot\,|\theta)$ satisfies the detailed balance condition.

Inferece

- Propose a random sample $\hat{\theta}$ from a simple distribution (very often a Gaussian).
- Accept with probability:

$$\min\left\{1, \frac{p\left(y\middle|u, \hat{\theta}\right)p(\hat{\theta})}{p(y|u, \theta_t)p(\theta_t)}\right\}$$

- If you accept $\hat{\theta}$, then $\theta_{t+1} = \hat{\theta}$.
- If you reject $\hat{\theta}$, then $\theta_{t+1} = \theta_t$.

Proposal distribution

– Gaussian!

$$\hat{\theta} = \theta_t + \varepsilon \qquad \varepsilon \sim N(0, \Sigma)$$

– To sample ε you can use the fact that $\mathrm{var}(Lx) = L^T\ \mathrm{var}(x) L$. If $L^T L = \Sigma$, and $var(x) = I$, then $\mathrm{var}(Lx) = \Sigma$.

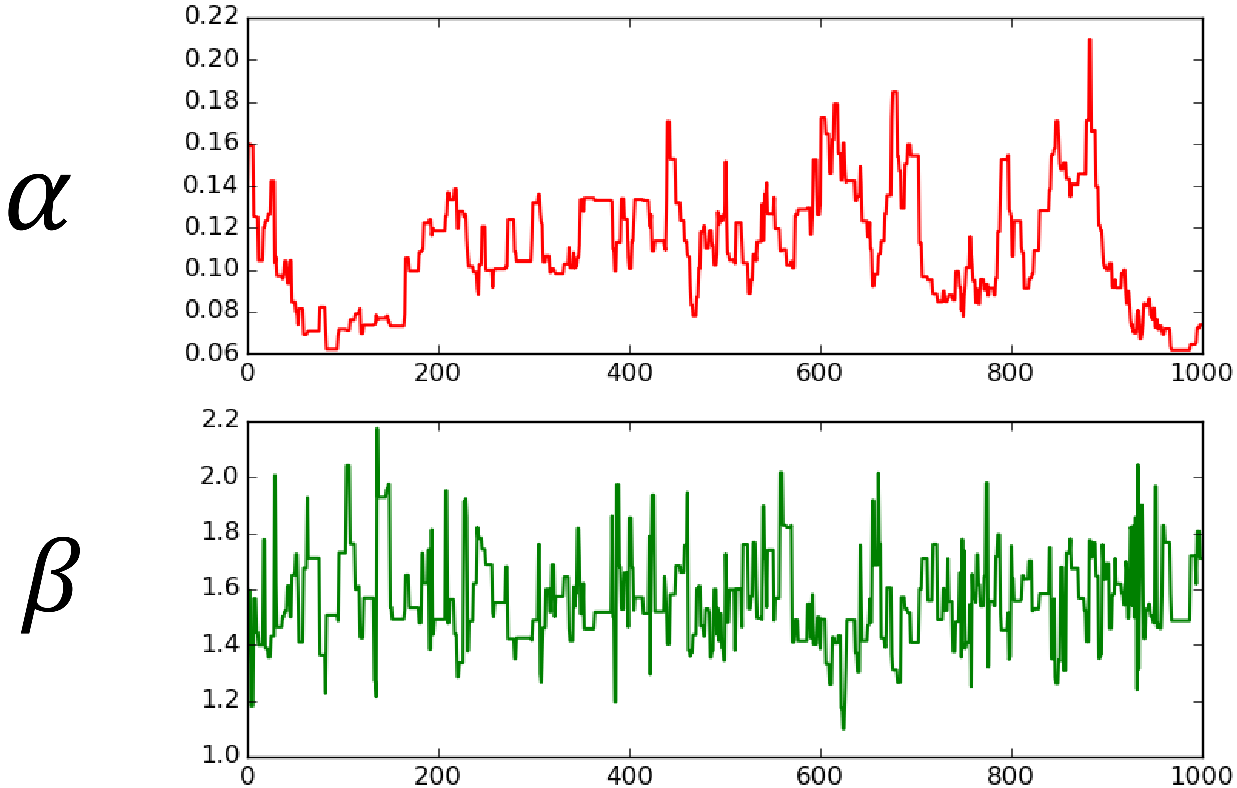
– **chol** computes the square root of Σ, L.

Main loop

- Repeat:
  - Sample from proposal distribution
  - Evaluate the join probability
  - Accept or reject new sample
  - Store the results

```
% Compute the acceptance rate
ar = 0;

for i = 1: nburnin + niter
    % Propose a new sample
    ntheta = otheta + htheta.csigma * randn(size(otheta));

    % Compute log likelihood conditioned on a proposed sample
    nllh = llh(y, u, ntheta, ptheta);
    % Compute log prior probability
    nlpp = lpp(u, ntheta, ptheta);

    % Probability of acceptance
    paccept = min(0, (nllh + nlpp) - (ollh + olpp));

    % If the samples is accepted keep it
    if exp(paccept) > rand
        otheta = ntheta;
        ollh = nllh;
        olpp = nlpp;

        ar = ar + 1;
    end

    % Store your samples
    if i > nburnin
        post.ljp(i - nburnin) = ollh + olpp;
        post.theta(:, i - nburnin) = otheta;
    end

end
```

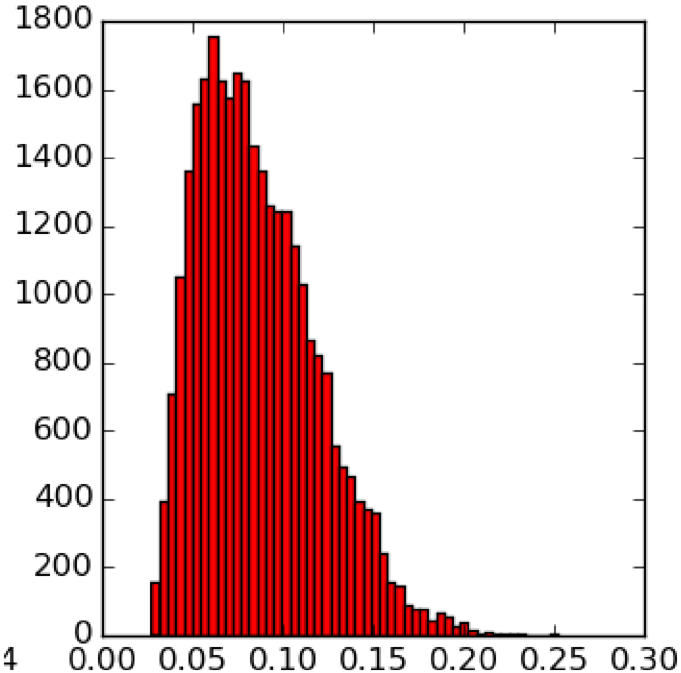
Samples



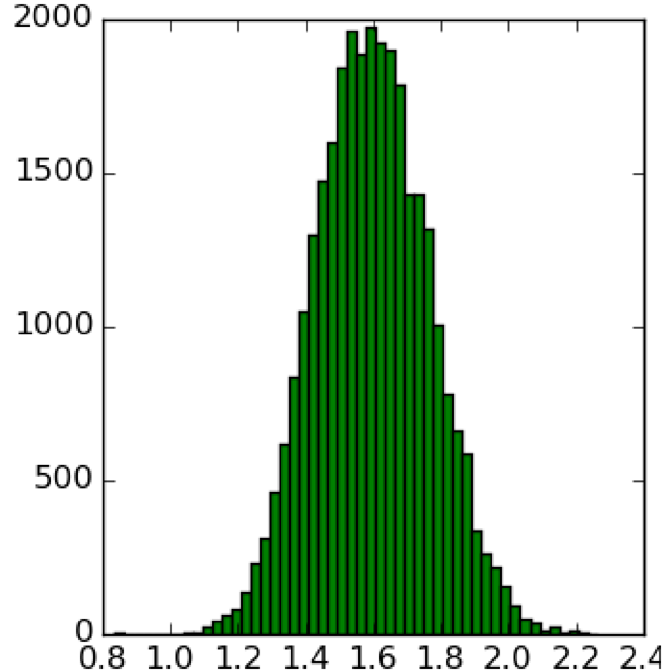
α
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0
200
400
600
800
1000
β
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0
200
400
600
800
1000

Samples

α



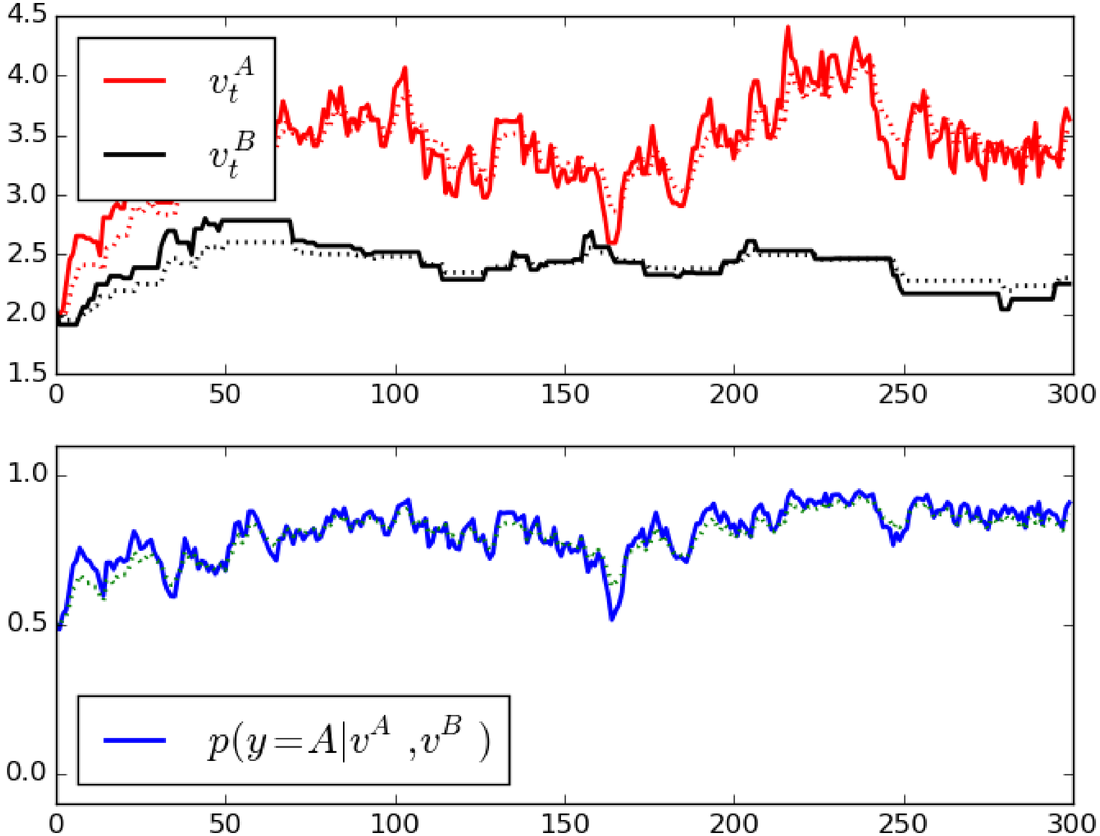
β



$\alpha = 0.05 \rightarrow E[\alpha|y, u, m] = 0.08$

$\beta = 1.5 \rightarrow E[\beta|y, u, m] = 1.58$

Samples



v_t^A
v_t^B
4.5
4.0
3.5
3.0
2.5
2.0
1.5
0
50
100
150
200
250
300
$p(y=A|v^A, v^B)$
1.0
0.5
0.0
0
50
100
150
200
250
300

Some notes!

- There is no universal, principled criterion to decide whether a chain has reached equilibrium. In practice, there are very good heuristics to test this.
- Convergence is only asympotically guaranteed.
- Efficiency: How many samples do you need to adequatedly sample the distribution?

Some tips

- There is a very large literature on MCMC!
- Some problems are harder than others!
- There is a trade-off between how efficient your solutions is, and how long you need to get it working. Your time is more valueable that computational time!
- Check tapas/mpdcm/ for more advanced code to use in your own problem.