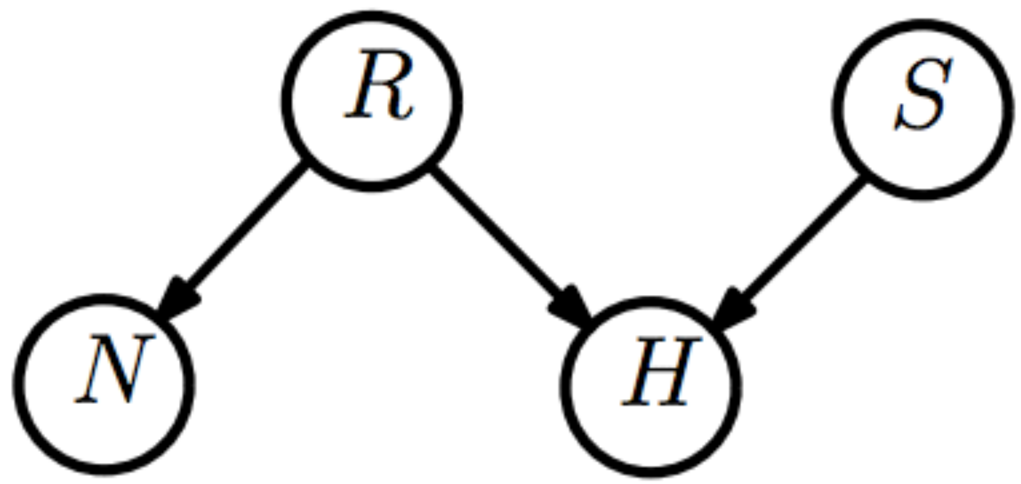
Approximate Inference Through Stochastic Simulation

Wei Li

Roadmap

- probabilistic inference
  - Maximum a posteriori (MAP) inference
  - **conditional probability inference**
    - exact and approximate inference algorithms
      - Variable elimination
      - Message passing algorithm
      - **Stochastic Sampling**
        - Rejection Sampling
        - Metropolis-Hastings Algorithm

Scenario



$R \in \{0, 1\}$, $R = 1$ means it has been raining
$S \in \{0, 1\}$, $S = 1$ means the sprinkler was left on
$N \in \{0, 1\}$, $N = 1$ means neighbours lawn is wet
$H \in \{0, 1\}$, $H = 1$ means Holmes lawn is wet

Roadmap

- probabilistic inference
  - Maximum a posteriori (MAP) inference
  - **conditional probability inference**
    - exact and approximate inference algorithms
      - Variable elimination
      - Message passing algorithm
      - **Stochastic Sampling**
        - Rejection Sampling
        - Metropolis-Hastings Algorithm

Inference Problem

- Conditional Probability Query
- MAP inference

Inference Problem

- Conditional Probability Query
  - Evidence: $E = e$
  - Query: a subset of variables y
  - Task: compute $P(y \mid E = e)$

Inference Problem

- Conditional Probability Query
- MAP inference
  - Evidence: $E = e$
  - Query: all other variables Y ($Y = \{X_1, X_2, \ldots\} - E$)
  - Task: computer $MAP(Y \mid E = e) = \text{argmax}_y P(Y = y \mid E = e)$

Approximate inference

- probabilistic inference
  - Maximum a posteriori (MAP) inference
  - **conditional probability inference**
    - exact and approximate inference algorithms
      - Variable elimination
      - Message passing algorithm
      - **Stochastic Sampling**
        - Rejection Sampling
        - Metropolis-Hastings Algorithm

Algorithms: Conditional Probability

- Push summaries into factor product
  - Variable elimination

Exact

- Message passing over a graph
  - Belief propagation
  - Variational approximations

Exact
&
Approximate

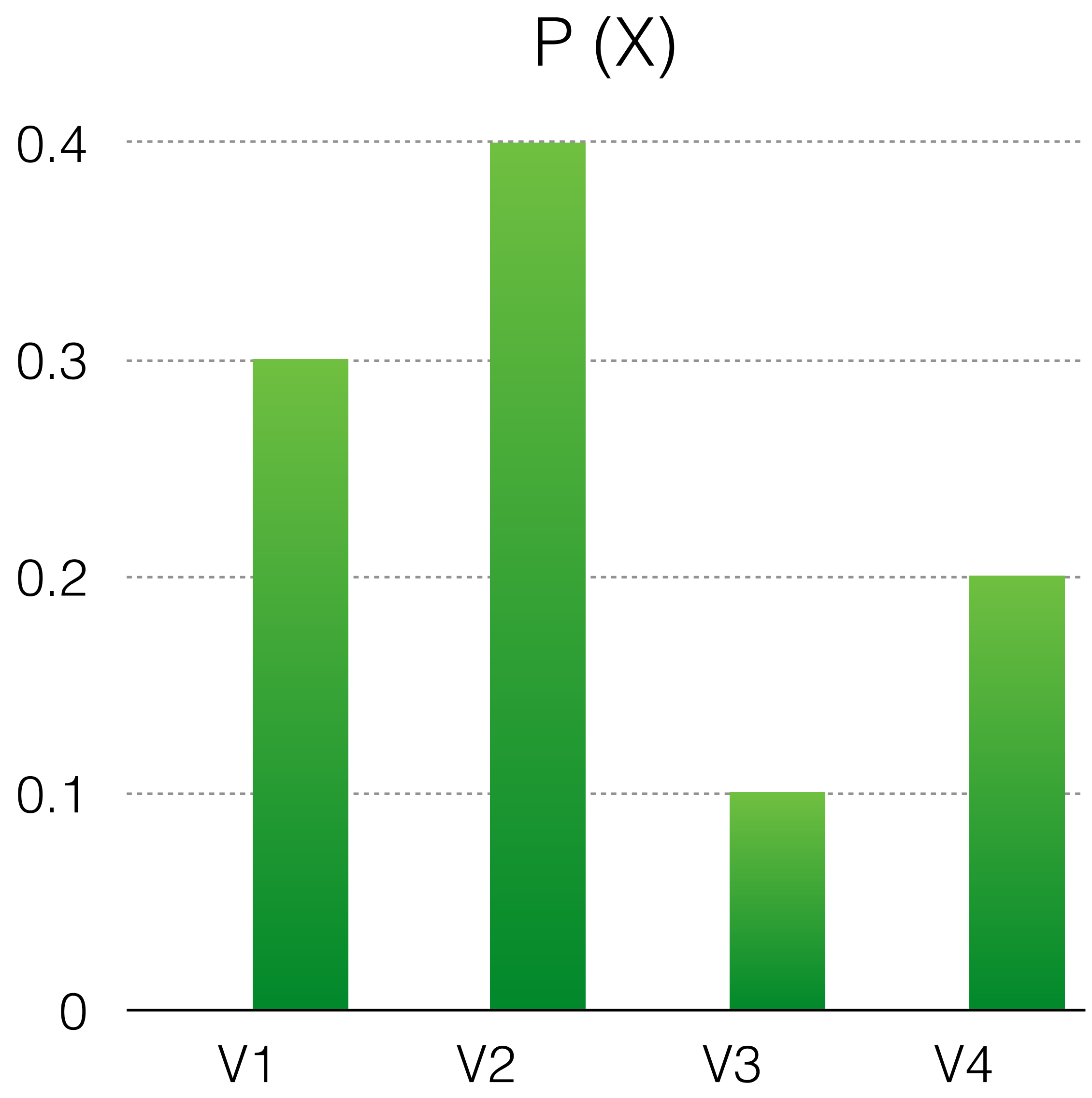
- Stochastic sampling
  - Markov chain Monte Carlo (MCMC)
  - Importance sampling

Approximate

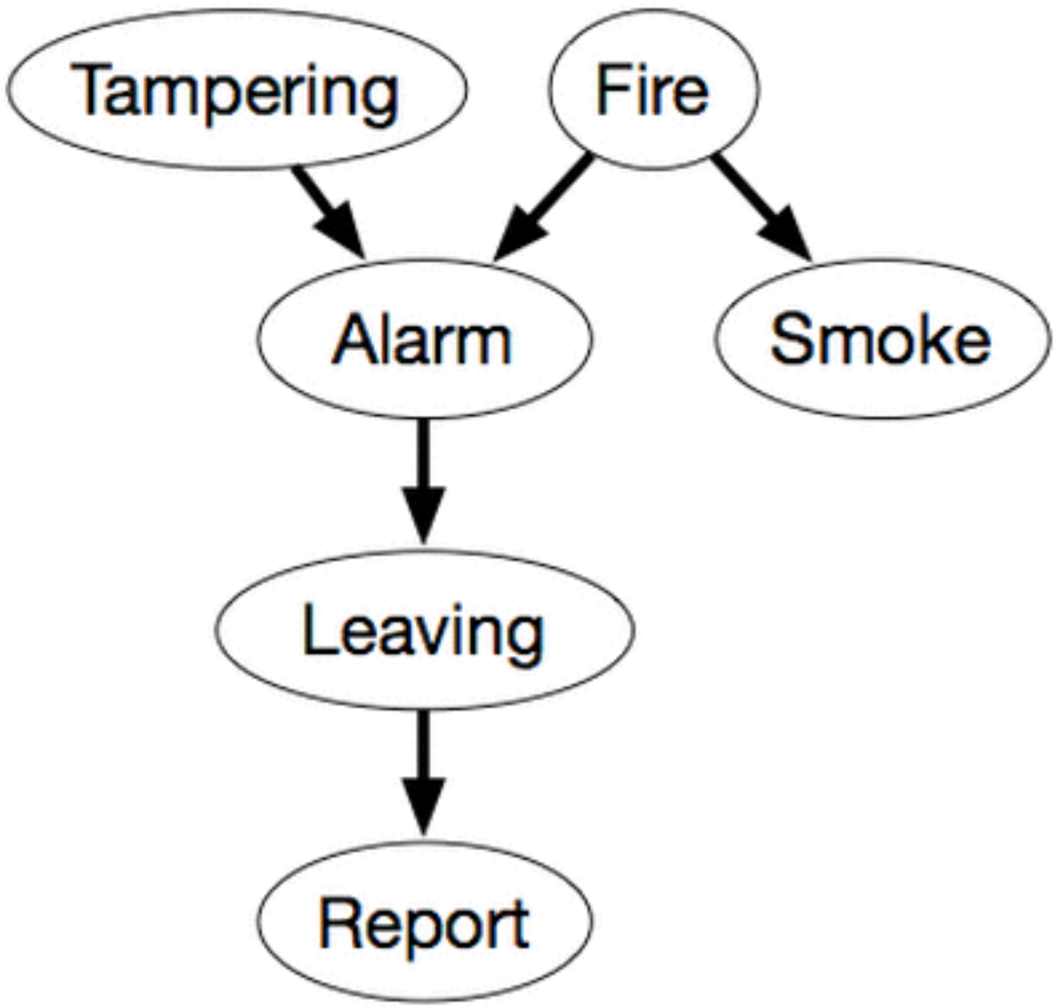
Stochastic Sampling

- how to generate samples
- how to incorporate observations
- how to infer probabilities from samples

Sampling from a Single Variable



Forward Sampling in a Bayesian Network



Sample	Tampering	Fire	Alarm	Smoke	Leaving	Report
s_1	false	true	true	true	false	false
s_2	false	false	false	false	false	false
s_3	false	true	true	true	true	true
s_4	false	false	false	false	false	true
s_5	false	false	false	false	false	false
s_6	false	false	false	false	false	false
s_7	true	false	false	true	true	true
s_8	true	false	false	false	false	true
...						
s_{1000}	true	false	true	true	false	false

From Samples to Probabilities

- Probabilities can be estimated from a set of examples using the **sample average**.
- The **sample average** of a proposition α is the number of samples where α is true divided by the total number of samples.
- The sample average approaches the true probability as the number of samples approaches infinity by the law of large numbers.

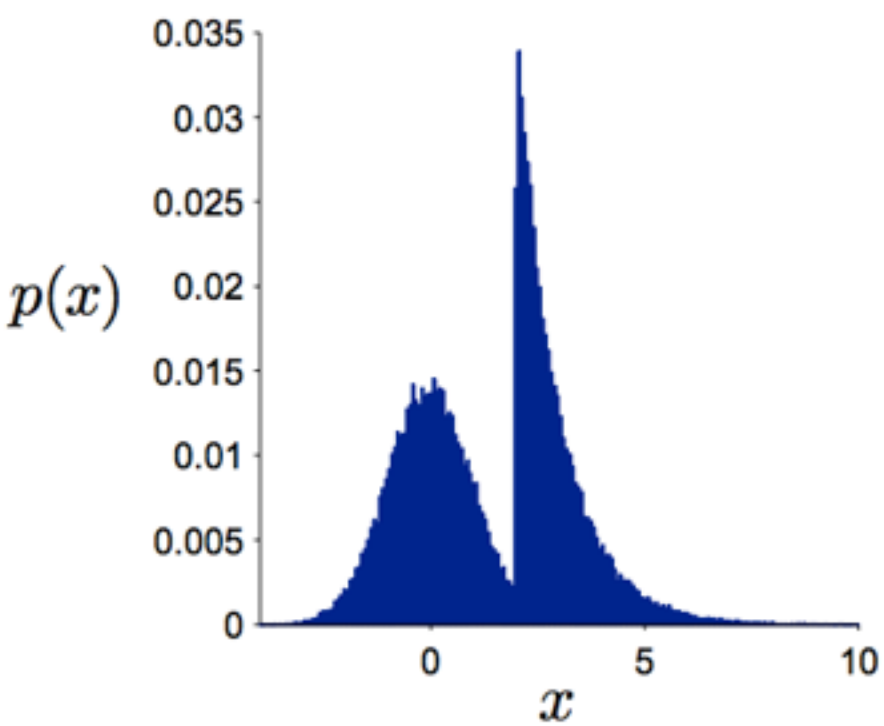
Two Methods of Stochastic Sampling

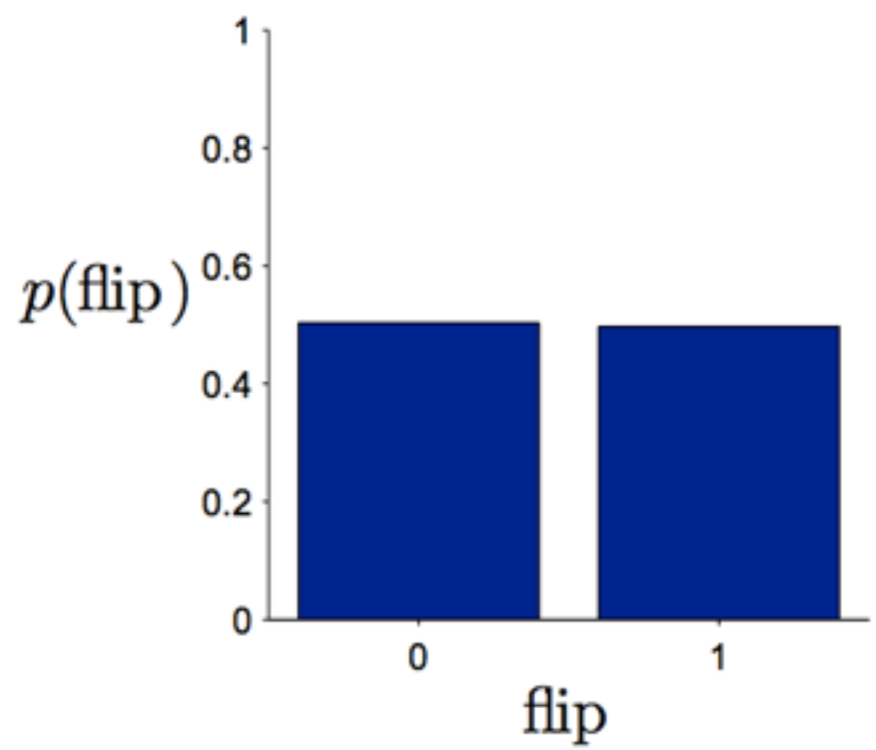
- probabilistic inference
  - Maximum a posteriori (MAP) inference
  - **conditional probability inference**
    - exact and approximate inference algorithms
      - Variable elimination
      - Message passing algorithm
      - **Stochastic Sampling**
        - Rejection Sampling
        - Metropolis-Hastings Algorithm

An example probabilistic program

```
flip = rand < 0.5
if flip
    x = randg + 2    % Random draw from Gamma(1,1)
else
    x = randn        % Random draw from standard Normal
end
```

Implied distributions over variables

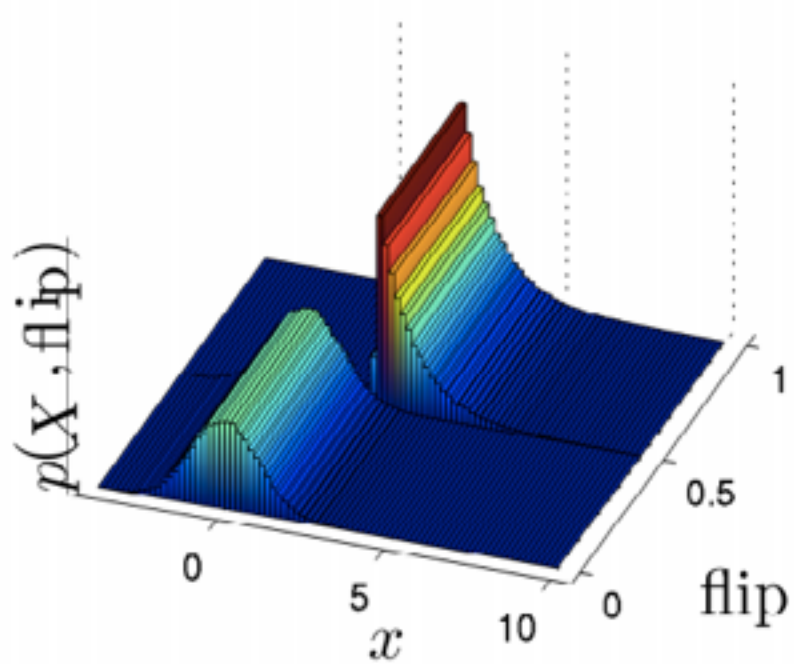


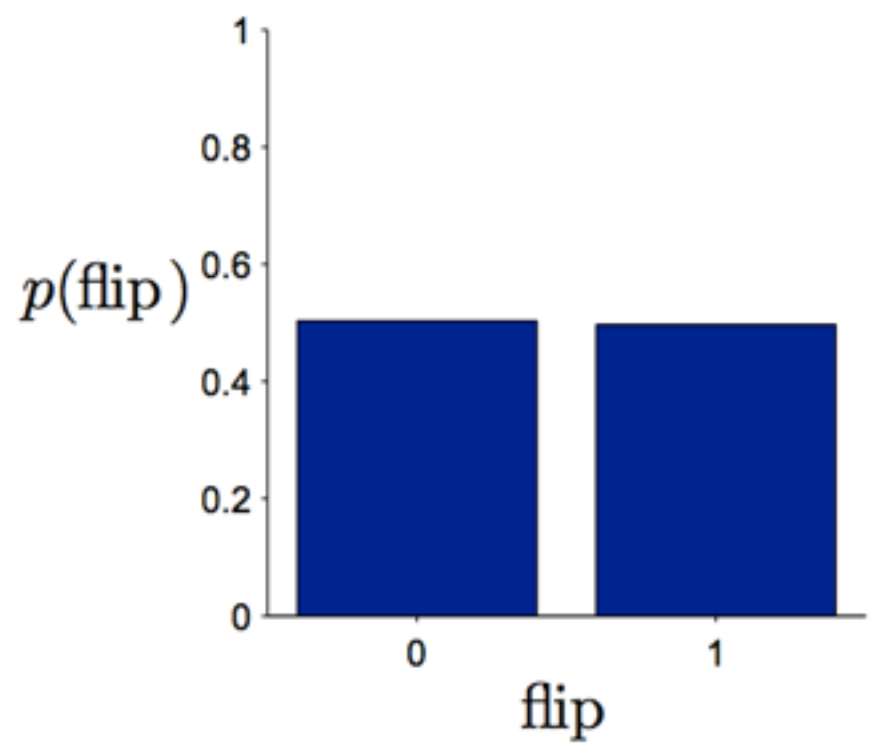


An example probabilistic program

```
flip = rand < 0.5
if flip
    x = randg + 2    % Random draw from Gamma(1,1)
else
    x = randn        % Random draw from standard Normal
end
```

Implied distributions over variables

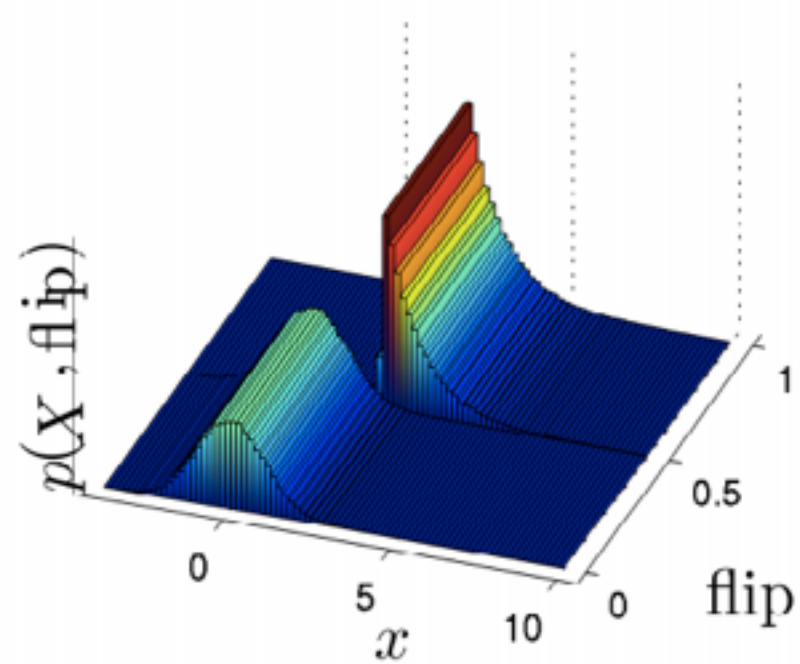


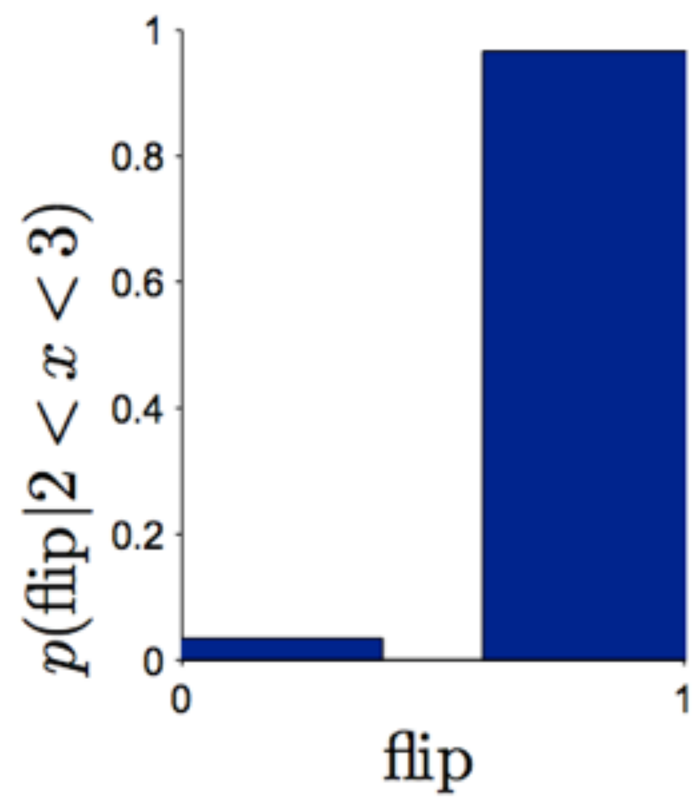


Condition

```
flip = rand < 0.5
if flip
    x = randg + 2    % Random draw from Gamma(1,1)
else
    x = randn        % Random draw from standard Normal
end
```

Implied distributions over variables





Rejection Sampling

- Run the program with a fresh source of random numbers
- If condition is true, record the sample, else ignore the sample
- Repeat

Rejection Sampling

Example

```
flip = rand < 0.5
if flip
    x = randg + 2
else
    x = randn
end
```

- True
- 2.7

This produces samples over the execution trace e.g. (True, 2.7),

Rejection Sampling

Example

```
flip = rand < 0.5
if flip
    x = randg + 2
else
    x = randn
end
```

- True
- 3.2

This produces samples over the execution trace e.g. (True, 2.7),

Rejection Sampling

Example

```
flip = rand < 0.5
if flip
    x = randg + 2
else
    x = randn
end
```

- True
- 2.1

This produces samples over the execution trace e.g. (True, 2.7), (True, 2.1)

Rejection Sampling

Example

```
flip = rand < 0.5
if flip
    x = randg + 2
else
    x = randn
end
```

- False

- -1.3

This produces samples over the execution trace
e.g. (True, 2.7), (True, 2.1)

Rejection Sampling

Example

```
flip = rand < 0.5
if flip
    x = randg + 2
else
    x = randn
end
```

- False
- 2.3

This produces samples over the execution trace
e.g. (True, 2.7), (True, 2.1), (False, 2.3), …

Rejection Sampling

Example

```
flip = rand < 0.5
if flip
    x = randg + 2
else
    x = randn
end
```

$P\,(\text{flip} = \text{true} \mid 2 < x < 3)$

$= 2 / 3$

This produces samples over the execution trace
e.g. (True, 2.7), (True, 2.1), (False, 2.3), …

Can we be more efficient?

- Given the current state x of the algorithm, **Metropolis-Hastings** chooses a new state x' from a "proposal distribution," which often simply involves picking a variable Xi at random and choosing a <u>new value</u> for that variable, again at random.

- Computes the acceptance probability α.

- With probability α the algorithm accepts the proposal and moves to x' and with probability 1 - α the algorithm remains in the state x.

Metropolis-Hastings Algorithm

1. Initialise $x^{(0)}$.
2. For $i = 0$ to $N - 1$

   – Sample $u \sim \mathcal{U}_{[0,1]}$.

   – Sample $x^\star \sim q(x^\star | x^{(i)})$.

   – If $u < \mathcal{A}(x^{(i)}, x^\star) = \min\left\{1, \frac{p(x^\star) q(x^{(i)} | x^\star)}{p(x^{(i)}) q(x^\star | x^{(i)})}\right\}$

   $$x^{(i+1)} = x^\star$$

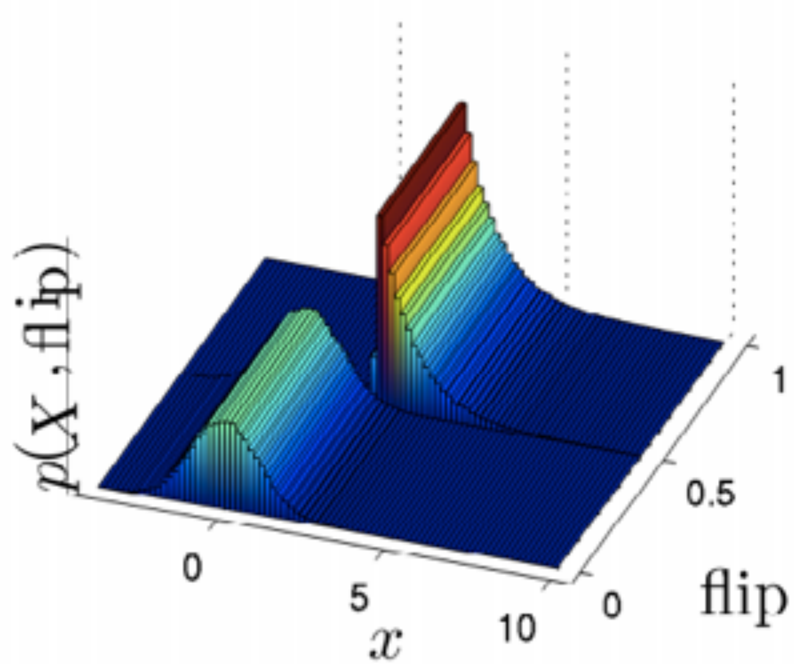
   else

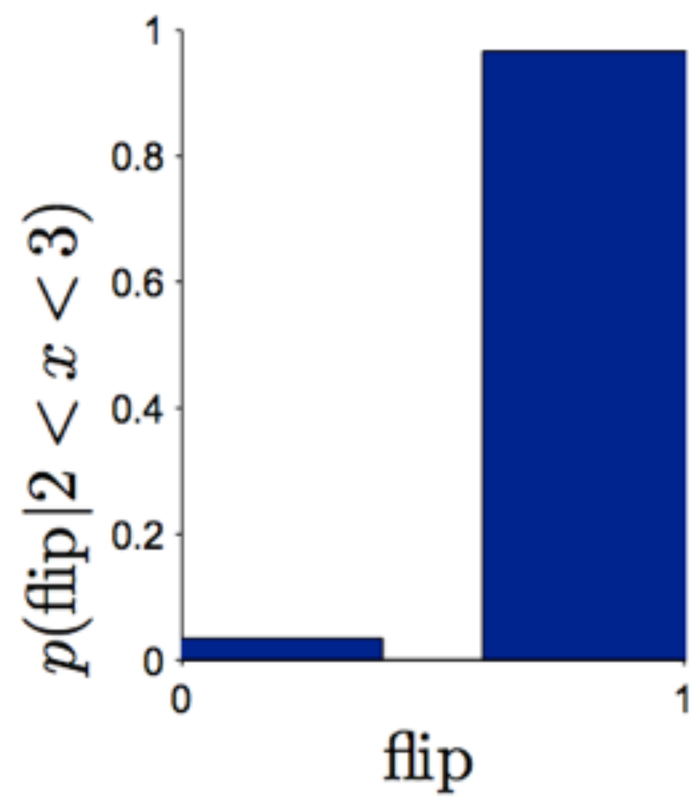
   $$x^{(i+1)} = x^{(i)}$$

An example probabilistic program

```
flip = rand < 0.5
if flip
    x = randg + 2    % Random draw from Gamma(1,1)
else
    x = randn        % Random draw from standard Normal
end
```

Implied distributions over variables





Metropolis-Hastings

- Start with a trace
  - (True, 2.3)
- Change one random decision, sample subsequent decisions
  - (False, -0.9)
- Accept with the appropriate acceptance probability
  - Reject, does not satisfy observation

Metropolis-Hastings

- Start with a trace
  - (True, 2.3)
- Change one random decision, sample subsequent decisions
  - (True, 2.9)
- Accept with the appropriate acceptance probability
  - Accept, maybe

Conclusion

- probabilistic inference
  - Maximum a posteriori (MAP) inference
  - **conditional probability inference**
    - exact and approximate inference algorithms
      - Variable elimination
      - Message passing algorithm
      - **Stochastic Sampling**
        - Rejection Sampling
        - Metropolis-Hastings Algorithm

Reference

- http://www.stanford.edu/~ngoodman/papers/WSG-AIStats11.pdf
  Lightweight implementation of Probabilistic Programming

- http://artint.info/html/ArtInt_150.html
  Artificial Intelligence

- http://mlg.eng.cam.ac.uk/zoubin/course04/hbtnn2e-I.pdf
  Michael Jordan: Probabilistic inference in graphical models

- http://www.cs.princeton.edu/courses/archive/spr06/cos598C/papers/AndrieuFreitasDoucetJordan2003.pdf
  An Introduction to MCMC for Machine Learning

Any Questions ?

- probabilistic inference
  - Maximum a posteriori (MAP) inference
  - **conditional probability inference**
    - exact and approximate inference algorithms
      - Variable elimination
      - Message passing algorithm
      - **Stochastic Sampling**
        - Rejection Sampling
        - Metropolis-Hastings Algorithm

Thanks for listening !