

'Final' Assignment

- Choose one of the following topics. Address the problem using statistical computing. Write up your work using Rmarkdown. Upload your report to the repository called “Final Project” on Github (preferably as a .md or .pdf)
- Due date: Monday May 11th @4pm.
- Note that this is a ‘hard’ deadline.
- Possible topics:
 - **Sudoku exercise (slides 5 onwards)**
 - **Tempered MCMC for Gamma rvs. (next slide)**
 - **Use simulation to tackle one of the “Riddler” problems on slides 3 or 4.**
 - **Report on a question from your own research that uses statistical computation.**

Examinable assignment 4 - part 1

- Use the Urn model starting with 2 red balls of weight 1, and one black (mutation) ball of weight w .
- Draw balls until you have 10 non-black balls.
- If all non-black balls are the same color at the end, what is the posterior distribution of the weight of the black ball?
- If we observe exactly 2 non-black colors at the end, what is the posterior distribution of the weight of the black ball?
-

Use a Uniform[0,20] prior for the weight of the black ball

Bayesian Importance Sampling 2

- Suppose we have parameter(s), θ , prior π , and importance sampling distribution ξ , and observed data D. Iterate the following:
 1. Sample θ' from ξ , **and simulate data D' using θ' .**
 2. **Accept θ' if $D' = D$.** Otherwise reject θ' and return to 1.
 3. Add a point with **(Importance Sampling) weight** of $\pi(\theta')/\xi(\theta')$ to the posterior at θ'
 4. Rejection method: sample from prior $\pi(\theta)$ and construct posterior;

Result: **empirical estimate of posterior distribution $P(\theta|D)$.**

The sampler will be more efficient than naive rejection.

How do we choose ξ ?

-

Examinable assignment - part 2

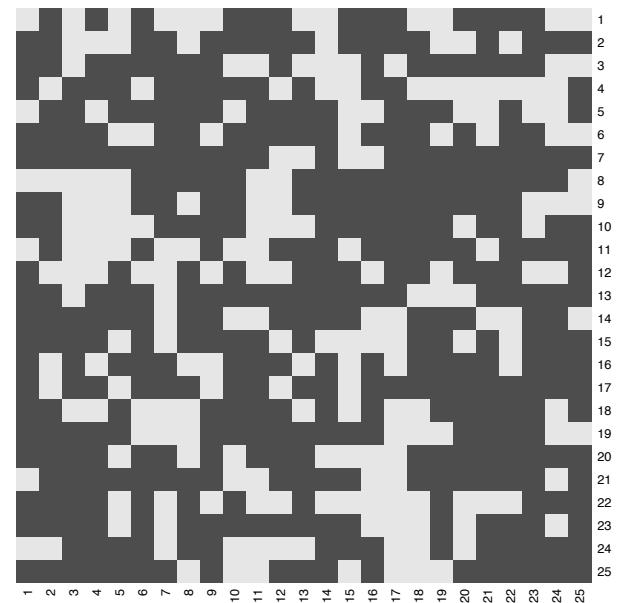
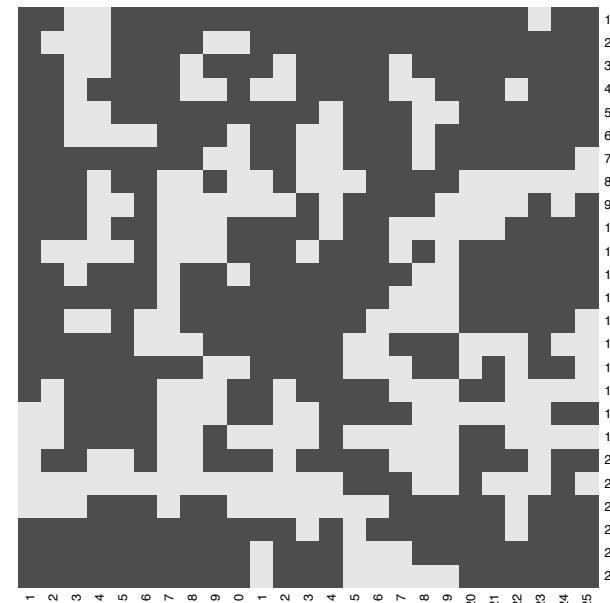
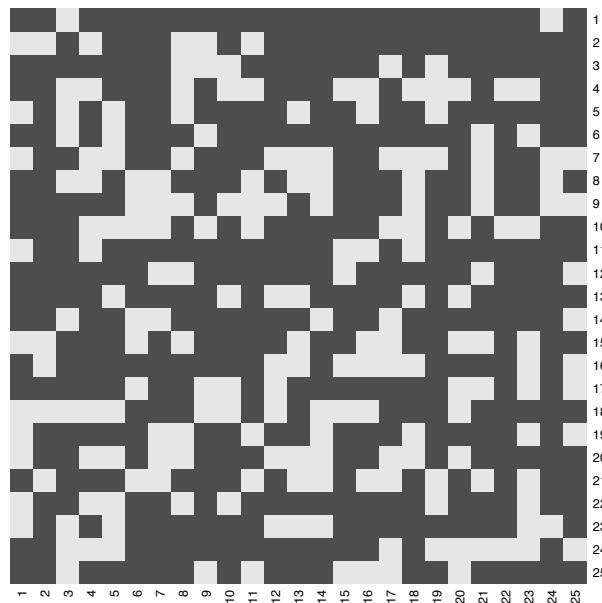
- Repeat the Urn model assignment to find the posterior **mass** of the black ball given that we observe just one (non-black) color in an Urn containing 10 (non-black) balls.
- This time, sample θ' (black ball **mass**) from ξ , an $\text{expo}(\lambda)$ distribution, rather than a Uniform distribution. (So this is Importance sampling.)
- If you accept θ' then add a point with (**Importance Sampling**) **weight** $\pi(\theta')/\xi(\theta')$ to the posterior distribution of θ .
- If we restrict θ' to be in the interval [0,20] (for simplicity's sake), then this means that we have:
 - $\pi(\theta') = 1/20$
 - $\xi(\theta') = \lambda e^{-\lambda\theta'} / (1-e^{-20\lambda})$
- Compare efficiency to a rejection method that samples directly from a Uniform[0,20] distribution, by comparing the number of iterations you need to run in order to collect 10000 accepted θ 's.

Remember: pseudocode on Blackboard “Pseudocode_Urn.R”

Examinable Assignment - Part 3

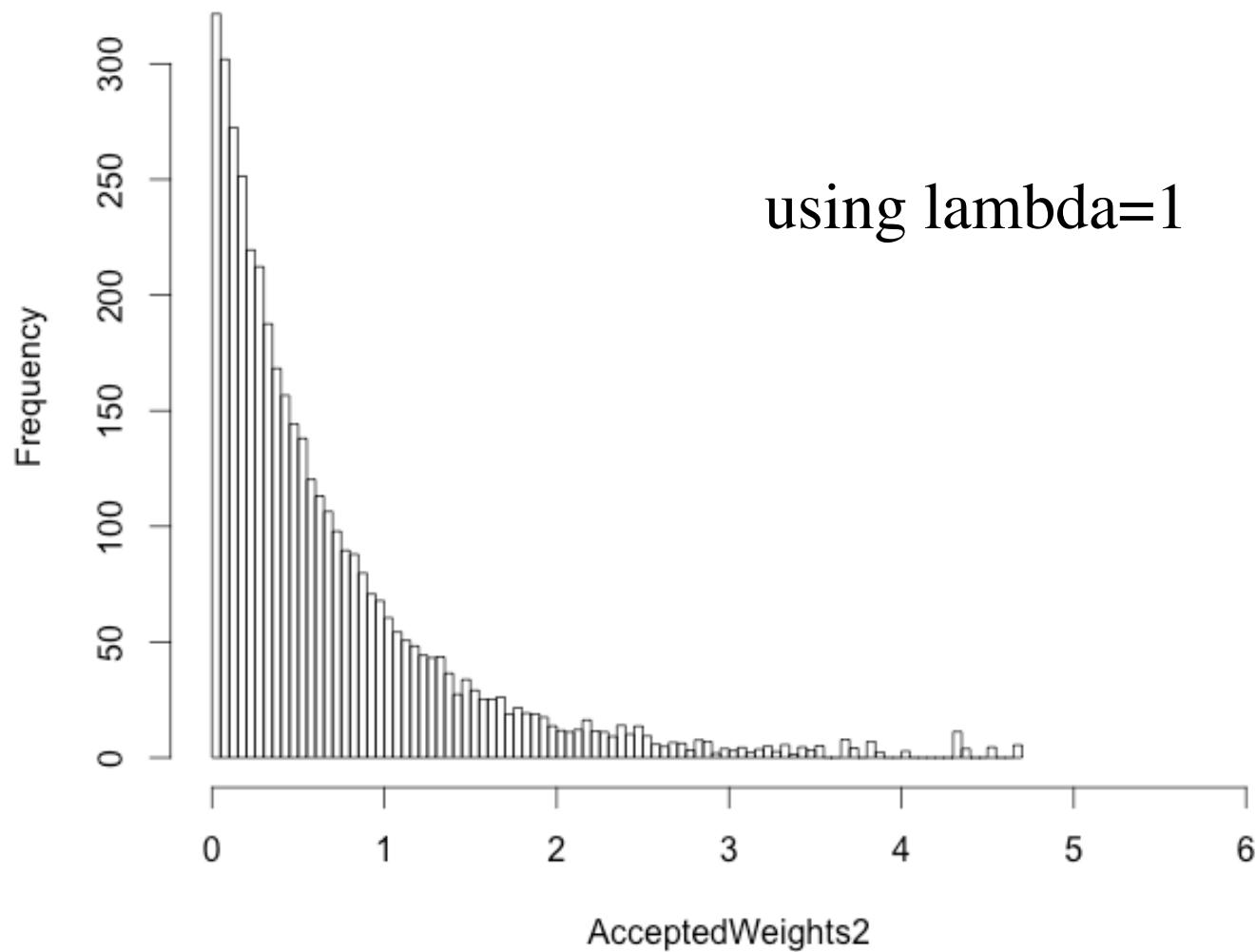
- Many tissues contain 2 (or more) cell-types.
- Investigator wants a way of testing whether each of 2 cell types in a given tissue is homogeneously distributed (in space). So:
 - H_0 : cell types are homogeneously (randomly) arranged
 - H_1 : cell types are not homogeneously arranged (i.e. they are clustered in some fashion).
- Your job, is to come up with such a test, using Monte Carlo methods.
 1. Formulate your test.
 2. Apply it to each of Grid1, Grid2, Grid3 and determine a p-value for rejecting the H_0 in favor of H_1 .
 3. Write-up your test (i.e. Methods) and your Results.

Examinable Assignment - Part 2



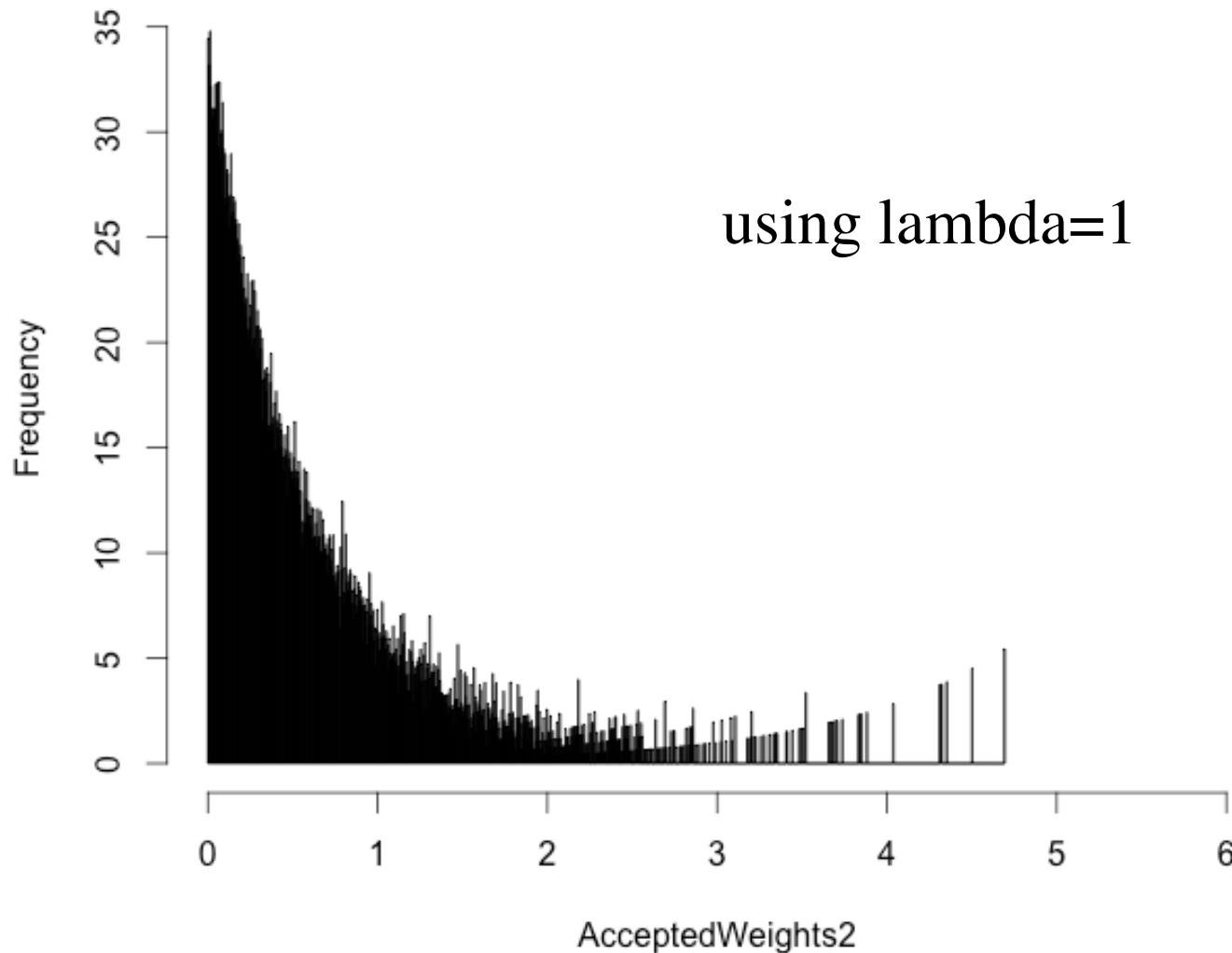
Importance sampling

Histogram of accepted mass



Importance sampling

Histogram of accepted mass

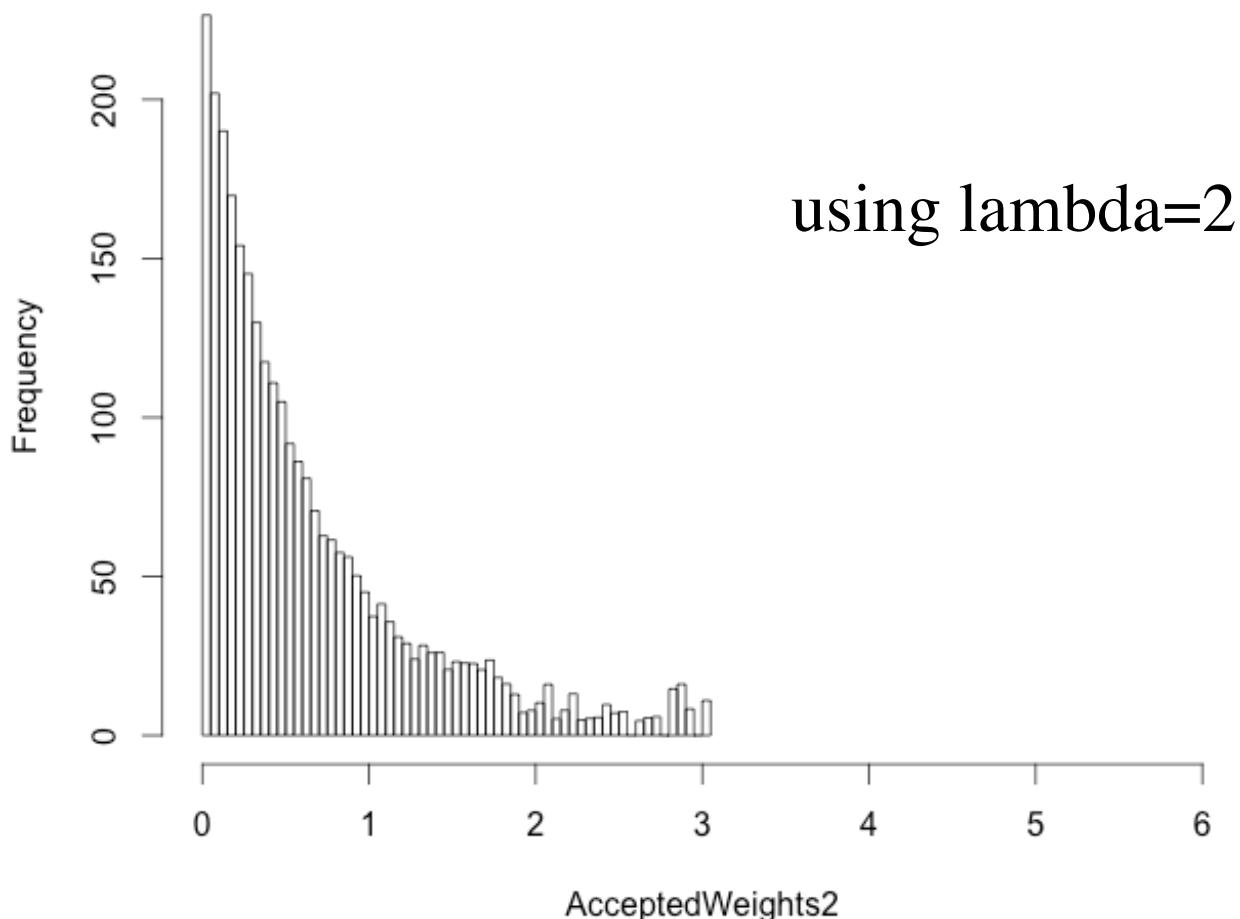


Importance sampling weight

- Let W denote the mass of the black ball
- Assume we sample W from $\exp(\lambda)$, and we truncate this distribution, and the prior, at 20.
- For an accepted parameter value of $W=w$, the importance sampling weight will be
 - $(1/20) / [\lambda \exp(-\lambda w) / (1 - \exp(-\lambda 20))]$
 - This number can get very big for high λ !

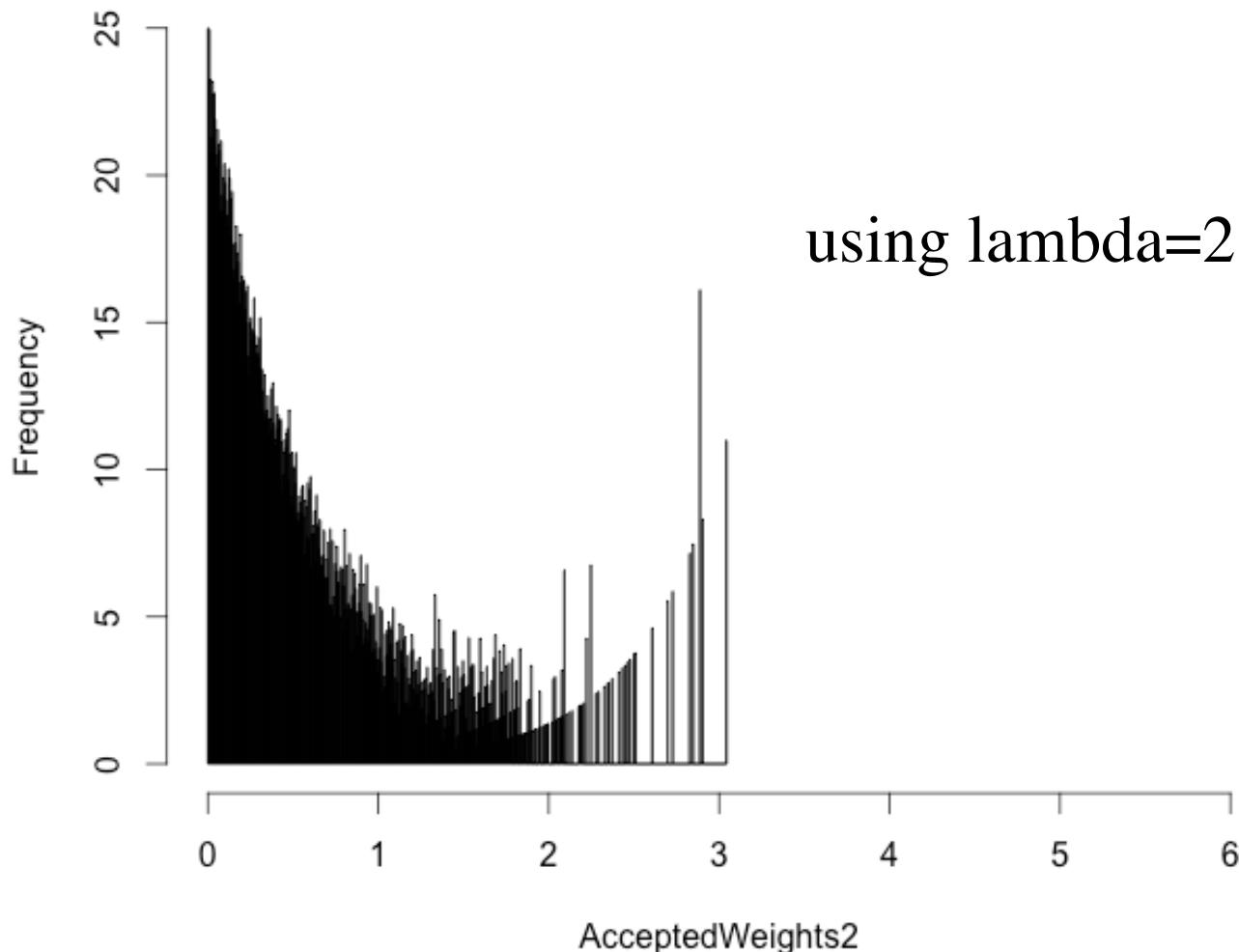
Importance sampling

Histogram of accepted mass



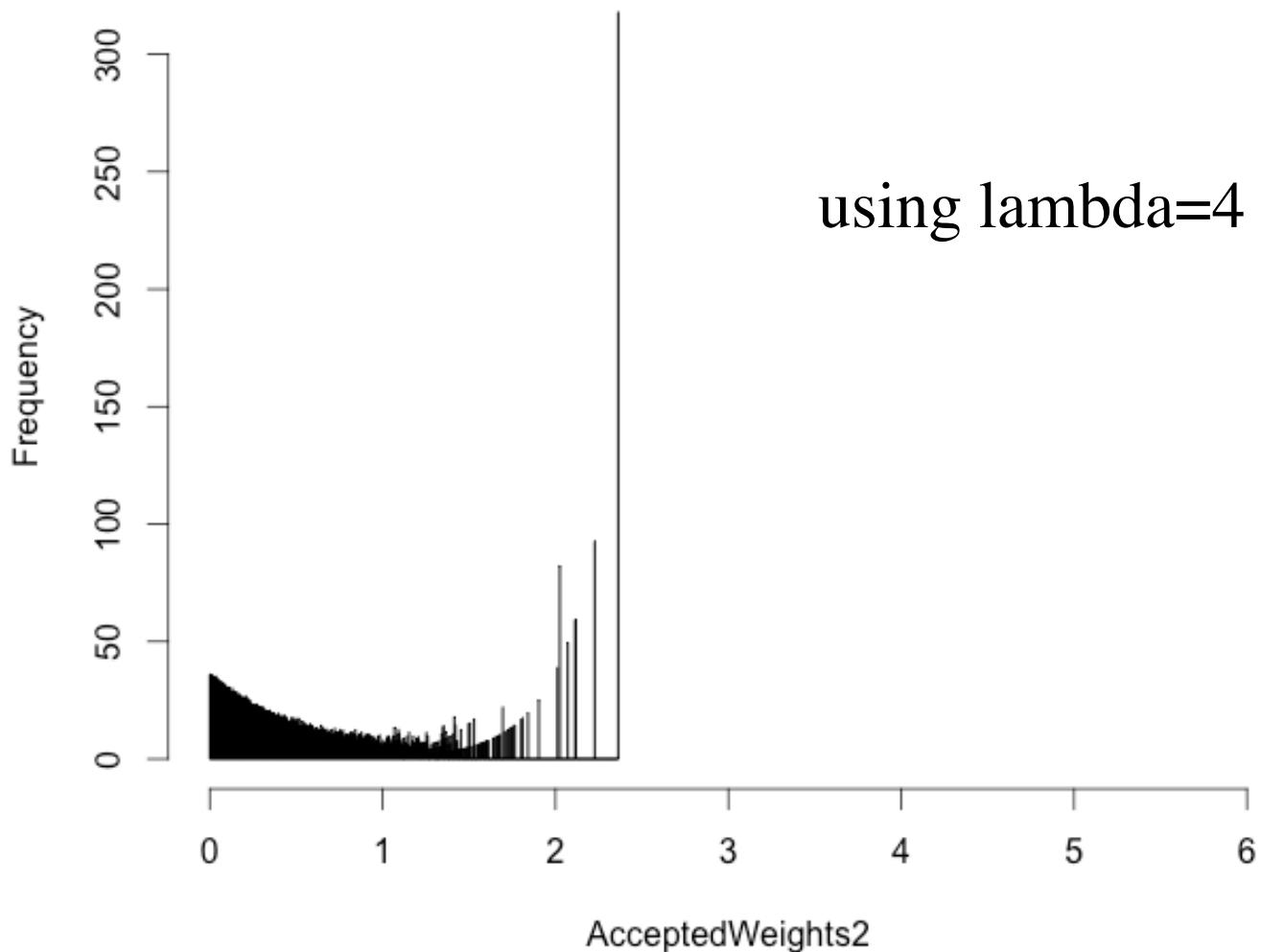
Importance sampling

Histogram of accepted mass



Importance sampling

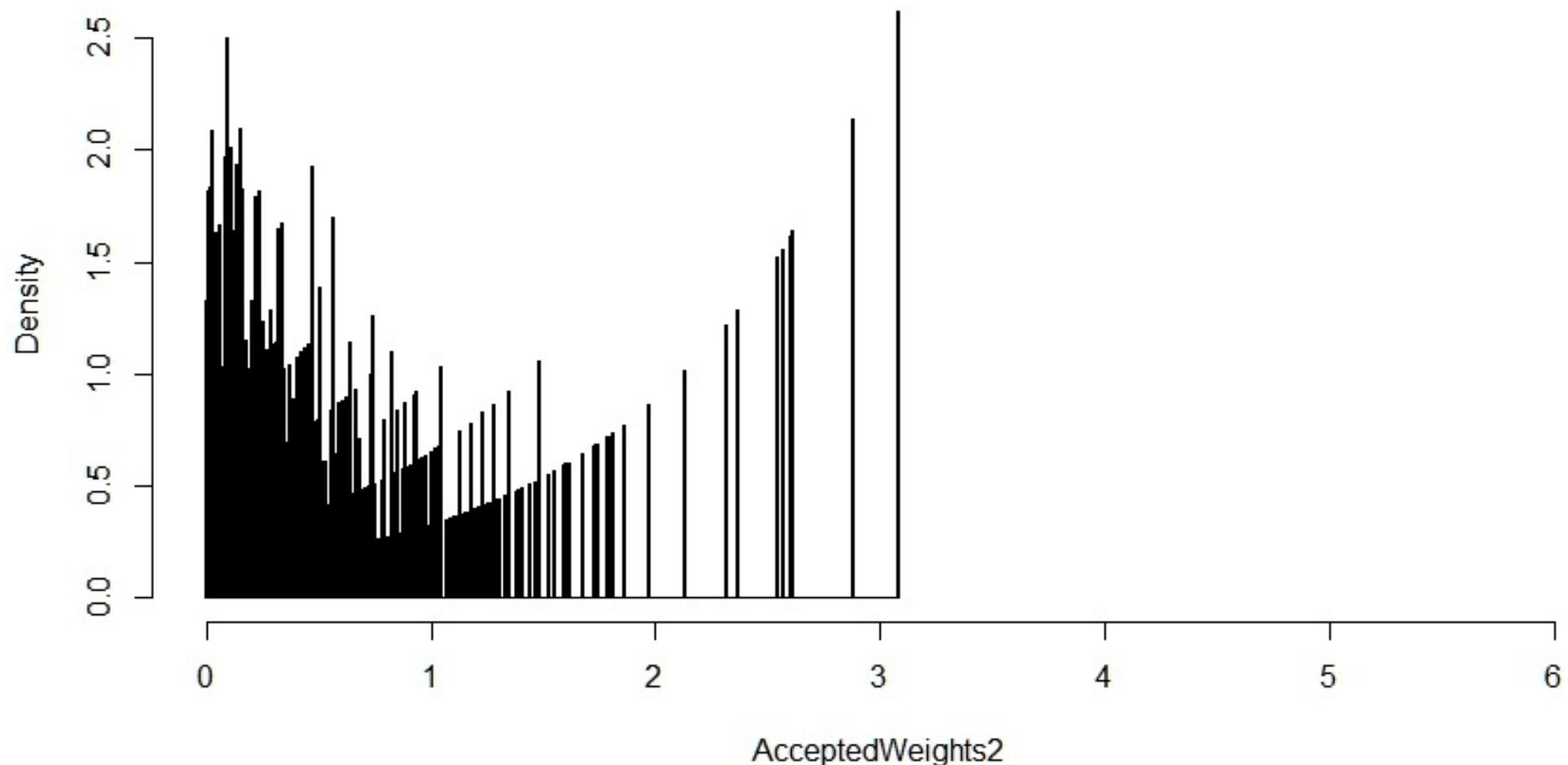
Histogram of accepted mass



Importance sampling

Histogram of accepted mass

1000 samples is generated by 2723 iterations



Hidden Markov Models [HMMs]

Markov Chains

- $X(t)$ is the state of the system at time t [t is discrete]
- The **state-space** S is the set of values that $X(t)$ can take.
- The **transition function** $P(x_1, x_2)$ gives the probability that the next value of X is x_2 , given that X is currently in state x_1 .
- Note, by assumption: $P(X(t)|X(t-1), X(t-2), \dots, X(0)) = P(X(t)|X(t-1))$.
 - So “the future is independent of the past, except through the present state”.
- Example:
 - $Y(t) = -1$ with prob $1/2$; $+1$ otherwise, for all t .
 - Define $X(0) = 0$; $X(t) = X(t-1) + Y(t)$ [$= Y(t) + Y(t-1) + \dots + Y(0)$]
 - So $P(X(t)=x_2|X(t-1)=x_1)=1/2$ if $|x_2-x_1|=1$ (for integers x_1, x_2)
 - This is a “symmetric random walk”.

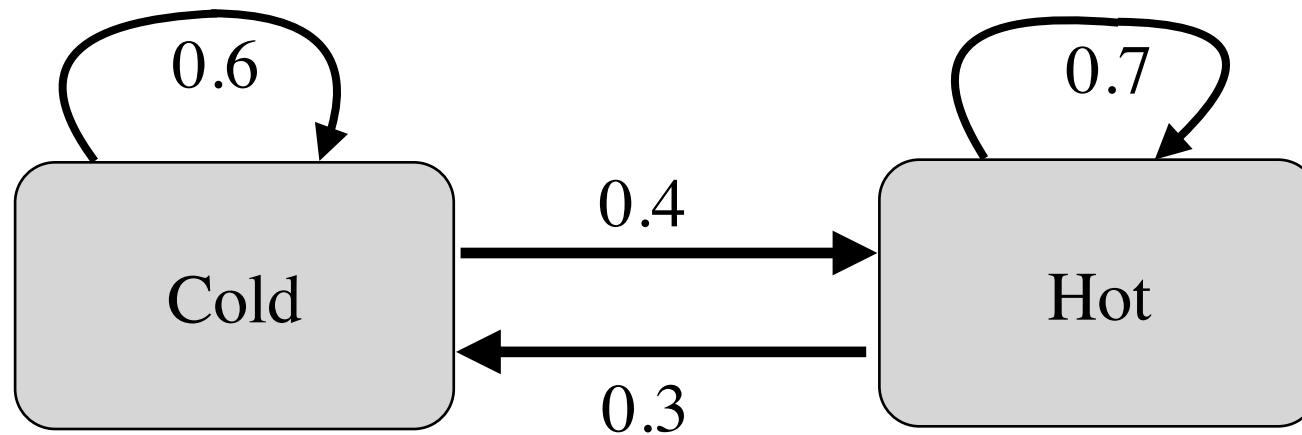
Elements of Discrete-time Hidden Markov Models

1. A finite number, N , of (hidden) states in the model.
2. At each clock time, t , a new (hidden) state is entered. (*Note that the new state may be the same as the last state*). This is controlled by the transition probability matrix. [**It is a Markov Chain.**]
3. After each (potential) transition some observation is made. The observation is made according to a probability distribution that depends on the current state of the system. [Often called the “emission probability” or “confusion” matrix.] Note that the set of observations is typically **not** a Markov chain.

HMM example

- Motivating example: (Stamp, Mark. "A revealing introduction to hidden Markov models." Department of Computer Science San Jose State University (2004).)
- Wish to infer weather conditions over a series of years. We aim to categorize years as 'hot' or 'cold'. Suppose that data records suggest the following:
 - the probability of a hot year being followed by another hot year is 0.7
 - the probability that a cold year is followed by another cold year is 0.6.

HMM example - hidden states



Described by a Markov chain:
 $X(t)$ is the type of weather in year t .

Transition matrix:

$$\begin{matrix} & \begin{matrix} H & C \end{matrix} \\ \begin{matrix} H \\ C \end{matrix} & \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \end{matrix}$$

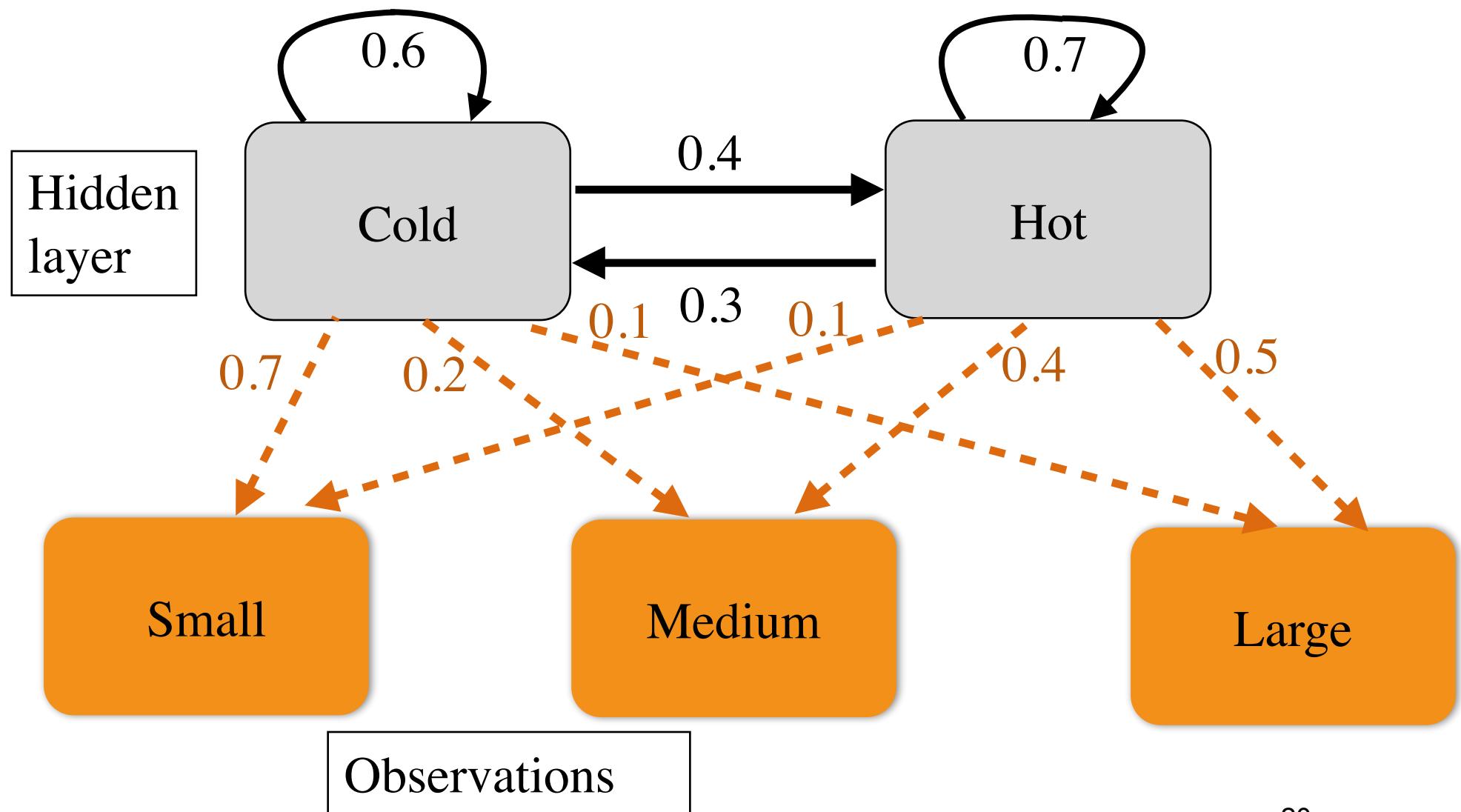
HMM example - observations

- But we can't go back and observe the weather conditions. Instead we look at the width of tree rings:
 - Suppose there is a correlation between the size of tree growth rings and temperature.
 - We summarize this using three different tree ring sizes, small [S], medium [M] and large [L].
 - Suppose the probabilistic relationship between annual temperature and tree ring sizes is given by

	S	M	L
H	0.1	0.4	0.5
C	0.7	0.2	0.1

Emission probabilities

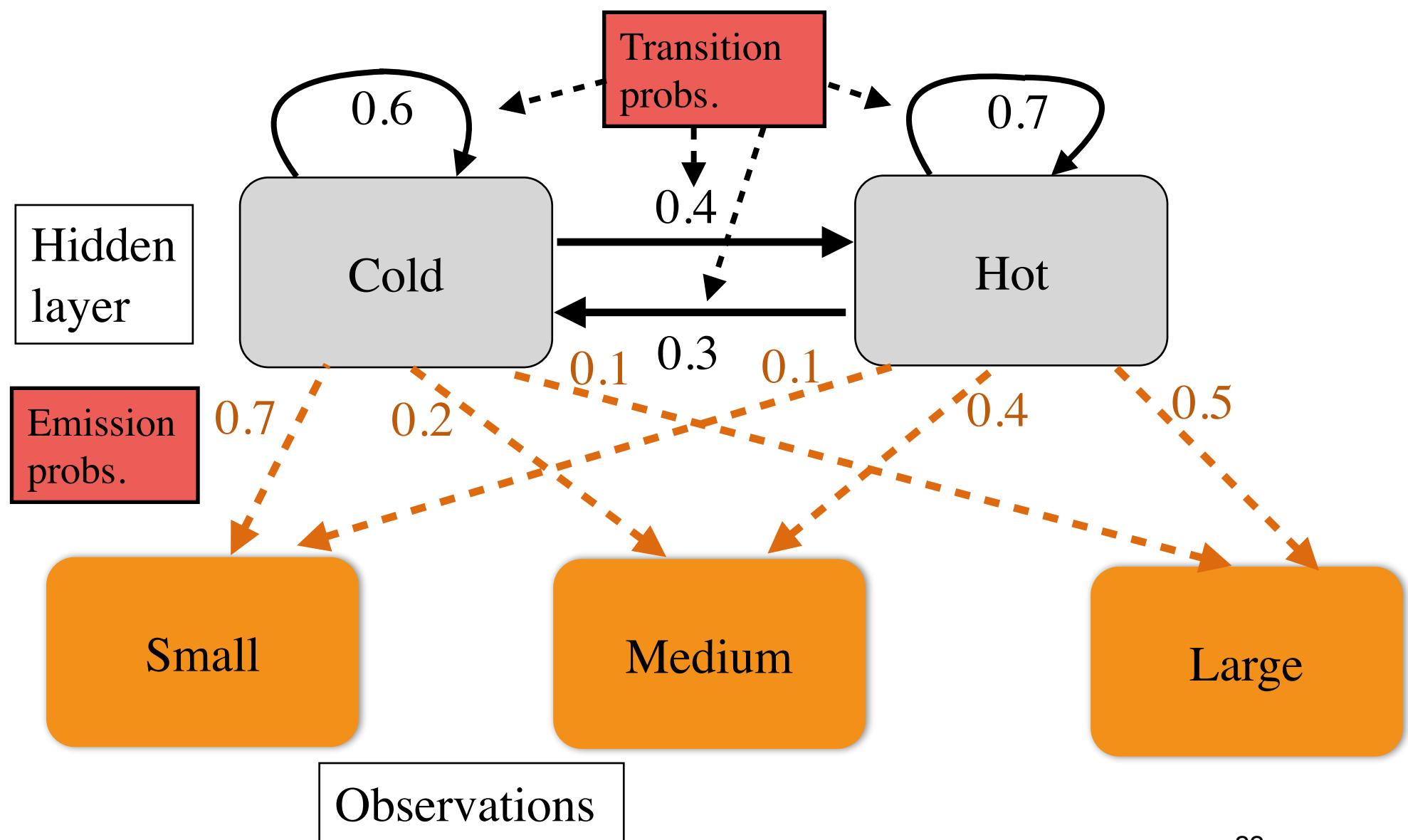
HMM example - graph



HMMs

- **Hidden states:** the underlying state of the process
 - **Observations:** the thing that is actually measured
-
- **Transition matrix:** Describes the way the hidden states change.
 - **Emission matrix** (or observed probability matrix, or confusion mx): describes the conditional probability of seeing each observation as a function of the true underlying state of the process.
-
- Note: The sequence of hidden states is Markovian. The sequence of observed states **is typically not**.

HMM example - graph



HMMs

- Our goal:
 - Given a sequence of observed states that depend upon an underlying hidden process, we want to say something about that underlying process.
 - e.g., which is the most likely sequence of underlying states?
 - Note, we may or may not know both the transition matrix and emission probabilities, so sometimes we also need to infer one of those [much more difficult to do].

HMMs - example from decision theory

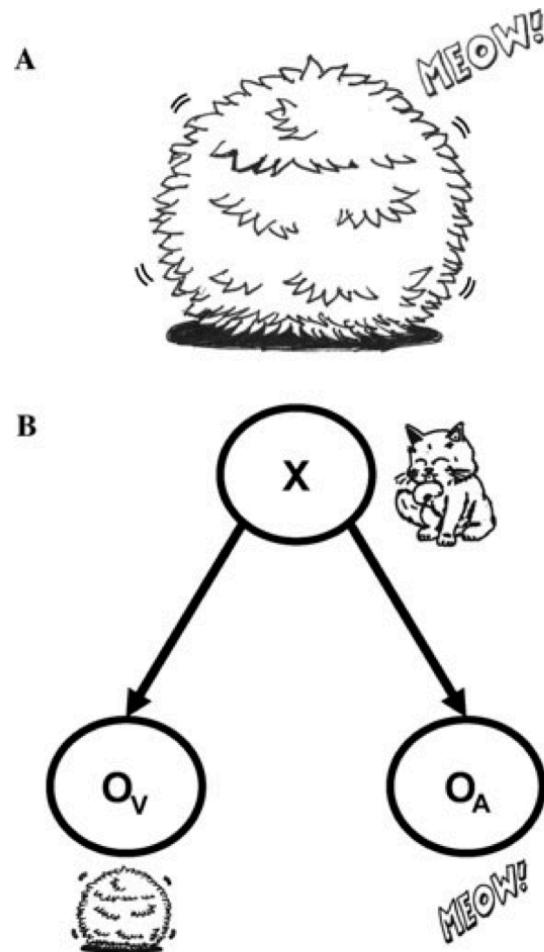
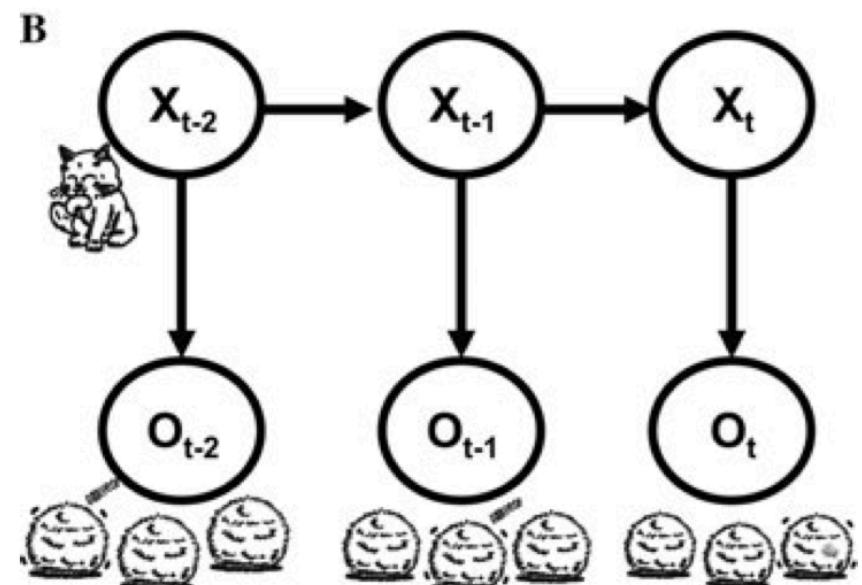
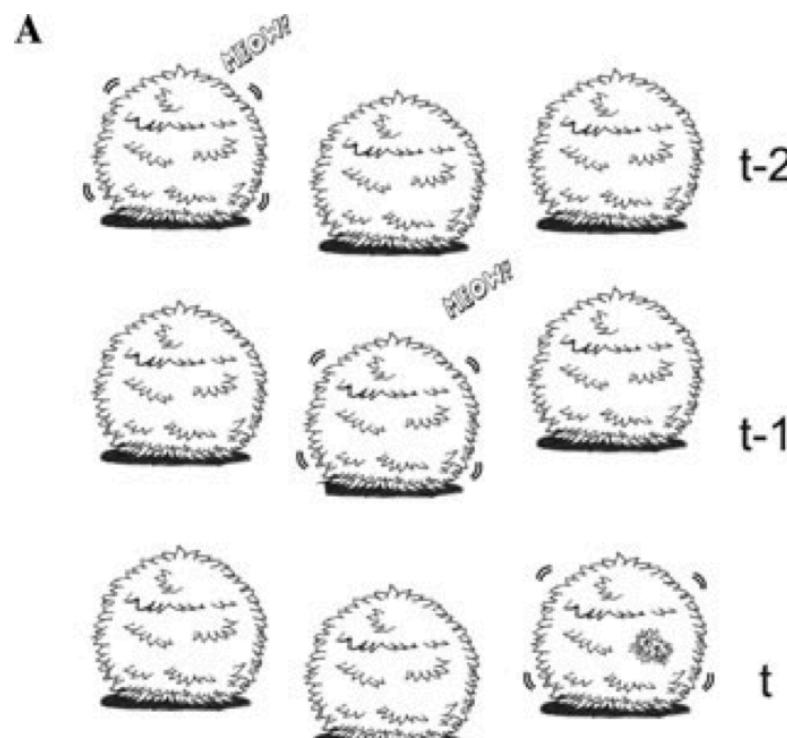


Figure 1. Cue combination. (A) Example of an indirect observation of a cat's position. You can see the bush moving and hear a "meow" sound, but you cannot directly observe the cat. (B) Graphical model of what is seen in A. The variable X (position of the cat) is unobserved, but it produces two observed variables: the moving bush, which provides a visual cue (O_V), and the "meow" sound, which provides an auditory cue (O_A). Cartoons made by Hugo M. Martins.

Vilares, I., & Konrad, K. (2011). Bayesian models: the structure of the world, uncertainty, behavior, and the brain. *Annals of the New York Academy of Sciences*, 1224(1), 22–39.

HMMs - example from decision theory



Vilares, I., & Konrad, K. (2011). Bayesian models: the structure of the world, uncertainty, behavior, and the brain. *Annals of the New York Academy of Sciences*, 1224(1), 22–39.

Figure 3. Combining information across time. (A) Example of an indirect observation of a cat's position, at different points of time ($t-2$, $t-1$, and t , t being the present time). (B) Graphical model of what is seen in A. The hidden variable X (position of the cat) at each point of time produces a variable O that is observed.

HMMs - Notation

T = length of the observation sequence

N = number of states in the model

M = number of observation symbols

X_t = the state of the hidden process at time t [A Markov chain]

O_t = the observation made at time t [Not a Markov chain]

Q = {q₁, q₂, ..., q_N}, the distinct states of the Markov process [The state-space of X]

V = {v₁, v₂, ..., v_M}, the set of possible observations [The state-space of O]

A = {a_{ij}}, the (hidden) state transition probabilities [a_{ij}=P(X_{t+1}=q_j | X_t=q_i)]

B = {b_{ij}}, the observation probability matrix [b_{ij}=P(O_t=v_j | X_t=q_i)]

π = initial (hidden) state distribution.

Example - Fair/unfair dice



- A casino has two dice:
 - Die 1: Each outcome (1,2,3,4,5,6) has prob. $1/6$.
 - Die 2: Rolls a 6 with prob. $1/2$; rolls a 1,2,3,4 or 5 with prob. $1/10$.
- If Die 1 was used last time, then the casino uses Die 2 next time with prob. 0.01; otherwise it uses Die 1 again.
- If Die 2 was used last time, then the casino uses Die 1 next time with prob. 0.02; otherwise it uses Die 2 again.
- You cannot tell which of the two dice is being used for any given roll.
- You observe the sequence of die-rolls.
- Goal: To infer which die is being used when.

Example - Fair/unfair dice

- A casino has two dice:
- If Die 1 was used last time, then the casino uses Die 2 next time with prob. 0.01; otherwise it uses Die 1 again.
- If Die 2 was used last time, then the casino uses Die 1 next time with prob. 0.02; otherwise it uses Die 2 again.



- Index the rolls with $t=1,2,3,\dots$
- Define X_t to be the die used at time t .
- $X_t=1$ if Die 1 was used; $X_t=2$ if Die 2 was used.
- Transition matrix, A :

Hidden layer

$$\begin{bmatrix} 0.99 & 0.01 \\ 0.02 & 0.98 \end{bmatrix}$$

[$a_{ij}=P(X_{t+1}=q_j | X_t=q_i)$]

Example - Fair/unfair dice

- A casino has two dice:
 - Die 1: Each outcome (1,2,3,4,5,6) has prob. 1/6.
 - Die 2: Rolls a 6 with prob. 1/2; rolls a 1,2,3,4 or 5 with prob. 1/10.

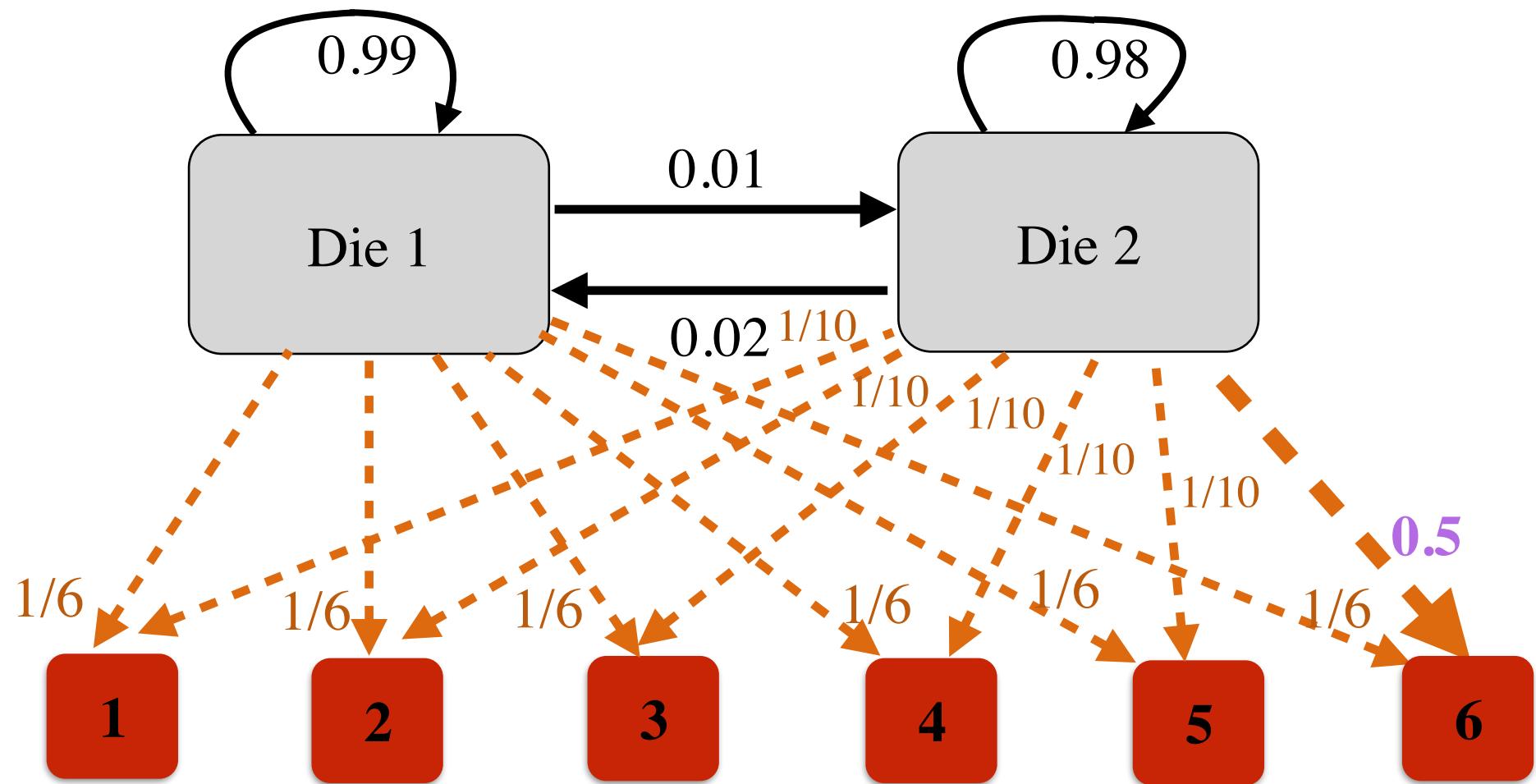


- Index the rolls with $t=1,2,3,\dots$
- Define O_t to be the die-roll observed at time t .
- Emission (confusion) matrix, B :

Observed layer

$$\begin{array}{cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
 1 & \left[\begin{array}{cccccc} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{array} \right] & & & & & \\
 2 & \left[\begin{array}{cccccc} 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/2 \end{array} \right] & & & & & \\
 & [b_{ij} = P(O_t=v_j | X_t=q_i)] & & & & &
 \end{array}$$

HMM example - graph



3 key HMM questions

1. Given the observation sequence $\mathbf{O} = (O_1, O_2, \dots, O_T)$, and a model $\lambda=(A,B,\pi)$ how do we efficiently compute $P(\mathbf{O} | A,B,\pi)$, the probability of the observation sequence, given the model? (Note, this is not conditioned on X , the state of the hidden layer.) **This can also be used to assess model fit.** - Forward-backward algorithm
2. Given the observation sequence $O = (O_1, O_2, \dots, O_T)$, and a model $\lambda=(A,B,\pi)$ how do we choose a corresponding state sequence $\mathbf{X}=X_1,X_2,\dots,X_T$ that is optimal in some meaningful sense (e.g., maximum likelihood). **(i.e., what sequence of hidden states best “explains” the observed data?)** - Viterbi algorithm
3. How do we adjust the model parameters (A,B,π) to maximize $P(O|A,B,\pi)$? **Parameter estimation.** - Baum-Welch algorithm

R package: HMM - Dice example

```
#install.packages('HMM')
library('HMM')
set.seed(985)
nSim = 2000
States = c("Fair", "Unfair")
Symbols = 1:6
# Define the transition matrix
transProbs = matrix(c(0.99, 0.01, 0.02, 0.98), c(length(States),length(States)), byrow = TRUE)
# Define the emission probabilities
emissionProbs = matrix(c(rep(1/6, 6), c(rep(0.1, 5), 0.5)),c(length(States), length(Symbols)), byrow = TRUE)
#Set up the HMM
hmm = initHMM(States, Symbols, transProbs = transProbs, emissionProbs = emissionProbs)
# Simulate some test data to play with
sim = simHMM(hmm, nSim)
# the resulting die-rolls are stored in sim$observation
plot(sim$observation,pch='.')
# the sequence recording which die was rolled is stored in sim$states. Let's look at the first few values
sim$states[1:20]
```

[‘HMMs.R’ on Github in Week13_HiddenMarkovModels]

Initialization

Usage

```
initHMM(States, Symbols, startProbs=NULL, transProbs=NULL, emissionProbs=NULL)
```

Arguments

States	Vector with the names of the states.
Symbols	Vector with the names of the symbols.
startProbs	Vector with the starting probabilities of the states.
transProbs	Stochastic matrix containing the transition probabilities between the states.
emissionProbs	Stochastic matrix containing the emission probabilities of the states.

Simulating test data

Usage

```
simHMM(hmm, length)
```

Arguments

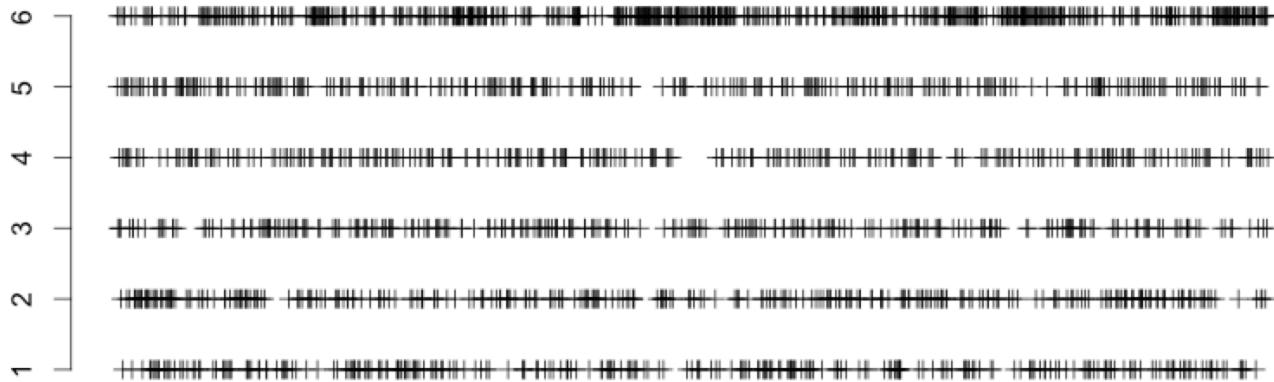
`hmm` A Hidden Markov Model.

`length` The length of the simulated sequence of observations and states.

Format

Dice - Example output

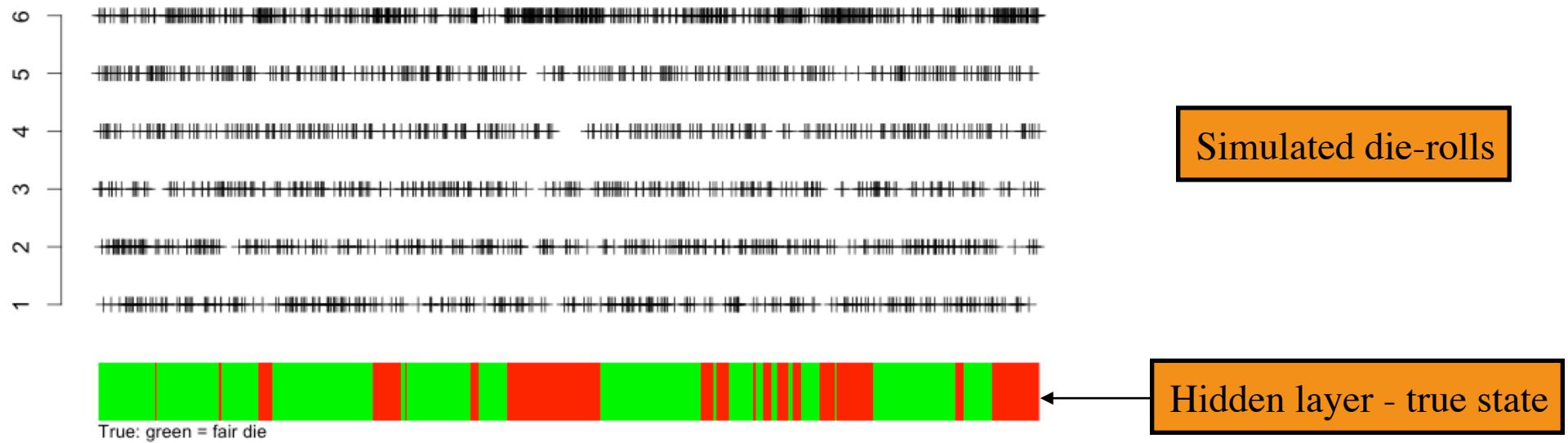
Fair and unfair die



Simulated die-rolls

Dice - Example output

Fair and unfair die



Viterbi algorithm to find most likely path

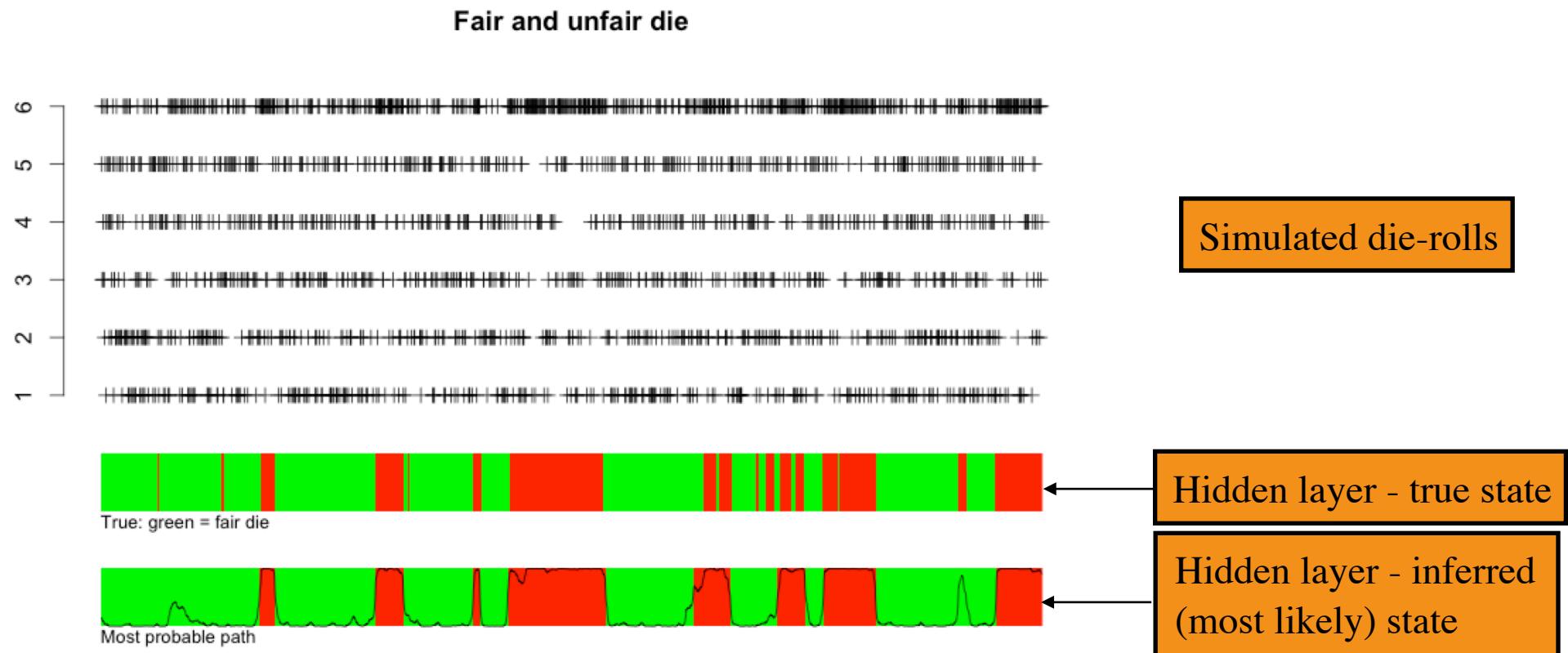
Usage

```
viterbi(hmm, observation)
```

Arguments

hmm	A Hidden Markov Model.
observation	A sequence of observations.

Dice - Example output



Posterior state probs

Description

This function computes the posterior probabilities of being in state X at time k for a given sequence of observations and a given Hidden Markov Model.

Usage

```
posterior(hmm, observation)
```

Arguments

hmm	A Hidden Markov Model.
observation	A sequence of observations.

This uses the output of the forwards (& backwards) algorithms for the probabilities of everything up to (after) a given time.

[It averages over all possible paths, rather than just taking the most likely path.]

Forward algorithm

Description

The `forward`-function computes the forward probabilities. The forward probability for state X up to observation at time k is defined as the probability of observing the sequence of observations e_1, \dots, e_k and that the state at time k is X . That is:

$f[X, k] := \text{Prob}(E_1 = e_1, \dots, E_k = e_k, X_k = X)$.

Where $E_1 \dots E_n = e_1 \dots e_n$ is the sequence of observed emissions and X_k is a random variable that represents the state at time k .

Usage

```
forward(hmm, observation)
```

Arguments

`hmm` A Hidden Markov Model.

`observation` A sequence of observations.

Backward algorithm

Description

The backward-function computes the backward probabilities. The backward probability for state X and observation at time k is defined as the probability of observing the sequence of observations e_{k+1}, \dots, e_n under the condition that the state at time k is X . That is:

$b[X, k] := \text{Prob}(E_{k+1} = e_{k+1}, \dots, E_n = e_n \mid X_k = X)$.

Where $E_1 \dots E_n = e_1 \dots e_n$ is the sequence of observed emissions and X_k is a random variable that represents the state at time k .

Usage

```
backward(hmm, observation)
```

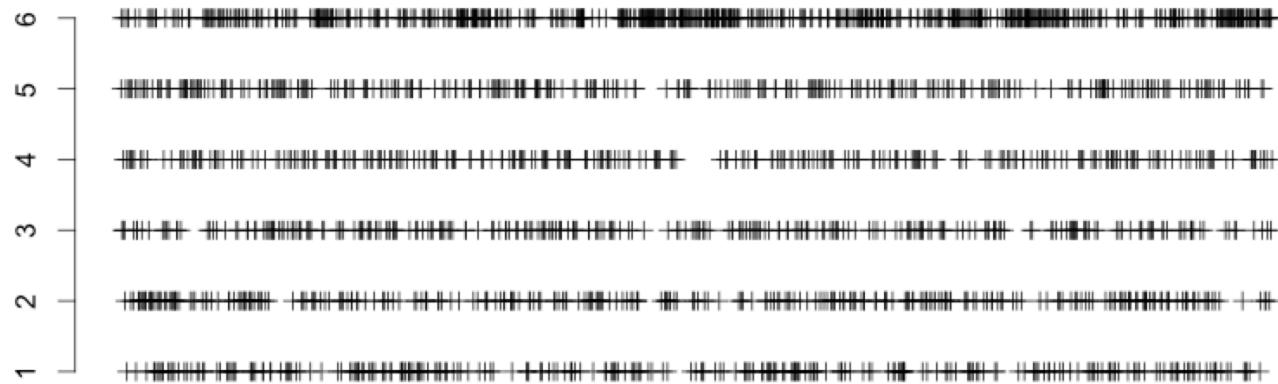
Arguments

`hmm` A Hidden Markov Model.

`observation` A sequence of observations.

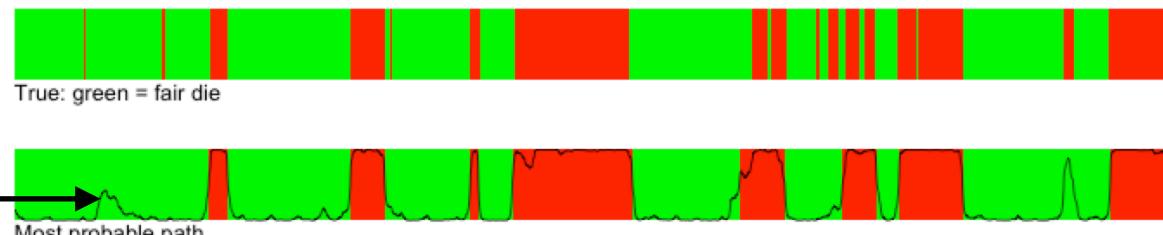
Dice - Example output

Fair and unfair die



Simulated die-rolls

Marginal prob. of using unfair Die

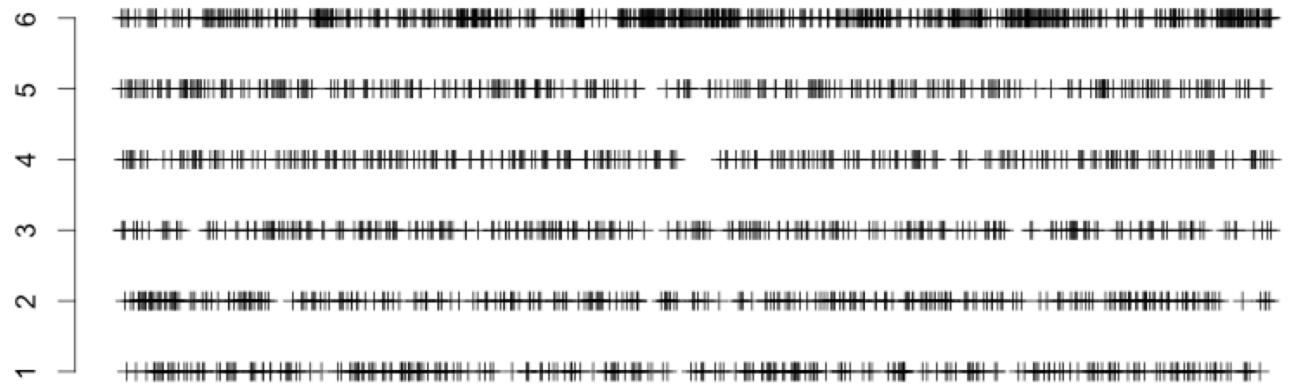


Hidden layer - true state

Hidden layer - inferred (most likely) state

Dice - Example output

Fair and unfair die



Simulated die-rolls

Marginal prob. of using unfair Die



Hidden layer - true state



Hidden layer - inferred (most likely) state



Difference between 'true' and 'inferred'



Difference between 'true' and 'inferred' marginally

0 500 1000 1500 2000

Throw nr.

Example 2 - DNA sequences

- Goal - to detect CpG regions ('methylation islands') in DNA sequence data.
- After <http://web.stanford.edu/class/stats366/exs/HMM1.html> (where you can find more details)

Table of Transition Probabilities for **CpG** Islands

Model +	A	C	G	T
A	.180	.274	.426	.120
C	.171	.368	.274	.188
G	.161	.339	.375	.125
T	.079	.355	.384	.182
station	0.155	0.341	0.350	0.154

Table of Transition Probabilities for **non CpG** Islands

Model -	A	C	G	T
A	.300	.205	.285	.210
C	.322	.298	.078	.302
G	.248	.246	.298	.208
T	.177	.239	.292	.292
station	0.262	0.246	0.239	0.253

- How should we set this up as an HMM?
 - What is the hidden layer?
 - What are the states of the hidden layer?
 - What is the observed layer?
 - What are the states of the observed layer?

Suggestion

- Hidden layer contains both the nucleotide and a flag for whether or not we are in a CpG island:
 - A+, C+, G+, T+, A-, C-, G-, T-
- Observed layer contains just the nucleotide:
 - A, C, G, T
- Code: HMMs_CpGs.R on Blackboard

Transition mx

to

from	A+	C+	G+	T+	A-	C-	G-	T-
A+	0.1620	0.2466	0.3834	0.1080	0.0250	0.0250	0.0250	0.0250
C+	0.1539	0.3312	0.2466	0.1683	0.0250	0.0250	0.0250	0.0250
G+	0.1449	0.3051	0.3375	0.1125	0.0250	0.0250	0.0250	0.0250
T+	0.0711	0.3195	0.3456	0.1638	0.0250	0.0250	0.0250	0.0250
A-	0.0250	0.0250	0.0250	0.0250	0.2700	0.1845	0.2565	0.1890
C-	0.0250	0.0250	0.0250	0.0250	0.2898	0.2682	0.0702	0.2718
G-	0.0250	0.0250	0.0250	0.0250	0.2232	0.2214	0.2682	0.1872
T-	0.0250	0.0250	0.0250	0.0250	0.1593	0.2151	0.2628	0.2628

Emission matrix

symbols	A	C	G	T
states				
A+	0.997	0.001	0.001	0.001
C+	0.001	0.997	0.001	0.001
G+	0.001	0.001	0.997	0.001
T+	0.001	0.001	0.001	0.997
A-	0.997	0.001	0.001	0.001
C-	0.001	0.997	0.001	0.001
G-	0.001	0.001	0.997	0.001
T-	0.001	0.001	0.001	0.997

Inferring of parameters - Baum-Welch algorithm

Description

For an initial Hidden Markov Model (HMM) and a given sequence of observations, the Baum-Welch algorithm infers optimal parameters to the HMM. Since the Baum-Welch algorithm is a variant of the Expectation-Maximisation algorithm, the algorithm converges to a local solution which might not be the global optimum.

Usage

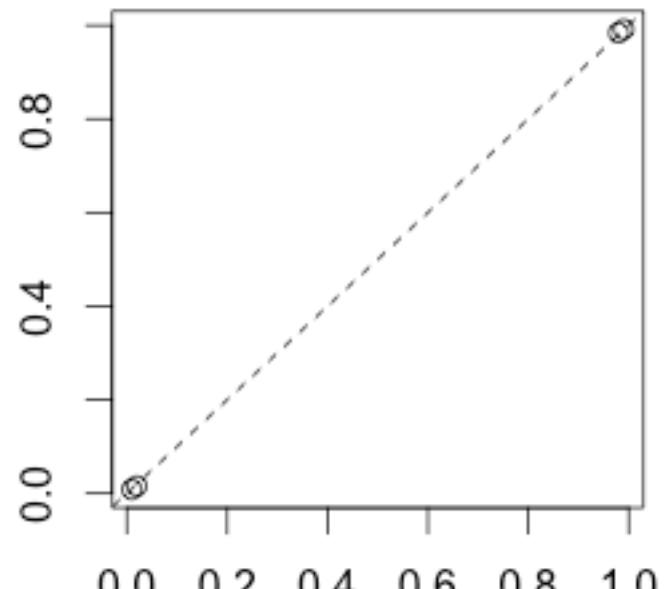
```
baumWelch(hmm, observation, maxIterations=100, delta=1E-9, pseudoCount=0)
```

Arguments

<code>hmm</code>	A Hidden Markov Model.
<code>observation</code>	A sequence of observations.
<code>maxIterations</code>	The maximum number of iterations in the Baum-Welch algorithm.
<code>delta</code>	Additional termination condition, if the transition and emission matrices converge, before reaching the maximum number of iterations (<code>maxIterations</code>). The difference of transition and emission parameters in consecutive iterations must be smaller than <code>delta</code> to terminate the algorithm.
<code>pseudoCount</code>	Adding this amount of pseudo counts in the estimation-step of the Baum-Welch algorithm.

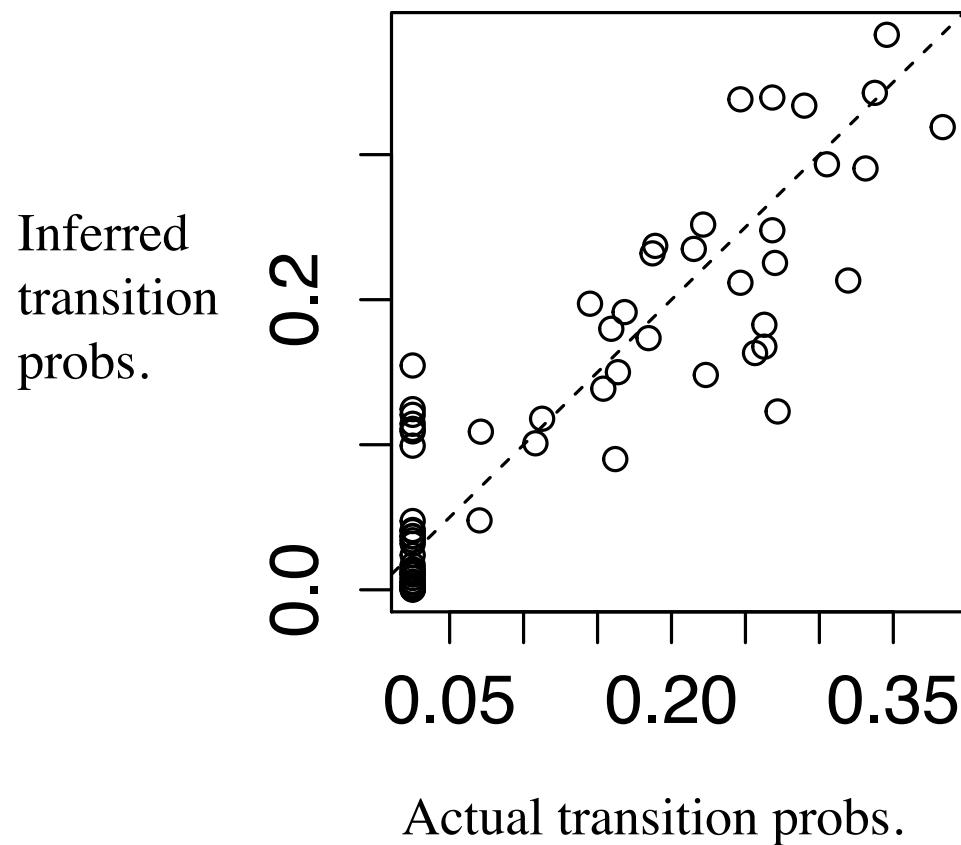
Application to Dice

Inferred
transition
probs.



Actual transition probs.

Application to CpG problem



HMM application

- A Martian lands on earth and starts reading books.
- What can she understand about the English language using an HMM?
- We use a **two-state** model for the underlying hidden state, X.
- Assume that **letter frequencies vary** according to underlying state X.

R. L. Cave and L. P. Neuwirth, Hidden Markov models for English, in J. D. Ferguson, editor, Hidden Markov Models for Speech, IDA-CRD, Princeton, NJ, October 1980.

- Initial conditions

$$\pi = [\begin{array}{cc} 0.51316 & 0.48684 \end{array}]$$

$$A = \begin{bmatrix} 0.47468 & 0.52532 \\ 0.51656 & 0.48344 \end{bmatrix}.$$

Each element of B was initialized to approximately $1/27$. The precise values in the initial B matrix (actually, the transpose of B) appear in the second and third columns of Figure 3.

After the initial iteration, we have

$$\log[P(\mathcal{O} | \lambda)] = -165097.29$$

	Initial	
a	0.03735	0.03909
b	0.03408	0.03537
c	0.03455	0.03537
d	0.03828	0.03909
e	0.03782	0.03583
f	0.03922	0.03630
g	0.03688	0.04048
h	0.03408	0.03537
i	0.03875	0.03816
j	0.04062	0.03909
k	0.03735	0.03490
l	0.03968	0.03723
m	0.03548	0.03537
n	0.03735	0.03909
o	0.04062	0.03397
p	0.03595	0.03397
q	0.03641	0.03816
r	0.03408	0.03676
s	0.04062	0.04048
t	0.03548	0.03443
u	0.03922	0.03537
v	0.04062	0.03955
w	0.03455	0.03816
x	0.03595	0.03723
y	0.03408	0.03769
z	0.03408	0.03955
space	0.03688	0.03397

Initial emission
probabilities

Now run the Baum-Welch algorithm:

After the initial iteration, we have

$$\log[P(\mathcal{O} | \lambda)] = -165097.29$$

and after 100 iterations,

$$\log[P(\mathcal{O} | \lambda)] = -137305.28.$$

After 100 iterations, the model $\lambda = (A, B, \pi)$ has converged to

$$\pi = \begin{bmatrix} 0.00000 & 1.00000 \end{bmatrix}$$

and

$$A = \begin{bmatrix} 0.25596 & 0.74404 \\ 0.71571 & 0.28429 \end{bmatrix}$$

	Initial		Final	
a	0.03735	0.03909	0.13845	0.00075
b	0.03408	0.03537	0.00000	0.02311
c	0.03455	0.03537	0.00062	0.05614
d	0.03828	0.03909	0.00000	0.06937
e	0.03782	0.03583	0.21404	0.00000
f	0.03922	0.03630	0.00000	0.03559
g	0.03688	0.04048	0.00081	0.02724
h	0.03408	0.03537	0.00066	0.07278
i	0.03875	0.03816	0.12275	0.00000
j	0.04062	0.03909	0.00000	0.00365
k	0.03735	0.03490	0.00182	0.00703
l	0.03968	0.03723	0.00049	0.07231
m	0.03548	0.03537	0.00000	0.03889
n	0.03735	0.03909	0.00000	0.11461
o	0.04062	0.03397	0.13156	0.00000
p	0.03595	0.03397	0.00040	0.03674
q	0.03641	0.03816	0.00000	0.00153
r	0.03408	0.03676	0.00000	0.10225
s	0.04062	0.04048	0.00000	0.11042
t	0.03548	0.03443	0.01102	0.14392
u	0.03922	0.03537	0.04508	0.00000
v	0.04062	0.03955	0.00000	0.01621
w	0.03455	0.03816	0.00000	0.02303
x	0.03595	0.03723	0.00000	0.00447
y	0.03408	0.03769	0.00019	0.02587
z	0.03408	0.03955	0.00000	0.00110
space	0.03688	0.03397	0.33211	0.01298

Initial and final
emission
probabilities

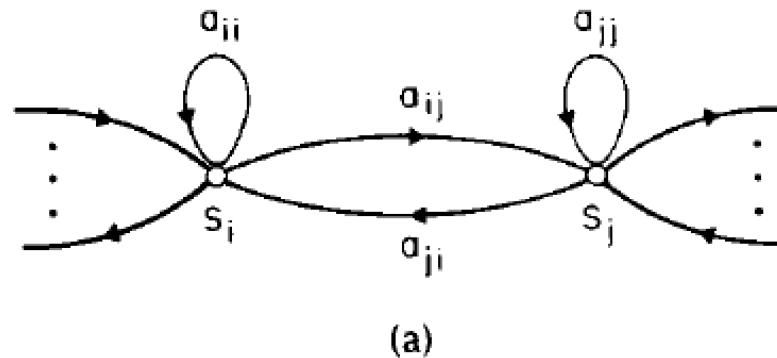
	Initial		Final	
a	0.03735	0.03909	0.13845	0.00075
b	0.03408	0.03537	0.00000	0.02311
c	0.03455	0.03537	0.00062	0.05614
d	0.03828	0.03909	0.00000	0.06937
e	0.03782	0.03583	0.21404	0.00000
f	0.03922	0.03630	0.00000	0.03559
g	0.03688	0.04048	0.00081	0.02724
h	0.03408	0.03537	0.00066	0.07278
i	0.03875	0.03816	0.12275	0.00000
j	0.04062	0.03909	0.00000	0.00365
k	0.03735	0.03490	0.00182	0.00703
l	0.03968	0.03723	0.00049	0.07231
m	0.03548	0.03537	0.00000	0.03889
n	0.03735	0.03909	0.00000	0.11461
o	0.04062	0.03397	0.13156	0.00000
p	0.03595	0.03397	0.00040	0.03674
q	0.03641	0.03816	0.00000	0.00153
r	0.03408	0.03676	0.00000	0.10225
s	0.04062	0.04048	0.00000	0.11042
t	0.03548	0.03443	0.01102	0.14392
u	0.03922	0.03537	0.04508	0.00000
v	0.04062	0.03955	0.00000	0.01621
w	0.03455	0.03816	0.00000	0.02303
x	0.03595	0.03723	0.00000	0.00447
y	0.03408	0.03769	0.00019	0.02587
z	0.03408	0.03955	0.00000	0.00110
space	0.03688	0.03397	0.33211	0.01298

Initial and final
emission
probabilities

vowels

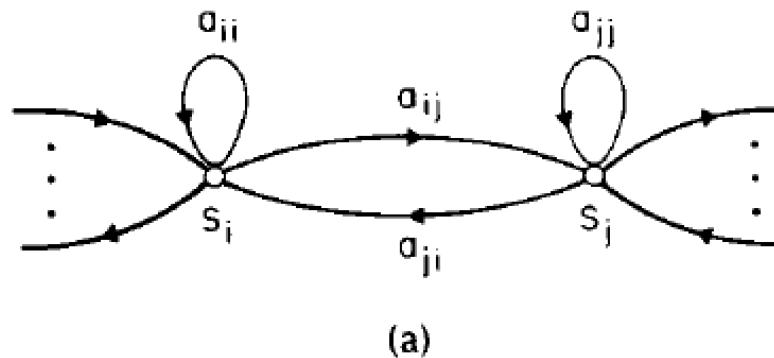
Figure 3: Initial and final B transpose

Duration in states



What is the distribution of time spent in each state?

Duration in states

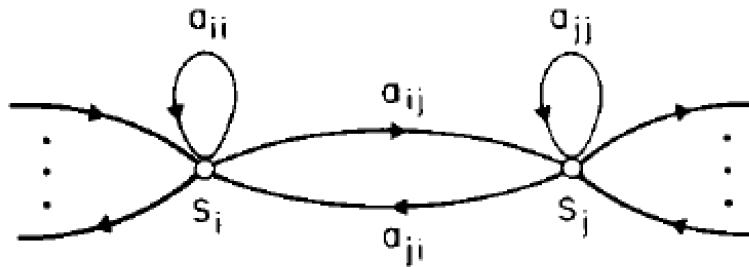


What is the distribution of time spent in each state?

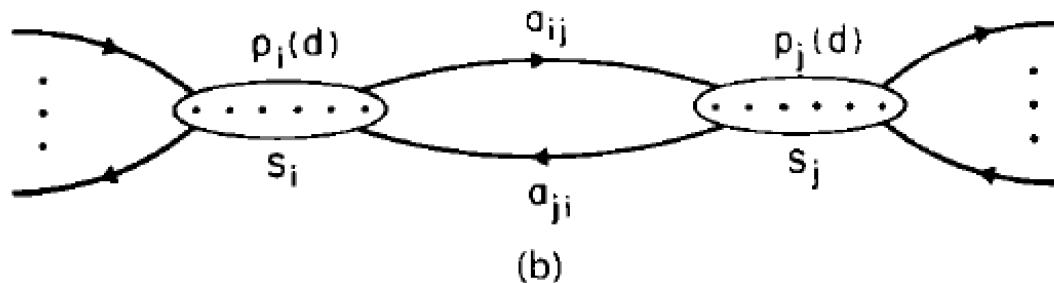
Discrete time -> duration is geometrically distributed

Continuous time -> duration is exponentially distributed

Duration in states



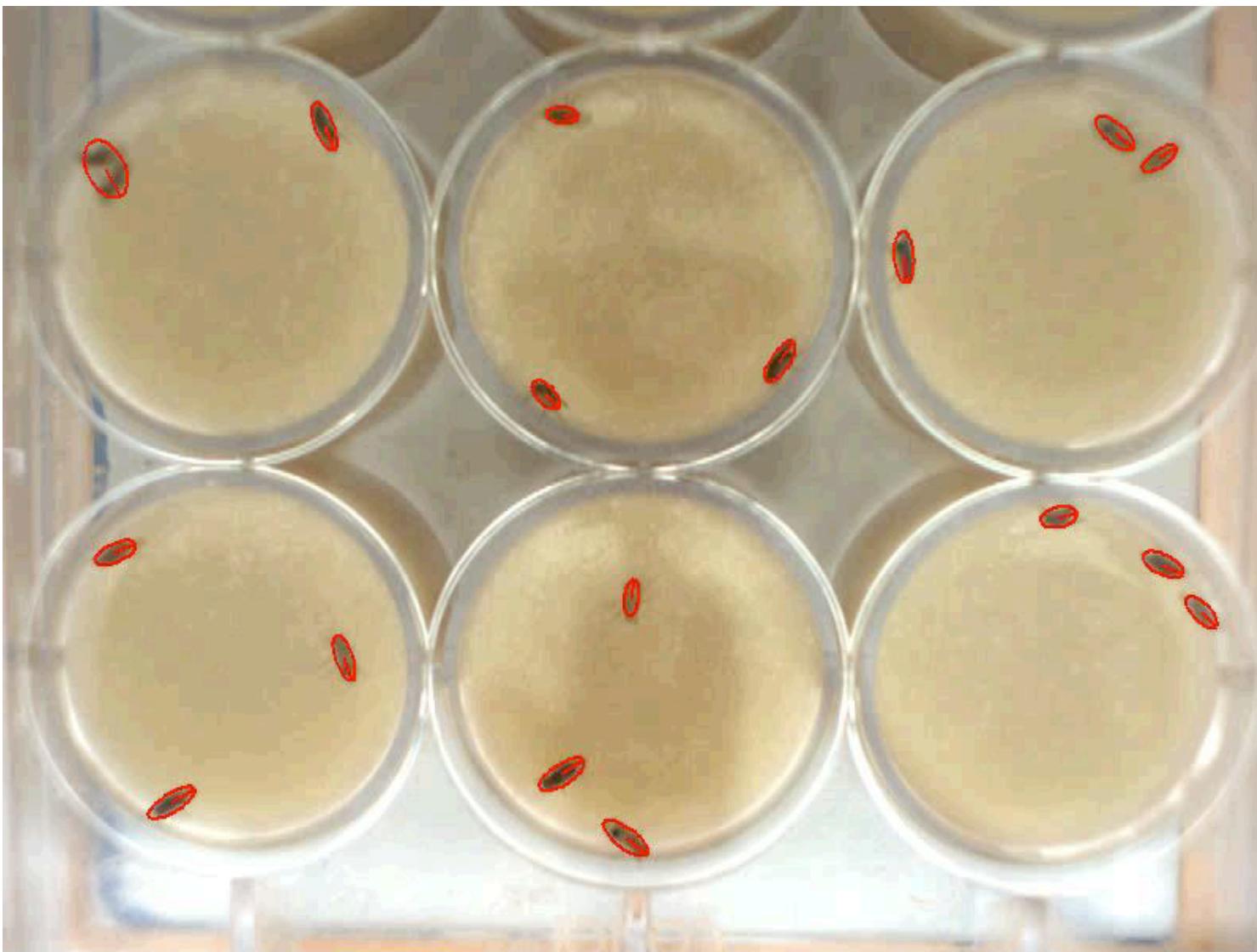
(a)



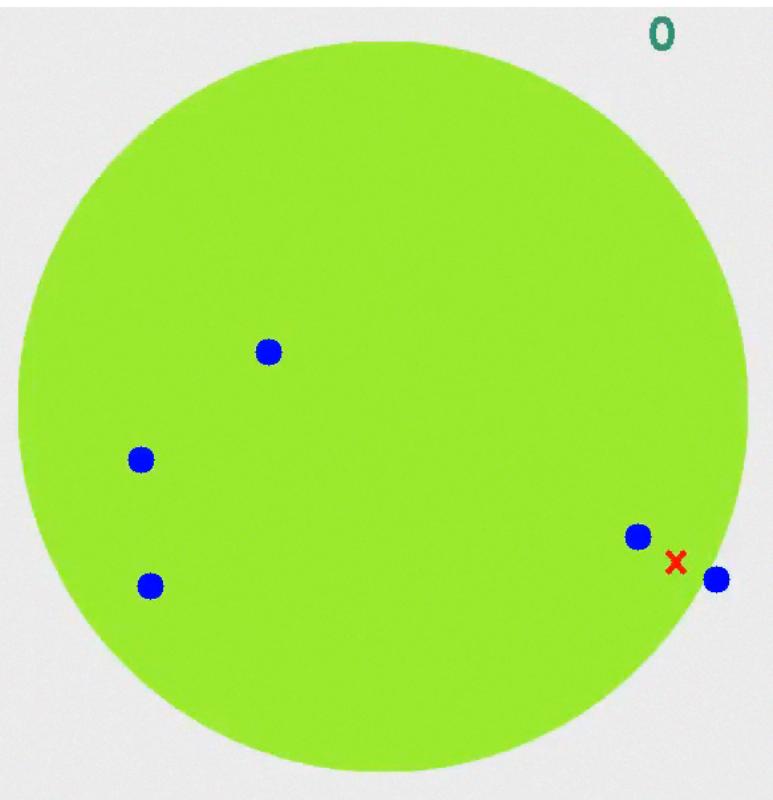
(b)

Generalized HMMs [(b) above] remove ‘self’ \rightarrow ‘self’ loops and instead define an arbitrary distribution for duration in each state.

Application



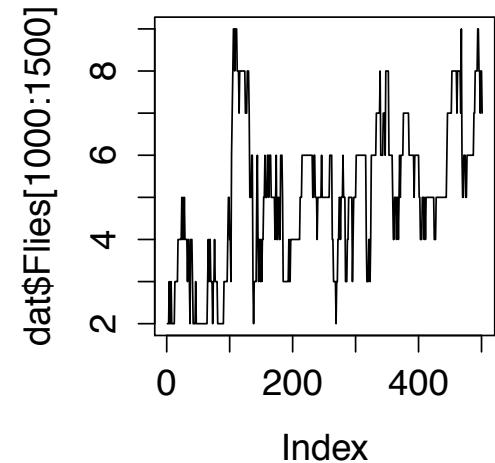
Application extension



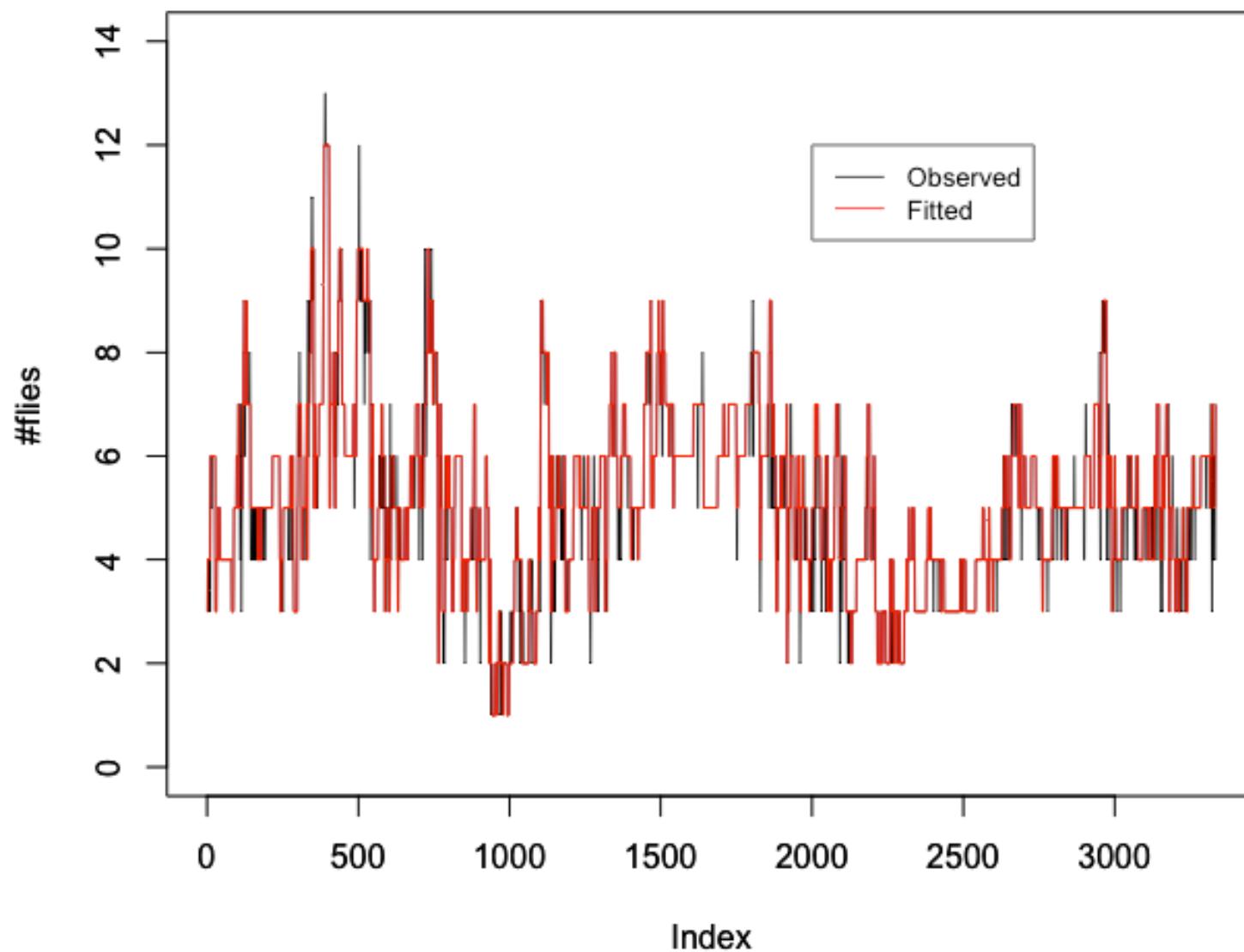
Goal: To automatically and accurately detect the number of flies on the food patch.

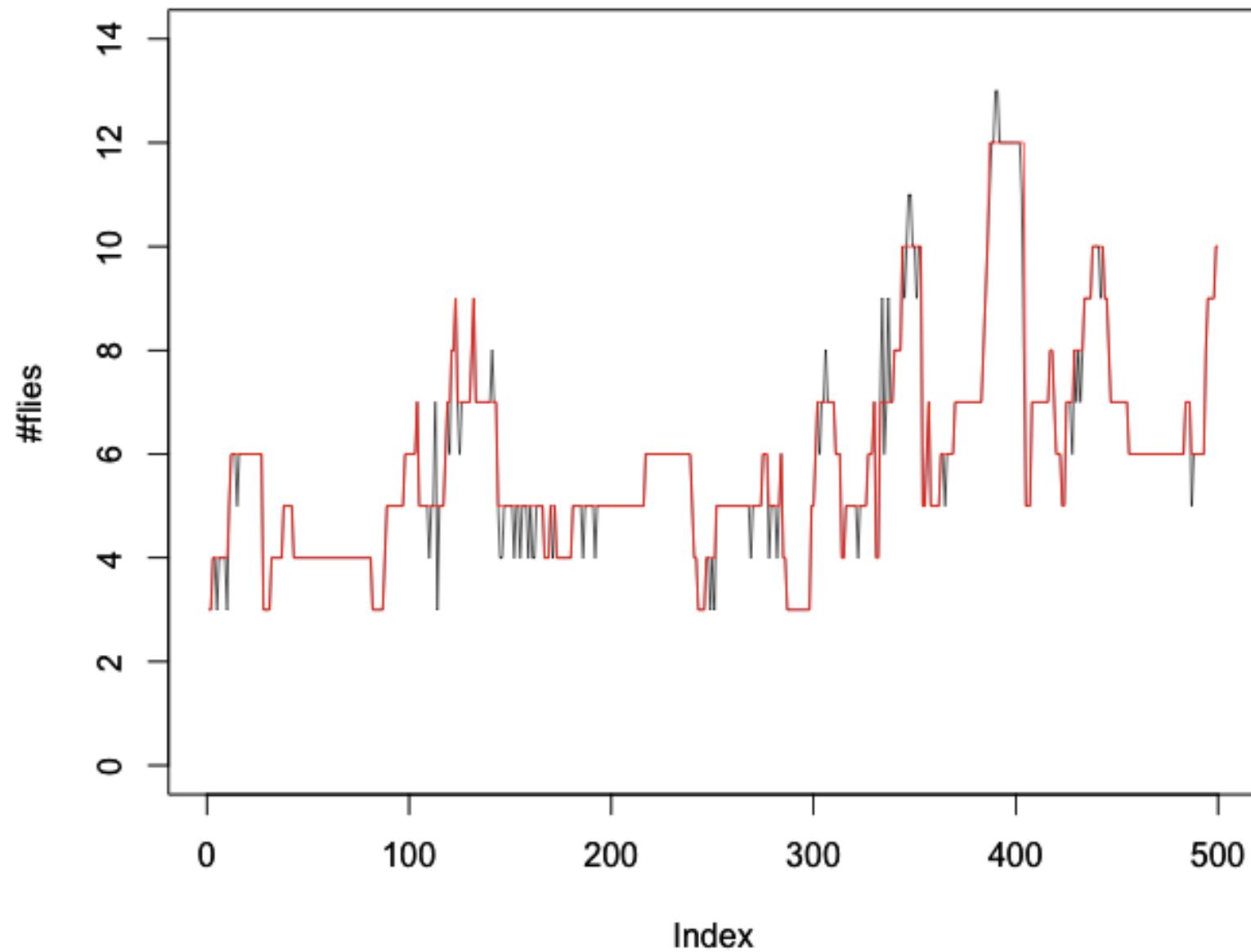
HMM elements for this problem

1. A finite number, N , of (hidden) states in the model. **The number of flies on the food patch.**
2. **A Markov Chain that describes how the number of flies changes from frame-to-frame.**
3. After each (potential) transition some observation is made. **The observation is the number of 'flies' (i.e., dark patches) that are detected by the image processing software.**



Can we use HMMs to smooth out the data (remove tracking noise)?





Further reading

Nice review paper:

Visser, I. (2011). “Seven things to remember about hidden Markov models: A tutorial on Markovian models for time series”. *Journal of Mathematical Psychology*, 55(6), 403–415.
doi:10.1016/j.jmp.2011.08.002

END