DD2380

Artificial Intelligence

HMM

Patric Jensfelt

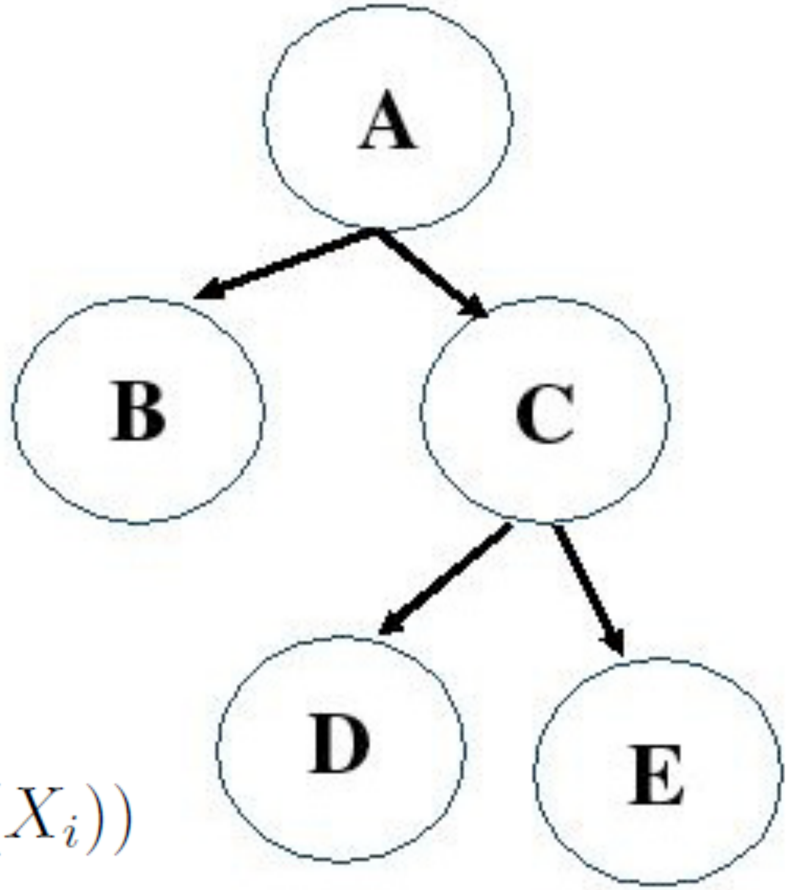
Reading instructions

- Chapters 13-15 in the book
- Stamp tutorial on HMMs on the course web page

Bayesian network

- Aka **Probabilistic Directed Acyclic Graphical Model**
- Represents joint probability distribution
- Arrow → "direct influence over"
  A has direct influence over B
- Each arrow is accompanied with a conditional probability distribution, e.g. p(B|A)
- Factorization

$$p(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{i=n} p(X_i | Parents(X_i))$$



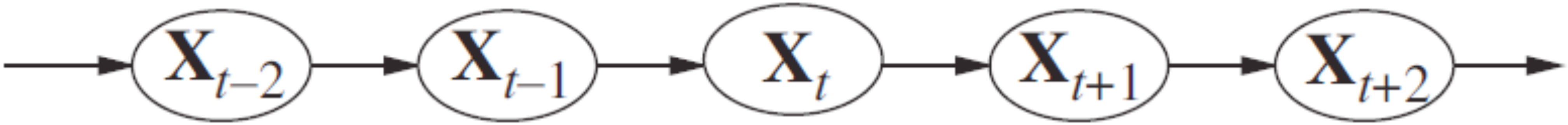
Bayesian network

- Compact representation of the joint distribution over a set of variables
- Graphical representation that helps
  - analyze prob. Information
  - structure prob. Information

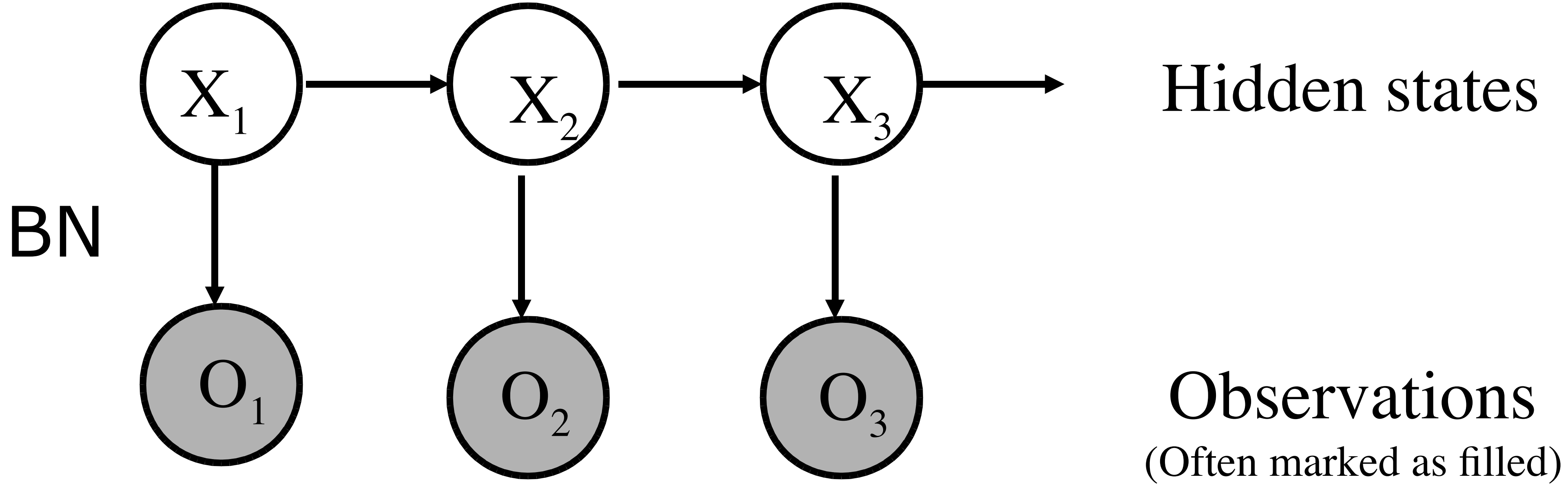
- Very hard to gather data to build a model for p(A,B,C,D,E), much easier to look at conditional probabilities such as p(B|A)

Markov model

- The (first order) Markov assumption
  - The distribution p(X_t) depends only on the distribution p(X_{t-1})
  - The present (current state) can be predicted using local knowledge of the past (state at the previous step)

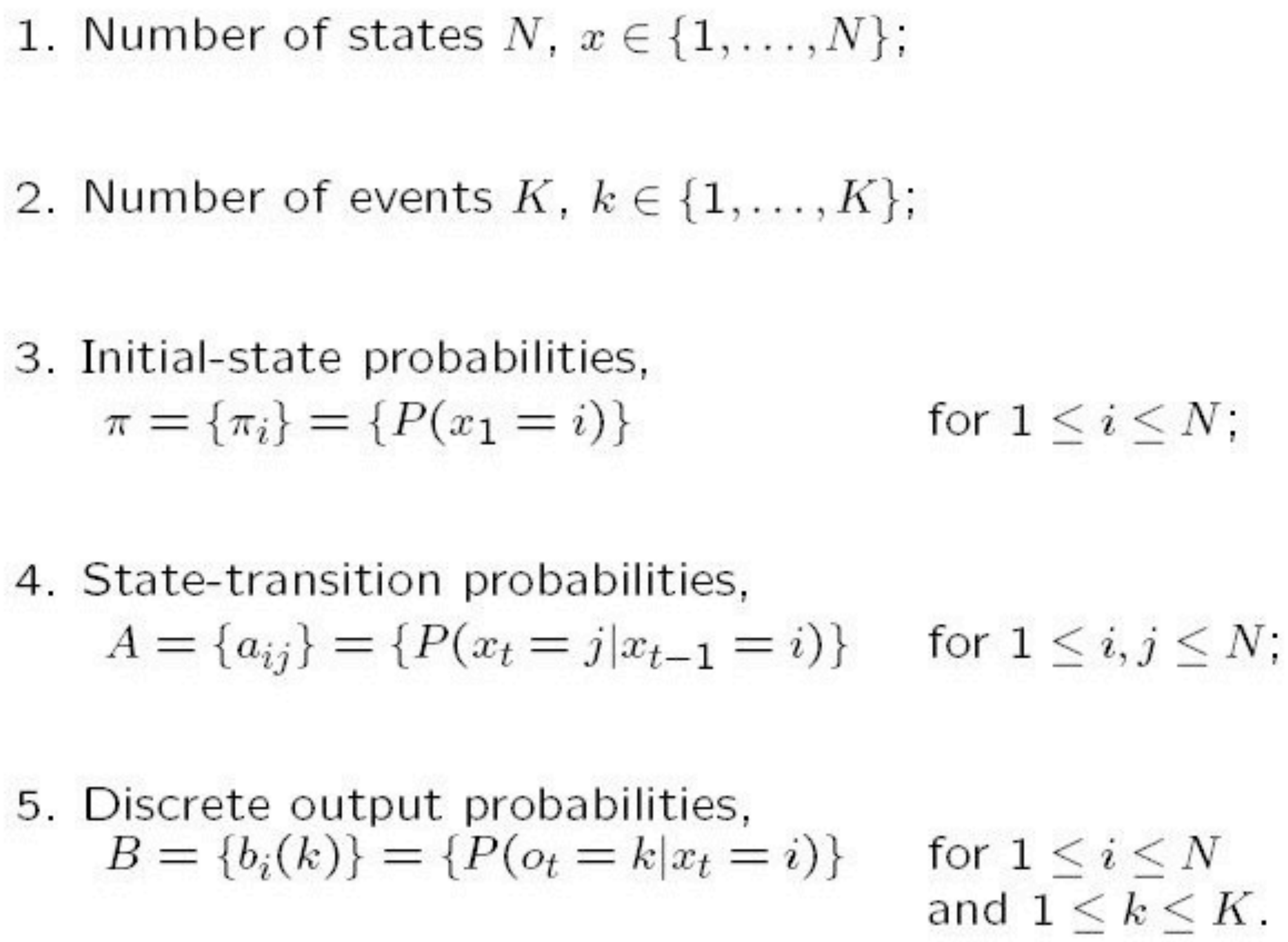


Hidden Markov Models (HMMs)



- State transition model: $p(X_t=j|X_{t-1}=i)=A(i,j)=a_{ij}$
- Observation model: $p(O_t=j|X_t=i)=b_{ij}$

Elements of discrete HMM



1. Number of states N, $x \in \{1, \ldots, N\}$;

2. Number of events K, $k \in \{1, \ldots, K\}$;

3. Initial-state probabilities,
   $\pi = \{\pi_i\} = \{P(x_1 = i)\}$ for $1 \leq i \leq N$;

4. State-transition probabilities,
   $A = \{a_{ij}\} = \{P(x_t = j | x_{t-1} = i)\}$ for $1 \leq i, j \leq N$;

5. Discrete output probabilities,
   $B = \{b_i(k)\} = \{P(o_t = k | x_t = i)\}$ for $1 \leq i \leq N$ and $1 \leq k \leq K$.

Elements of discrete HMM: π

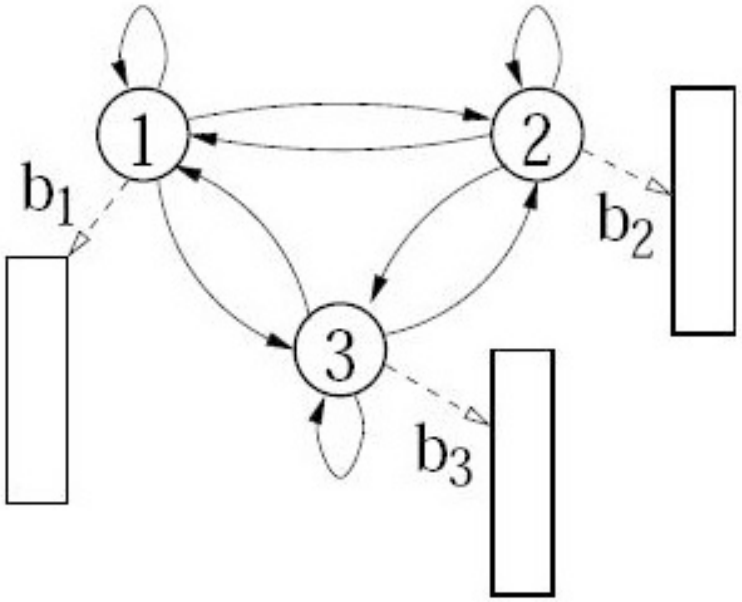
- Initial Distribution : contains the probability of the (hidden) model being in a particular hidden state at time t = 1 (sometimes t=0).
- Often referred to as π

- Ex: $\pi=[0.5\ 0.2\ 0.3]$, i.e.,

  $p(X_1=1)=0.5$

  $p(X_1=2)=0.2$

  $p(X_1=3)=0.3$



Elements of discrete HMM: A

- State transition matrix : holding the probability of transitioning from one hidden state to another hidden state.
- Ex: a_{21} gives $p(X_{t+1}=1|X_t=2)$, i.e. probability to transition from state 2 to state 1

	$X_{t+1}=1$	$X_{t+1}=2$	…	$X_{t+1}=N$
$X_t=1$	a_{11}	a_{12}	…	a_{1N}
$X_t=2$	a_{21}	a_{22}	…	a_{2N}
…	…	…	…	…
$X_t=N$	a_{N1}	a_{N2}	…	a_{NN}

Elements of discrete HMM: B

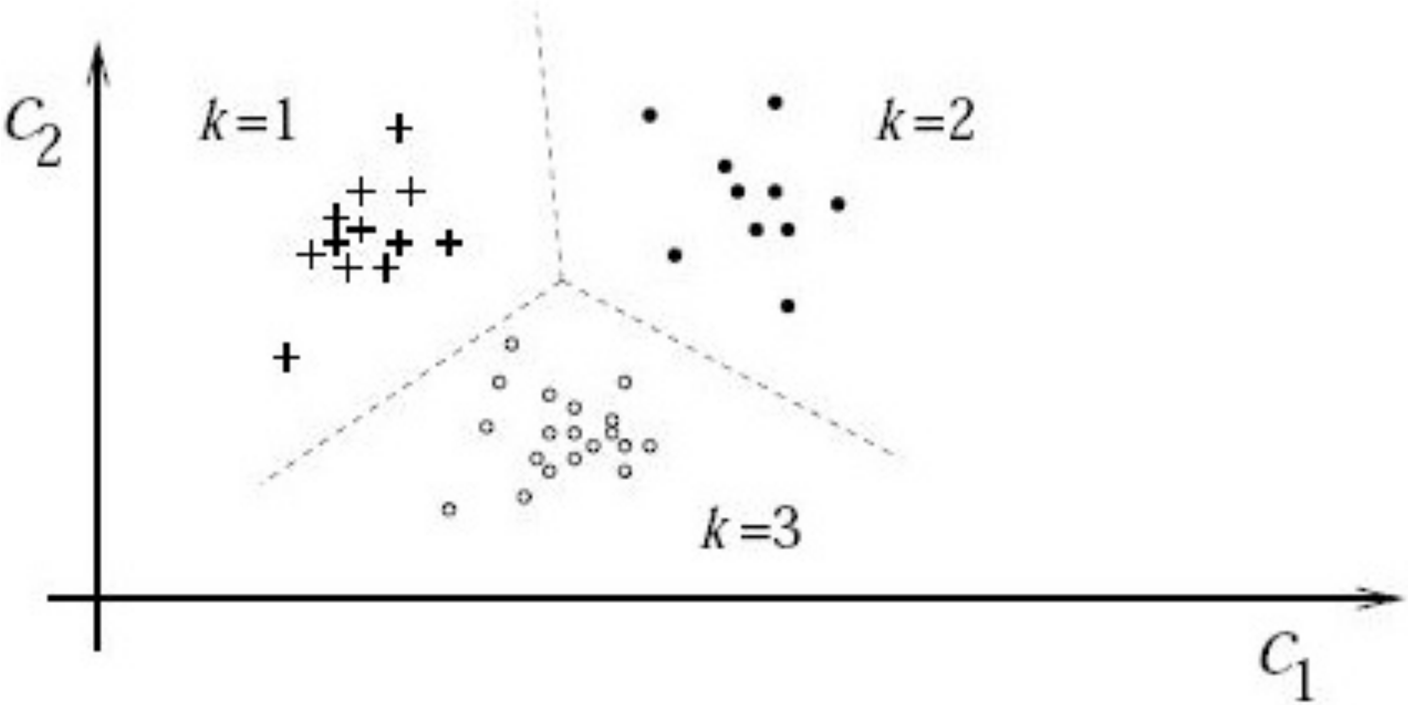
- Output matrix : containing the probability of observing a particular measurement given that the hidden model is in a particular hidden state.

	$O_t=1$	$O_t=2$	…	$O_t=K$
$X_t=1$	$b_1(1)$	$b_1(2)$	…	$b_1(K)$
$X_t=2$	$b_2(1)$	$b_2(2)$	…	$b_2(K)$
…	…	…	…	…
$X_t=N$	$b_N(1)$	$b_N()$	…	$b_N(K)$

- $b_j(O_t)$ is the probability to observe O_t in state j

Discretised observations

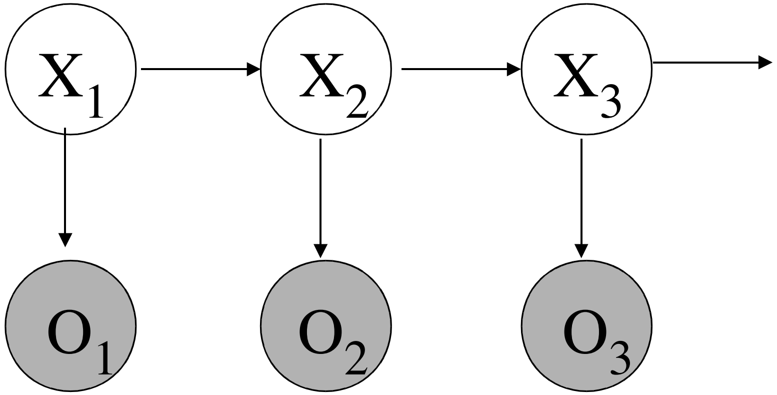
- If the observations are not discrete we can make them discrete by assigning them to different classes



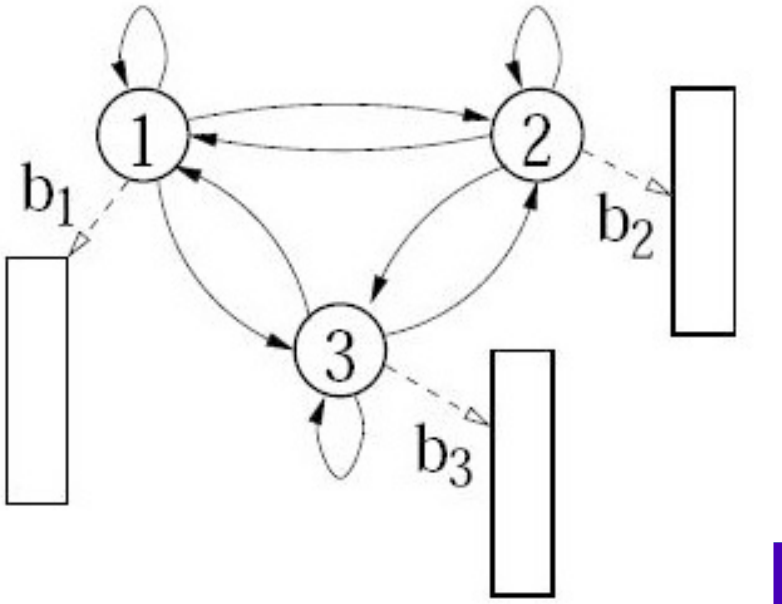
Connect BN and HMMs

- The two diagrams below show different things

BN
Arrows: Dependency between variables
Circles: Variables



HMM
Arrows: State transitions and observation probs
Circles: The different states

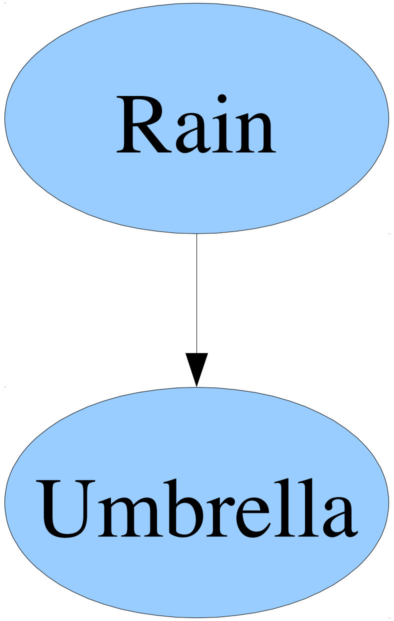


Example: Umbrella world

- A person tries to infer the weather outside (rain={true,false}) by observing if a certain person has an umbrella={true,false} that day.
- Draw Bayesian network!
  - What are the variable?
  - Which cause which?

Example: Umbrella world

- A person tries to infer the weather outside (rain={true,false}) by observing if a certain person has an umbrella={true,false} that day.
- Draw Bayesian network



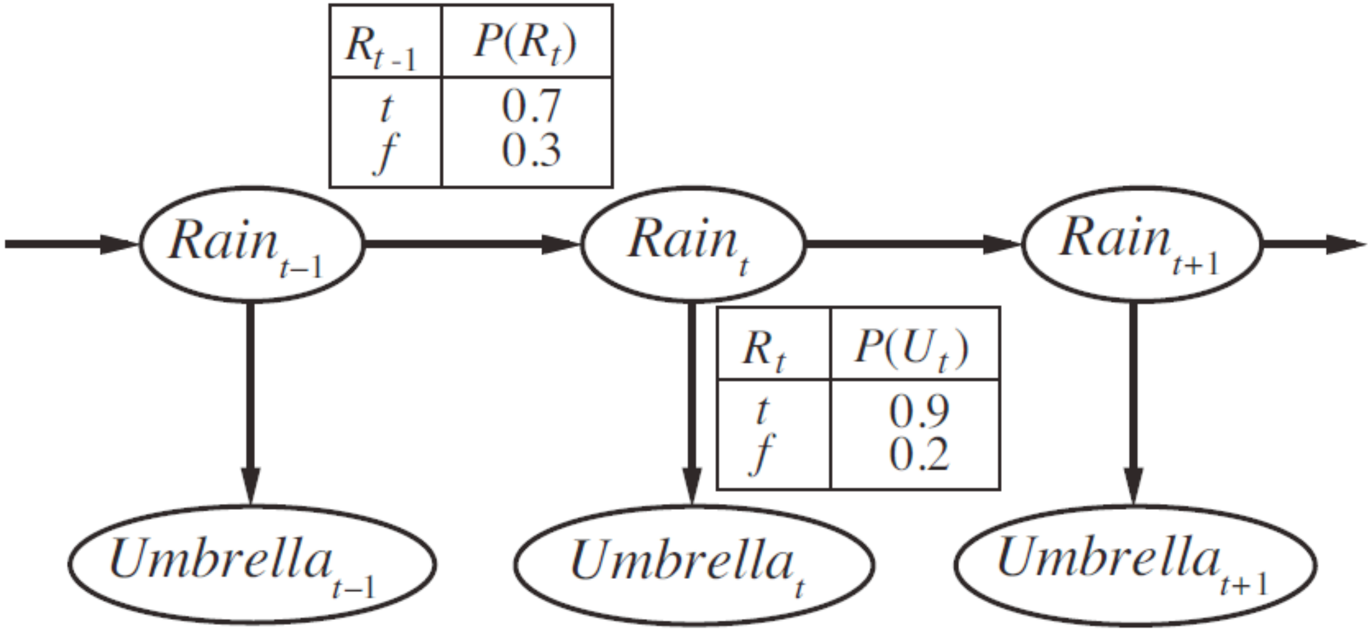
Here, we cannot directly observe if it rains. This is **hidden** to us. We observe the umbrella and infer if it rains or not. Rain causes the person to use an umbrella

Example: Umbrella world

- A person tries to infer the weather outside (rain={true,false}) **every day** by observing if a certain person has an umbrella={true,false} that day.
- Draw the Bayesian network that corresponds to the time sequence (the person makes observations over consecutive days).
- What additional assumptions did you make?

Example: Umbrella world

- At each time step the state can be either Rain=true or Rain=false. The observation depends directly on the state.
- Additional assumption: The next state depends only on the previous state (Markov assumption).



R_{t-1}	$P(R_t)$
t	0.7
f	0.3

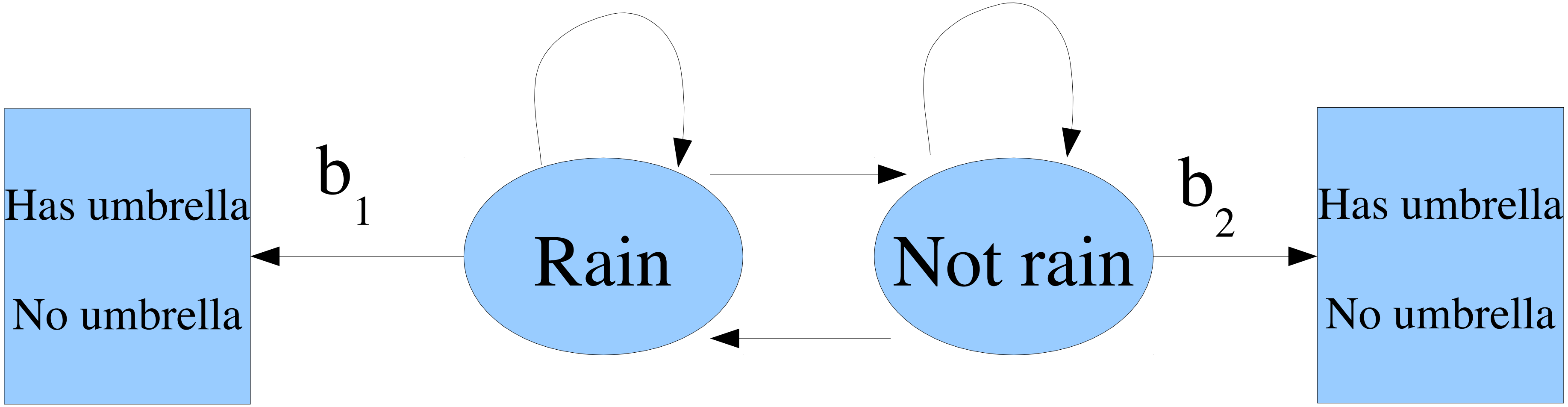
R_t	$P(U_t)$
t	0.9
f	0.2

Example: Umbrella world

- A person tries to infer the weather outside (rain={true,false}) **every day** by observing if a certain person has an umbrella={true,false} that day. Formulate as an HMM.
  - How many states? Which?
  - How many observations? Which?

Example: Umbrella world

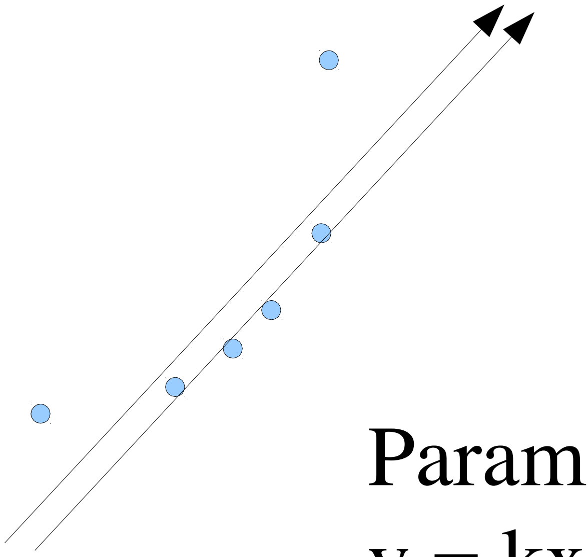
- A person tries to infer the weather outside (rain={true,false}) **every day** by observing if a certain person has an umbrella={true,false} that day. Formulate as an HMM.
  - States: {Rain=true, Rain=false}
  - Observations: {Umbrella=true, Umbrella=false}



What is an HMM, again...

- It is a **model** (not necessarily a perfect one) to describe a system / data from a system
- It can be used to (e.g.)
  - Generate predictions about how the system will behave. Ex: stock market
  - Analyze if a sequence of measurements match a certain model, e.g., did the person say “Bayesian” (ie match our model for how that sounds) or “Bearnaise” (match that model)?
  - Learn something about a system by learning the model parameters.

How to model the data?

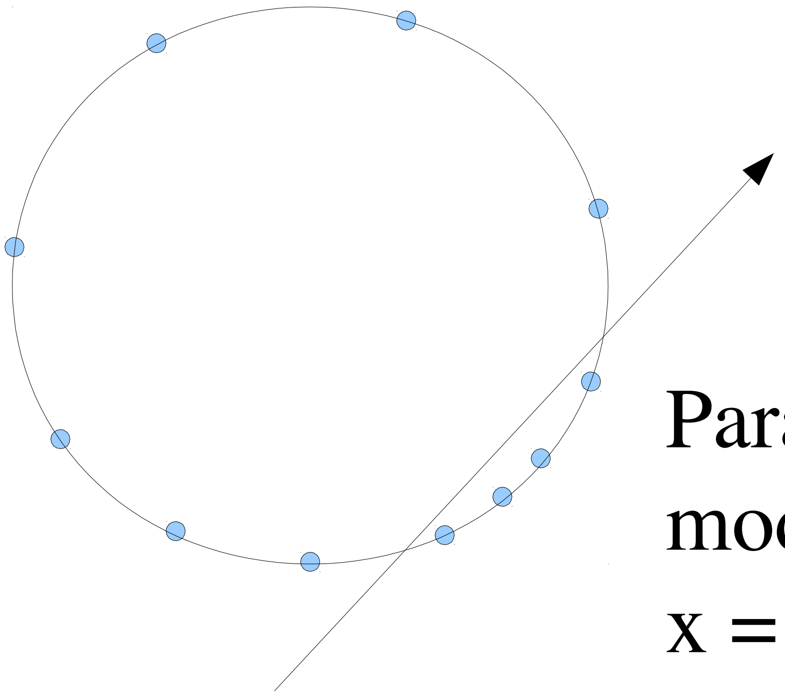


Parametric model

$y = kx + m$

How to model the data?

- Still a line?



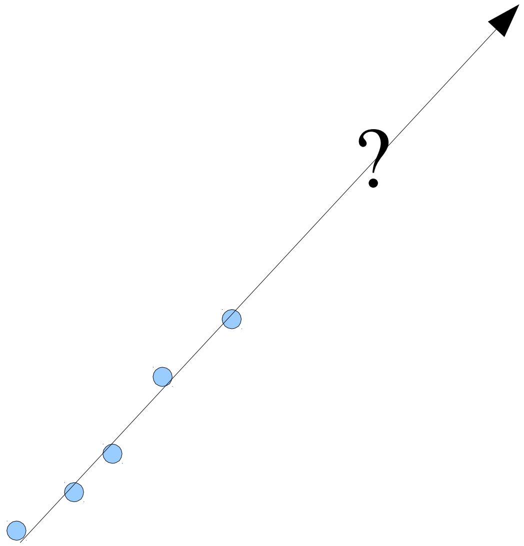
Parametric model

$x = x_0 + r\cos(\theta)$

$y = y_0 + r\sin(\theta)$

Predict using the model

- What happens next?



Look at
Ex: Usain Bolt

What is A,B,π?
Answer Q1-4

Ex: Usain Bolt

- What does A, B and π look like? (4th state non-emitting / silent)

$$A=\begin{bmatrix} 0.3 & 0.7 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B=\begin{bmatrix} 0.5 & 0.2 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.7 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0.5 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\pi=\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

$p(X_2$=ACCELERATE)?

- Start with π and propagate probability one step with A!
- $\pi=X_1=[1\ 0\ 0\ 0]$ → $X_2=\pi A=[0.3\ 0.7\ 0\ 0]$

→ $p(X_2$=ACCELERATE)=0.7

$p(O_2=F2)$?

- We know the distribution for X_2

  $p(X_2)=[0.3\ 0.7\ 0\ 0]$

  and we know how probable each measurement is in each state. Use sum rule!!

$$p(0_2=F_2)=\sum_{i=1}^{N} p(O_2=F_2|X_2=i)\,p(X_2=i)$$

$$=0.2\cdot 0.3+0\cdot 0.7+0\cdot 0+0\cdot 0=0.06$$

p(X$_2$=ACCELERATE | O$_2$=F$_2$)?

- Use Bayes rule

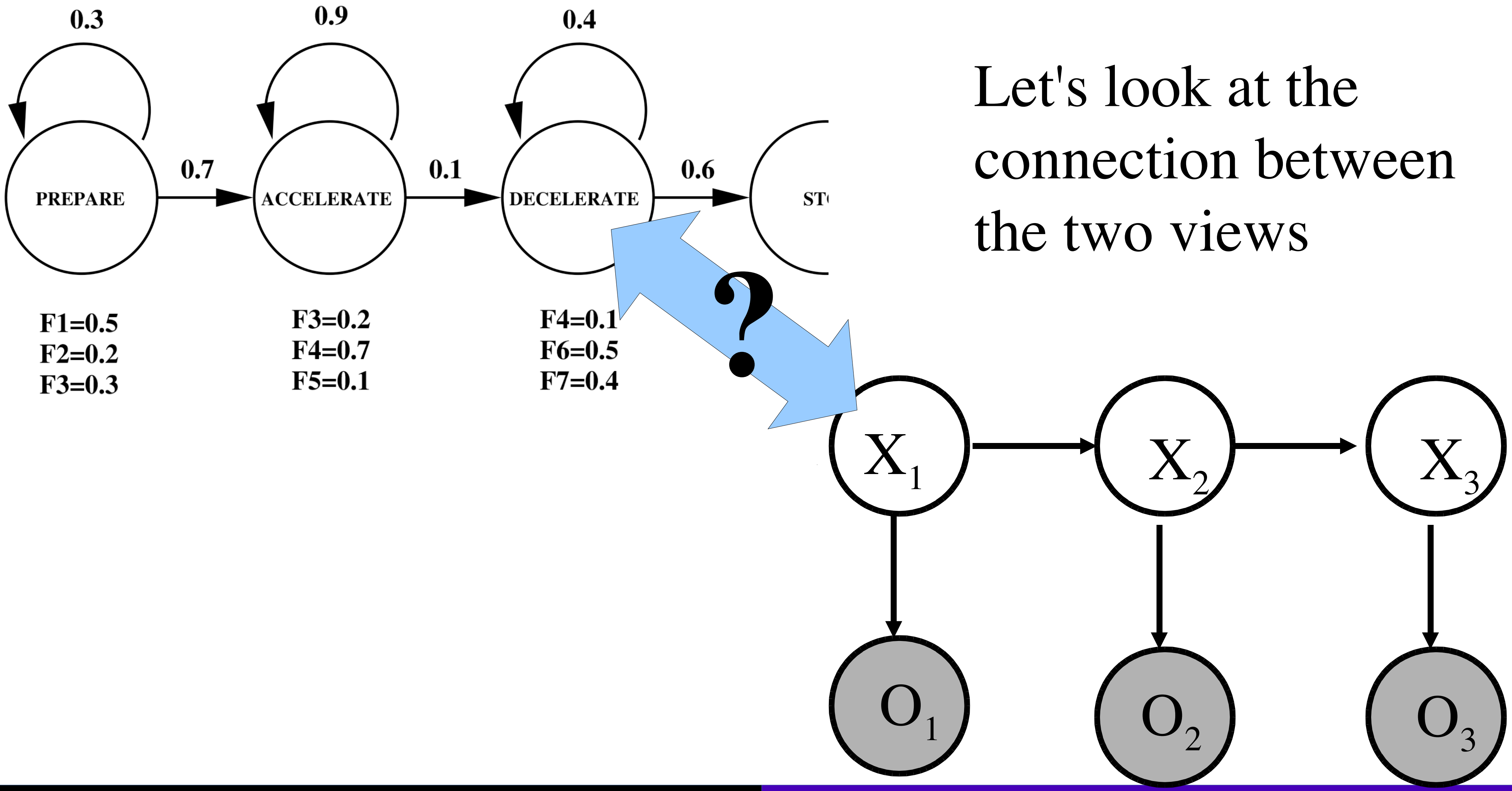
$$p\left(X_2=A \mid O_2=F_2\right)=\frac{p\left(O_2=F_2 \mid X_2=A\right)p\left(X_2=A\right)}{p\left(O_2=F_2\right)}$$

$$=\frac{0\cdot 0.7}{0.06}=0$$

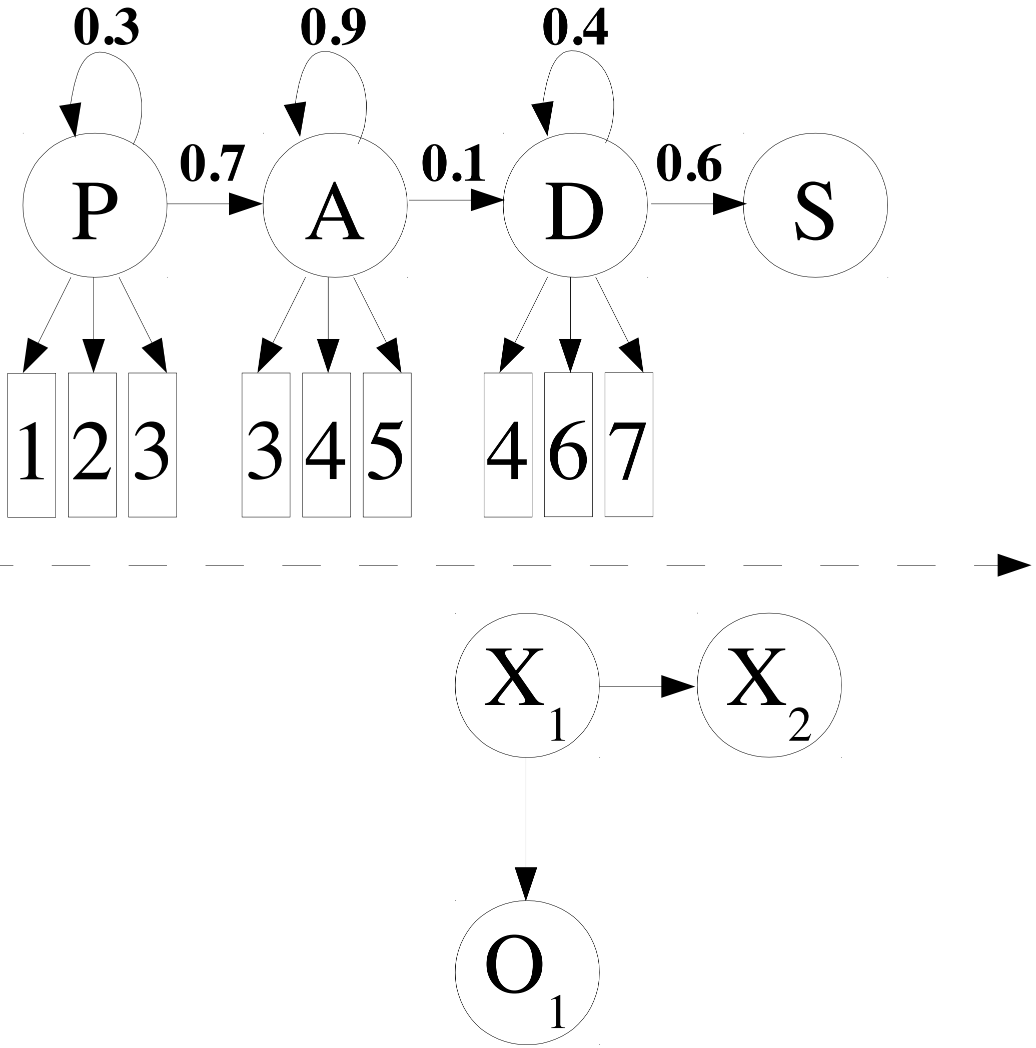
- Could we have seen this immediately?
  - YES!! Cannot measure F2 in Accelerate state, i.e., p=0

Ex: Usain Bolt

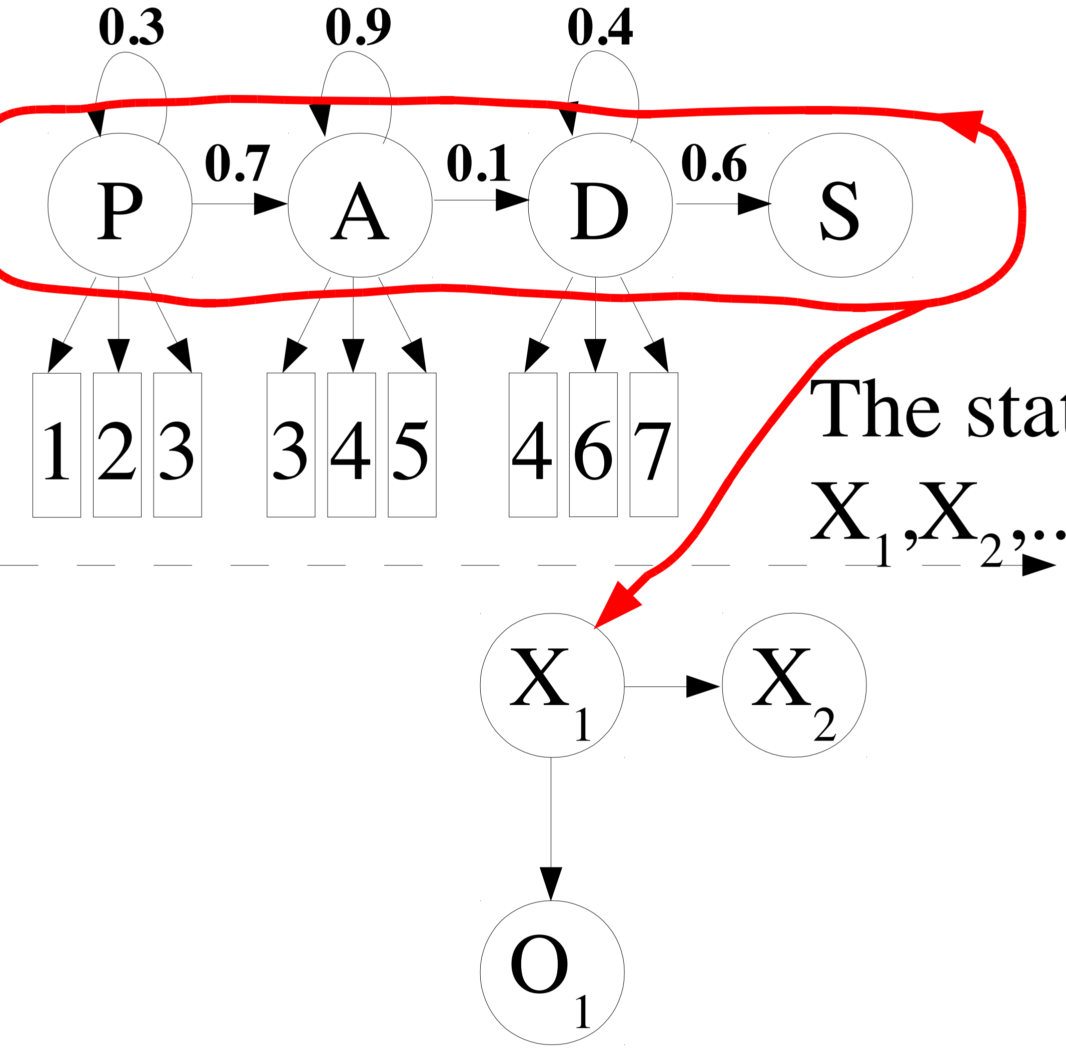
Let's look at the connection between the two views



Let's have a closer look!

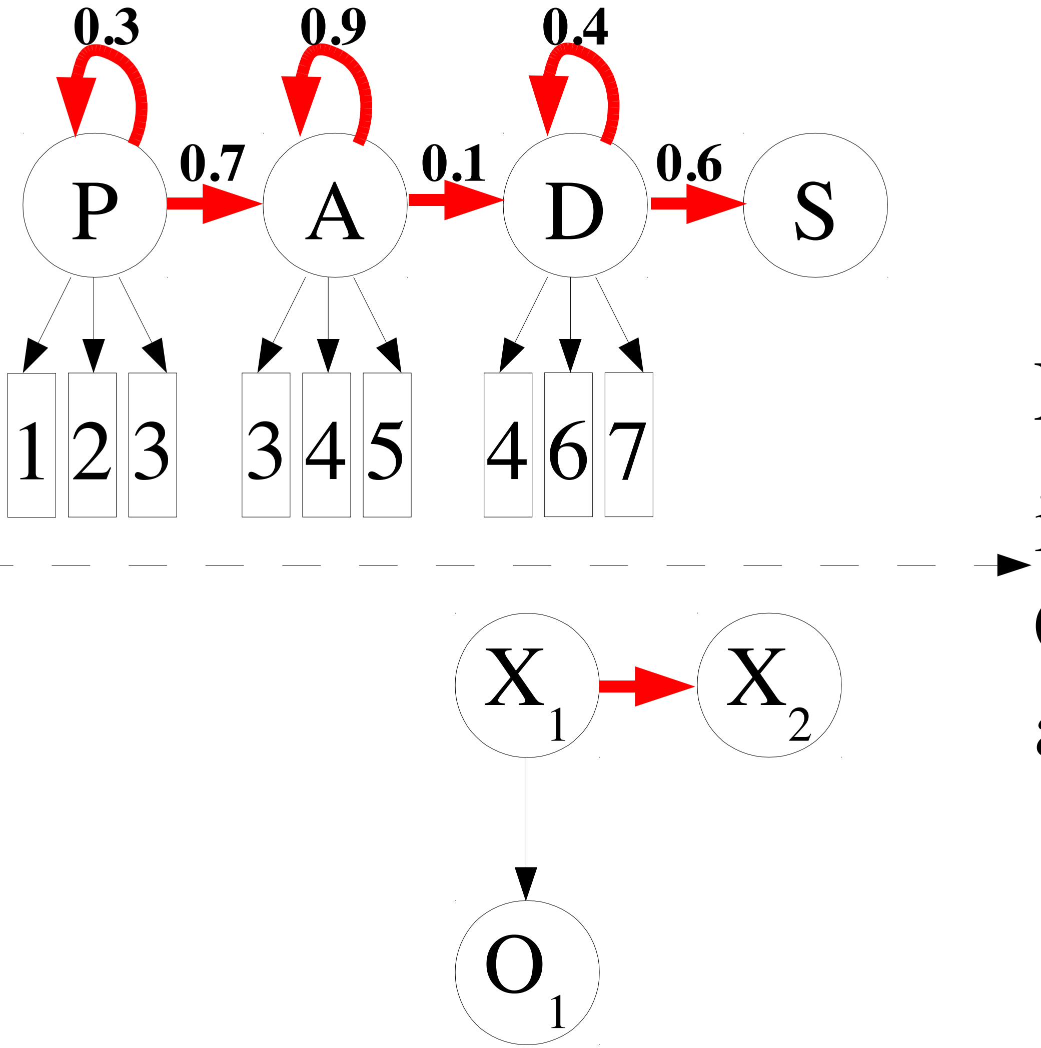


The states



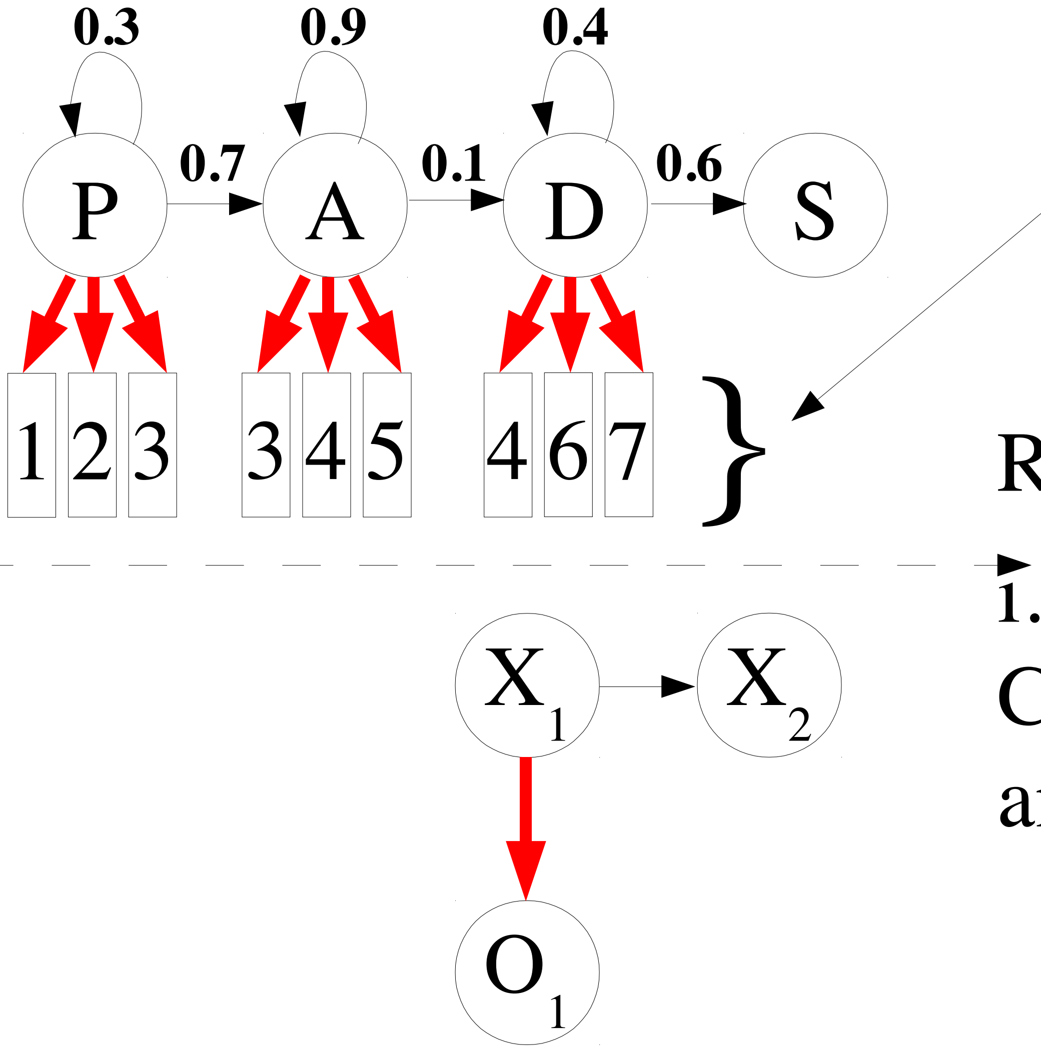
The states that variables X_1,X_2,... can be in (remember discrete)

The state transitions



Red lines describe $p(x_{k+1}|x_k)$ i.e., the state transitions. Captured with the A matrix (top) and $p(x_{k+1}|x_k)$ (bottom)

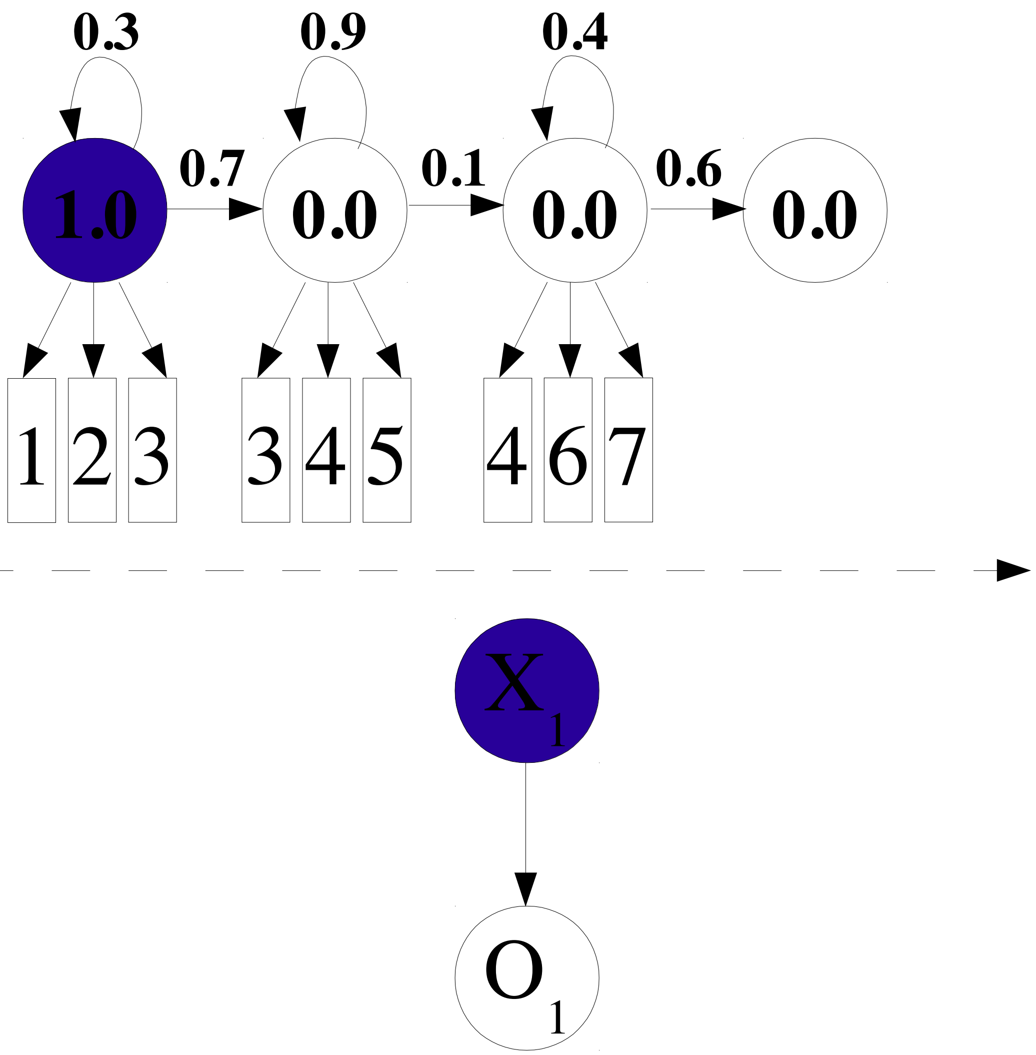
The observation/emission model



Different possible observations, (aka “symbols”)

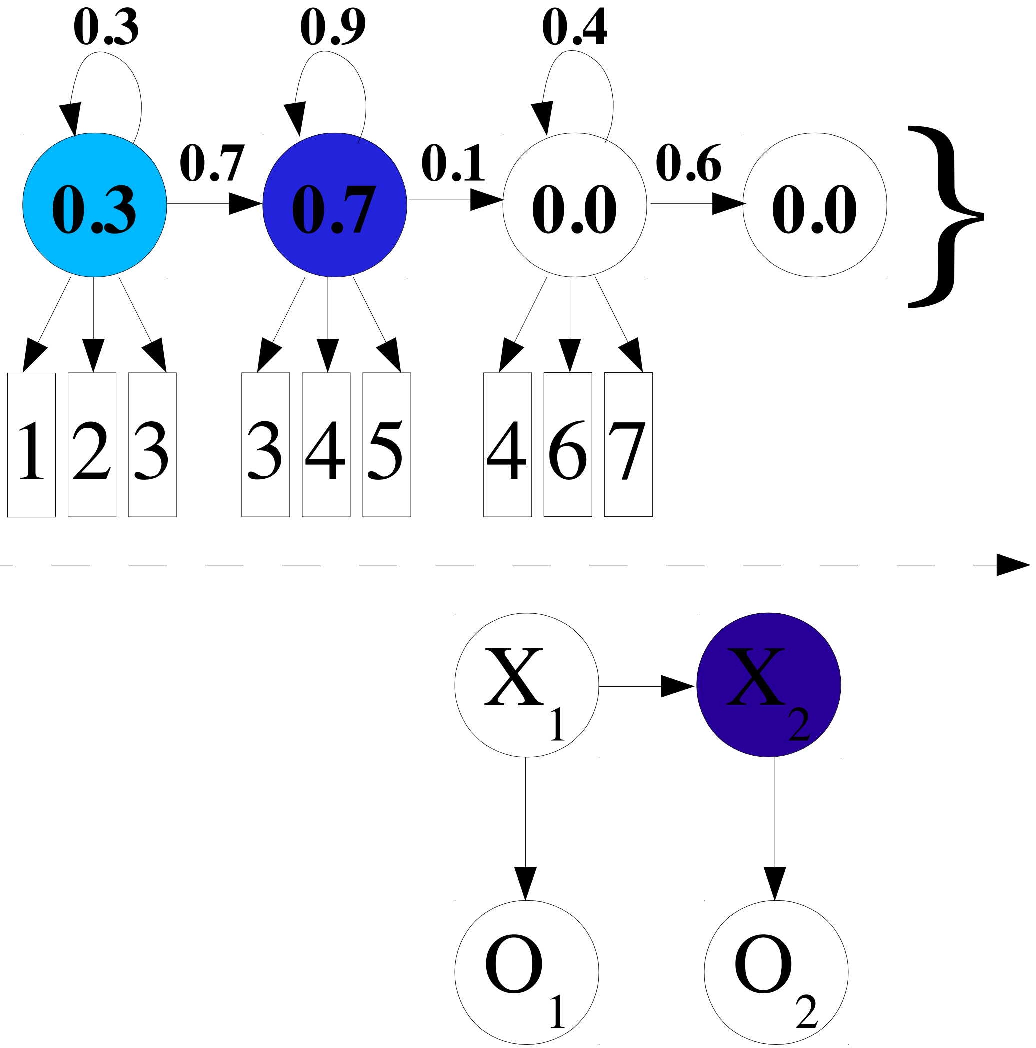
Red lines describe $p(O_k|X_k)$ i.e., observation model. Captured by matrix B (top) and $p(O_k|X_k)$ (bottom).

Initial state π=[1 0 0 0]



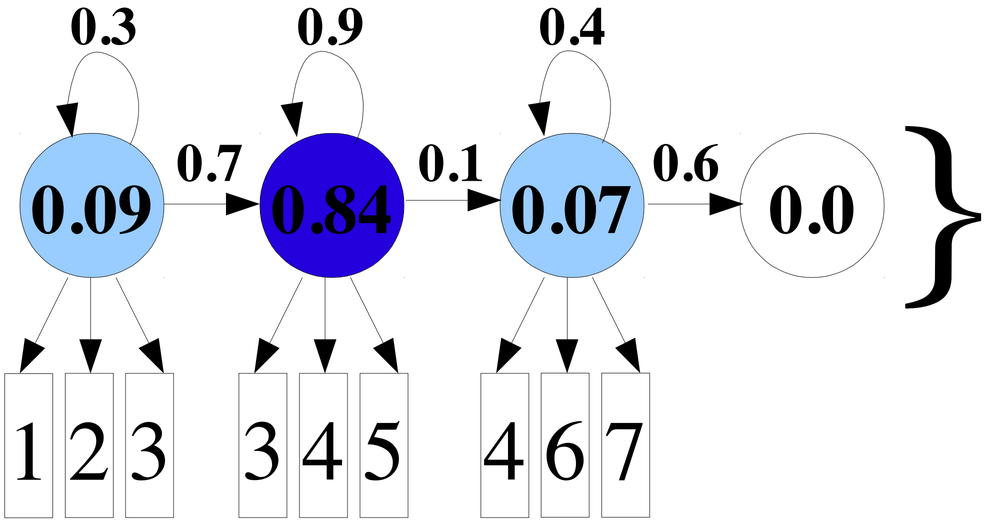
The darker the higher probability

Prediction for X_2

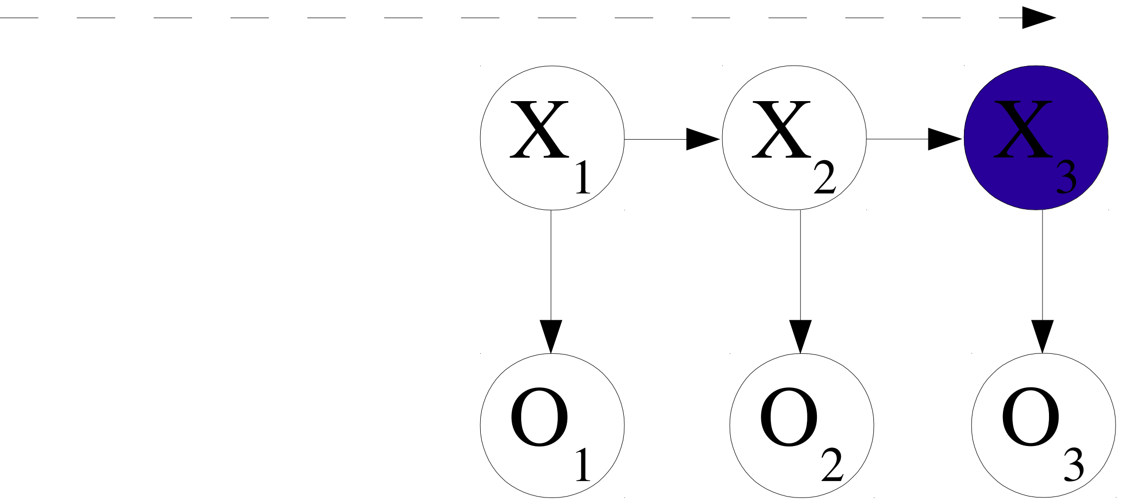


} Must sum to 1 i.e., the system is in some state!!

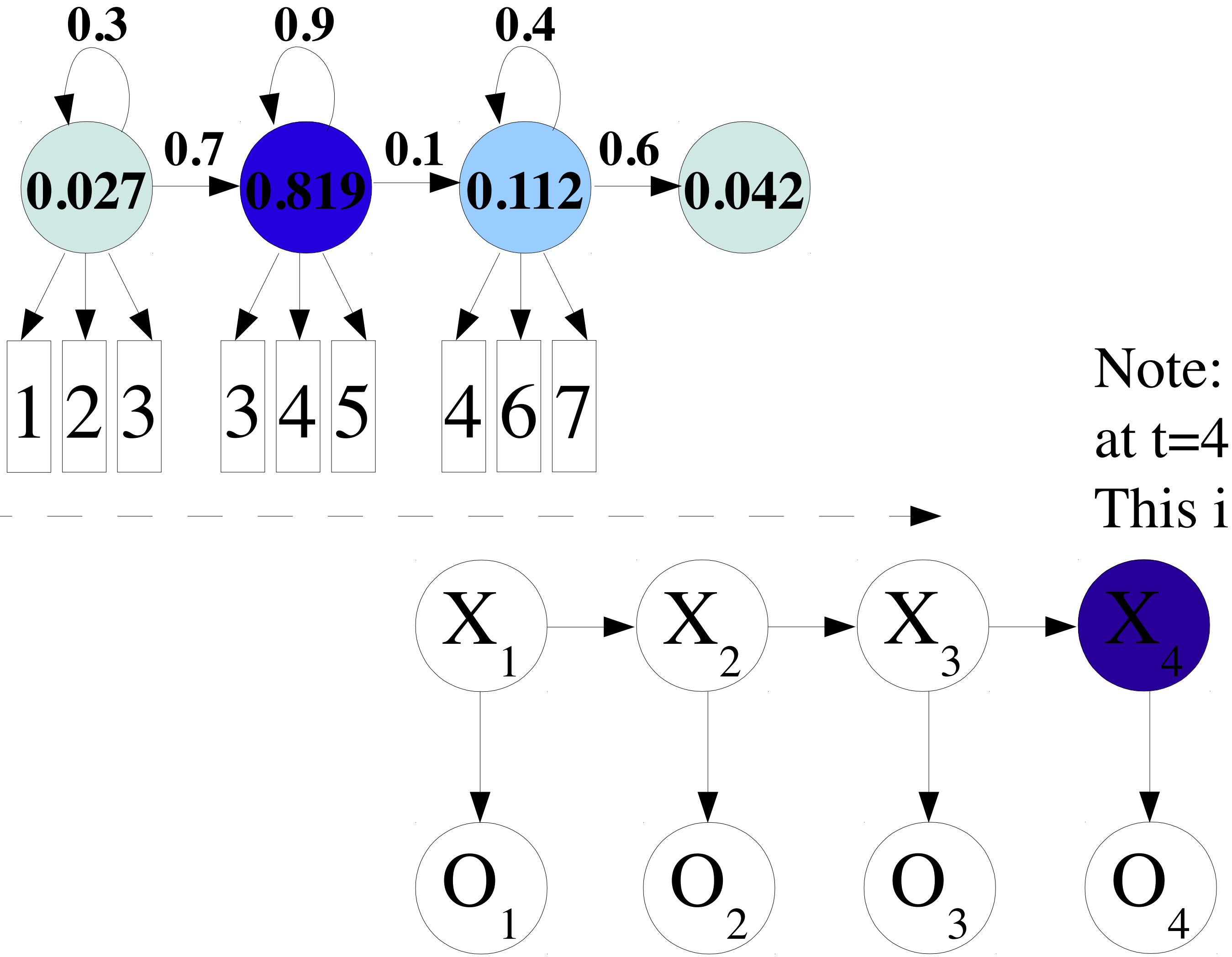
Prediction for X_3



Must sum to 1 i.e., the system is in some state!!



Prediction for X_4

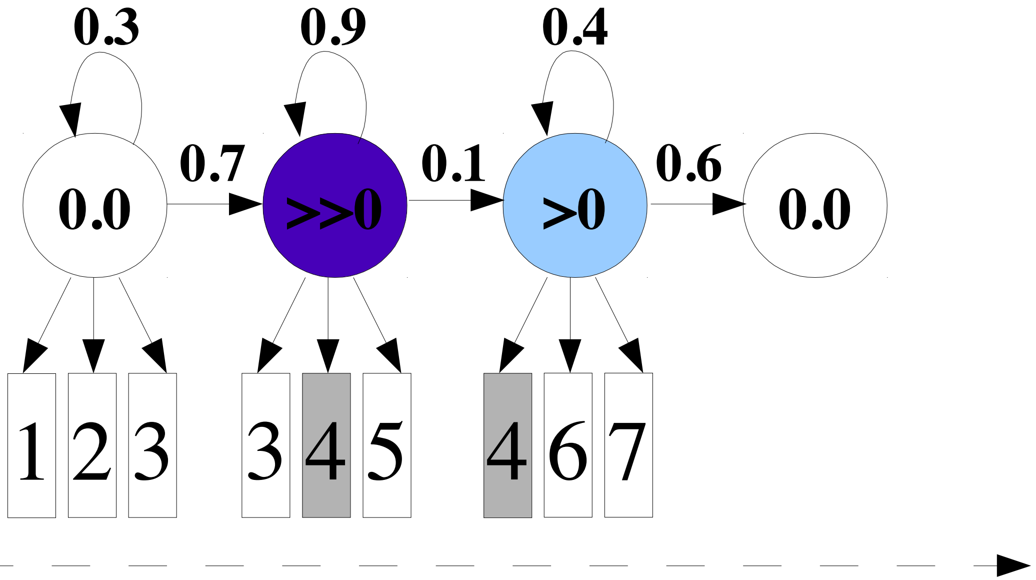


Note: We know for sure that we are at t=4 but not in which state we are. This is what we often try to estimate!

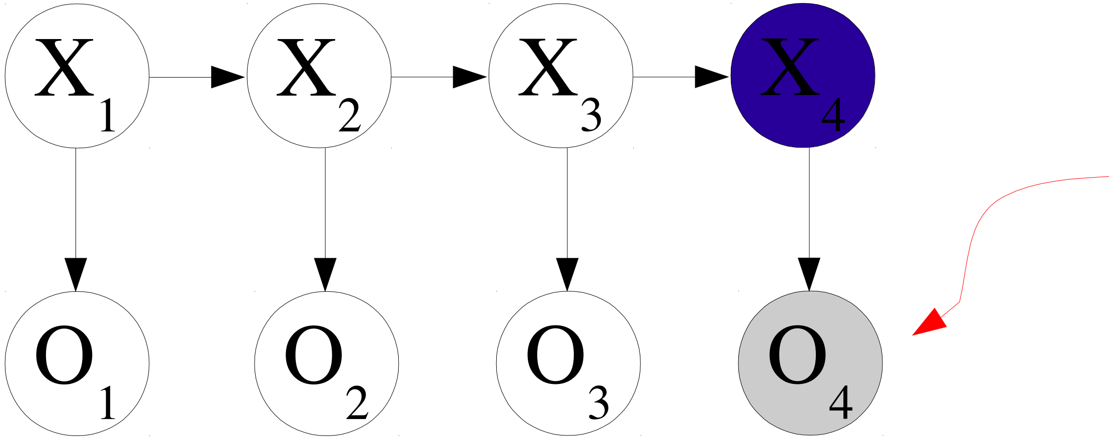
Measurements?

- Without measurements we are doing pure prediction
  - can do that ahead of time!
- Measurements gives clues to what state we are in

Assume we measure O_4=4 (i.e. obs 4 at t=4)



Weigh with p(O|X)
→ not in state 1 for sure because we cannot measure 4 there



Observed (grey)

How to calculate p(x$_4$|O$_4$)?

- We have p(X$_4$) from the prediction
- For each j we can calculate p(X$_4$=j|O$_4$)

$$
\begin{aligned}
p(X_4=j|O_4) &= \{Bayes\,rule\} \\
&= \frac{p(O_4|X_4=j)\,p(X_4=j)}{p(0_4)} \\
&= \frac{b_j(4)\,p(X_4=j)}{p(0_4)} \\
&= \eta\, b_j(4)\, p(X_4=j)
\end{aligned}
$$

- Calculate η by normalization so that the above terms sum to 1 for all j.

To think about

- Which state will the system most likely be in longest?
- What does lowering $p(Acc_k|Acc_{k-1})$ correspond to?

Three problems solved with HMMs

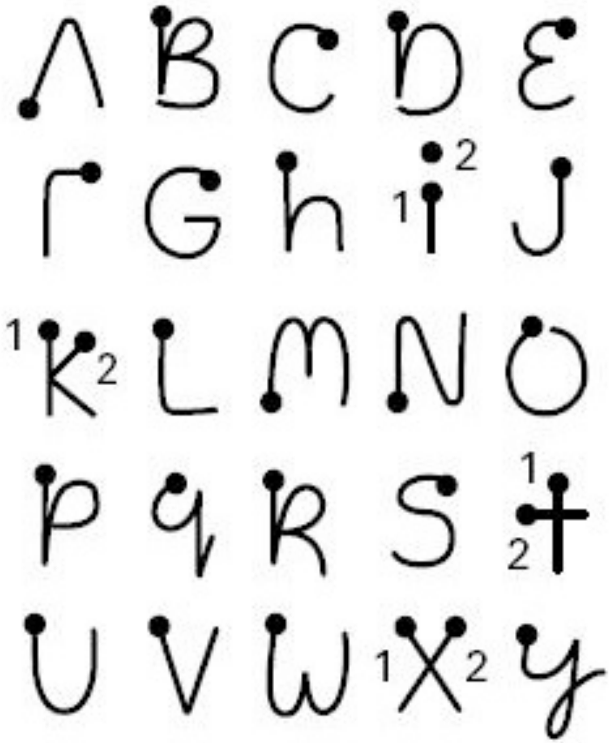
1. Estimate model parameters $\Lambda=\{\lambda\}$ given $O_{1:T}$ such that $p(O_{1:T}|\lambda)$ is maximized
   → A and B matrices and π

2. Compute likelihood $p(O_{1:t}|\lambda)$ of observation sequence $(O_{1:T}=\{O_1,O_2,\ldots,O_t,\ldots,O_T\})$ given λ

3. Find best fitting state sequence $X^*_{1:T}$ given $O_{1:T}$ and λ

Example: Character recognition

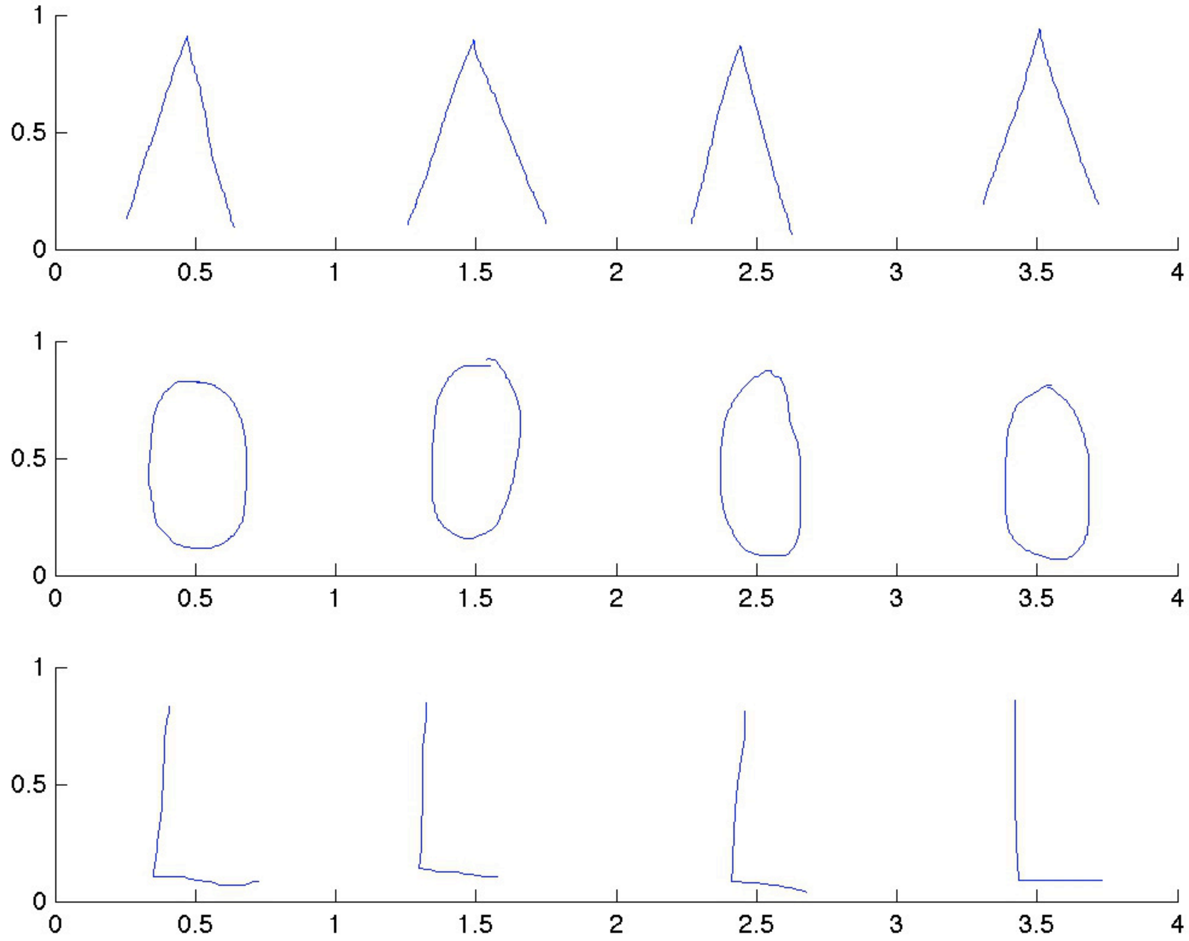
- First step of reading hand written text
- Graffitti language for Palm OS developed to standardize the characters and make them easier to separate.



With HMM

- Collect training data
- Train models for the characters
- Write new character
- Compare to all models and pick the one that fits the best

Training data

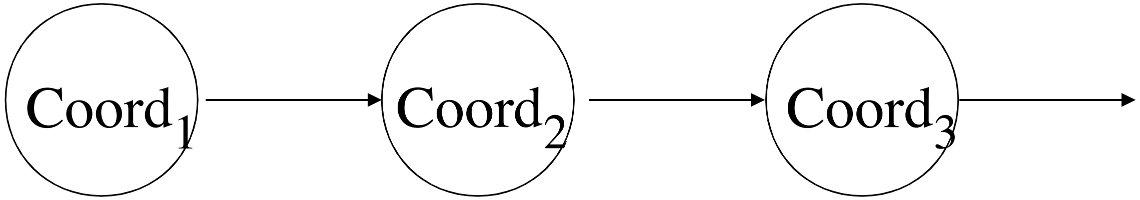


What can the computer “see”?

- The position of the black pixels
- Order in which they were drawn

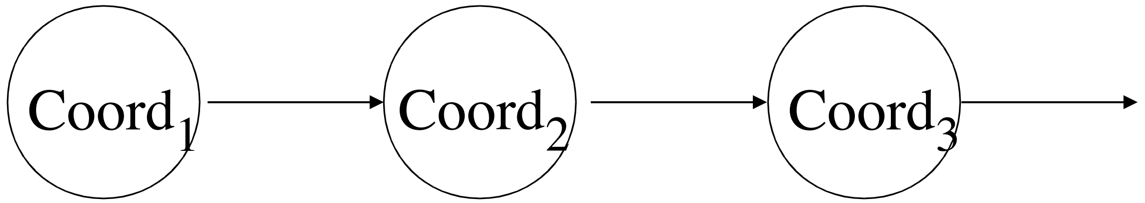
Bayesian network?

- We could draw it as



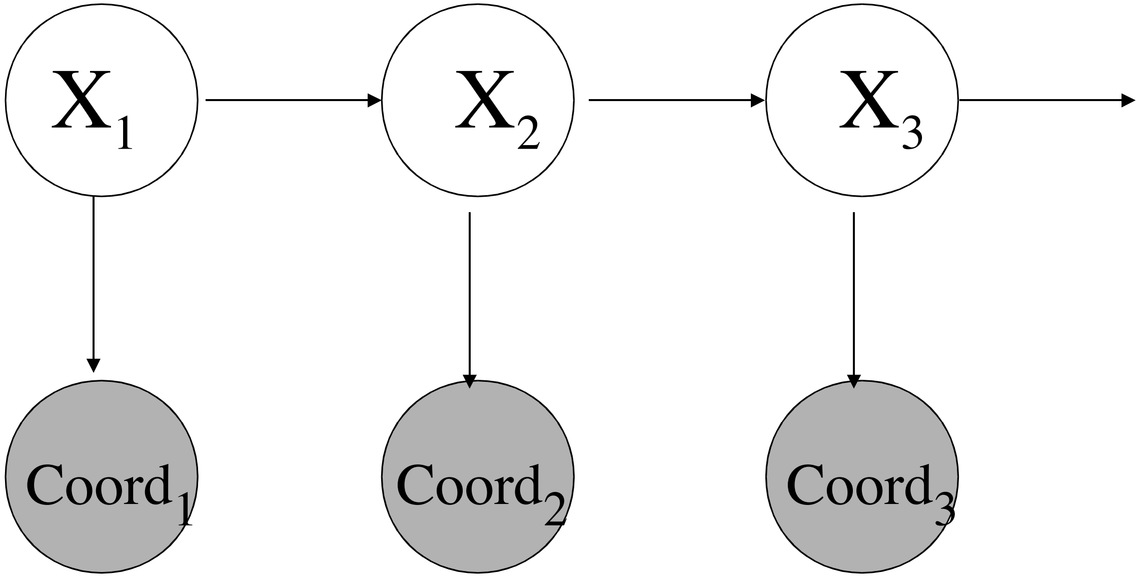
Bayesian network?

- We could draw it as
- BUT it would not be great for our task
  - Does not really capture the structure of the problem. Not just previous coordinate that tells what happens next
- What to do?



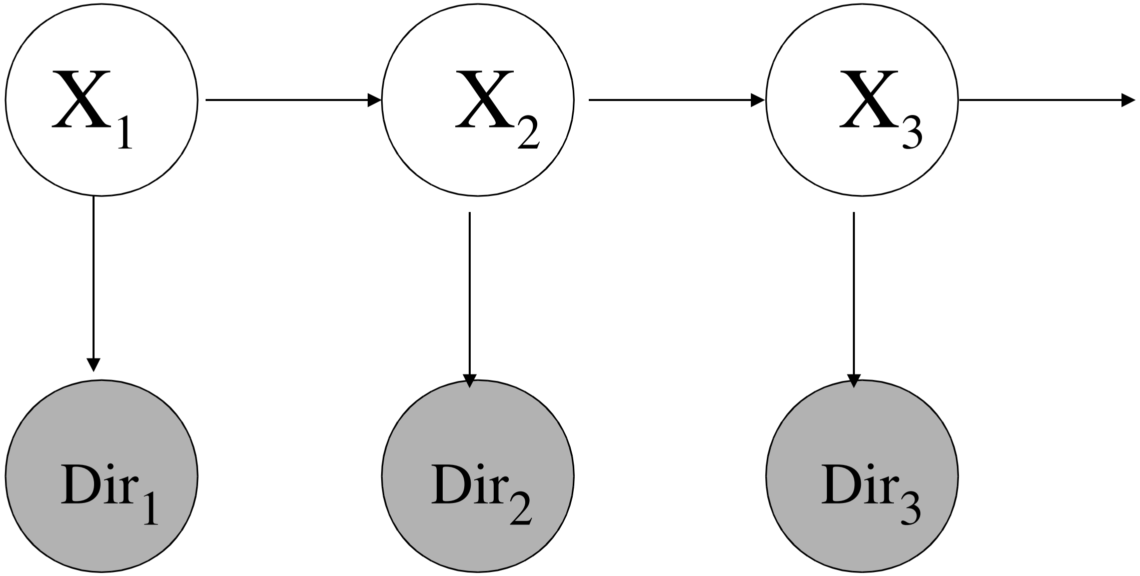
Bayesian network?

- Introduce **hidden** states X_k to capture this structure
- What is this state?
  - Not so clear, and we do not need to know. We will learn how it behaves from data.



Observations revisited

- Coordinates as observations → very large set of different observations. How to change?
- For example: Use direction of motion. Discretize into K direction. I used 8.



How many different states?

- Intuition does not help us so much here!
- We are trying to find a model that explains the data well.

  → Evaluate different values for N!

Matlab demo

- If there is enough time (and courage)

Three problems solved with HMMs cont'd

- Note:
  - I will display formulas for completeness but not derive them and not spend time on explaining most them
  - I will gloss over the details and target a conceptual understanding
  - Go back later for more details
    - HMM tutorials by Mark Stamp
    - Course book

1. Estimate model parameters $\Lambda=\{\lambda\}$ given $O_{1:T}$

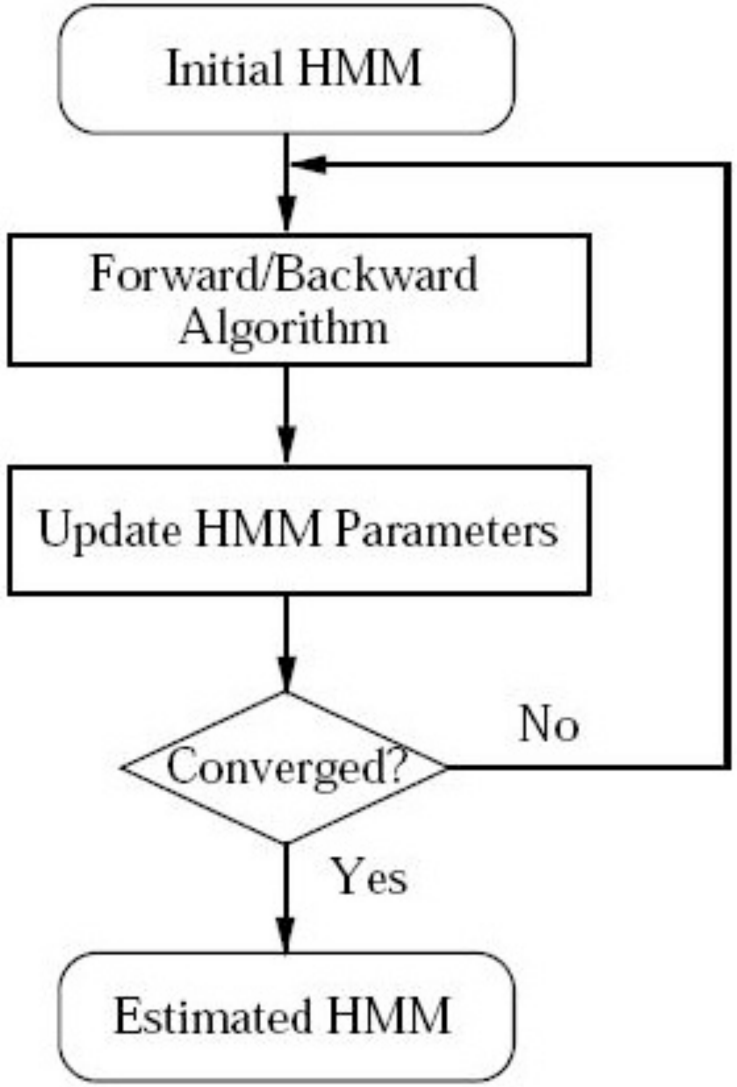
- Motivating examples:
  - Learn a model for a letter for recognizing hand written text
  - Learn a model for the word “Bayesian” from audio data
  - A1: Learn a model for the movement of a bird in the game duck hunt

The Baum-Welch Algorithm

- Given an observation sequence $O_{1:T}$, the number of states, N, and the number of observation outcomes, M.

1. Initialize $\lambda=(A,B,\pi)$
2. Compute $\alpha_t(i)$, $\beta_t(k)$, $\gamma_t(i,j)$ and $\gamma_t(i)$
3. Re-estimate the model $\lambda=(A,B,\pi)$
4. Repeat from 2 until $p(O|\lambda)$ converges

Model estimation with Baum-Welch



Forward algorithm (aka α-pass)

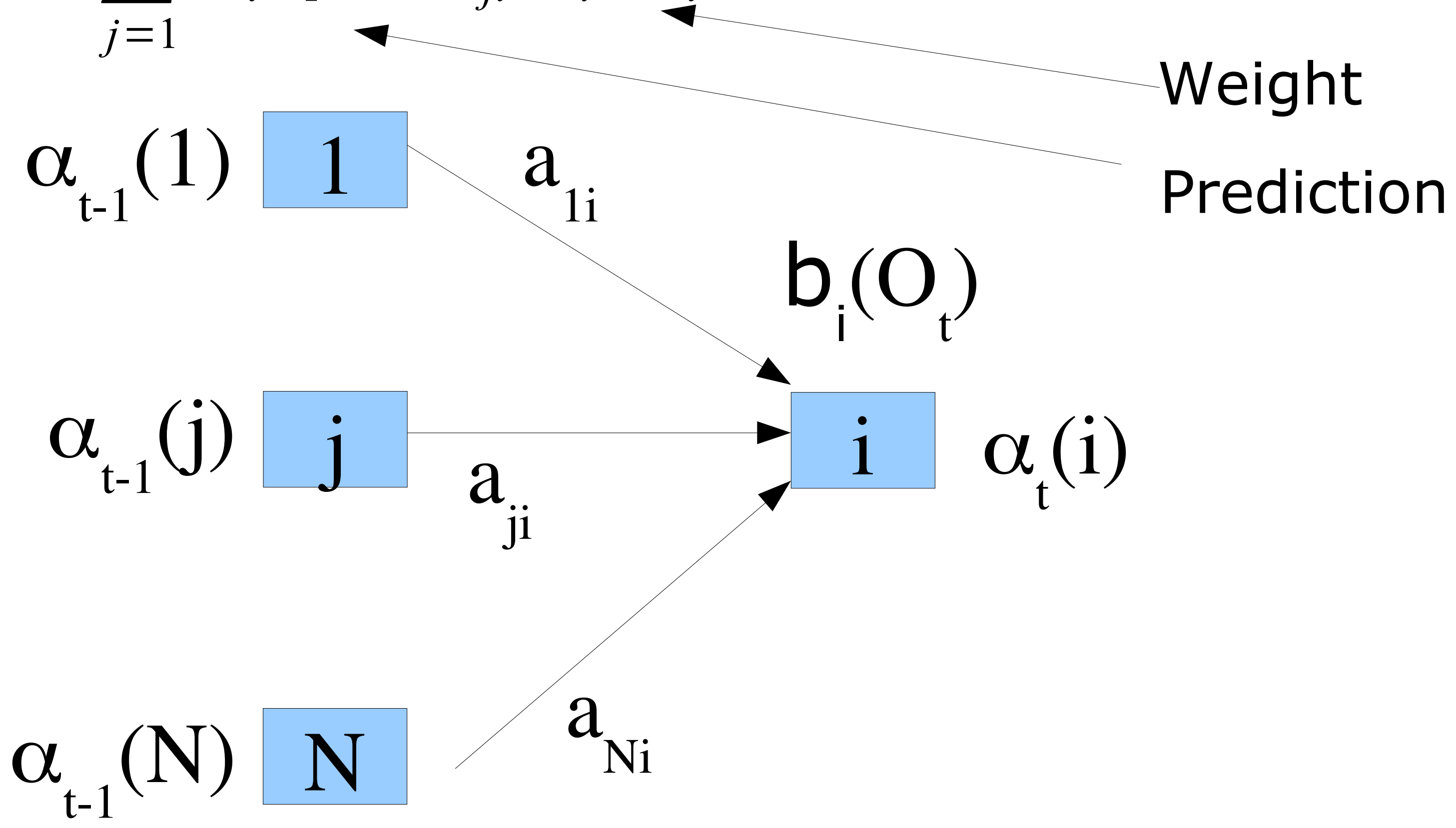
- Introduce: $\alpha_t(i) = p(O_{1:t}, X_t = i | \lambda) \quad \forall \quad t = 1, \ldots, T$
- Initialize as: $\alpha_1(i) = \pi_i b_i(O_1)$
- For 2≤t≤T: $\alpha_t(i) = [\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}] b_i(O_t)$
- Which gives us: $p(O_{1:T} | \lambda) = \sum_{i=1}^{N} p(O_{1:T}, X_t = i | \lambda) = \sum_{i=1}^{N} \alpha_T(i)$

→ a recursive way to calculate likelihood with only N²T multiplications (compare to 2TN^T)

One step in the forward algorithm

$$\alpha_t(i) = p(O_{1:t}, X_t = i | \lambda) \quad \forall \quad t = 1, \ldots, T$$

$$\alpha_t(i) = [\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}] b_i(O_t)$$



Backward algorithm (aka β-pass)

- Assume measurement sequence $O_{1:T}$
- Introduce: $\beta_t(i) = p(O_{t+1:T} | X_t = i, \lambda)$
- Initialize: $\beta_T(i) = 1, \forall\, i = 1, \ldots, N$
- For t<T: $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$

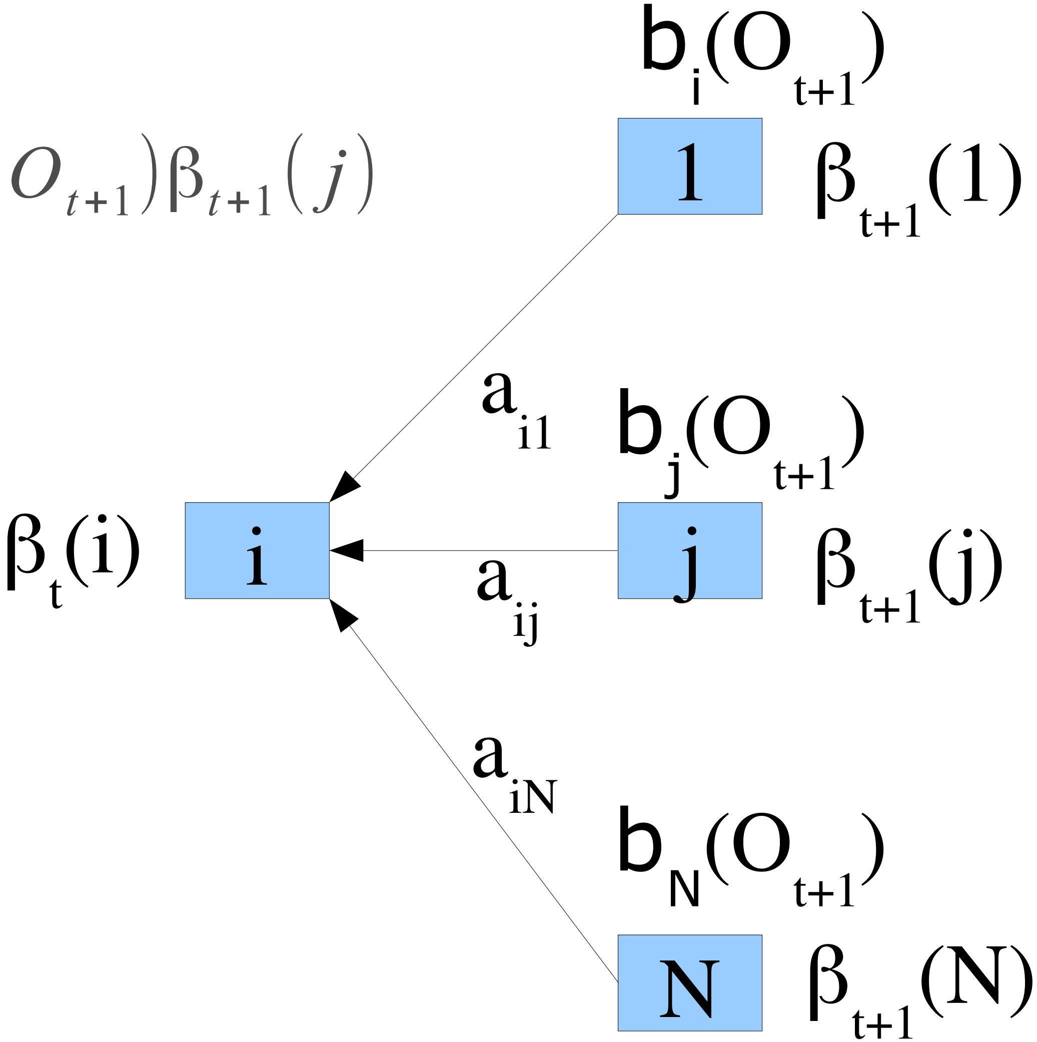
Backward algorithm (aka β-pass)

$$
\begin{aligned}
p(O_{1:T}|\lambda) &= \sum_{i=1}^{N} p(O_{1:T}, X_1=i) = \{\mathit{product\ rule}\} \\
&= \sum_{i=1}^{N} p(X_1=i)\, p(0_{1:T}|X_1=i) = \{O_{1:T} = 0_1 \quad , O_{2:T}\} \\
&= \sum_{i=1}^{N} p(X_1=i)\, p(0_{1,} O_{2:T}|X_1=i) = \{0_1\, \mathit{indep.\ of}\ O_{2:T}\} \\
&= \sum_{i=1}^{N} p(X_1=i)\, p(0_1|X_1=i)\, p(O_{2:T}|X_1=i) \\
&= \sum_{i=1}^{N} \pi_i b_i(O_1) \beta_1(i)
\end{aligned}
$$

- An alternative way to calculate $p(O_{1:T}|\lambda)$

One step in the backward algorithm

$$\beta_t(i)=\sum_{j=1}^{N} a_{ij} b_j(O_{t+1})\beta_{t+1}(j)$$



γ_t(i) and γ_t(i,j)

$$\gamma_t(i) = p(X_t = i | O_{1:T}, \lambda)$$

$$\gamma_t(i, j) = p(X_t = i, X_{t+1} = j | O_{1:T}, \lambda)$$

- Relation between the two (marginalize out X_{t+1})

$$\gamma_t(i) = \sum_{j=1}^{N} \gamma_t(i, j)$$

The function $\gamma_t(i)$

$$\gamma_t(i) = p(X_t = i | O_{1:T}, \lambda) = \{product\ rule\}$$

$$= \frac{p(X_t = i, O_{1:T} | \lambda)}{p(O_{1:T} | \lambda)} = \frac{p(X_t = i, O_{1:t}, O_{t+1:T} | \lambda)}{p(O_{1:T} | \lambda)}$$

$$= \{product\ rule\} = \frac{p(O_{1:t}, X_t = i | \lambda)\, p(O_{t+1:T} | X_t = i, O_{1:t}, \lambda)}{p(O_{1:T} | \lambda)}$$

$$= \{O_{t+1:T}\ independent\ of\ O_{1:t}\} = \frac{p(X_t = i, O_{1:t} | \lambda)\, p(O_{t+1:T} | X_t(i), \lambda)}{p(O_{1:T} | \lambda)}$$

$$= \frac{\alpha_t(i)\beta_t(i)}{p(O_{1:T} | \lambda)} = \boxed{\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_T(i)}}$$

- Expresses the probability of being in state i at time t given the measurement sequence $O_{1:T}$

The di-gamma function, $\gamma_t(i,j)$

$$
\begin{aligned}
\gamma_t(i,j) &= p(X_t=i, X_{t+1}=j | O_{1:T}, \lambda) = \{\mathit{product\ rule}\}\\
&= \frac{p(X_t=i, X_{t+1}=j, O_{1:T}|\lambda)}{p(O_{1:T}|\lambda)} = \frac{p(X_t=i, X_{t+1}=j, O_{1:t}, O_{t+1:T}|\lambda)}{p(O_{1:T}|\lambda)}\\
&= \{\mathit{product\ rule}\} = \frac{p(O_{1:t}, X_t=i, X_{t+1}=j|\lambda)\, p(O_{t+1:T}, X_t=i, X_{t+1}=j, O_{1:t}, \lambda)}{p(O_{1:T}|\lambda)}\\
&= \{0_{t+1:T}\ \mathit{cond.\ indep.\ of}\ O_{1:t}\} = \frac{\alpha_t(i)\, a_{ij}\, p(O_{t+1:T}|X_t=i, X_{t+1}=j, \lambda)}{p(O_{1:T}|\lambda)}\\
&= \frac{\alpha_t(i)\, a_{ij}\, p(O_{t+1}, O_{t+2:T}|X_t=i, X_{t+1}=j, \lambda)}{p(O_{1:T}|\lambda)} = \{\mathit{all}\ O\ \mathit{indep.}\}\\
&= \frac{\alpha_t(i)\, a_{ij}\, p(O_{t+1}|X_t=i, X_{t+1}, \lambda)\, p(O_{t+2:T}|X_t=i, X_{t+1}, \lambda)}{p(O_{1:T}|\lambda)}\\
&= \{O_{t+1}\ \mathit{indep.\ of}\ X_t \wedge O_{t+2:T}\ \mathit{cond.\ indep.\ of}\ X_t\ \mathit{given}\ X_{t+1}\}\\
&= \frac{\alpha_t(i)\, a_{ij}\, p(O_{t+1}|X_{t+1}, \lambda)\, p(O_{t+2:T}|X_{t+1}, \lambda)}{p(O_{1:T}|\lambda)} = \frac{\alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j)}{p(O_{1:T}|\lambda)}\\
&= \boxed{\frac{\alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_T(i)}}
\end{aligned}
$$

Learn the model, i.e. calculating A,B,π

- Given γ and di-gamma, estimate A,B,π using:

$$\pi_i = \gamma_1(i) \quad \forall \quad i = 1, \ldots, N$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \forall \quad i, j = 1, \ldots, N$$

$$b_j(k) = \frac{\sum_{\substack{t \in 1,2,\ldots T-1 \\ O_t = k}} \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \forall \quad i = 1, \ldots, N, \quad j = 1, \ldots, M$$

Interpretation of model estimation

- a_{ij} is calculated as the ratio of all state transitions from i to j and the total number of times we are in state i.
- $b_j(k)$ is calculated as the ratio of the number of times observation k is given in state i and the total number of times we are in state i.

Model initialization

- Need to make sure that A,B,π are all row stochastic (rows sum to 1)
- Use whatever knowledge you have to provide good initial guesses
- If you have no clue, assign the values randomly as

$$a_{ij} \approx 1/N$$

$$b_j(k) \approx 1/M$$

$$\pi_i \approx 1/N$$

Important considerations

- Stop if too many iterations to avoid deadlocks
- Make sure that A,B and π are **not** uniform, ie that the values are not exactly 1/N and 1/M resp.
  - Otherwise we are in a local maximum that we cannot get out of and the method will not converge
  - If B is uniform, a measurement gives no info!
- Calculating long products of probabilities
  → very small numbers
  → underflow problems
  → use scaling and log likelihoods

Important considerations

- How much data is needed?
  - Remember that we are estimating A and B by calculating statistics for how frequent transitions/emissions are.
    - Little data → bad statistics → bad model
  - No general rule, it depends on the problem.
    - More parameters → more data

2. Compute likelihood $p(O_{1:t}|\lambda)$ of observation sequence given λ

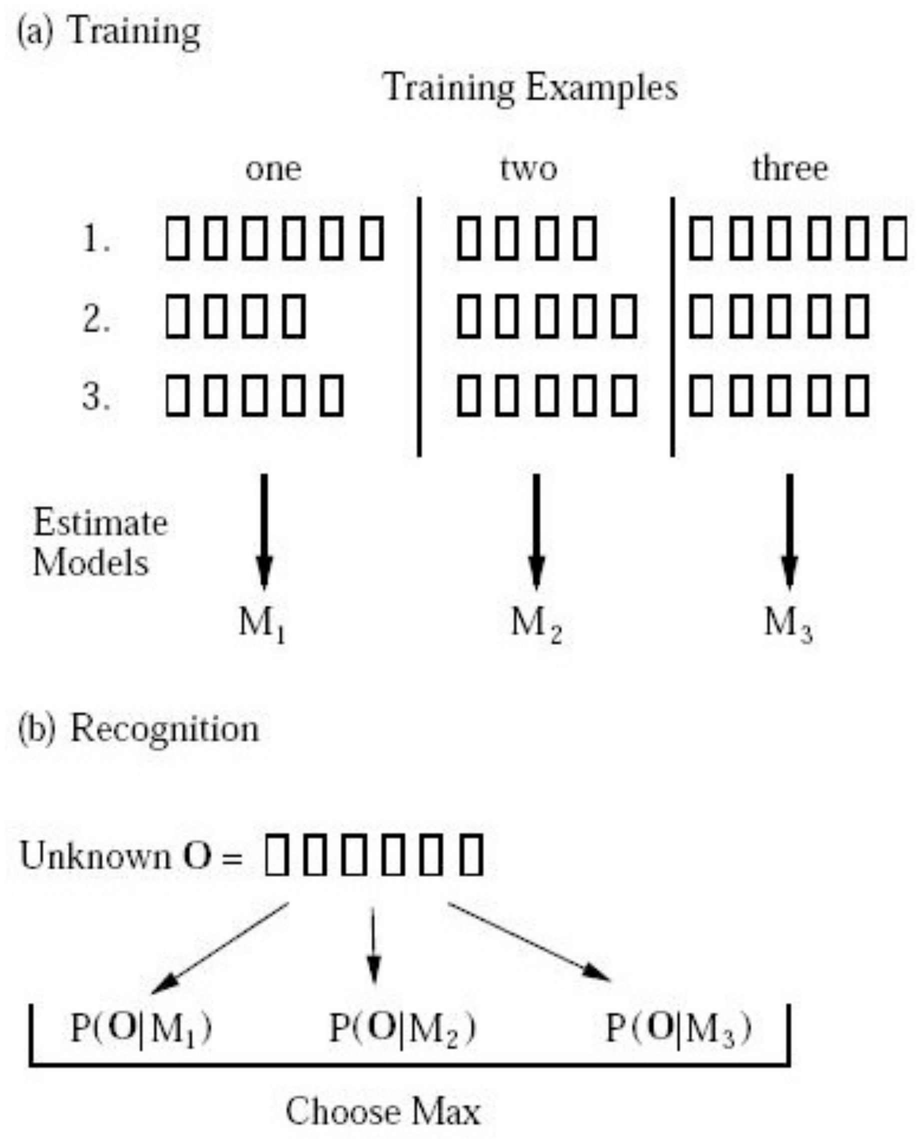
- Motivating examples
  - Character recognition:
    - Have models for characters in an app
    - Draw a character and **compare it to models of different characters**
    - Pick model that fits best → Recognized char
  - A1: Identify birds species
    - Model movement of known birds, **compare new birds to the models**

Likelihood of $p(O_{1:T}|\lambda)$

$$\begin{aligned}
p(O_{1:T}|\lambda) &= \{\textit{summation rule}\} = \sum_{X_{1:T}} p(O_{1:T}, X_{1:T}|\lambda) \\
&= \{\textit{product rule}\} = \sum_{X_{1:T}} p(O_{1:T}|X_{1:T}, \lambda)\, p(X_{1:T}|\lambda) \\
&= \sum_{X_{1:T}} \underbrace{\pi_{X_1} a_{X_1 X_2} a_{X_2 X_3} \ldots a_{X_{T-1} X_T}}_{p(X_{1:T}|\lambda)} \underbrace{b_{X_1}(O_1)\, b_{X_2}(O_3) \ldots b_{X_T}(O_T)}_{p(O_{1:T}|X_{1:T}, \lambda)}
\end{aligned}$$

- Note that we are summing over all possible permutations of $X_{1:T}$
- Evaluating this requires $2TN^T$ multiplications
- Can be formulated recursively using the forward (and backward) algorithm

Training and testing procedure



3. Calc most likely state sequence

- Cannot solve individually for each time step
  - Could come up with states at t and t+1 for which $p(X_{t+1}|X_t)=0$
- Need to optimize the sequence not just individual states

$$X^*_{1:T} = \underset{X_{1:T}}{argmax}\, p(X_{1:T}|O_{1:T}, \lambda)$$

Most likely sequence (Viterbi alg.)

- Solved using dynamic programming (DP) with an algorithm called Viterbi.
    - Initialize: $\delta_1(i)=\pi_i b_i(O_1),\;\; i=1,\ldots,N$
    - For each t>1: $\delta_t(i)=\max_{j\in\{1,\ldots,N\}}[\delta_{t-1}(j)a_{ji}b_i(O_t)]$
    - Probability of best path: $\max_{j\in\{1,\ldots,N\}}[\delta_{T-1}(j)]$
    - Find path by keeping book of preceding states and trace back from highest-scoring final state.

Viterbi cont'd

- Watch out for underflows!!! Multiplying many small numbers (probabilities)

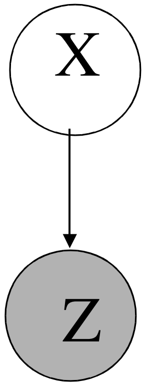
- Better to use
  - Initialize: $\delta_0(i) = \log[\pi_i b_i(O_0)], \quad i = 1, \ldots, N$
  - For t>1: $\hat{\delta}_t(i) = \max\limits_{j \in \{1, \ldots, N\}} [\hat{\delta}_{t-1}(j) + \log[a_{ji}] + \log[b_i(O_t)]]$
  - Probability of best path: $\max\limits_{j \in \{1, \ldots, N\}} [\hat{\delta}_{T-1}(j)]$

Ex: Positioning

- We want to find the position of something
- Our variable is the position X
- The position is not directly observable → X is hidden to us

Ex: Positioning

- Can measure something, Z, that depends on the positions
- Graphical model?



Bayesian network

Ex: Positioning

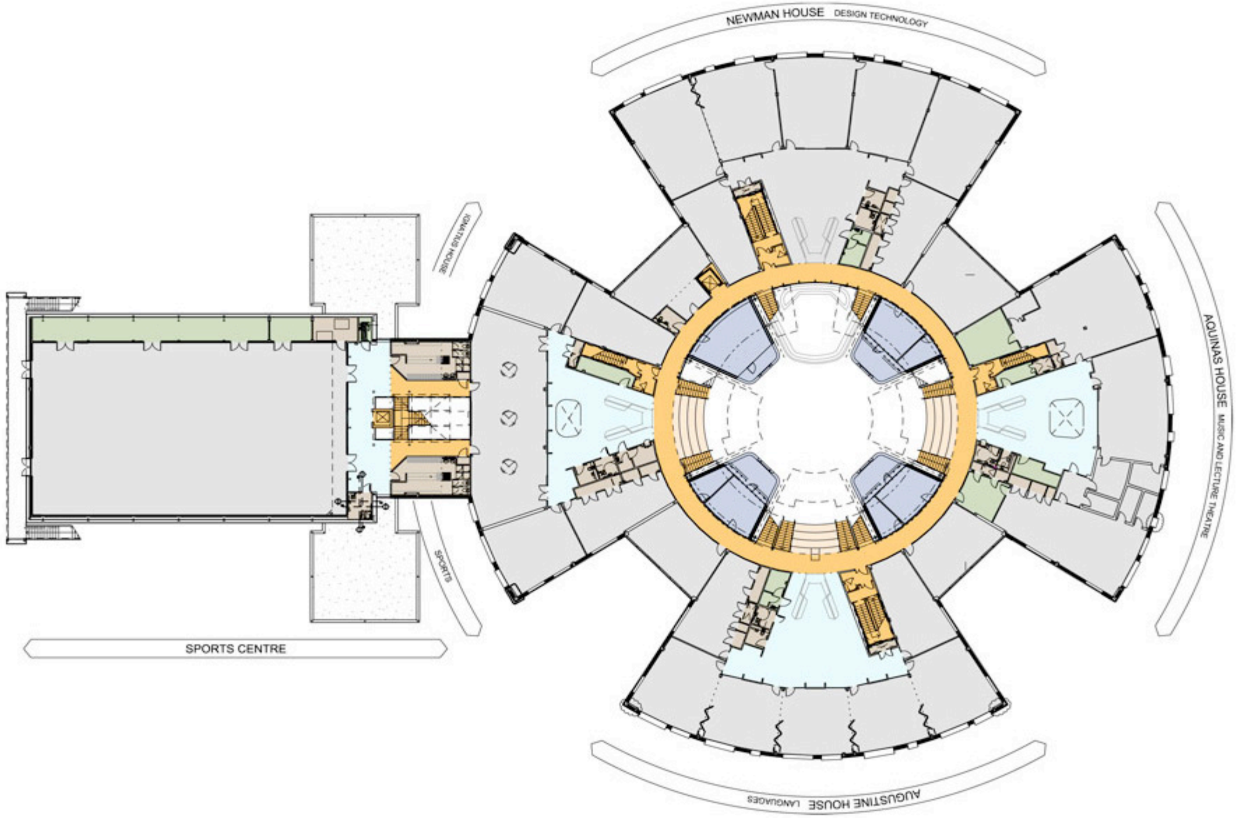
- What to measure?
- GPS: Is not the GPS making the position observable?
  - No! There are errors in the measurement.
- Camera: What we see depends on where we are.
- We learn about X, since Z depends on X!

Ex: Positioning

- What if we want to use more measurements?
- Get a time series!
- Assume: Positions form first order Markov chain (does not say anything about the motion, just that the next position only depends on the curren and not previous positions).

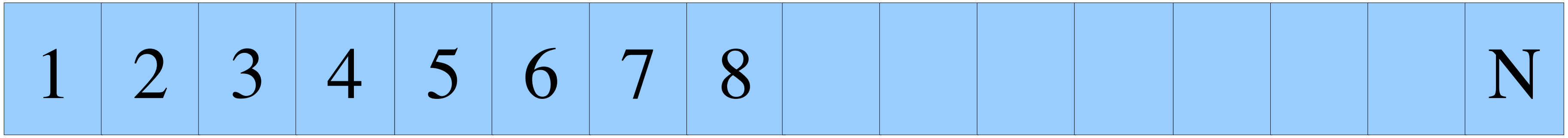
Ex: Robot localization

- Want to localize in the circular corridor


NEWMAN HOUSE DESIGN TECHNOLOGY
IGNATIUS HOUSE
AQUINAS HOUSE MUSIC AND LECTURE THEATRE
SPORTS
SPORTS CENTRE
AUGUSTINE HOUSE LANGUAGES

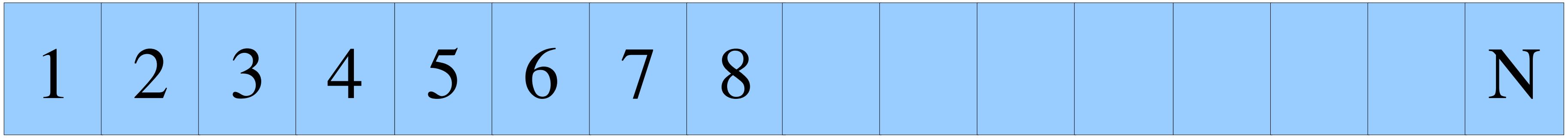
Robot localization: Modeling

- We divide the corridor into N segments
  - Segments 1 and N are connected.
  - The state of the robot corresponds to being in a certain segment.
- Let us say we start in segment 1.
  - What is the initial state distribution?
- We command the robot to move to the right.
  - State transition model?



Robot localization

- Let us say that there are some doors in the corridor that we can observe.
- Observation model?



Robot localization

- Let us say that there are some doors in the corridor that we can observe.
- Observation model?

  p(door | position)

  Depends on if there is a door at the **position** (i.e. state) and how good our detector is

1	2	3	4	5	6	7	8								N

Robot localization

- Now we can localize the robot even if we do not know where it started by looking at the sequence of observations.
- One measurement does not tell us where we are but combined they can (with high probability).

1	2	3	4	5	6	7	8								N

Robot localization

- What happens if all doors look the same and if there is a door in every segment?
  - All states have the same observation model! We do not get any information about what state we are in!! (assuming that we can only see the doors in the segment we are in).

1	2	3	4	5	6	7	8								N

Time for A1!

Let's play duck hunt!

A1

- Work in pairs
- Practice using HMMs
- Have to implement your HMM
  - Using an existing implementation not allowed

Motivation A1

- Solve an actual task, i.e., use the AI methods in a context
- Covers
  - Probabilistic reasoning
  - Machine learning
  - Decision making
  - Implementation
  - Testing and evaluation

Generate observations using Matlab

- Matlab: `[O,X]=hmmgenerate(T,A,B)`

- NOTE: Model always starts in state 1 with the Matlab HMM functions.
  - Can be handled by introducing a new dummy state 1 with transitions to the other states corresponding to π, i.e.,

$$A' = \begin{bmatrix} 0 & \pi \\ 0 & A \end{bmatrix} \qquad B' = \begin{bmatrix} 0 & 0 \\ 0 & B \end{bmatrix}$$

Baum-Welch in Matlab

- [A,B] = hmmtrain(O,A_{guess},B_{guess})

Viterbi in Matlab

- X_t={true,false}={1,2}
  O_t={true,false}={1,2}

```
A=[0.7 0.3;0.3 0.7];
B=[0.9 0.1;0.2 0.8];
O=[1 1 2 1 1];
hmmviterbi(O,A,B)
```

→ ans = 1   1   2   1   1

→ true,true,false,true,true