Hidden Markov Models

NIKOLAY YAKOVETS

A Markov System

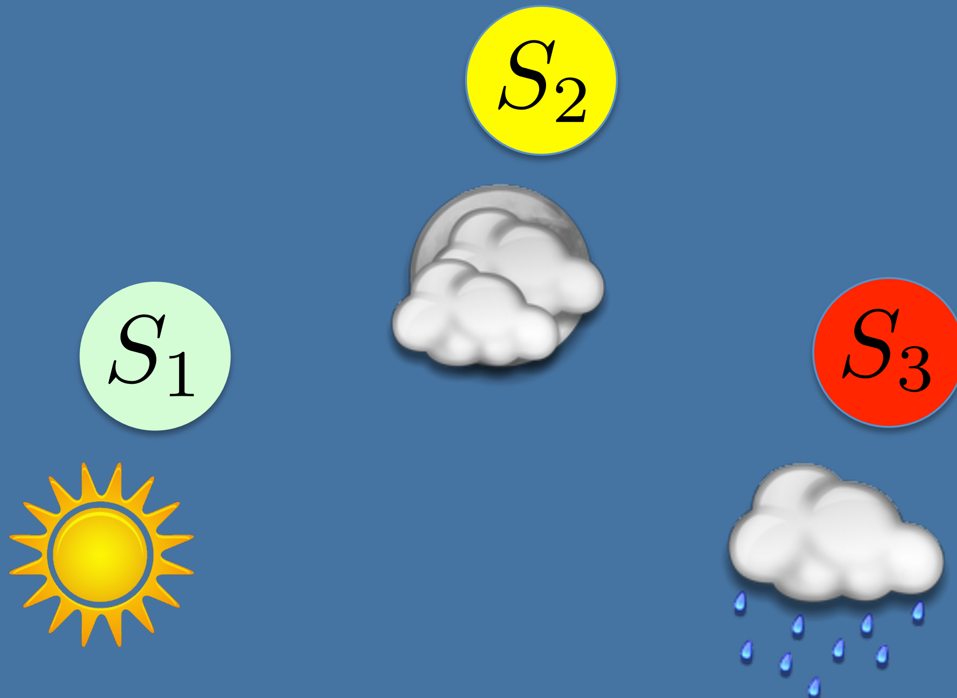
N **states**

$$s_1, .., s_N$$

A Markov System

N **states**

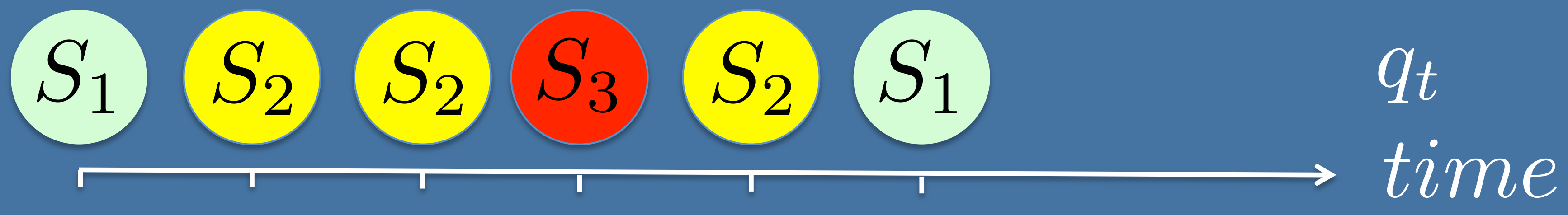
$s_1, .., s_N$



modeling weather

A Markov System

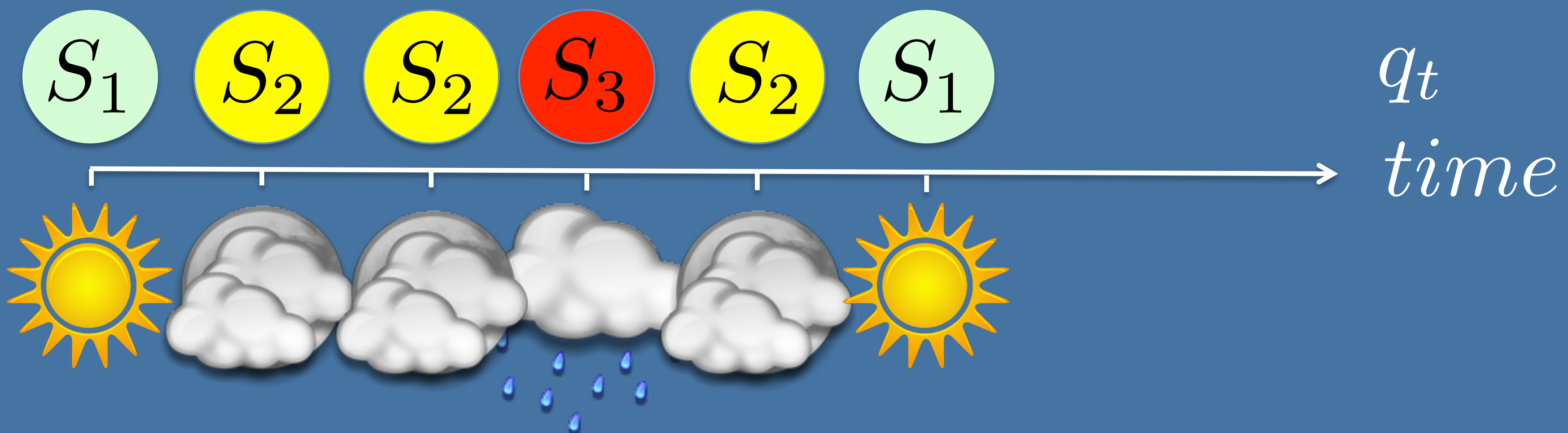
state changes over time..



$$q_t \in \{s_1, \ldots, s_N\}$$

A Markov System

state changes over time..

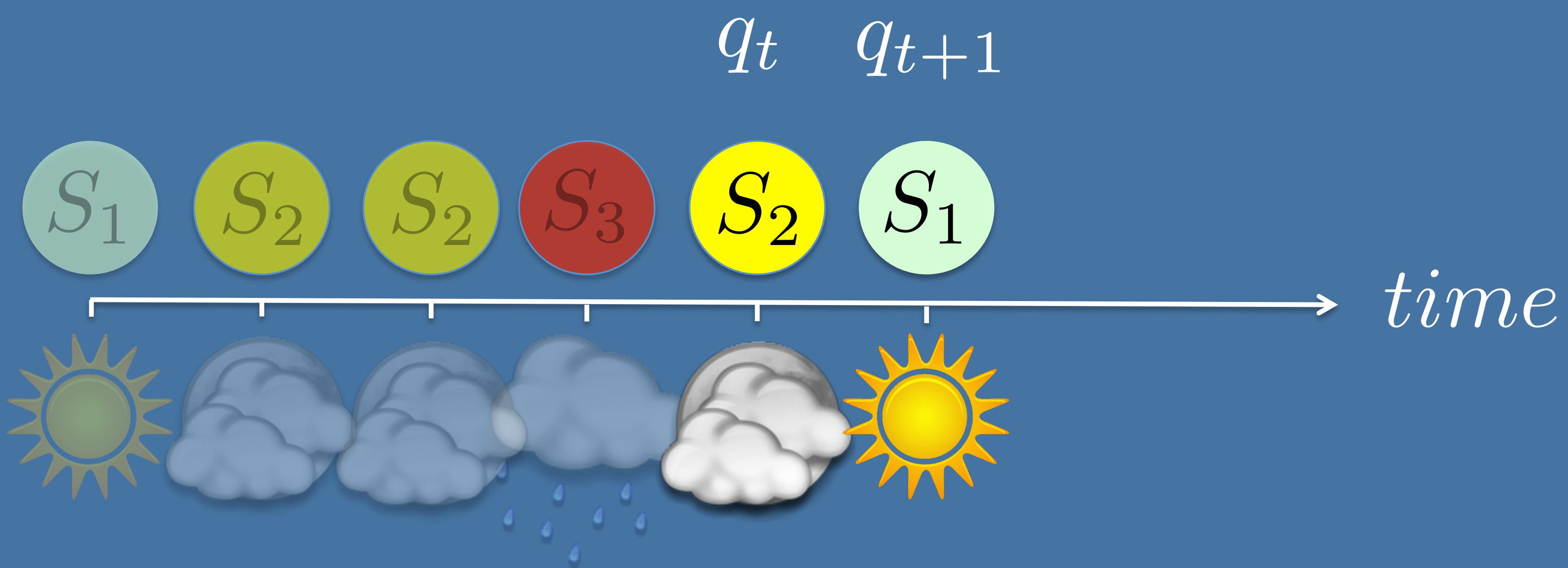


$$q_t \in \{s_1, \ldots, s_N\}$$

modeling weather

A Markov Property

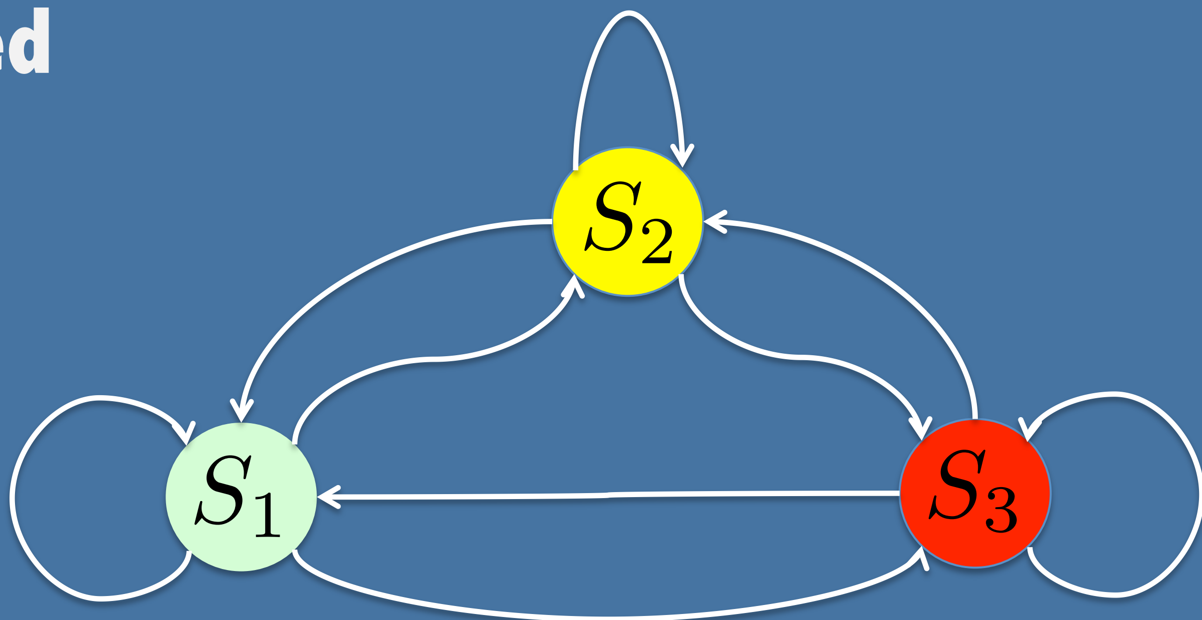
system is memory less..



$$P(q_{t+1} = S_j | q_t = S_i) = P(q_{t+1} = S_j | q_t = S_i, \text{any earlier history})$$

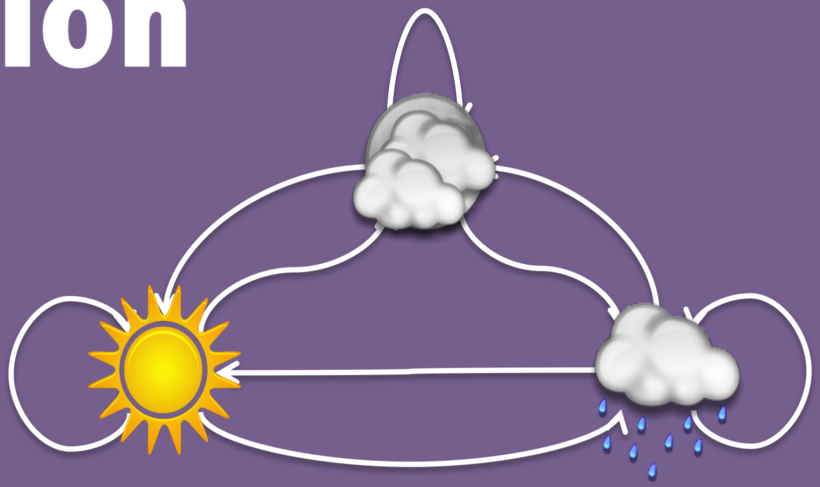
A Markov System

Directed Graph



$$P(q_{t+1} = S_j | q_t = S_i)$$

Weather Prediction



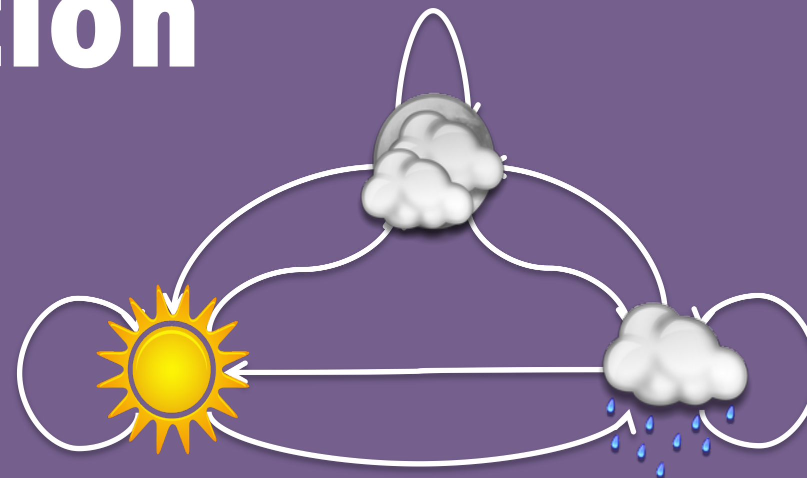
Initial P

0.5	0.2	0.1

Transitional P

	0	0	1
	0.5	0.5	0
	0.3	0.7	0

Weather Prediction



Initial P

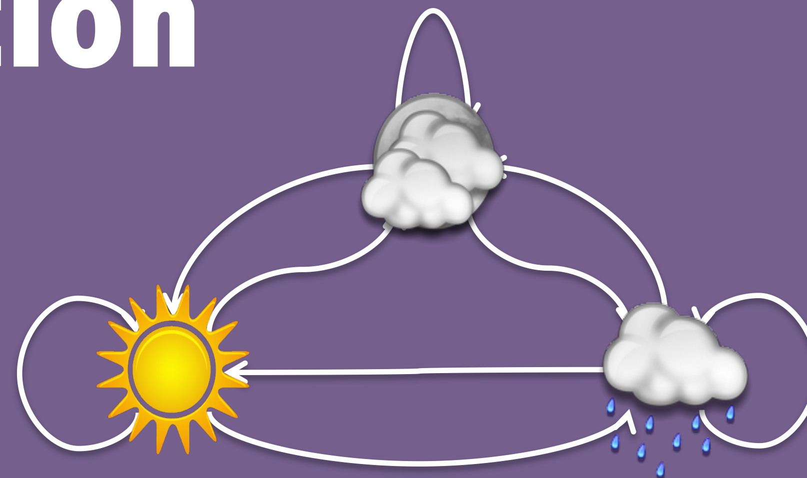
0.5	0.2	0.1

Transitional P

	0	0	1
	0.5	0.5	0
	0.3	0.7	0

Probability of 3-day forecast?: 

Weather Prediction

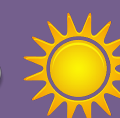


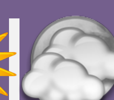
Initial P

		
0.5	0.2	0.1

Transitional P

			
	0	0	1
	0.5	0.5	0
	0.3	0.7	0

Probability of 3-day forecast?:   

P()P(|)P(|) =

0.1 * 0.7 * 0.3 = 0.021

Towards Hidden Markov

what if can't observe the current state?

for example…

CRAZY VENDING MACHINE

Prefers dispensing either Coke or Iced Tea

CRAZY VENDING MACHINE

Prefers dispensing either Coke or Iced Tea

Changes its mind all the time

CRAZY VENDING MACHINE

Prefers dispensing either Coke or Iced Tea

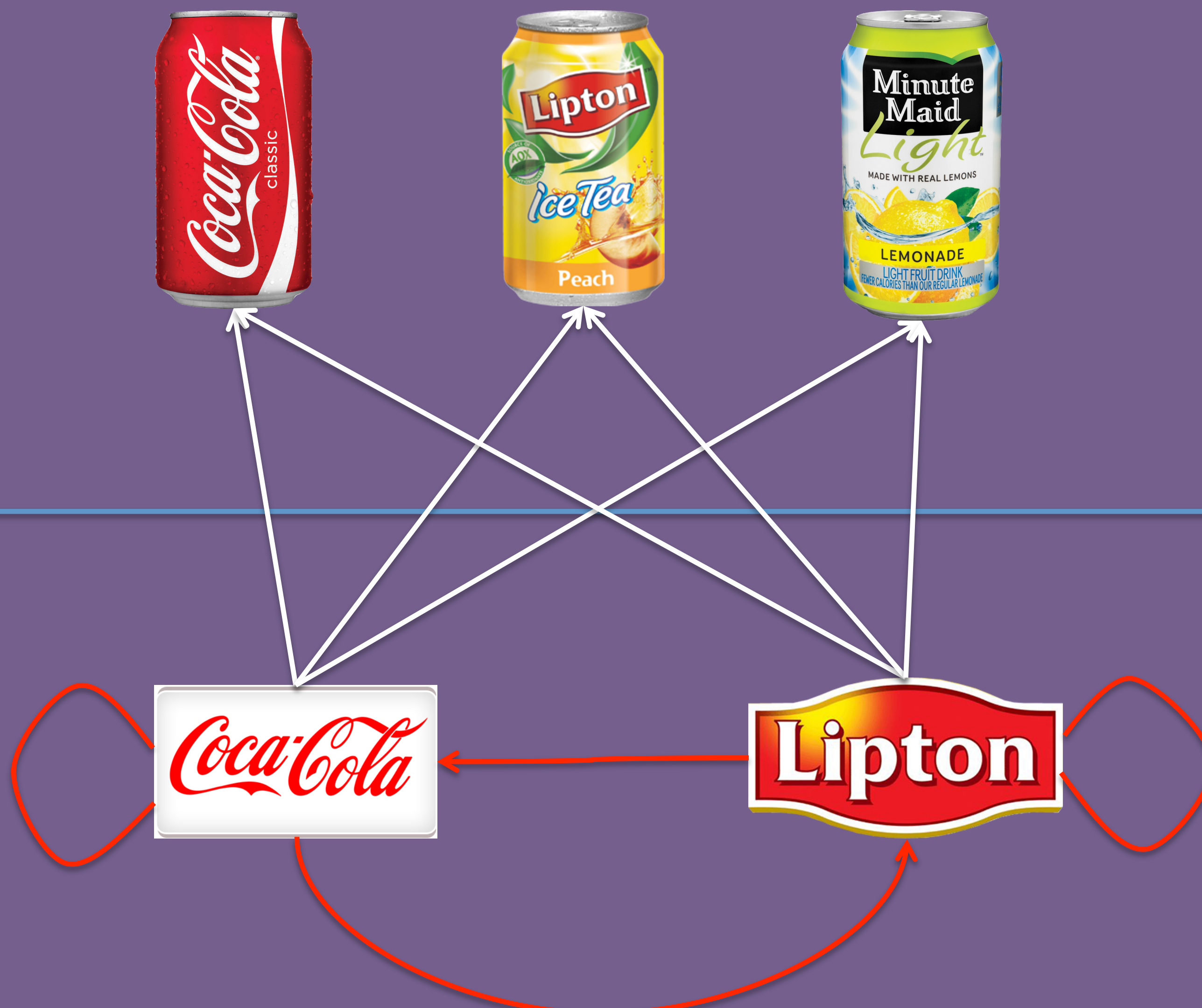
Changes its mind all the time

We don't know its preference at a given moment

CRAZY VENDING MACHINE

observations

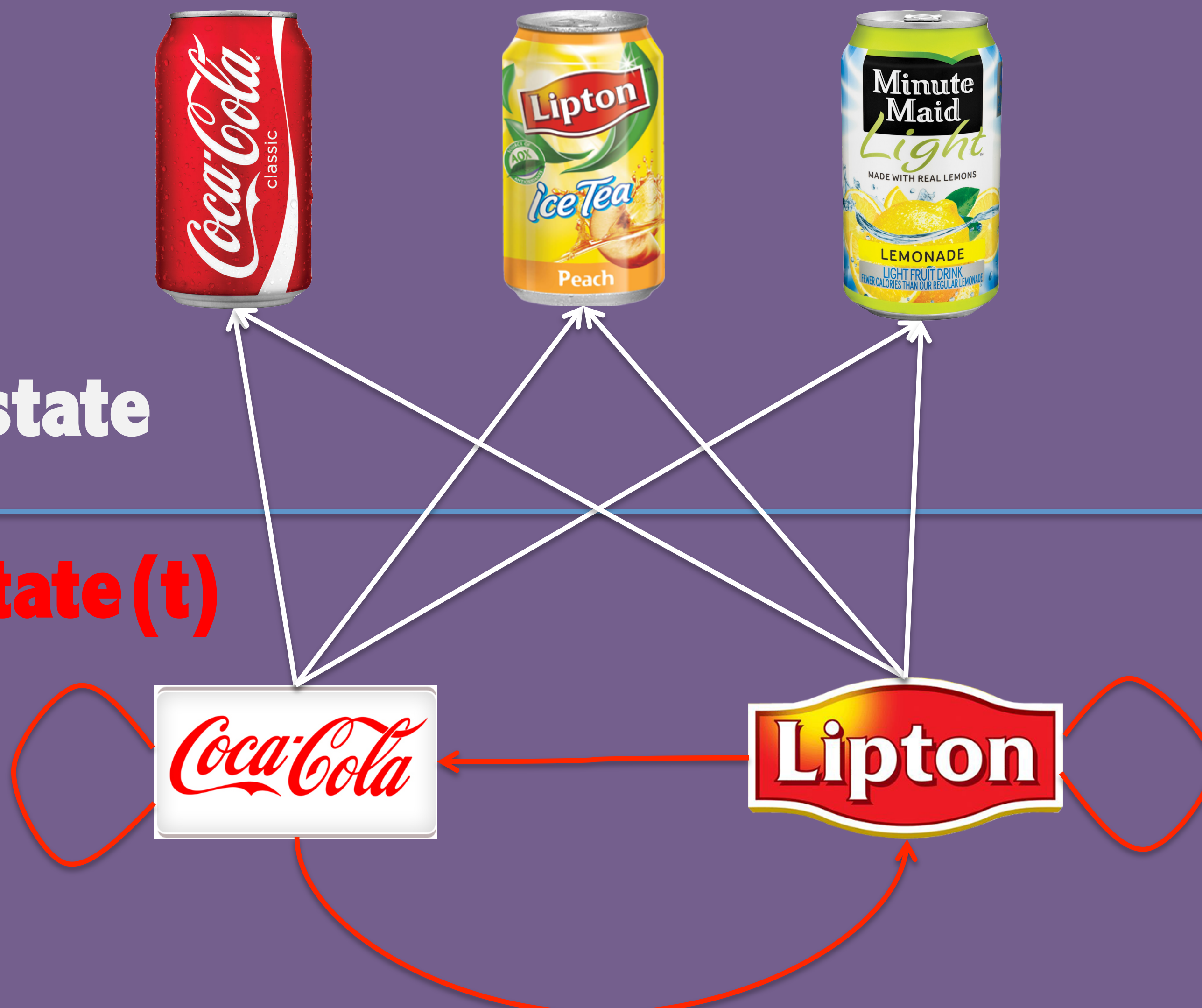
hidden states



CRAZY VENDING MACHINE

observation | state

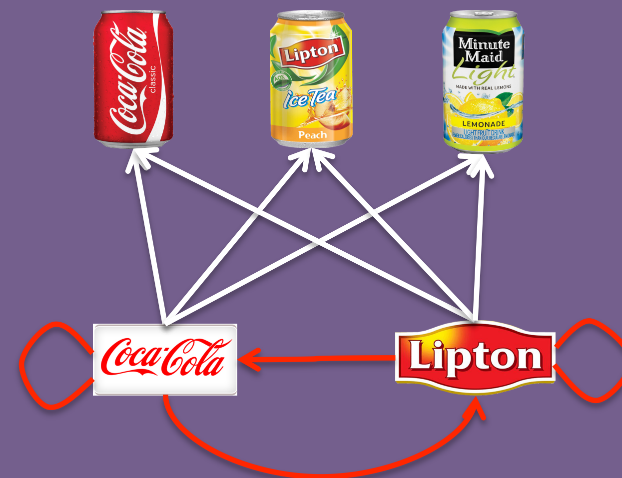
state(t+1) | state(t)



e.g.

Initial P

Coca-Cola	Lipton
1	0



Transitional P

	Coca-Cola	Lipton
Coca-Cola	0.7	0.3
Lipton	0.5	0.5

Output P

	Coca-Cola	Lipton Ice Tea	Minute Maid Light
Coca-Cola	0.6	0.1	0.3
Lipton	0.1	0.7	0.2

e.g.

Probability of vending?:

e.g.

Probability of vending?:

Consider all HMM paths:

T(|)O( |) T(|)O( |) +

e.g.

Probability of vending?:

Consider all HMM paths:

T(Coca-Cola | Coca-Cola) O(Minute Maid | Coca-Cola) T(Coca-Cola | Coca-Cola) O(Lipton | Coca-Cola) +

T(Coca-Cola | Coca-Cola) O(Minute Maid | Coca-Cola) T(Lipton | Coca-Cola) O(Lipton | Lipton) +

e.g.

Probability of vending?:

Consider all HMM paths:

T(Coca-Cola | Coca-Cola) O(Minute Maid | Coca-Cola) T(Coca-Cola | Coca-Cola) O(Lipton Ice Tea | Coca-Cola) +

T(Coca-Cola | Coca-Cola) O(Minute Maid | Coca-Cola) T(Lipton | Coca-Cola) O(Lipton Ice Tea | Lipton) +

T(Lipton | Coca-Cola) O(Minute Maid | Lipton) T(Coca-Cola | Lipton) O(Lipton Ice Tea | Coca-Cola) +

e.g.

Probability of vending?:

Consider all HMM paths:

T(Coca-Cola | Coca-Cola) O(Minute Maid | Coca-Cola) T(Coca-Cola | Coca-Cola) O(Lipton | Coca-Cola) +

T(Coca-Cola | Coca-Cola) O(Minute Maid | Coca-Cola) T(Lipton | Coca-Cola) O(Lipton | Lipton) +

T(Lipton | Coca-Cola) O(Minute Maid | Lipton) T(Coca-Cola | Lipton) O(Lipton | Coca-Cola) +

T(Lipton | Coca-Cola) O(Minute Maid | Lipton) T(Lipton | Lipton) O(Lipton | Lipton) = ...

Hidden Markov

Set of **states** **S**:

$$S = \{s_1, .., s_N\}$$

Hidden Markov

Set of **states S**:

$$S = \{s_1, .., s_N\}$$

Output **alphabet K**:

$$K = \{k_1, \ldots, k_M\} = \{1, \ldots, M\}$$

Hidden Markov

Initial state probabilities **Π**:

$$\Pi = \{\pi_i\}, i \in S$$

Hidden Markov

Initial state probabilities **Π**:

$$\Pi = \{\pi_i\}, i \in S$$

State **transition** probabilities **A**:

$$A = \{a_{ij}\}, i, j \in S$$

Hidden Markov

Initial state probabilities **Π**:

$$\Pi = \{\pi_i\}, i \in S$$

State **transition** probabilities **A**:

$$A = \{a_{ij}\}, i, j \in S$$

Symbol **emission** probabilities **B**:

$$B = \{b_{ijk}\}, i, j \in S, k \in K$$

Hidden Markov

State sequence **X**:

$$X = (X_1, .., X_{T+1})$$

Hidden Markov

State sequence **X**:

$$X = (X_1, .., X_{T+1})$$

Output sequence **O**:

$$O = (o_1, .., o_T)$$

Fundamental Problems

Evaluation:

how **likely** is certain observation **O**?

Given:

$\mu = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$

O

Find:

$P(\mathbf{O}|\mu)$?

Naïve Evaluation

$$
\begin{aligned}
\boxed{P(O|X,\mu)} &= \prod_{t=1}^{T} P(o_t|X_t, X_{t+1}, \mu) \\
&= b_{X_1 X_2 o_1} b_{X_2 X_3 o_2} \cdots b_{X_T X_{T+1} o_T}
\end{aligned}
$$

Naïve Evaluation

$$P(O|X,\mu) = \prod_{t=1}^{T} P(o_t|X_t, X_{t+1}, \mu)$$

$$= b_{X_1 X_2 o_1} b_{X_2 X_3 o_2} \cdots b_{X_T X_{T+1} o_T}$$

$$P(X|\mu) = \pi_{X_1} a_{X_1 X_2} a_{X_2 X_3} \cdots a_{X_T X_{T+1}}$$

Naïve Evaluation

$$P(O|X,\mu) = \prod_{t=1}^{T} P(o_t|X_t, X_{t+1}, \mu)$$

$$= b_{X_1 X_2 o_1} b_{X_2 X_3 o_2} \cdots b_{X_T X_{T+1} o_T}$$

$$P(X|\mu) = \pi_{X_1} a_{X_1 X_2} a_{X_2 X_3} \cdots a_{X_T X_{T+1}}$$

$$P(O, X|\mu) = P(O|X,\mu) P(X|\mu)$$

Naïve Evaluation

$$P(O|X,\mu) = \prod_{t=1}^{T} P(o_t|X_t, X_{t+1}, \mu)$$

$$= b_{X_1 X_2 o_1} b_{X_2 X_3 o_2} \cdots b_{X_T X_{T+1} o_T}$$

$$P(X|\mu) = \pi_{X_1} a_{X_1 X_2} a_{X_2 X_3} \cdots a_{X_T X_{T+1}}$$

$$P(O, X|\mu) = P(O|X,\mu)P(X|\mu)$$

$$P(O|\mu) = \sum_{X} P(O|X,\mu)P(X|\mu)$$

$$= \sum_{X_1 \cdots X_{T+1}} \pi_{X_1} \prod_{t=1}^{T} a_{X_t X_{t+1}} b_{X_t X_{t+1} o_t}$$

Naïve Evaluation

$$P(O|X,\mu) = \prod_{t=1}^{T} P(o_t|X_t, X_{t+1}, \mu)$$

$$= b_{X_1 X_2 o_1} b_{X_2 X_3 o_2} \cdots b_{X_T X_{T+1} o_T}$$

$$P(X|\mu) = \pi_{X_1} a_{X_1 X_2} a_{X_2 X_3} \cdots a_{X_T X_{T+1}}$$

$$P(O, X|\mu) = P(O|X,\mu)P(X|\mu)$$

$$P(O|\mu) = \sum_X P(O|X,\mu)P(X|\mu)$$

$$= \sum_{X_1 \cdots X_{T+1}} \pi_{X_1} \prod_{t=1}^{T} a_{X_t X_{t+1}} b_{X_t X_{t+1} o_t}$$

$$(2T+1) \cdot N^{T+1}$$

calculations!

Smarter Evaluation

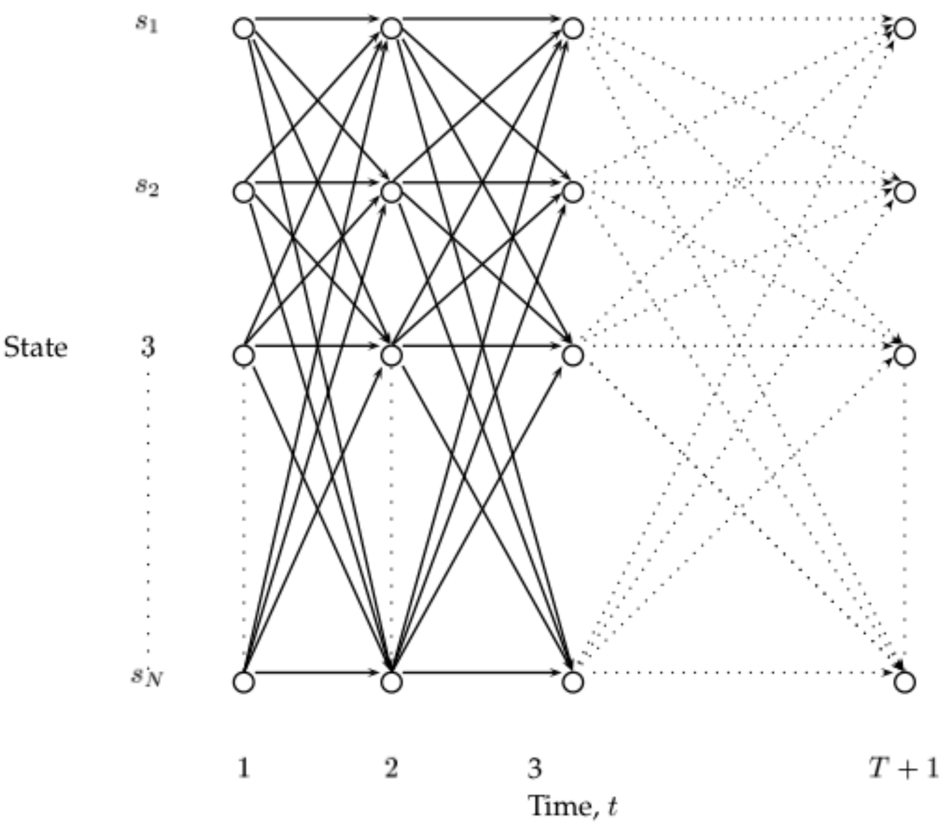
Use DP! FW-BW Alg.

Smarter Evaluation

Use DP! FW-BW Alg.

DP Table:

state over time

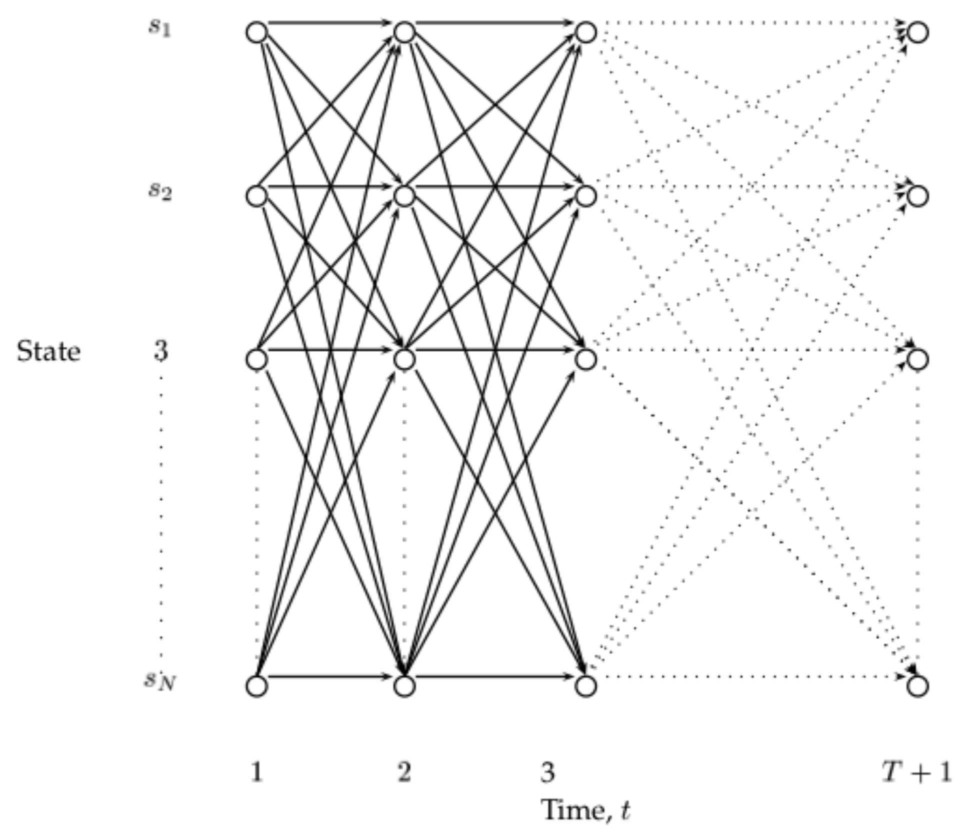


Smarter Evaluation

Use DP! FW-BW Alg.

DP Table:

state over time



store forward variables:

$$\alpha_i(t) = P(o_1 o_2 \cdots \cdots o_{t-1}, X_t = i|\mu)$$

Smarter Evaluation

compute forward variables:

1. initialization:

$$\alpha_i(1) = \pi_i$$

Smarter Evaluation

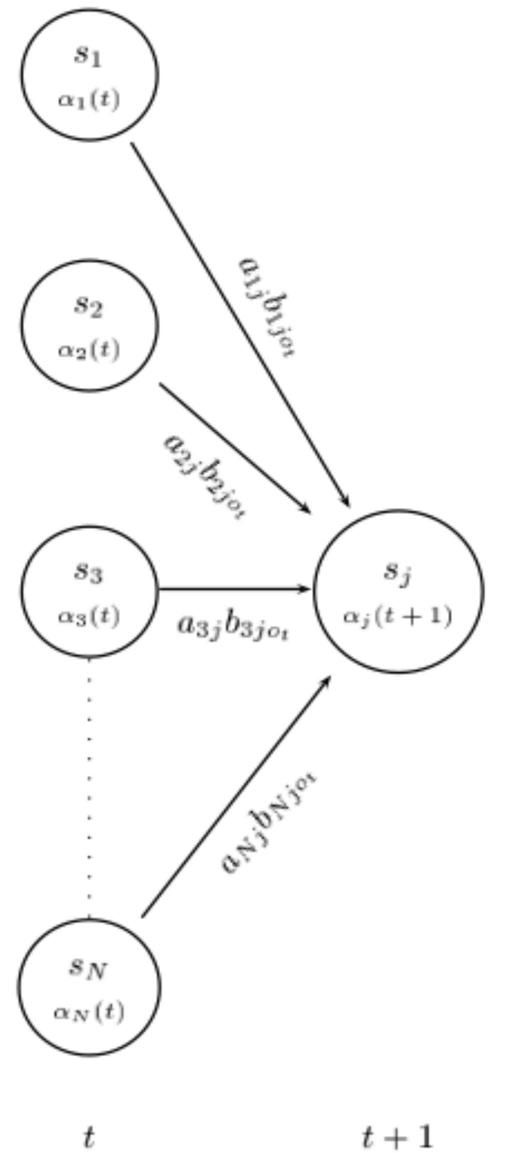
compute forward variables:

1. initialization:

$$\alpha_i(1) = \pi_i$$

2. induction:

$$a_j(t+1) = \sum_{i=1}^{N} \alpha_i(t) a_{ij} b_{ijo_t}$$



Smarter Evaluation

compute forward variables:

1. initialization:

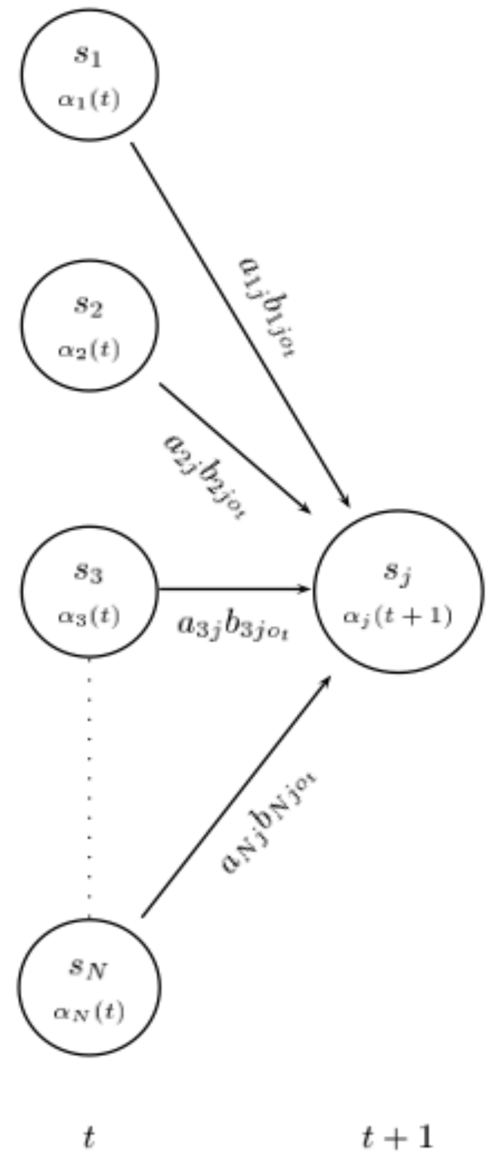
$$\alpha_i(1) = \pi_i$$

2. induction:

$$a_j(t+1) = \sum_{i=1}^{N} \alpha_i(t) a_{ij} b_{ijo_t}$$

3. total:

$$P(O|\mu) = \sum_{i=1}^{N} \alpha_i(T+1)$$



Smarter Evaluation

much lower complexity than naïve:

$$2N^2T$$ calculations! vs. $$(2T+1) \cdot N^{T+1}$$ calculations!

Smarter Evaluation

much lower complexity than naïve:

$$2N^2T$$ calculations! vs. $$(2T+1) \cdot N^{T+1}$$ calculations!

similarly, can work backwards:

$$\beta_i(t) = P(o_t \cdots o_T | X_t = i, \mu)$$

Fundamental Problems

Inference:

finding **X** that best explains **O**?

Given:

$\mu = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$

O

Find:

$$\underset{X}{\operatorname{argmax}}\, P(\mathbf{X} | \mathbf{O}, \mu)$$

Smarter Inference

Again, use DP! Viterbi Algorithm

Smarter Inference

Again, use DP! Viterbi Algorithm

Store:

probability of the most probable path that leads to a node

$$\delta_j(t) = \max_{X_1 \cdots X_{t-1}} P(X_1 \cdots X_{t-1}, o_1 \cdots o_{t-1}, X_t = j | \mu)$$

Smarter Inference

Again, use DP! Viterbi Algorithm

Store:

- **probability of the most probable path that leads to a node**

$$\delta_j(t) = \max_{X_1 \cdots X_{t-1}} P(X_1 \cdots X_{t-1}, o_1 \cdots o_{t-1}, X_t = j | \mu)$$

- **backtrack through max solution to find the path**

Smarter Evaluation

compute the variables (fill in the DP table):

1 initialization:

$$\delta_i(1) = \pi_i$$

Smarter Evaluation

compute the variables (fill in the DP table):

1 initialization:

$$\delta_i(1) = \pi_i$$

2.2 induction:

$$\delta_j(t+1) = \max_{1 \leq i \leq N} \delta_i(t) a_{ij} b_{ij o_t}$$

Smarter Evaluation

compute the variables (fill in the DP table):

1 initialization:

$$\delta_i(1) = \pi_i$$

2.2 induction:

$$\delta_j(t+1) = \max_{1 \leq i \leq N} \delta_i(t) a_{ij} b_{ijo_t}$$

2.2 store backtrace:

$$\psi_j(t+1) = arg \max_{1 \leq i \leq N} \delta_i(t) a_{ij} b_{ijo_t}$$

Smarter Evaluation

3 termination and path readout:

$$
\begin{aligned}
\hat{X}_{T+1} &= \arg\max_{1 \le i \le N} \delta_i(T+1) \\
\hat{X}_t &= \psi_{\hat{X}_{t+1}}(t+1) \\
P(\hat{X}) &= \max_{1 \le i \le N} \delta_i(T+1)
\end{aligned}
$$

Fundamental Problems

Estimation:

finding **μ** that best explains **O**?

Given:

$\mathbf{O}_{\mathbf{training}}$

Find:

$$\underset{\mu}{\operatorname{argmax}}\ P(\mathbf{O}_{\mathbf{training}}, \mu)$$

Estimation: MLE

no known analytic method

Estimation: MLE

no known analytic method

find local max using iterative hill-climb

Estimation: MLE

no known analytic method

find local max using iterative hill-climb

Baum-Welch: (outline)

1 choose a model μ (perhaps randomly)

Estimation: MLE

no known analytic method

find local max using iterative hill-climb

Baum-Welch: (outline)

1 choose a model μ (perhaps randomly)

2 estimate $P(O \mid \mu)$

Estimation: MLE

no known analytic method

find local max using iterative hill-climb

Baum-Welch: (outline)

1 choose a model μ (perhaps randomly)

2 estimate P(O | μ)

3 choose a revised model μ to maximize the values of the paths used a lot…

Estimation: MLE

no known analytic method

find local max using iterative hill-climb

Baum-Welch: (outline)

1 **choose a model μ (perhaps randomly)**

2 **estimate P(O | μ)**

3 **choose a revised model μ to maximize the values of the paths used a lot…**

4 **repeat 1-3, hope to converge on values of μ**

When HMMs are good..

Observations are **ordered**

Random process can be represented by a **stochastic finite state machine** with emitting states

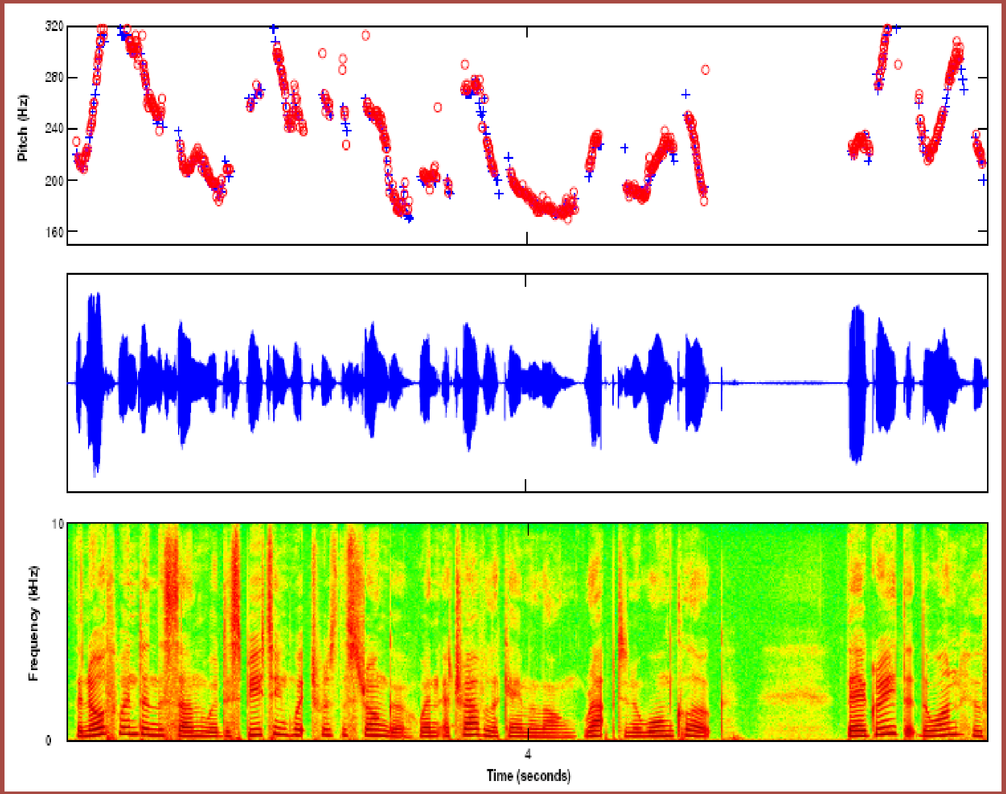
Why HMMs are good..

1. Statistical Grounding
2. Modularity
3. Transparency of a Model
4. Incorporation of Prior Knowledge

Why HMMs are bad..

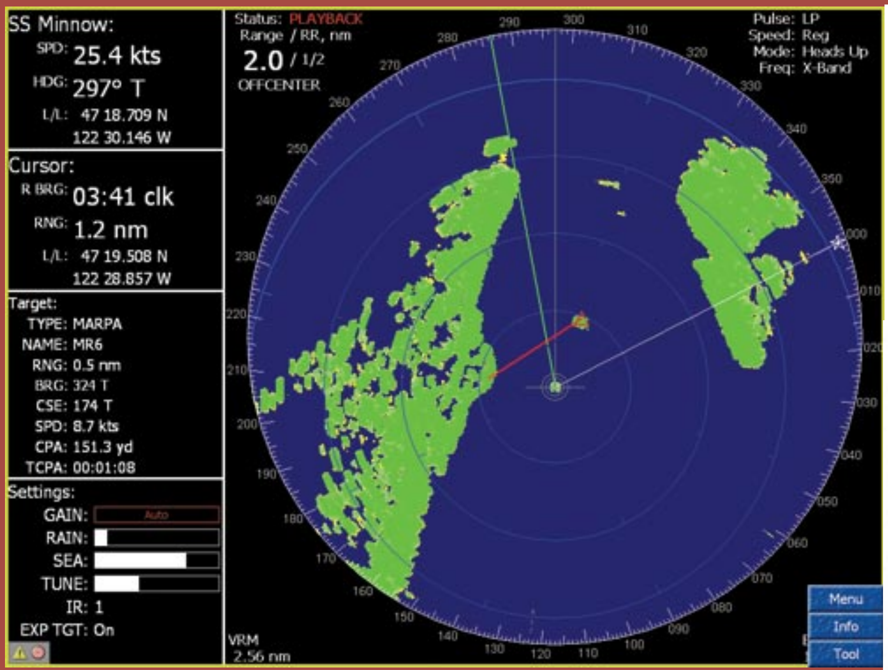
1. Markov Chains
2. Local Maxima/Over Fitting
3. Slower Speed

Speech Recognition

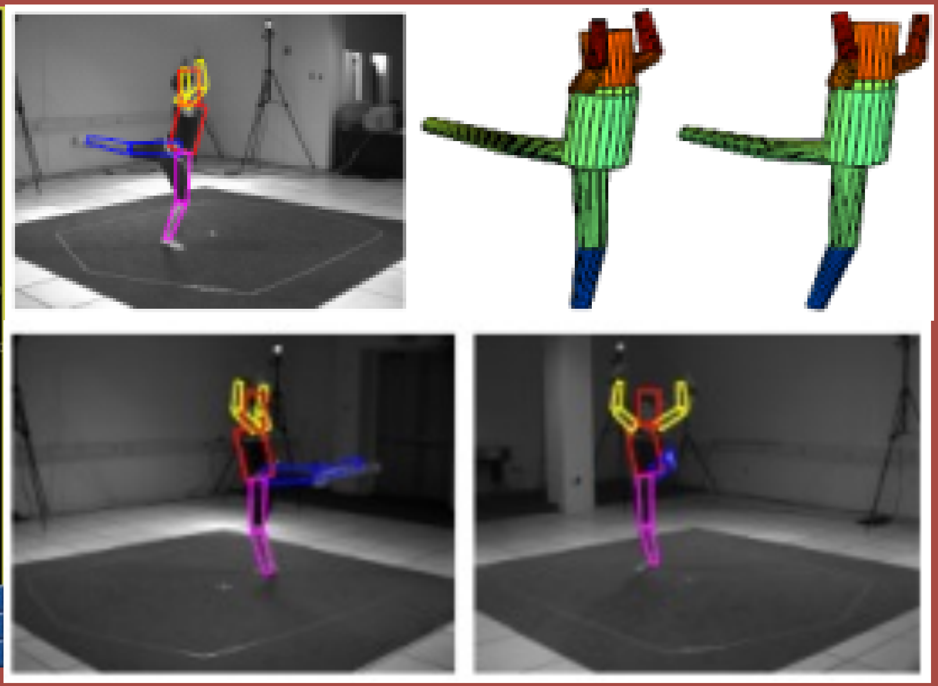


given an audio waveform, would like to robustly extract & recognize any spoken words

Target Tracking



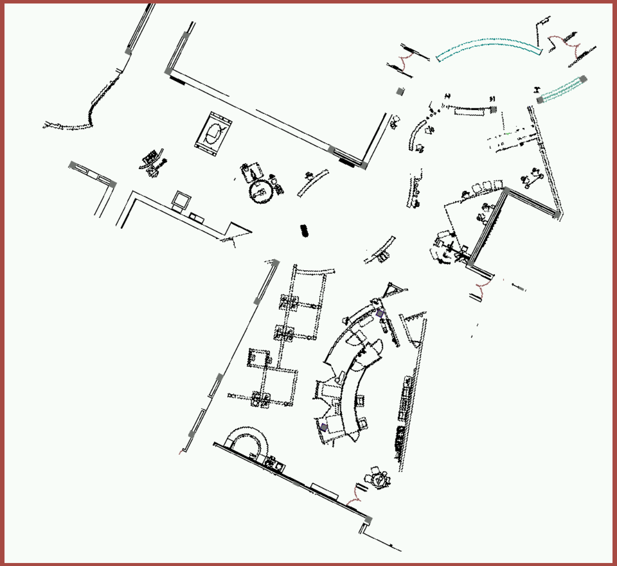
Radar-based tracking of multiple targets



Visual tracking of articulated objects

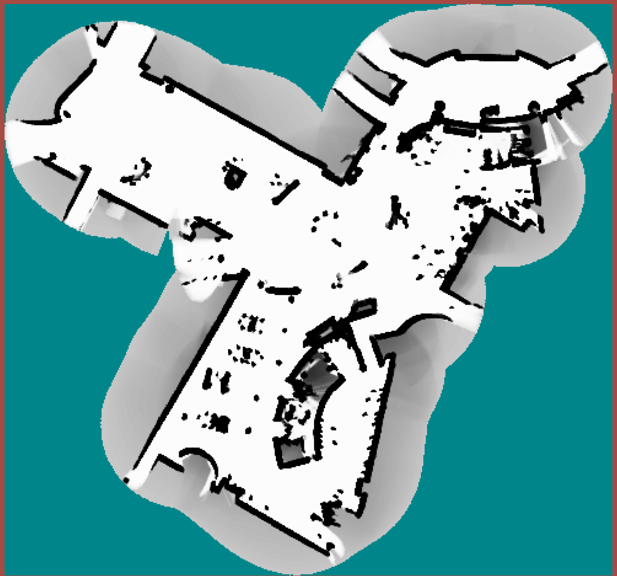
estimate motion of targets in 3D world from indirect, potentially noisy measurements

Robot Navigation



CAD Map

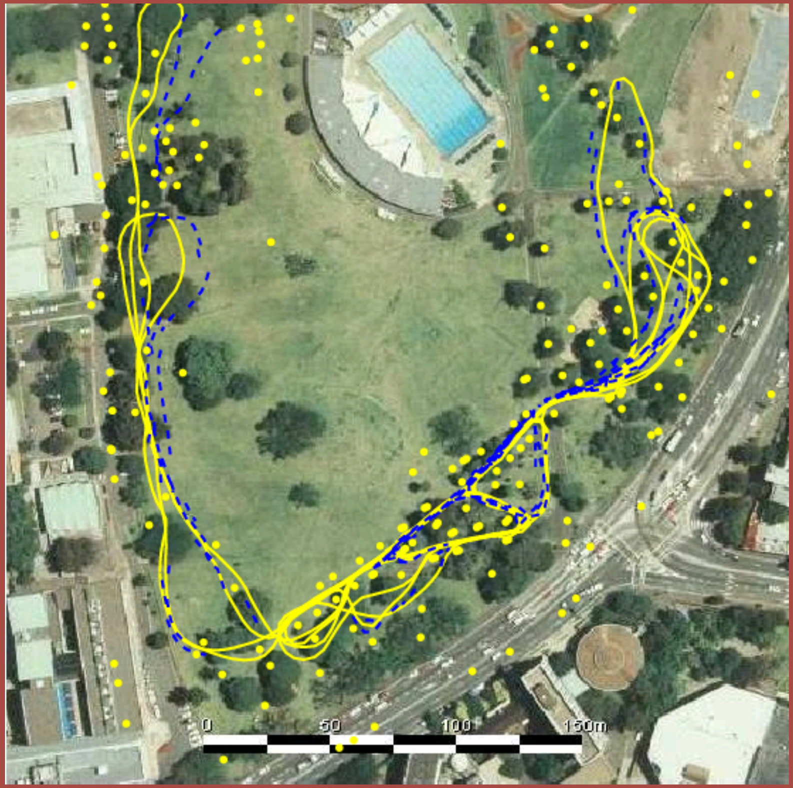
(S. Thrun, San Jose Tech Museum)



Estimated Map

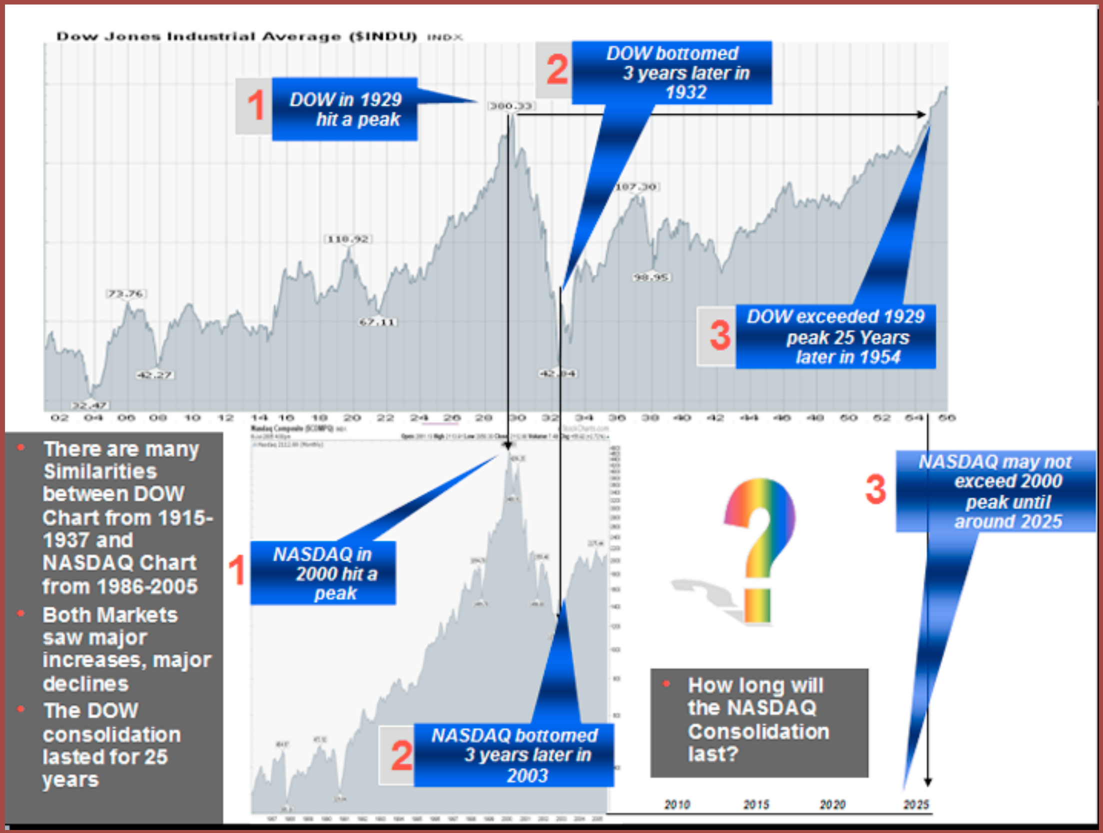
Landmark SLAM

(E. Nebot, Victoria Park)



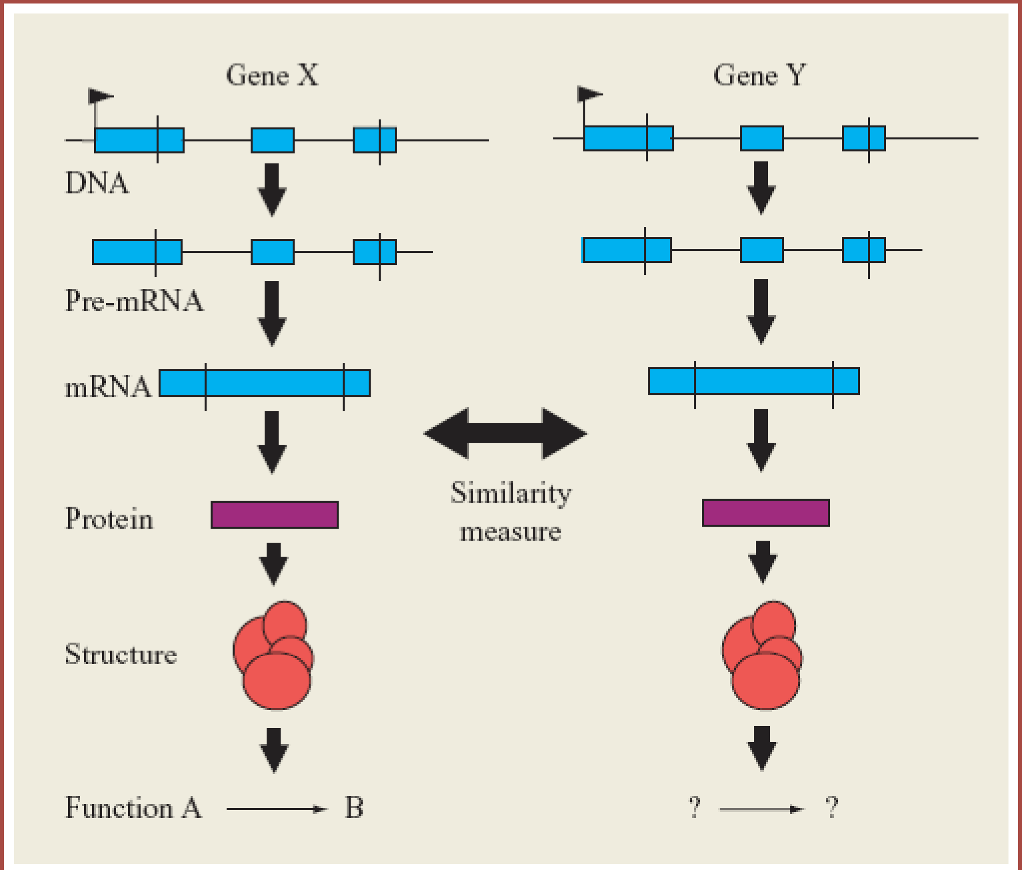
as robot moves, estimate its world geometry

Financial Forecasting



predict future market behavior from historical data, news reports, expert opinions,..

Bioinformatics



multiple sequence alignment, gene finding, motif/promoter region finding..

HMM Applications

HMM can be applied in many more fields where the goal is to recover sequence that is not immediately observable:

- cryptoanalysis
- POS tagging
- MT
- activity recognition
- etc.

Thank You