Hidden Markov Models

GU 4241/GR 5241

Statistical Machine Learning

Xiaofei Shi

Generative models are powerful

- Conditional generative model P(zebra images| horse images)

- Style Transfer

Input Image Monet Van Gogh

Zhou el al., Cycle GAN 2017

COLUMBIA UNIVERSITY

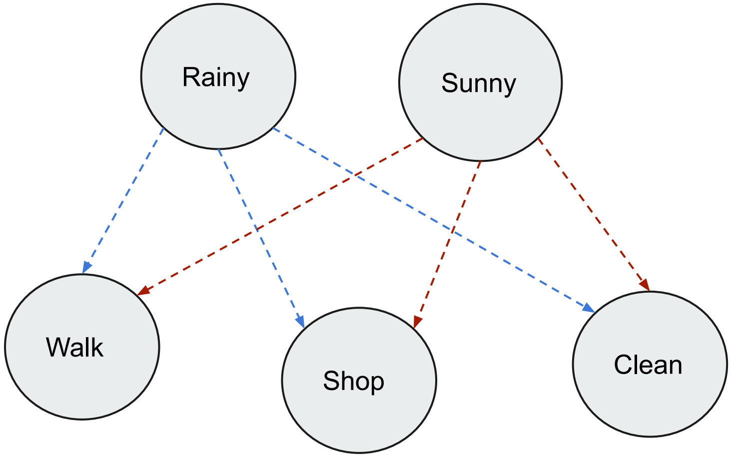
Generative models are powerful

Example of the Progression in the Capabilities of GANs From 2014 to 2017. Taken from The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation, 2018.

Recall Bayesian networks

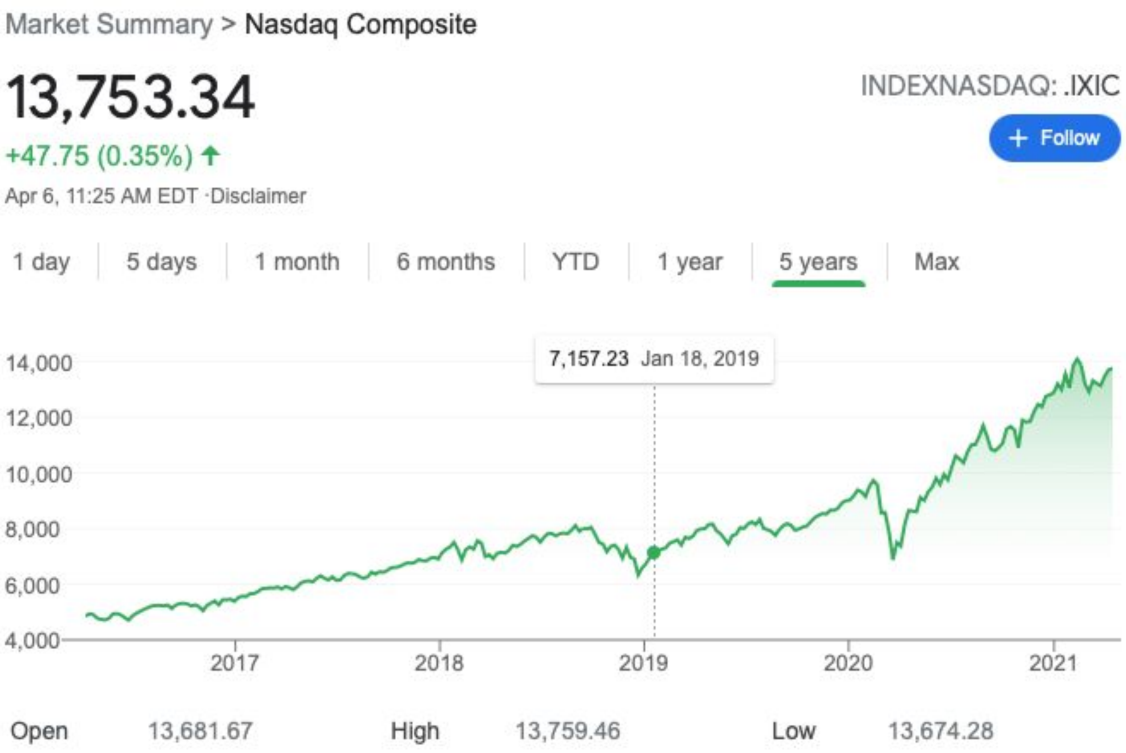
Consists of 2 parts:

- Probability belongs to each class
- Class conditional probability



There exists a direction!

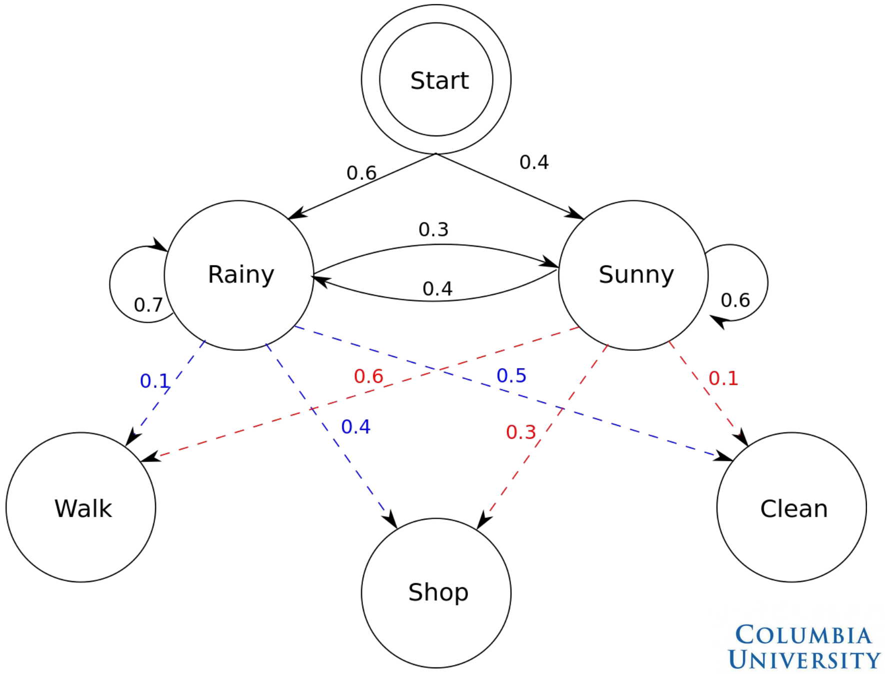
Generative models are powerful



COLUMBIA UNIVERSITY

Limitation of Bayesian networks

- Doing full Bayesian networks are computationally expensive;
- Performance is suboptimal on high-dimensional data;
- Bayesian networks are directed acyclic graphical models, where the directed edges are used to capture causal relationship between random variables, but we need undirected graphical model with potential loops in between nodes to model correlations.
- Bayesian networks cannot be used to model temporal/sequence data.

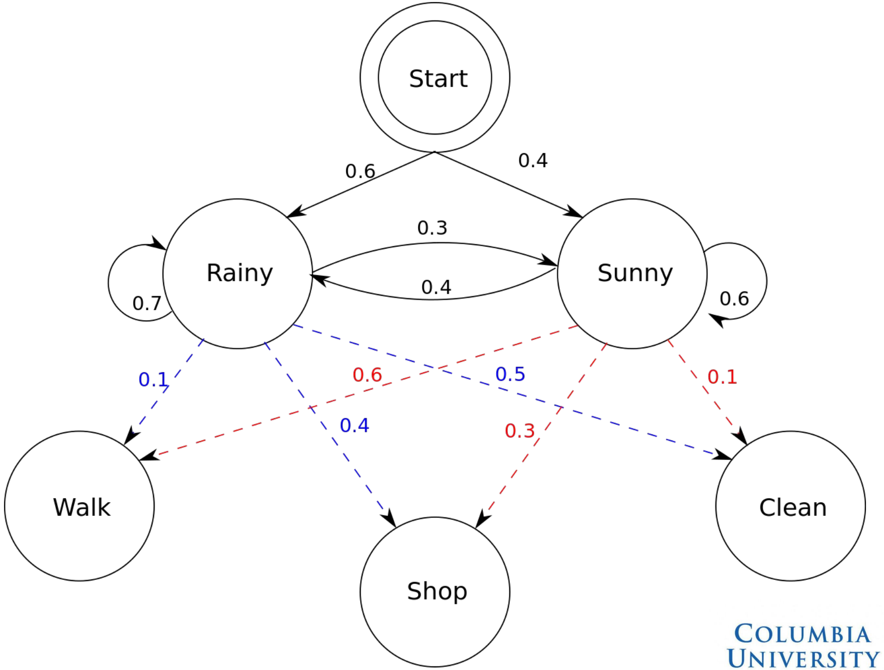


Hidden Markov models

Model a set of observations using a set of hidden states:

(observations, hidden states) such as:

- (pixel inputs, features)
- (range/visual sensor, location)
- (sound/visual signal, language/situation/words)
- (facial expression, results from the driving test)



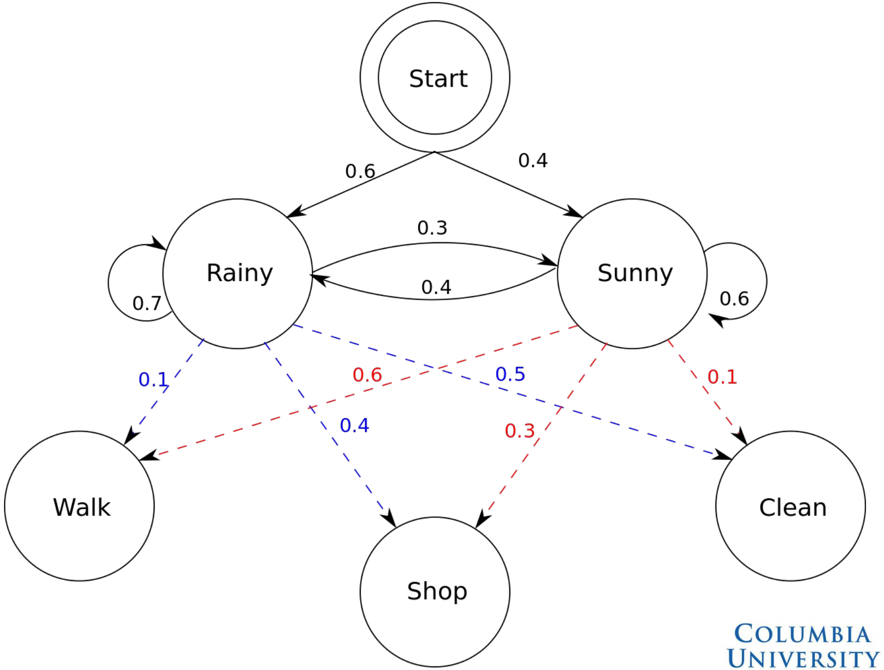
Hidden Markov models

Model a set of observations using a set of hidden states:

(observations, hidden states) such as:

- (pixel inputs, features)
- (range/visual sensor, location)
- (sound/visual signal, language/situation/words)
- (facial expression, results from the driving test)

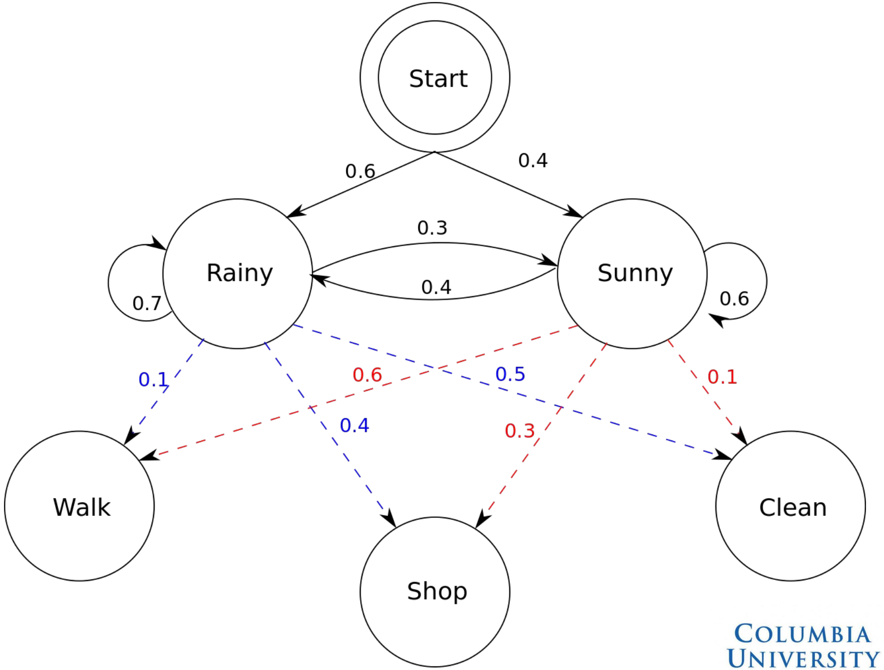
hidden state generates observation;
hidden state transitions to other hidden states



Example: Daisy’s diary

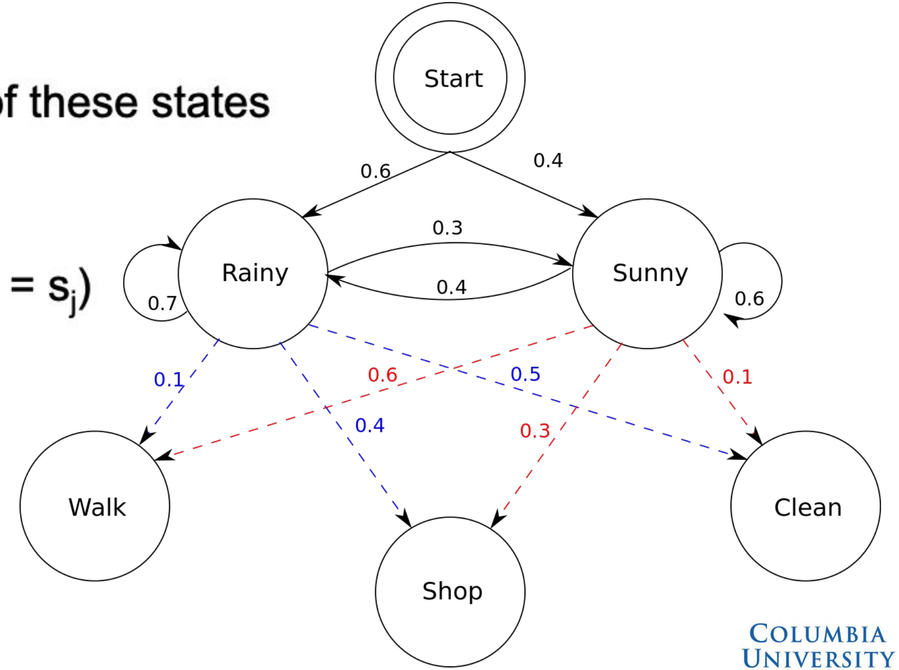
Daisy kept a diary on what she was doing but she forgot to keep track of the weather. Let’s treat the weather conditions as hidden states:

- With 60%/40%, today is rainy/sunny
- If today is rainy, it would either remain rainy (70%), or could be sunny tomorrow (30%).



Definition of a hidden Markov model

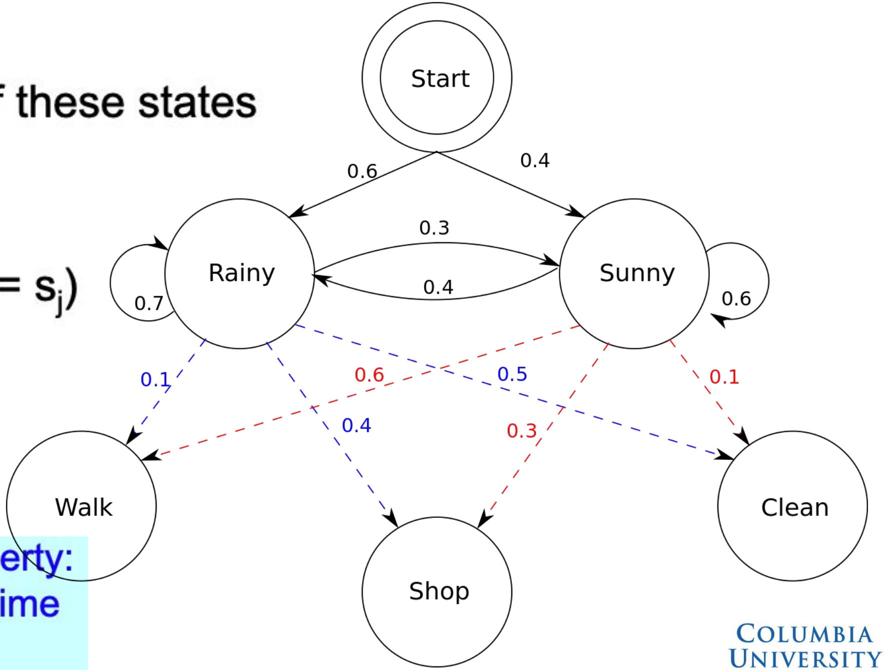
- A set of states $\{s_1 \ldots s_n\}$
  - In each time point we are in exactly one of these states denoted by q_t
- Π_i, the probability that we *start* at state s_i
- A transition probability model, $P(q_t = s_i \mid q_{t-1} = s_j)$
- A set of possible outputs Σ
  - At time *t* we emit a symbol $\sigma \in \Sigma$
- An emission probability model, $p(o_t = \sigma \mid s_i)$



COLUMBIA UNIVERSITY

Definition of a hidden Markov model

- A set of states $\{s_1 \dots s_n\}$
  - In each time point we are in exactly one of these states denoted by q_t
- Π_i, the probability that we *start* at state s_i
- A transition probability model, $P(q_t = s_i \mid q_{t-1} = s_j)$
- A set of possible outputs Σ
  - At time *t* we emit a symbol $\sigma \in \Sigma$
- An emission probability model, $p(o_t = \sigma \mid s_i)$



An important aspect of this definition is the Markov property: q_{t+1} is conditionally independent of q_{t-1} (and any earlier time points) given q_t

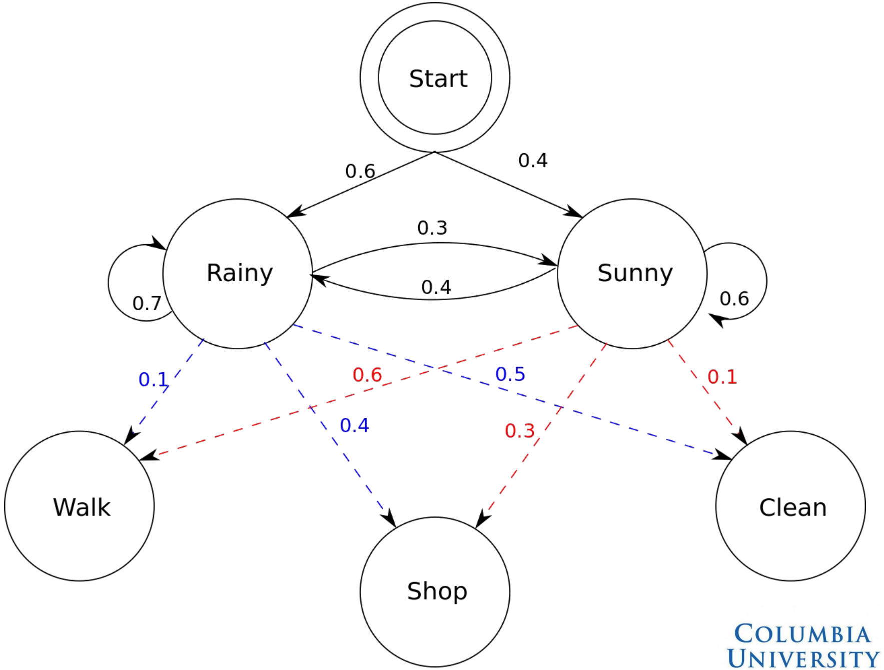
More formally $P(q_{t+1} = s_i \mid q_t = s_j) = P(q_{t+1} = s_i \mid q_t = s_j, q_{t-1} = s_j)$

Columbia University

Inference in HMM

Several questions people can ask:

- What is the current weather?
- What is the probability for shopping tomorrow?
- What is the probability for shopping next week?



Inference in HMM

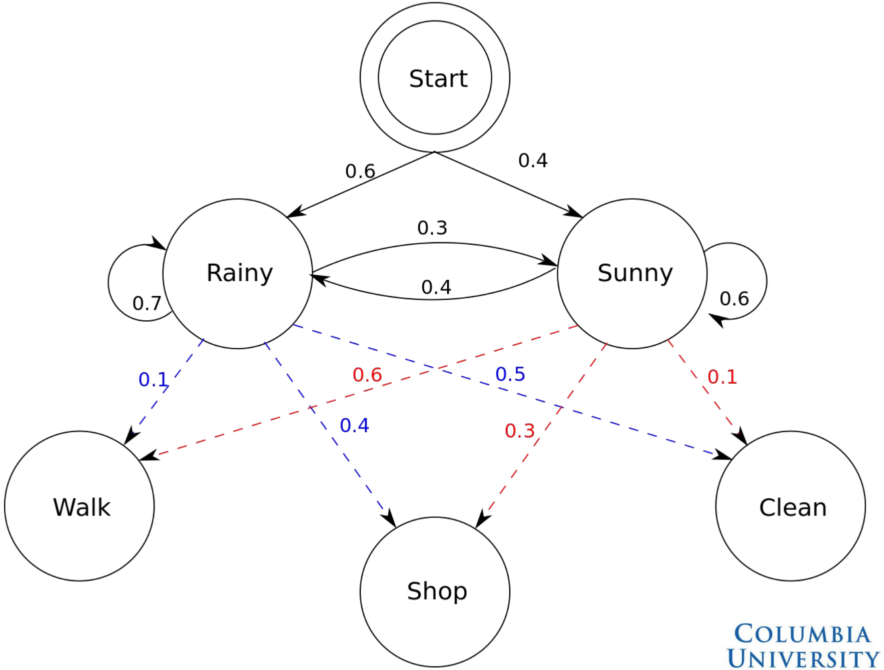
When no observation:

- What is the current weather?

  P[rainy] = 0.6 P[sunny]=0.4
- What is the weather tomorrow?

  P[rainy] = 0.6 x 0.7 + 0.4 x 0.4

  P[sunny] = 0.6 x 0.3 + 0.4 x 0.6
- What is the weather on the t-th day?



Inference in HMM

When no observation:

- What is the current weather?

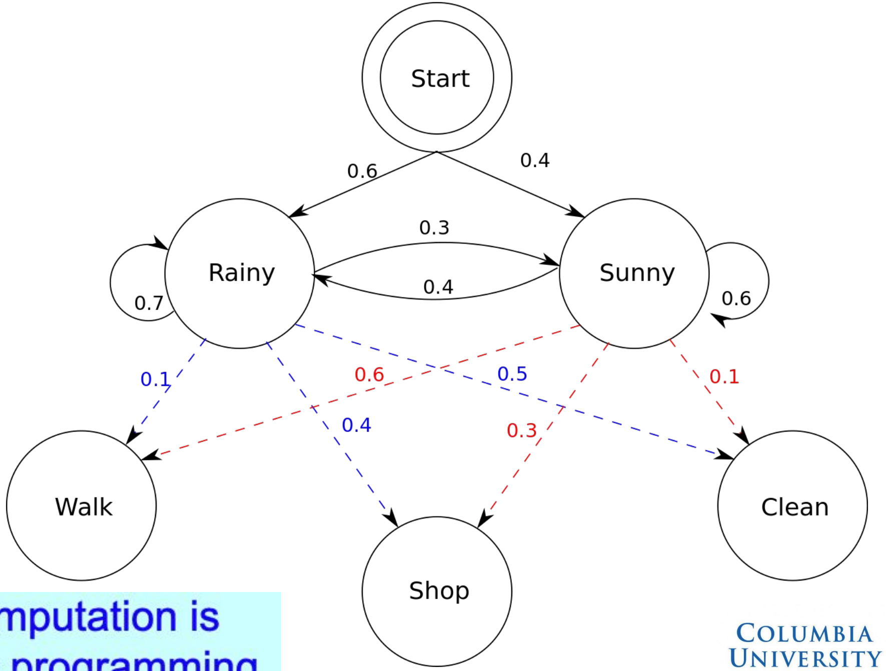
  P[rainy] = 0.6   P[sunny]=0.4

- What is the weather tomorrow?

  P[rainy] = 0.6 x 0.7 + 0.4 x 0.4

  P[sunny] = 0.6 x 0.3 + 0.4 x 0.6

- What is the weather on the t-th day?



Lets define $p_t(i)$ = probability state i at time t = $p(q_t = s_i)$

We can determine $p_t(i)$ by induction

1. $p_1(i) = \Pi_i$
2. $p_t(i) = \Sigma_j \, p(q_t = s_i \mid q_{t-1} = s_j) p_{t-1}(j)$

This type of computation is called dynamic programming

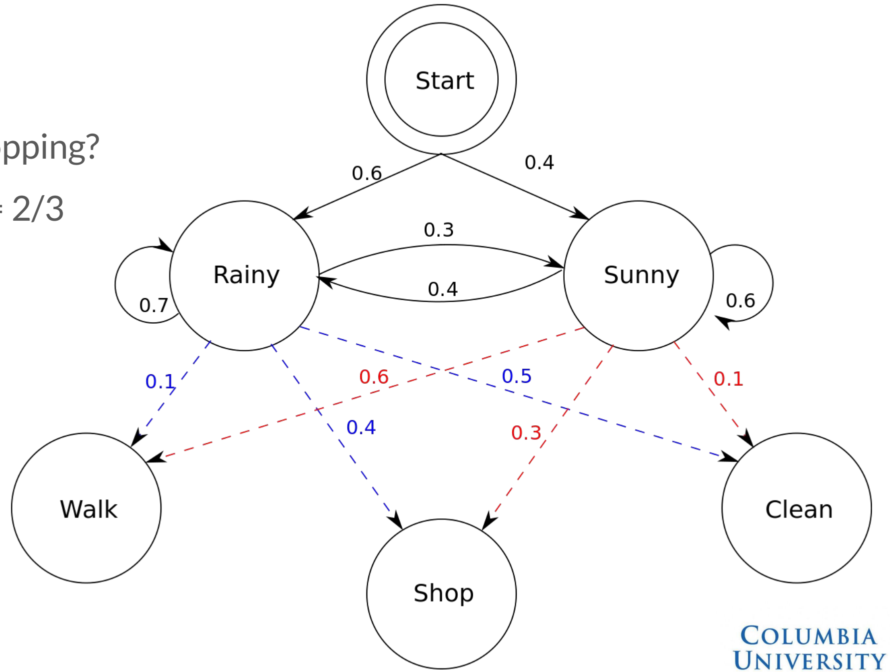
Complexity: $O(n^2 * t)$

Inference in HMM

When with observations:

- What is the current weather given that Daisy is shopping?

  P[rainy|shopping] = 0.6 x 0.4/(0.6 x 0.4 +0.4 X 0.3) = 2/3



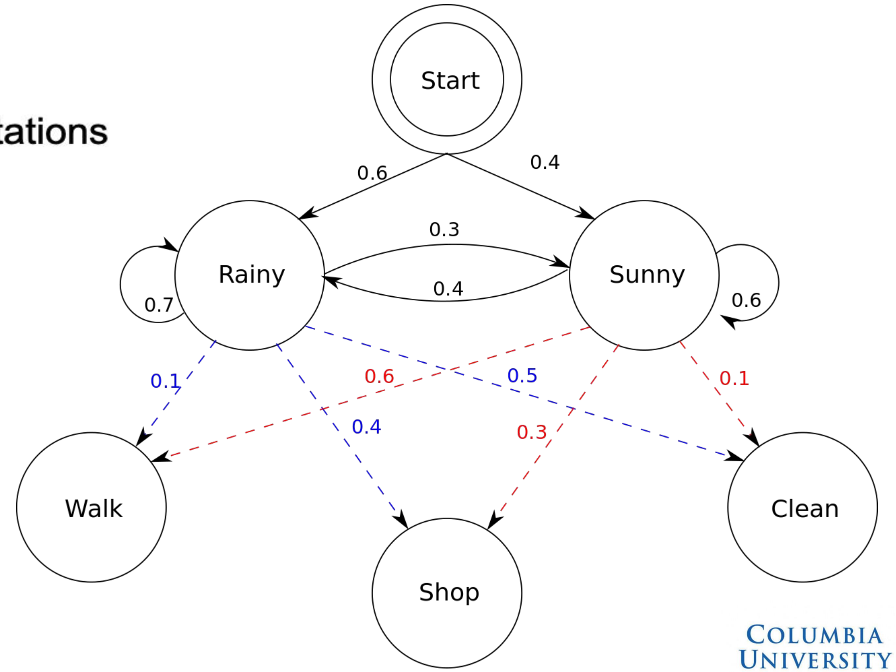
COLUMBIA UNIVERSITY

Inference in HMM

- We want to compute $P(q_t = A \mid O_1 \ldots O_t)$
- For ease of writing we will use the following notations (commonly used in the literature)
- $a_{j,i} = P(q_t = s_i \mid q_{t-1} = s_j)$
- $b_i(o_t) = P(o_t \mid s_i)$

Transition probability

Emission probability



Columbia University

Inference in HMM

- We want to compute $P(q_t = A \mid O_1 \ldots O_t)$
- Lets start with a simpler question. Given a sequence of states Q, what is $P(Q \mid O_1 \ldots O_t) = P(Q \mid O)$?

  - It is pretty simple to move from $P(Q|O)$ to $P(q_t = A \mid O)$

  - In some cases $P(Q \mid O)$ is the more important question

    - Speech processing

    - NLP

COLUMBIA UNIVERSITY

Inference in HMM

$$P(Q|O) = \frac{P(O \mid Q)P(Q)}{P(O)}$$

Easy, $P(O \mid Q) = P(o_1 \mid q_1)\, P(o_2 \mid q_2) \ldots P(o_t \mid q_t)$

$$P(Q|O) = \frac{P(O \mid Q)P(Q)}{P(O)}$$

$P(Q) = P(q_1)\, P(q_2 \mid q_1) \ldots P(q_t \mid q_{t-1})$

But it is comparatively hard to compute P(O), i.e. the probability of seeing a set of observations.

Columbia University

Inference in HMM

- What is the probability of seeing a set of observations:
  - An important question in it own rights, for example classification using two HMMs
- Define $\alpha_t(i) = P(o_1, o_2 \ldots, o_t \wedge \ q_t = s_i)$
- $\alpha_t(i)$ is the probability that we:

  1. Observe $o_1, o_2 \ldots, o_t$
  2. End up at state i

COLUMBIA UNIVERSITY

Inference in HMM

- We want to compute P(Q | O)
- For this, we only need to compute P(O)
- We know how to compute $\alpha_t(i)$

From now its easy

$\alpha_t(i) = P(o_1, o_2 \ldots, o_t \wedge q_t = s_i)$

so

$P(O) = P(o_1, o_2 \ldots, o_t) = \Sigma_i P(o_1, o_2 \ldots, o_t \wedge q_t = s_i) = \Sigma_i \alpha_t(i)$

note that

$$p(q_t = s_i \mid o_1, o_2 \ldots, o_t) = \frac{\alpha_t(i)}{\sum_j \alpha_t(j)}$$

$P(A \mid B) = P(A \wedge B) / P(B)$

COLUMBIA UNIVERSITY

Computational complexity

- How long does it take to compute P(Q | O)?
- P(Q): O(t)
- P(O|Q): O(t)
- P(O): $O(n^2t)$

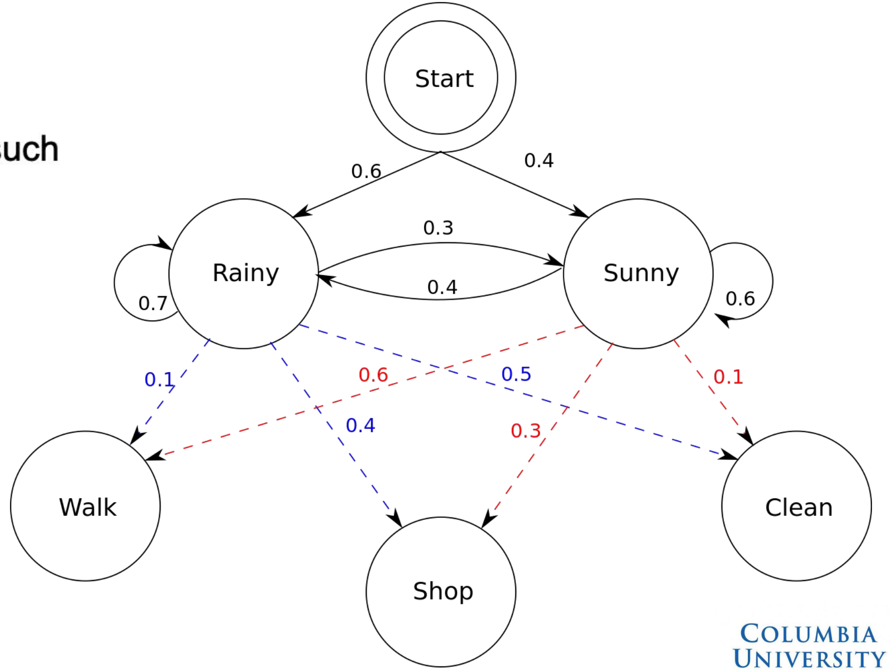
Inference in HMM

- We are almost done …
- One final question remains

  How do we find the most probable path, that is Q* such that

  $$P(Q^* \mid O) = \text{argmax}_Q P(Q|O)?$$

- This is an important path
  - The words in speech processing
  - The set of genes in the genome
  - etc.



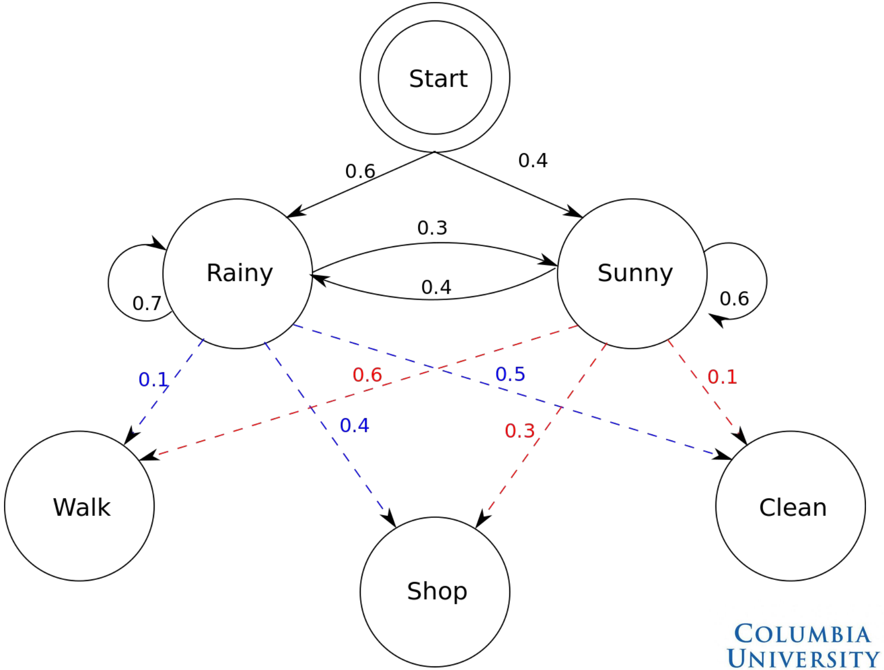
COLUMBIA UNIVERSITY

Inference in HMM

- What's the most likely sequence for you to see Daisy goes out for walking for 7 days in a week?

$$\arg\max_Q P(Q \mid O) = \arg\max_Q \frac{P(O \mid Q)P(Q)}{P(O)}$$

$$= \arg\max_Q P(O \mid Q)P(Q)$$



Inference in HMM

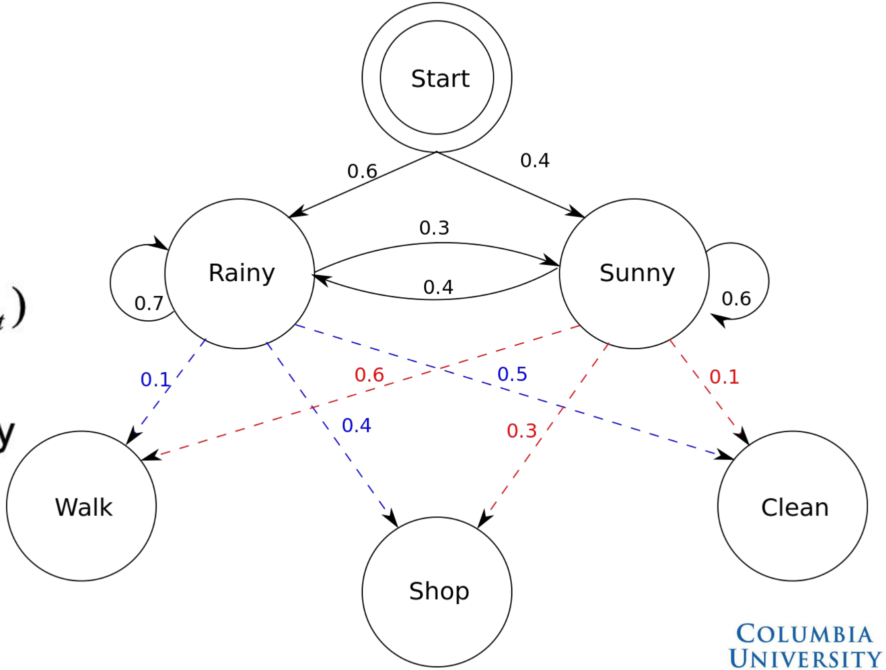
$$\arg\max_Q P(Q \mid O) = \arg\max_Q \frac{P(O \mid Q)P(Q)}{P(O)}$$

We will use the following definition:

$$\delta_t(i) = \max_{q_1 \ldots q_{t-1}} p(q_1 \ldots q_{t-1} \wedge q_t = s_i \wedge O_1 \ldots O_t)$$

In other words we are interested in the most likely path from 1 to t that:

1. Ends in S_i
2. Produces outputs $O_1 \ldots O_t$



COLUMBIA UNIVERSITY

Inference in HMM

$$\arg\max_Q P(Q \mid O) = \arg\max_Q \frac{P(O \mid Q)P(Q)}{P(O)}$$

We will use the following definition:

$$\delta_t(i) = \max_{q_1 \dots q_{t-1}} p(q_1 \dots q_{t-1} \wedge q_t = s_i \wedge O_1 \dots O_t)$$

In other words we are interested in the most likely path from 1 to t that:

1. Ends in S_i
2. Produces outputs $O_1 \dots O_t$

Q: Given $\delta_t(i)$, how can we compute $\delta_{t+1}(i)$?

A: To get from $\delta_t(i)$ to $\delta_{t+1}(i)$ we need to

1. Add an emission for time t+1 (O_{t+1})
2. Transition to state s_i

$$\begin{aligned}
\delta_{t+1}(i) &= \max_{q_1 \dots q_t} p(q_1 \dots q_t \wedge q_{t+1} = s_i \wedge O_1 \dots O_{t+1}) \\
&= \max_j \delta_t(j) p(q_{t+1} = s_i \mid q_t = s_j) p(O_{t+1} \mid q_{t+1} = s_i) \\
&= \max_j \delta_t(j) a_{j,i} b_i(O_{t+1})
\end{aligned}$$

Inference in HMM: the Viterbi algorithm

$$\delta_{t+1}(i) = \max_{q_1 \cdots q_t} p(q_1 \ldots q_t \wedge q_{t+1} = s_i \wedge O_1 \ldots O_{t+1})$$
$$= \max_j \delta_t(j) p(q_{t+1} = s_i \mid q_t = s_j) p(O_{t+1} \mid q_{t+1} = s_i)$$
$$= \max_j \delta_t(j) a_{j,i} b_i(O_{t+1})$$

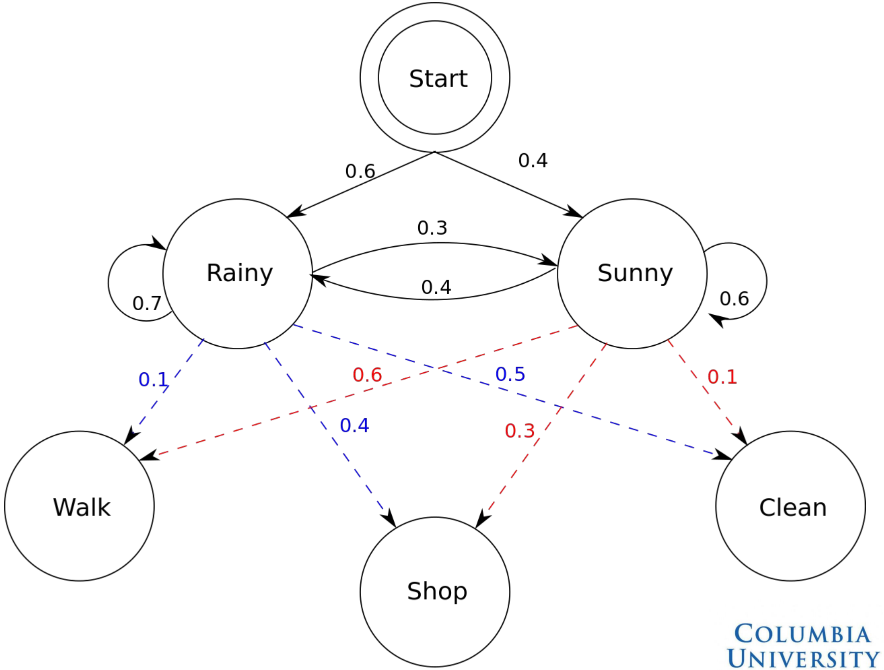
- Once again we use dynamic programming for solving $\delta_t(i)$
- Once we have $\delta_t(i)$, we can solve for our P(Q*|O)

By:

P(Q* | O) = argmax_QP(Q|O) = Path defined by max_i $\delta_t(i)$

Takeaways

- Why HMMs? Which applications are suitable?
- Inference in HMMs
  - No observations
  - Probability of next state with observations
  - Maximum scoring path (Viterbi)



References

- Christopher Bishop: Pattern Recognition and Machine Learning, Chapter 5
- Ziv Bar-Joseph, Tom Mitchell, Pradeep Ravikumar and Aarti Singh: CMU 10-701
- Ryan Tibshirani: CMU 10-725
- Ruslan Salakhutdinov: CMU 10-703
- https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/

COLUMBIA UNIVERSITY