CS 10:
Problem solving via Object Oriented Programming

Winter 2017

Tim Pierson
260 (255) Sudikoff

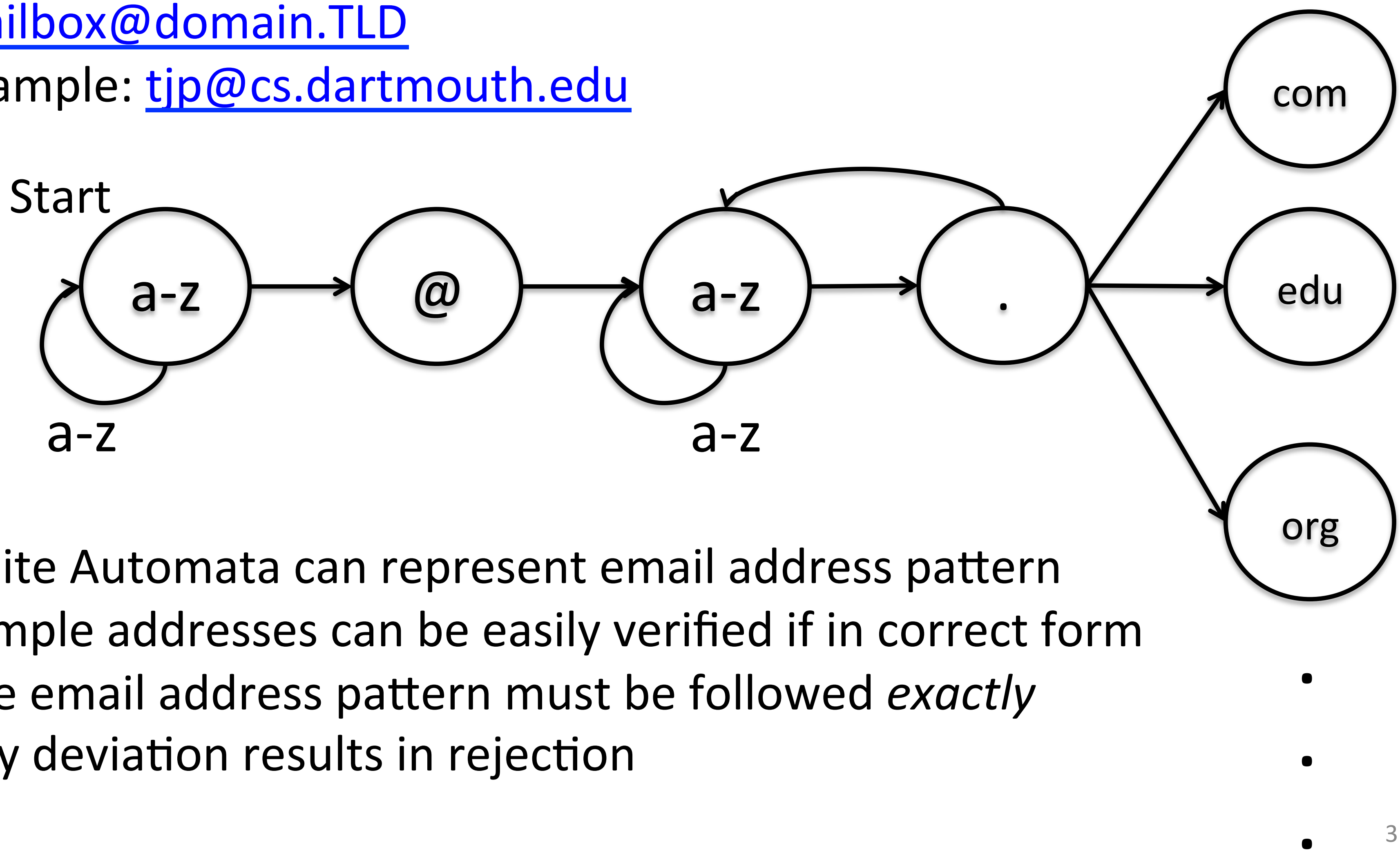
Day 20 – Pattern Recognition

Agenda

1. Pattern matching vs. recognition
2. From Finite Automata to Hidden Markov Models
3. Decoding: Viterbi algorithm
4. Training

Last class we discussed how to use a Finite Automata to match a pattern

Email addresses follow a pattern:
mailbox@domain.TLD
example: tjp@cs.dartmouth.edu



Finite Automata can represent email address pattern
Sample addresses can be easily verified if in correct form
The email address pattern must be followed *exactly*
Any deviation results in rejection

Sometimes our input is noisy and does not exactly match a pattern

Pattern matching vs. recognition

Is this a duck?

Matching	Recognition

Image: duckrace.com

Sometimes our input is noisy and does not exactly match a pattern

Pattern matching vs. recognition

Is this a duck?

	Matching	**Recognition**
Looks like a duck	✓	✓

Image: duckrace.com

Sometimes our input is noisy and does not exactly match a pattern

Pattern matching vs. recognition

Is this a duck?

	Matching	Recognition
Looks like a duck	✓	✓
Quacks like a duck	✓	✓

Image: duckrace.com

Sometimes our input is noisy and does not exactly match a pattern

Pattern matching vs. recognition

Is this a duck?

	Matching	Recognition
Looks like a duck	✓	✓
Quacks like a duck	✓	✓
Does not wear cool eyewear	✗	✗

Image: duckrace.com

Sometimes our input is noisy and does not exactly match a pattern

Pattern matching vs. recognition

Is this a duck?

	Matching	Recognition
Looks like a duck	✓	✓
Quacks like a duck	✓	✓
Does not wear cool eyewear	✗	✗
Is it a duck?	✗	✓

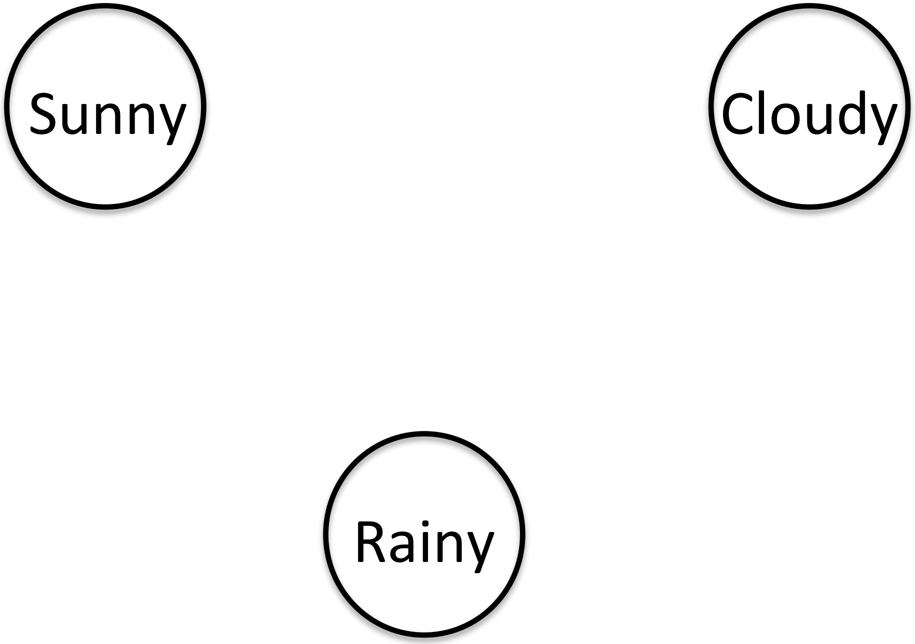
Image: duckrace.com

Agenda

1. Pattern matching vs. recognition
2. From Finite Automata to Hidden Markov Models
3. Decoding: Viterbi algorithm
4. Training

We can model systems using Finite Automata

Weather model: possible states



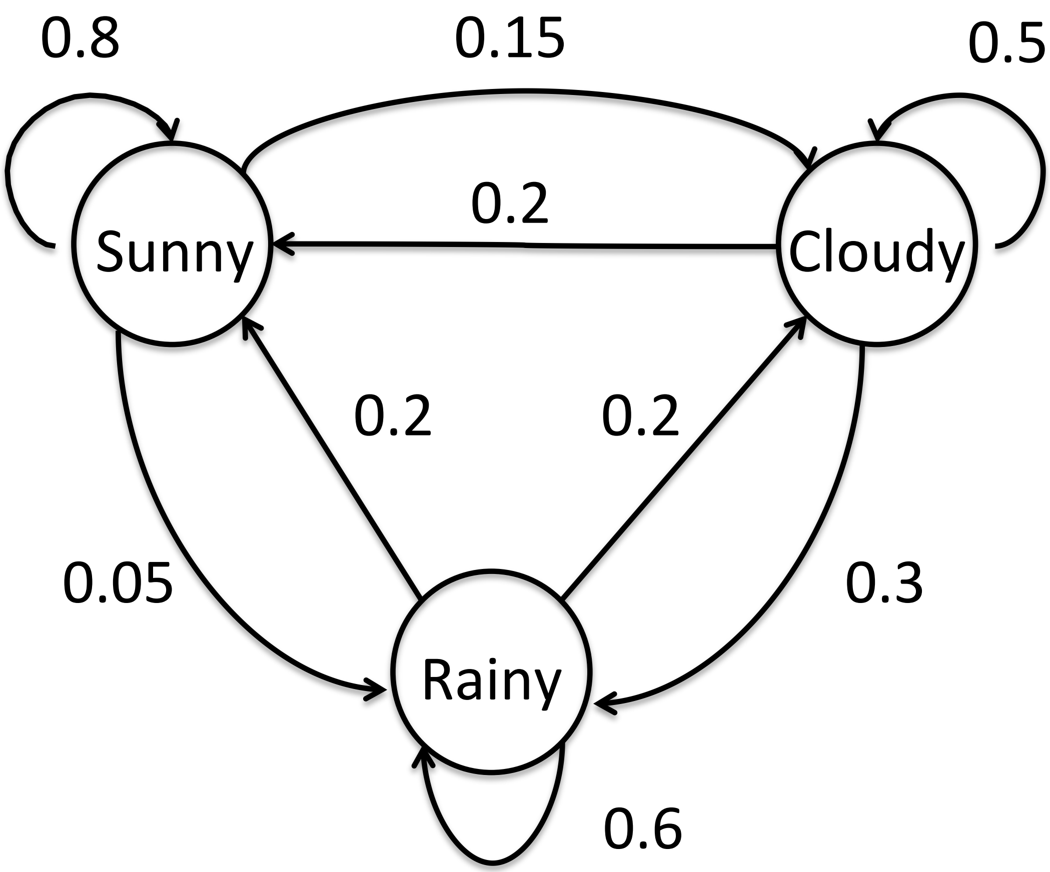
The **State** of the weather can be:

- Sunny
- Cloudy
- Rainy

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

We can model systems using Finite Automata

Weather model: transitions

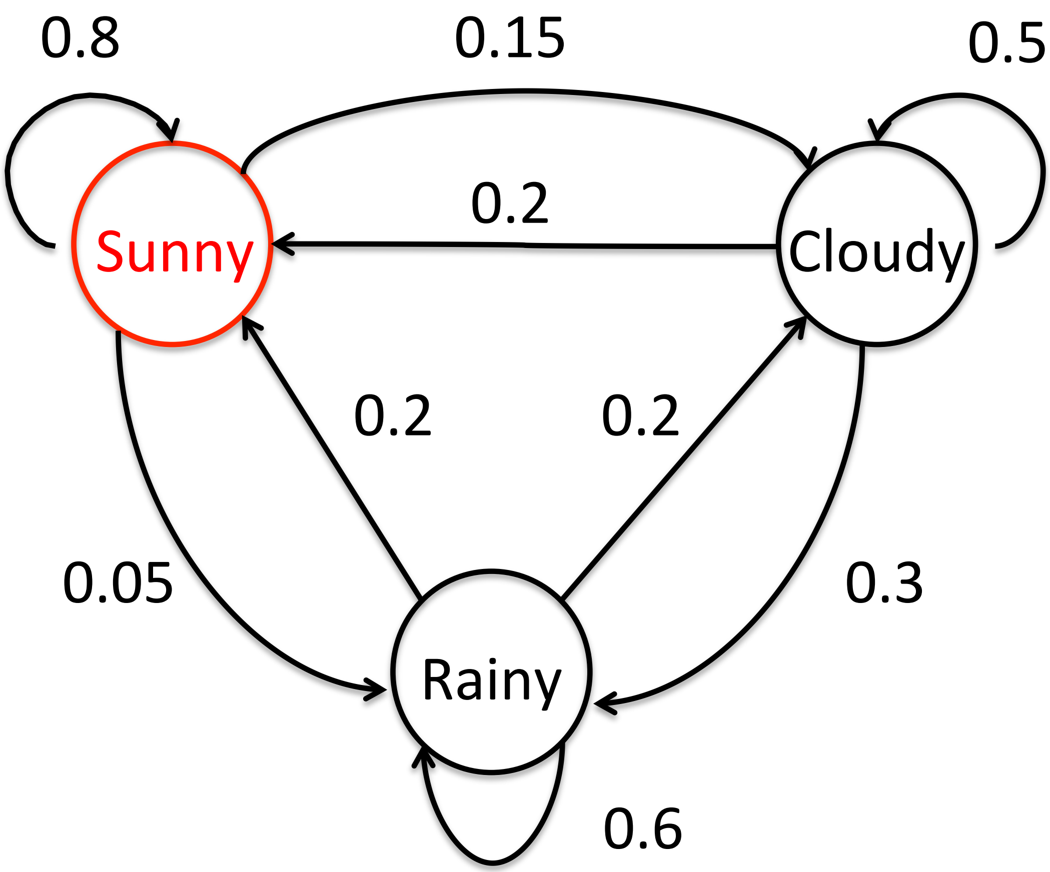


We can observe weather patterns and determine probability of **transition** between states

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

We can model systems using Finite Automata

Weather model: Sunny day example

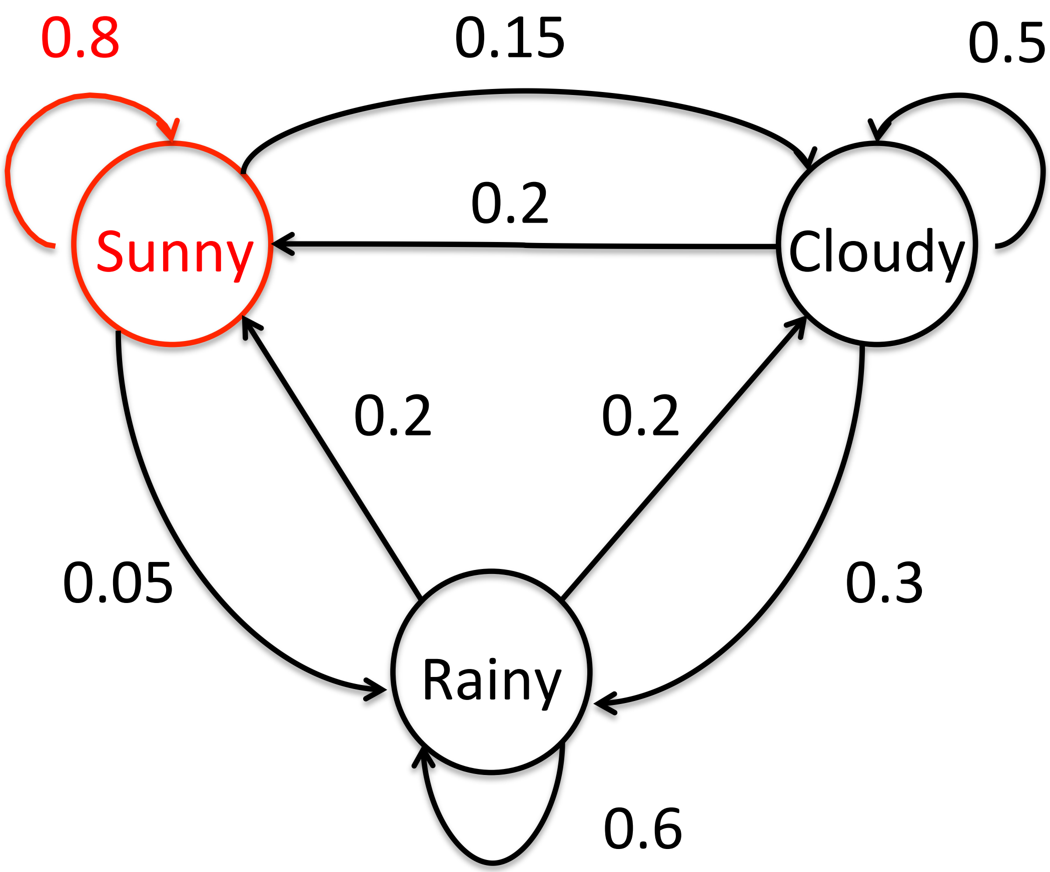


Probability a sunny day is followed by:

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

We can model systems using Finite Automata

Weather model: Sunny day example



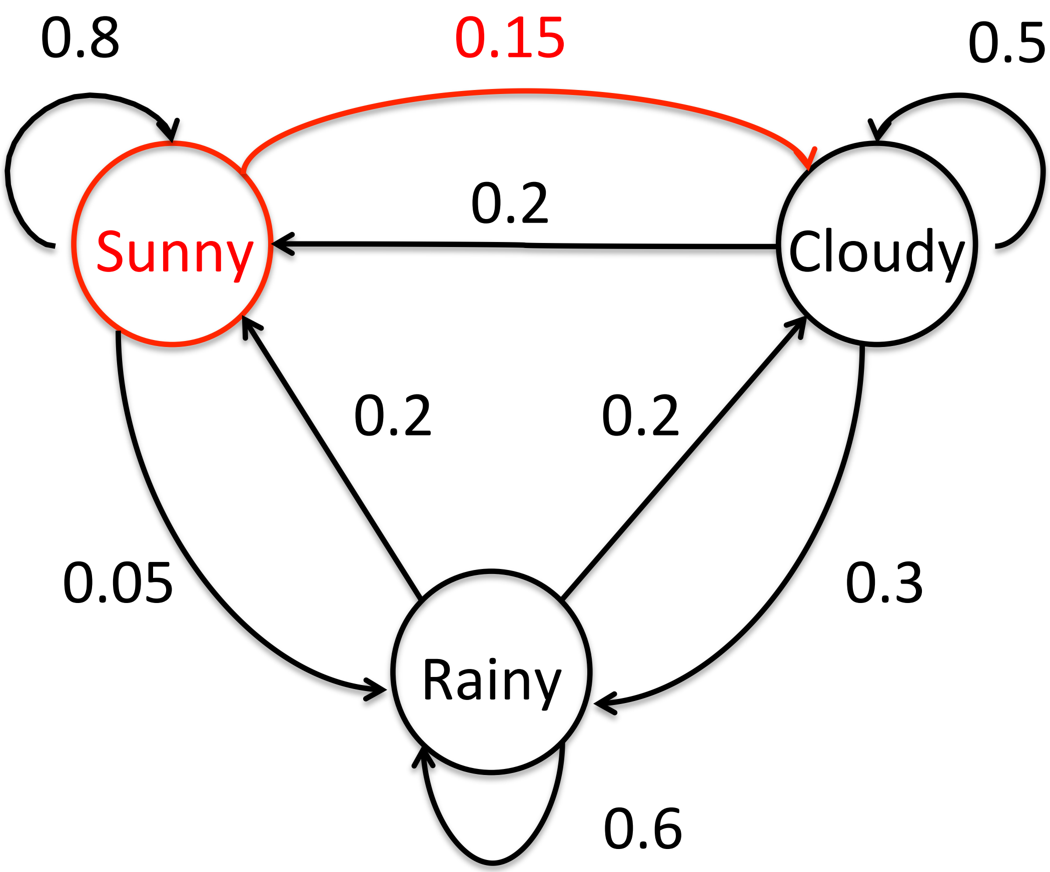
Probability a sunny day is followed by:

- Another sunny day 80%

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

We can model systems using Finite Automata

Weather model: Sunny day example



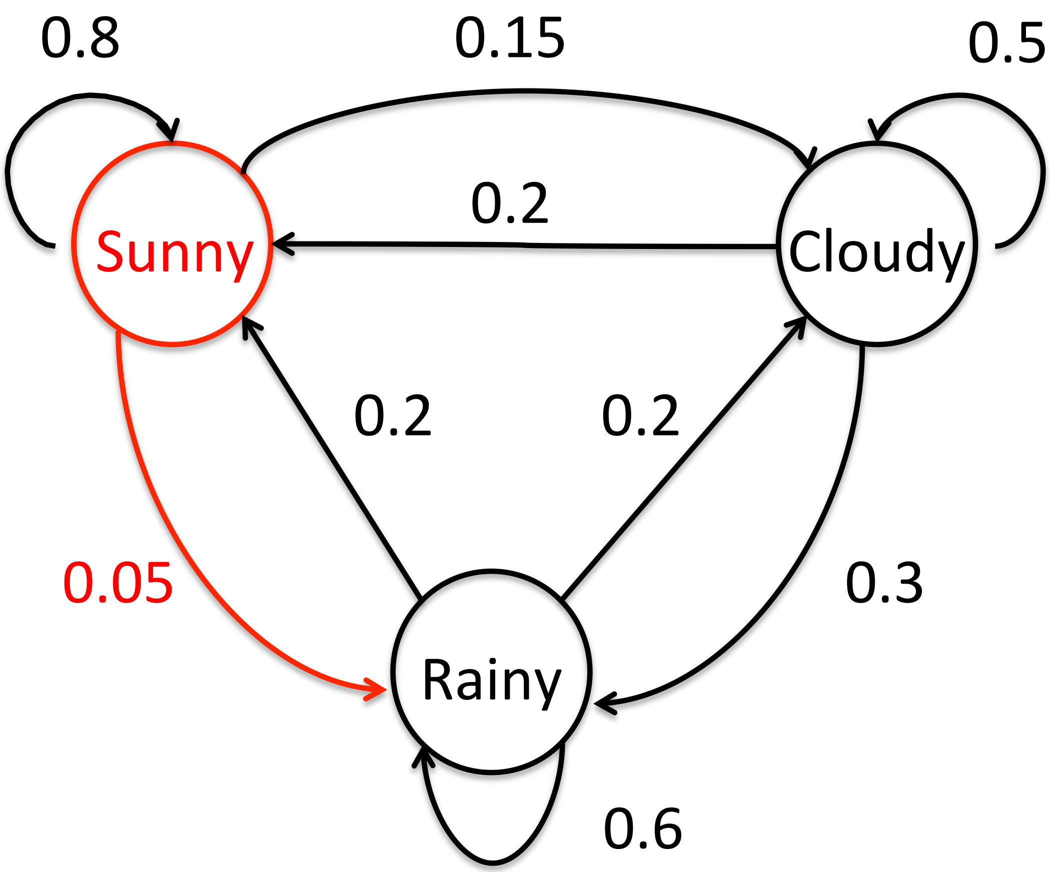
Probability a sunny day is followed by:

- Another sunny day 80%
- A cloudy day 15%

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

We can model systems using Finite Automata

Weather model: Sunny day example



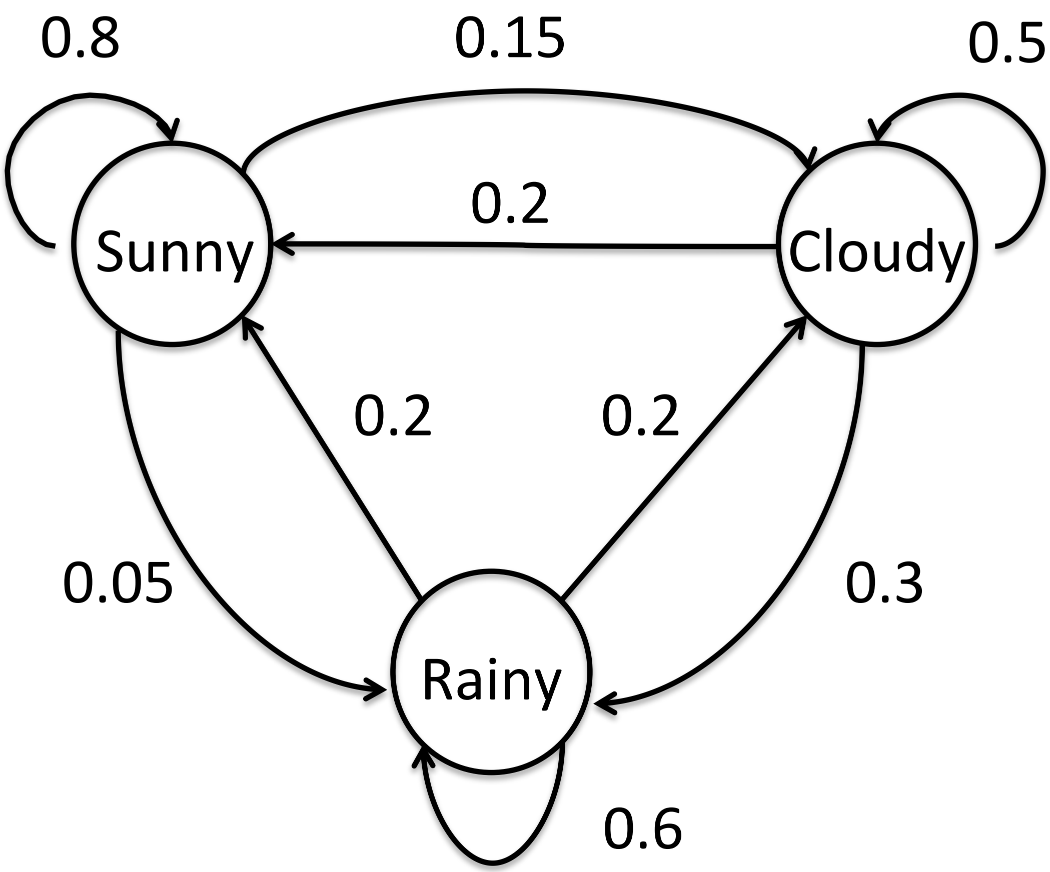
Probability a sunny day is followed by:

- Another sunny day 80%
- A cloudy day 15%
- A rainy day 5%

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

Model allows us to answer questions about the probability of events occurring

Weather model: predict two days in advance

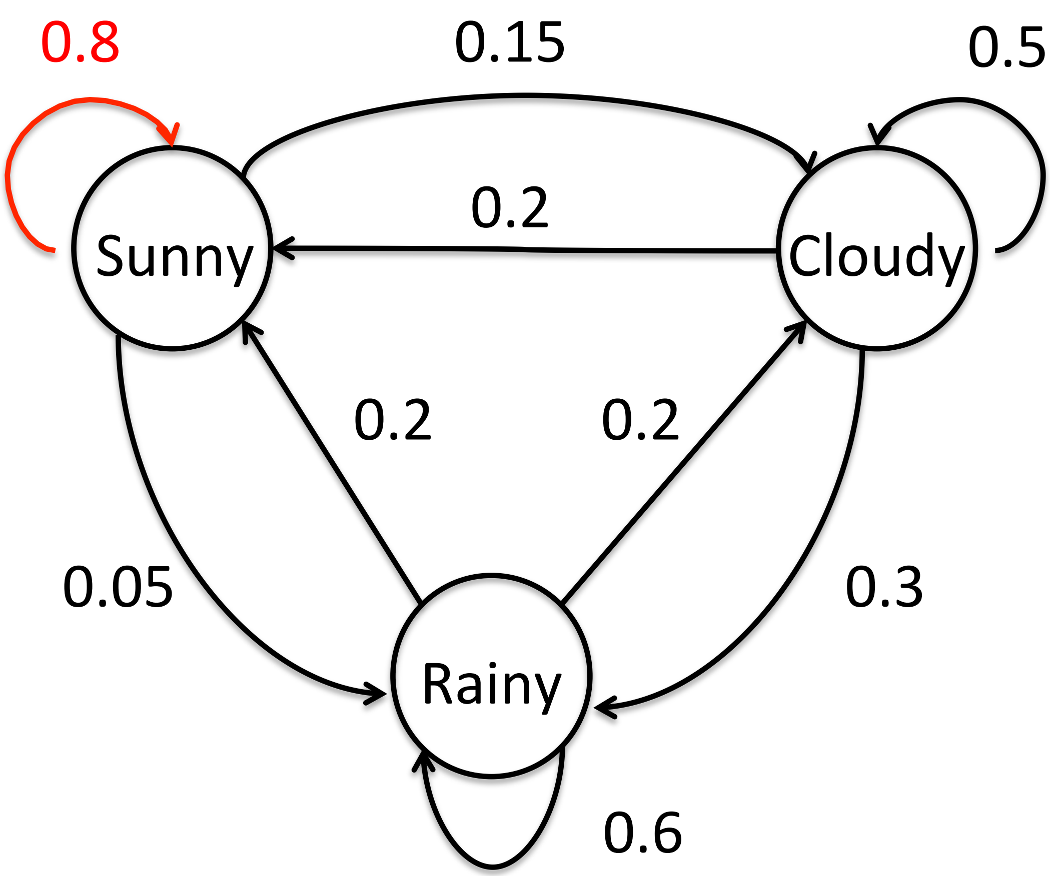


Given today is sunny, what is the probability it will be rainy two days from now?

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

FA model allows us to answer questions about the probability of events occurring

Weather model: predict two days in advance



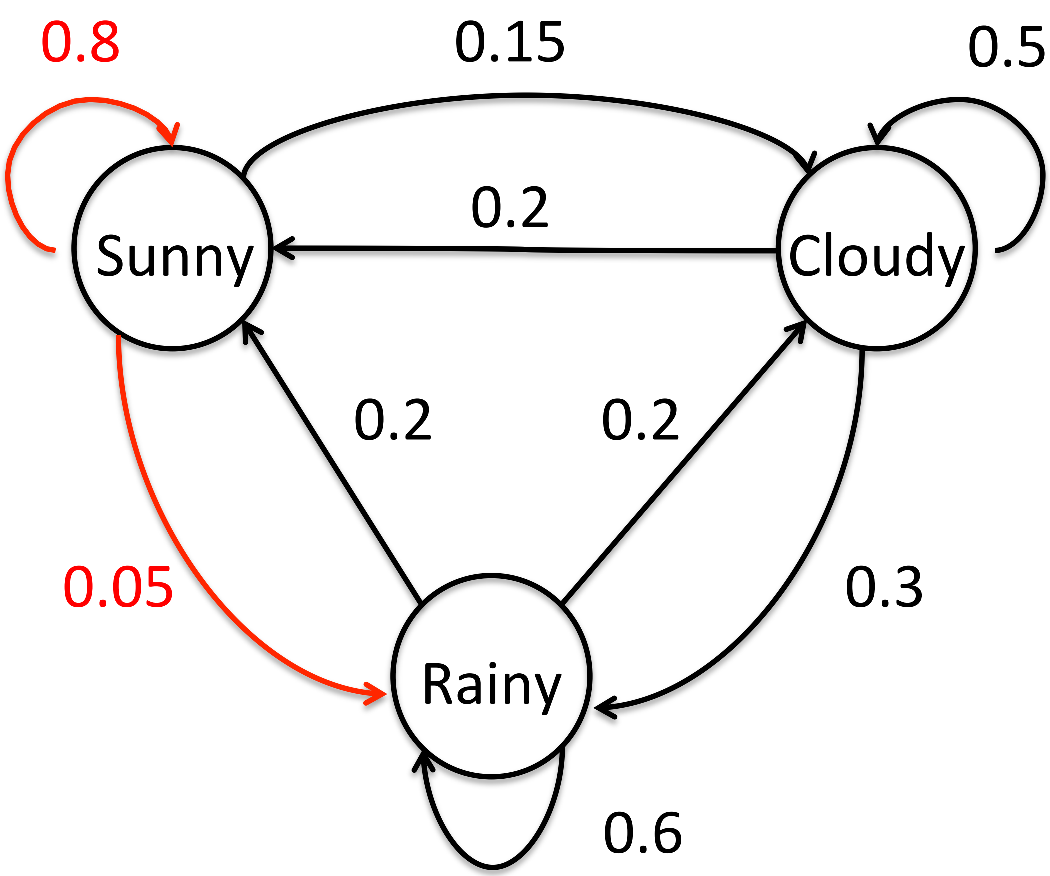
Given today is sunny, what is the probability it will be rainy two days from now?

- Could be sunny, then rainy

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

FA model allows us to answer questions about the probability of events occurring

Weather model: predict two days in advance



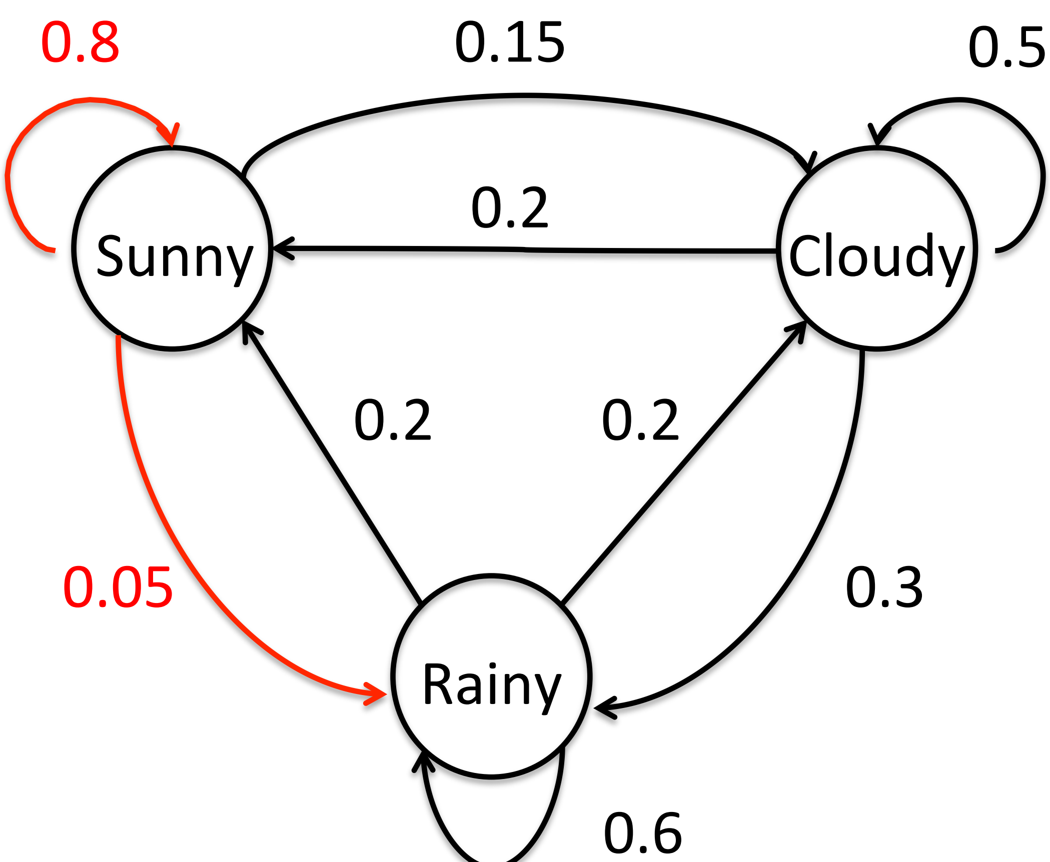
Given today is sunny, what is the probability it will be rainy two days from now?

- Could be sunny, then rainy

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

FA model allows us to answer questions about the probability of events occurring

Weather model: predict two days in advance



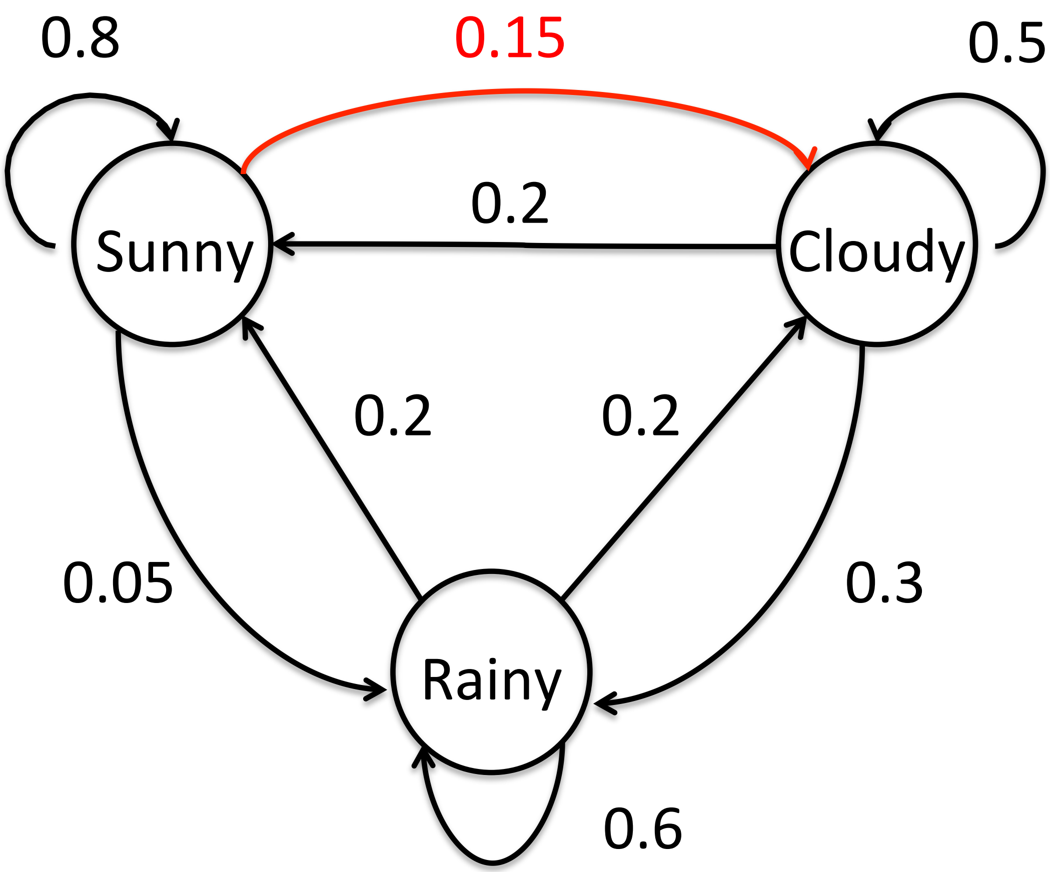
Given today is sunny, what is the probability it will be rainy two days from now?

- Could be sunny, then rainy (0.8*0.05)

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

FA model allows us to answer questions about the probability of events occurring

Weather model: predict two days in advance



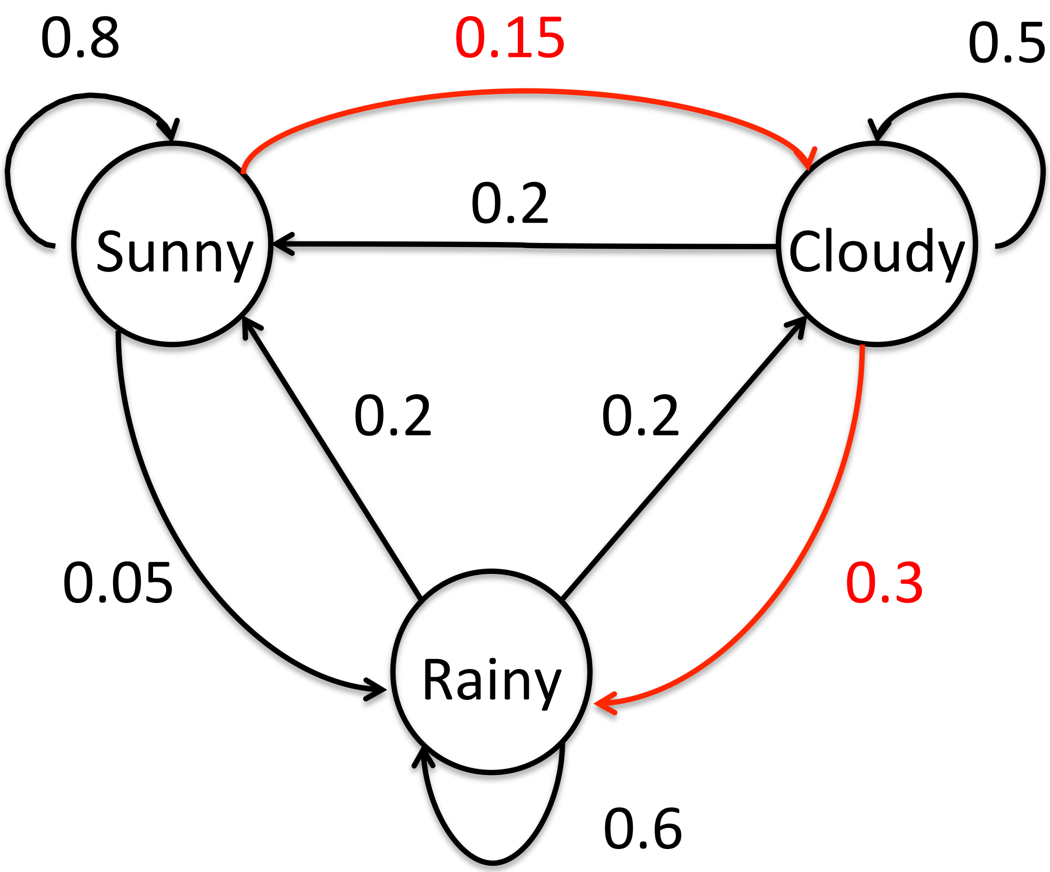
Given today is sunny, what is the probability it will be rainy two days from now?

- Could be sunny, then rainy (0.8*0.05)
- Could be cloudy, then rainy

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

FA model allows us to answer questions about the probability of events occurring

Weather model: predict two days in advance



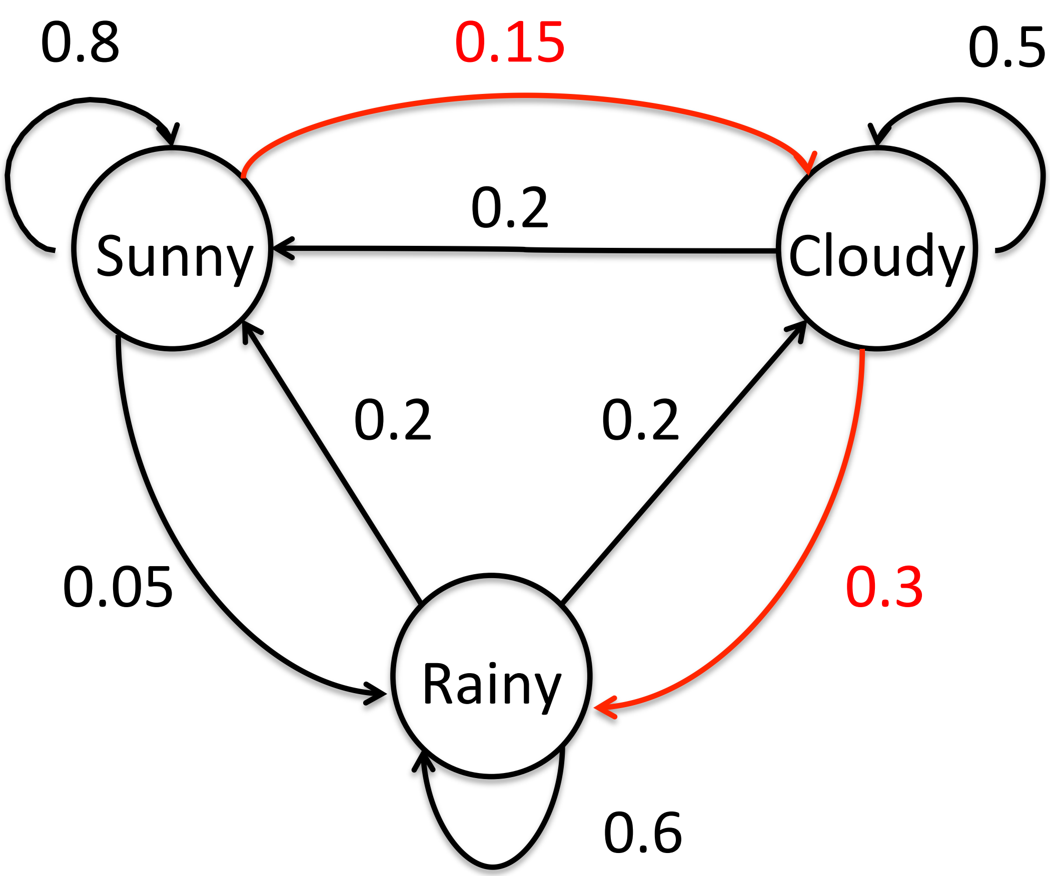
Given today is sunny, what is the probability it will be rainy two days from now?

- Could be sunny, then rainy (0.8*0.05)
- Could be cloudy, then rainy

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

FA model allows us to answer questions about the probability of events occurring

Weather model: predict two days in advance



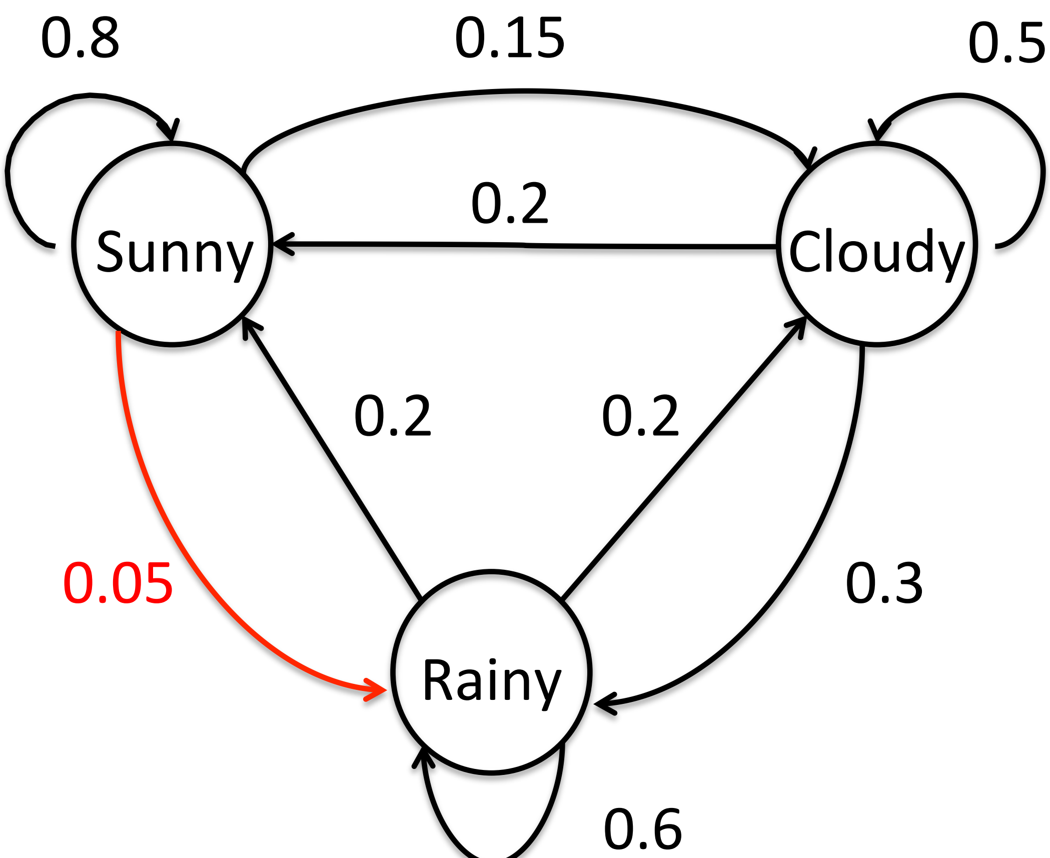
Given today is sunny, what is the probability it will be rainy two days from now?

- Could be sunny, then rainy (0.8*0.05)
- Could be cloudy, then rainy (0.15*0.3)

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

FA model allows us to answer questions about the probability of events occurring

Weather model: predict two days in advance



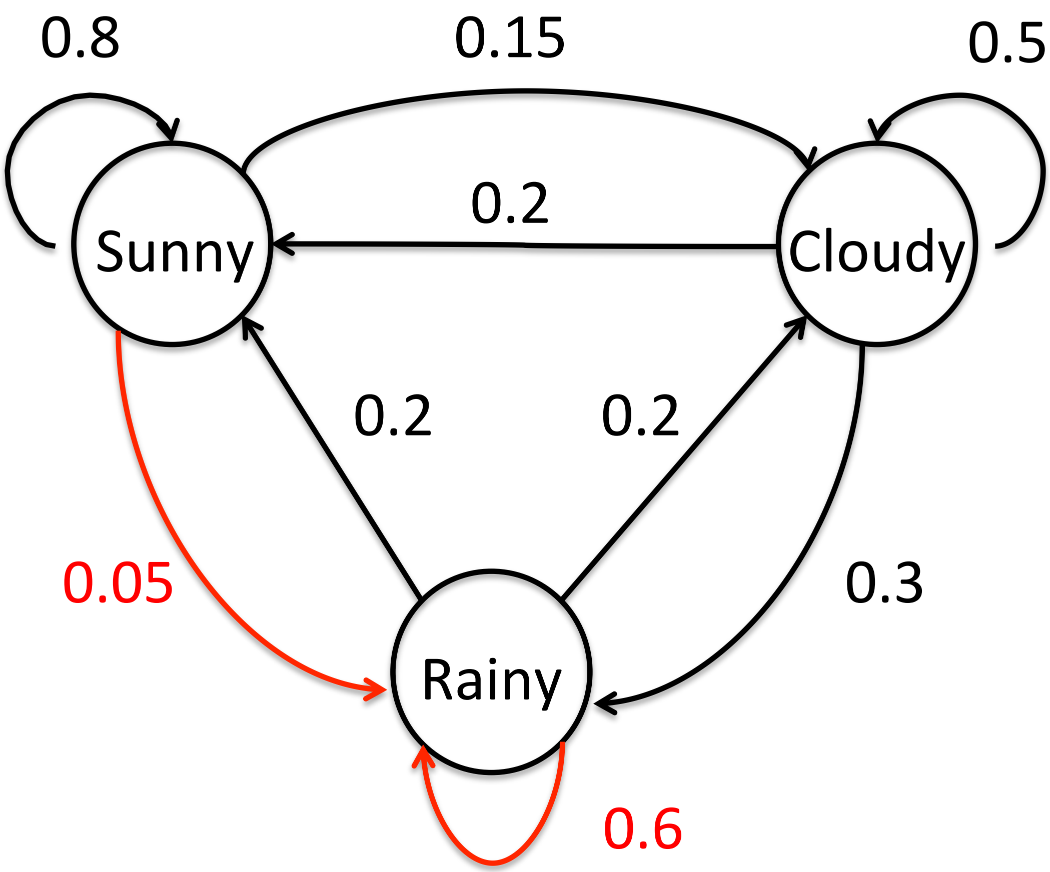
Given today is sunny, what is the probability it will be rainy two days from now?

- Could be sunny, then rainy (0.8*0.05)
- Could be cloudy, then rainy (0.15*0.3)
- Could be rainy, then rainy

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

FA model allows us to answer questions about the probability of events occurring

Weather model: predict two days in advance



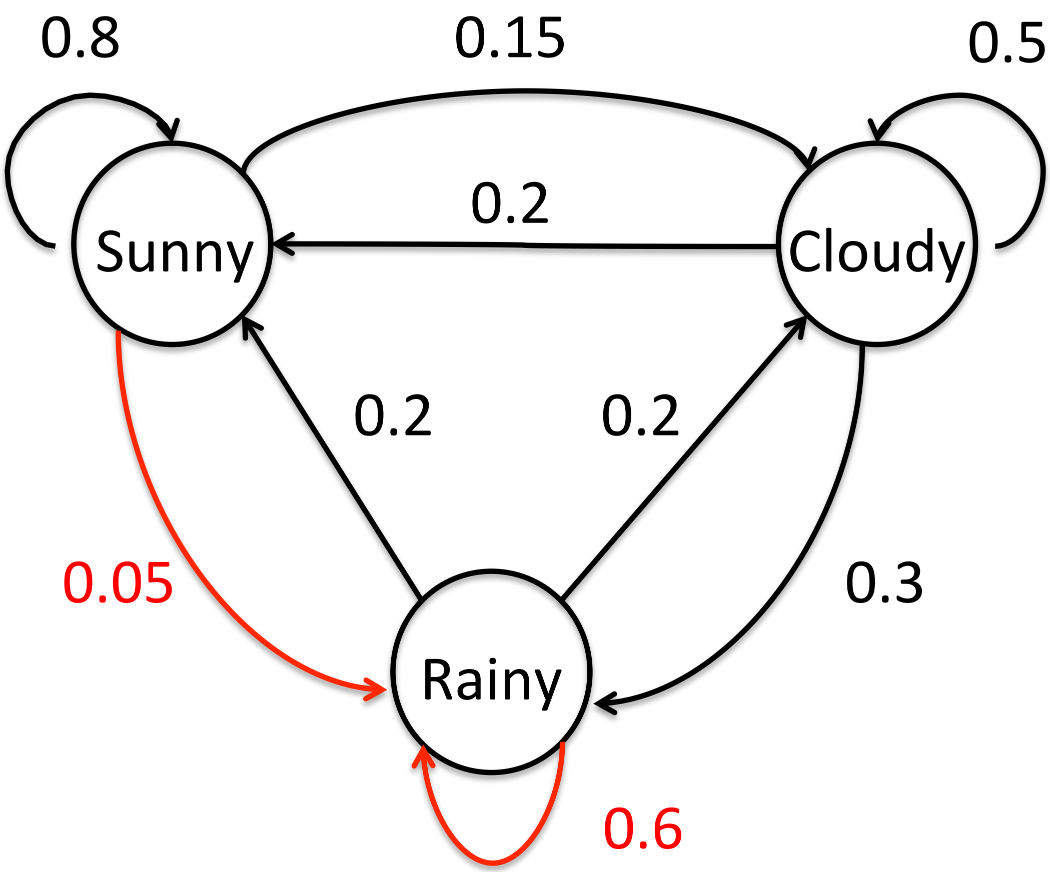
Given today is sunny, what is the probability it will be rainy two days from now?

- Could be sunny, then rainy (0.8*0.05)
- Could be cloudy, then rainy (0.15*0.3)
- Could be rainy, then rainy

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

FA model allows us to answer questions about the probability of events occurring

Weather model: predict two days in advance



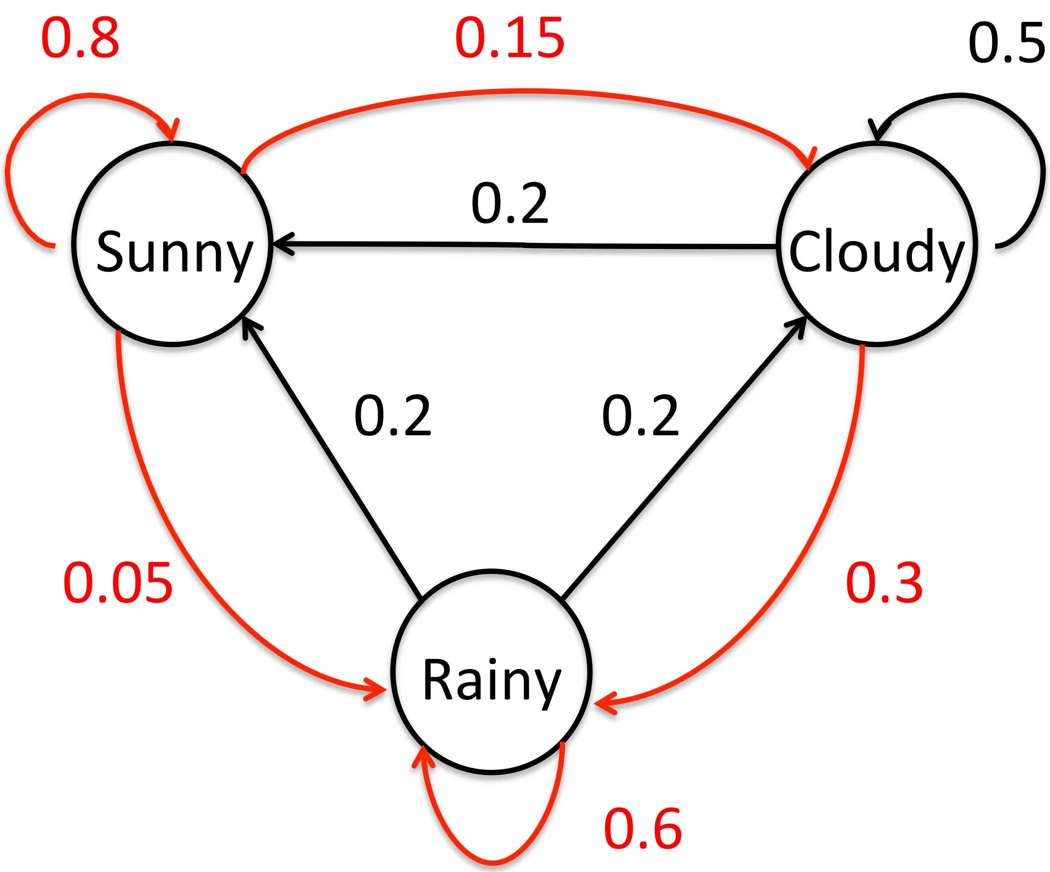
Given today is sunny, what is the probability it will be rainy two days from now?

- Could be sunny, then rainy (0.8*0.05)
- Could be cloudy, then rainy (0.15*0.3)
- Could be rainy, then rainy (0.05*0.6)

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

FA model allows us to answer questions about the probability of events occurring

Weather model: predict two days in advance



Given today is sunny, what is the probability it will be rainy two days from now?

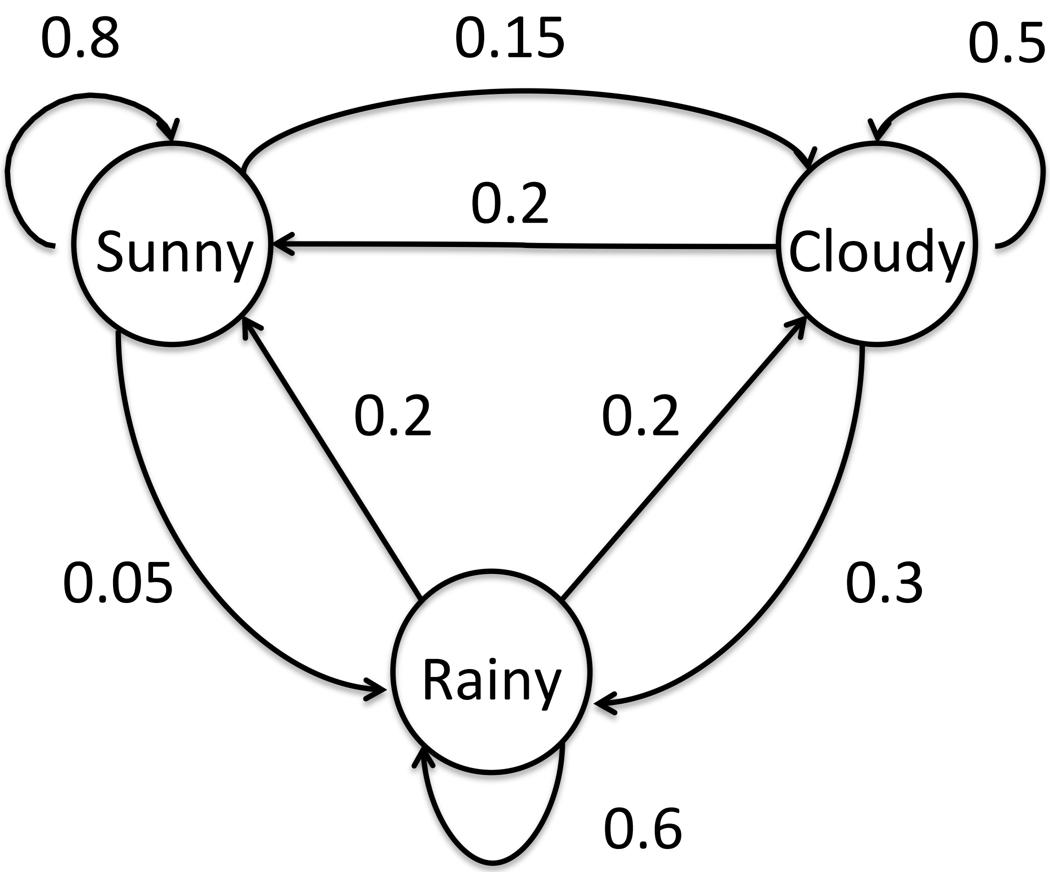
- Could be sunny, then rainy (0.8*0.05)
- Could be cloudy, then rainy (0.15*0.3)
- Could be rainy, then rainy (0.05*0.6)

Total = (0.8*0.05) + (0.15*0.3) + (0.05*0.6) = 0.115

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

Markov property suggests it doesn't really matter how we got into the current State

Given current State, can predict likelihood of future states



Given that we can observe the state we are in, it doesn't really matter how we got there:

- Probability of weather at time n, given the weather at time n-1, and at n-2, …
- Is approximately equal to the probability of weather at time n given *only* the weather at n-1
- $P(w_n | w_{n-1}, w_{n-2}, w_{n-3}) \approx P(w_n | w_{n-1})$

Adapted from: https://pdfs.semanticscholar.org/b328/2eb0509442b80760fea5845e158168daee62.pdf

Model works well if we can directly observe the state, what if we cannot?

Sometimes we cannot directly observe the state

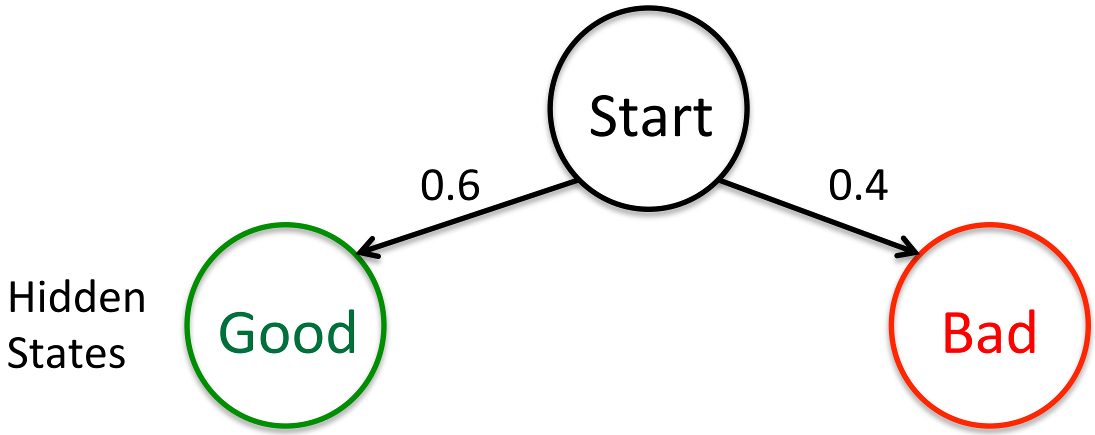
- You're being held prisoner and want to know the weather outside. You can't see outside, but you can observe if the guard brings an umbrella.

- You observe photos of your friends. You don't know what city they were in, but do know something about the cities. Can you guess what cities they visited?

- You want to ask for a raise, but only if the boss is in a good mood. How can you tell if the boss is in a good mood if you can't tell by looking?

Want to ask the boss for raise when the boss's state is a Good mood

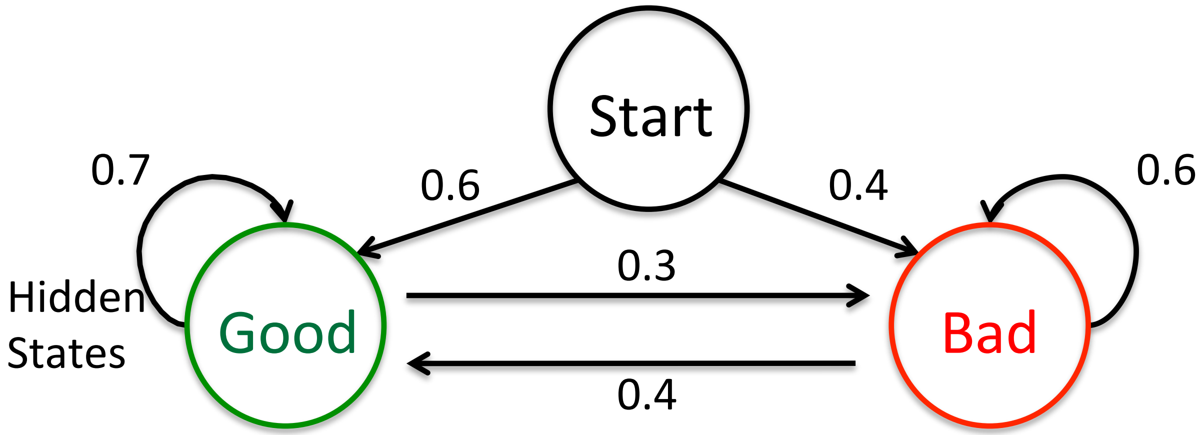
Gather stats about likelihood of states



- Can't know boss's mood for sure simply by looking (state is hidden)
- Want to know current state (Good or Bad)
- Could ask everyday and record statistics about it
- Assume boss answers truthfully
  - Ask 100 times
  - 60 times good
  - 40 times bad
- Boss slightly more likely to be in good mood (60% chance)

In addition to states, find likelihood of *transitioning* from one state to another

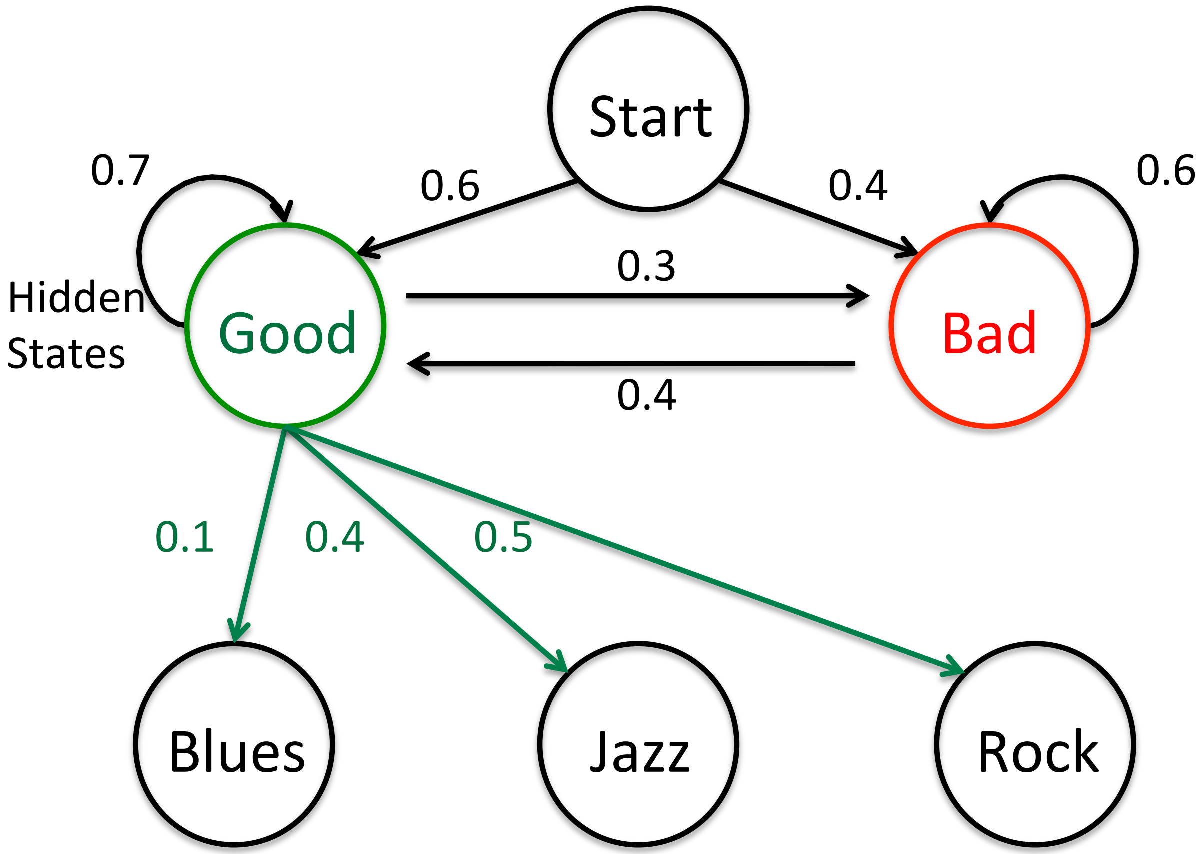
Gather stats about state transitions



- Watch boss on day after asking about mood, ask again next day
- Calculate probability of staying in same mood or transitioning to another mood (hidden state)
- Similar to how weather transitioned states

Once have states and transitions, might find something we *can* directly observe

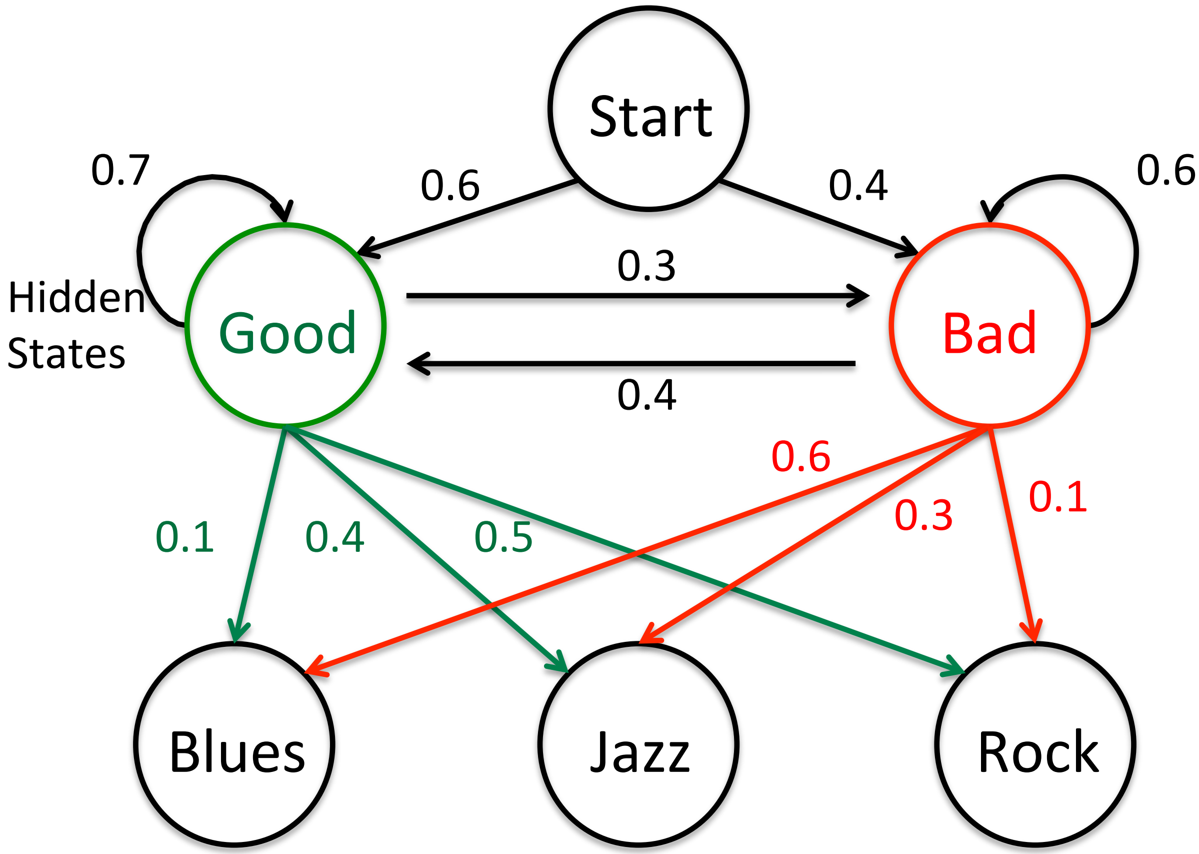
Might be able to observe music playing



- Might observe what music the boss plays
- Blues, Jazz or Rock
- Record stats about music choice when in either mood (hidden states)

Once have states and transitions, might find something we *can* directly observe

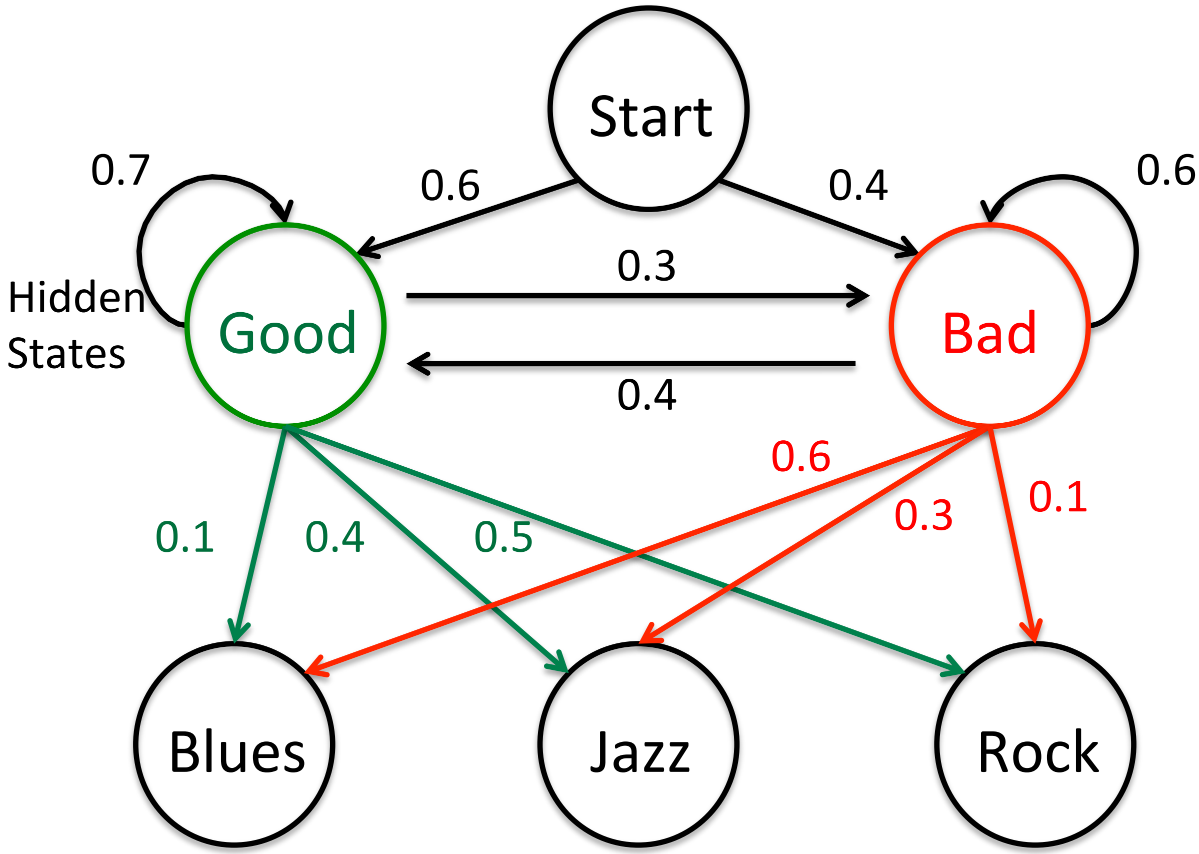
Might be able to observe music playing



- Might observe what music the boss plays
- Blues, Jazz or Rock
- Record stats about music choice when in either mood (hidden states)

This is a Hidden Markov Model (HMM)

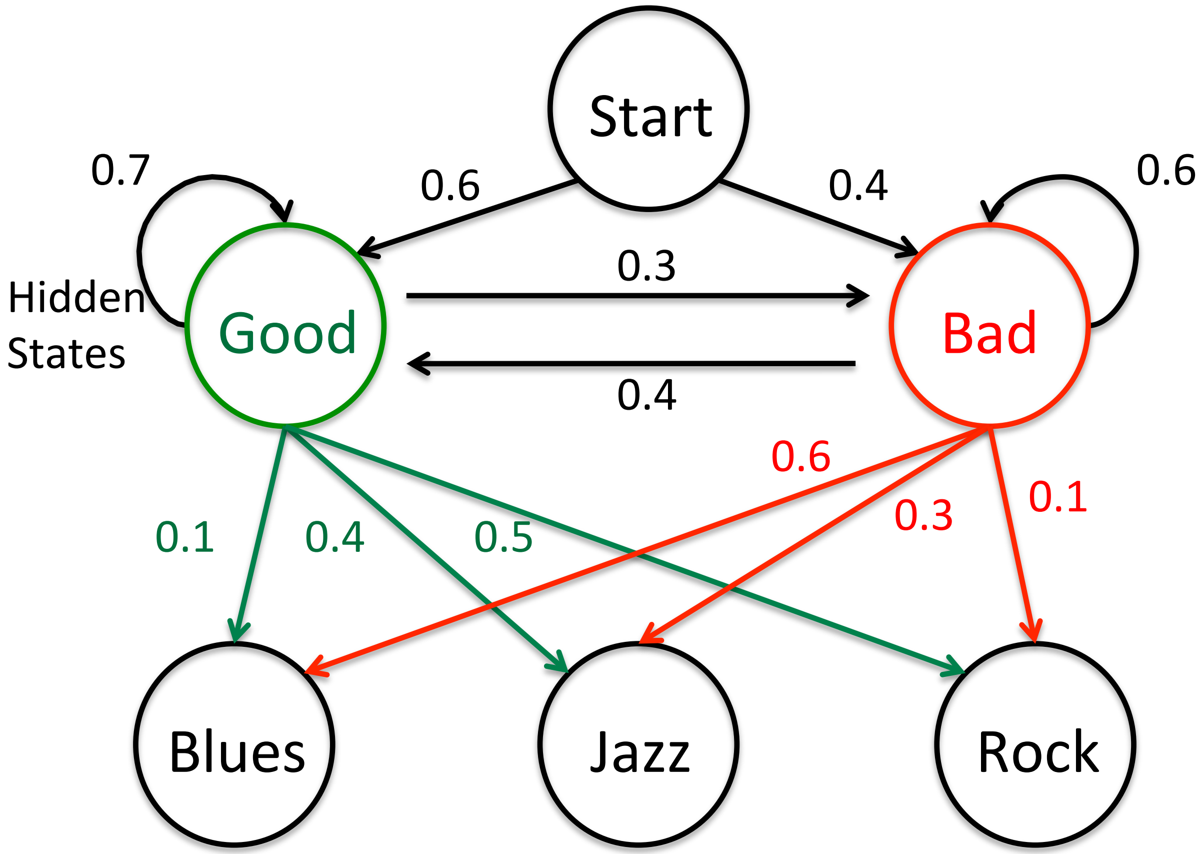
Hidden Markov Model



- States (boss's mood) are hidden, can't be directly observed
- But, we *can* observe something (music) that can help us calculate the most *likely* hidden state

So is today a good day to ask for a raise?

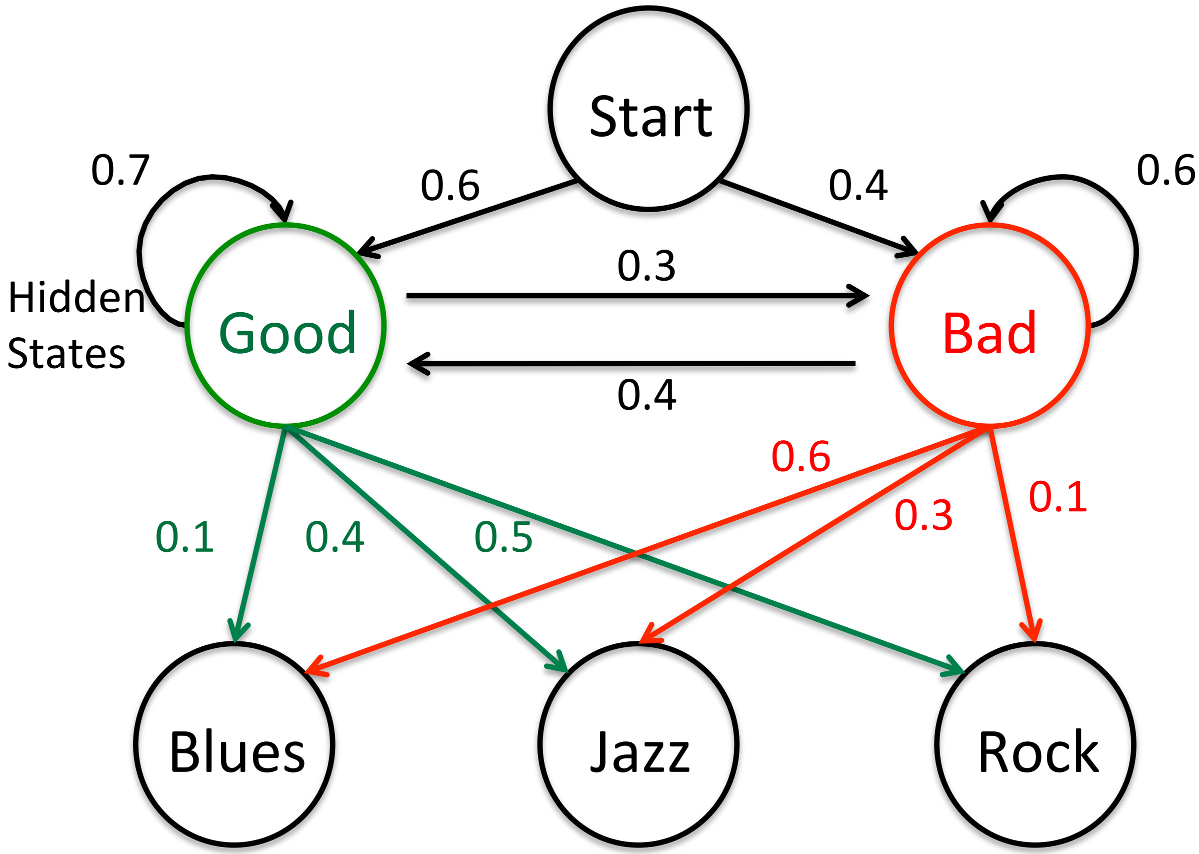
So far we have no music observation



- Given no other information, it's a pretty good bet the boss in Good mood
- Good mood = 0.6
- Bad mood = 0.4
- Yes, on any given day boss is slightly more likely to be in a good mood

By observing music, we might be able to get a better sense of the boss's mood!

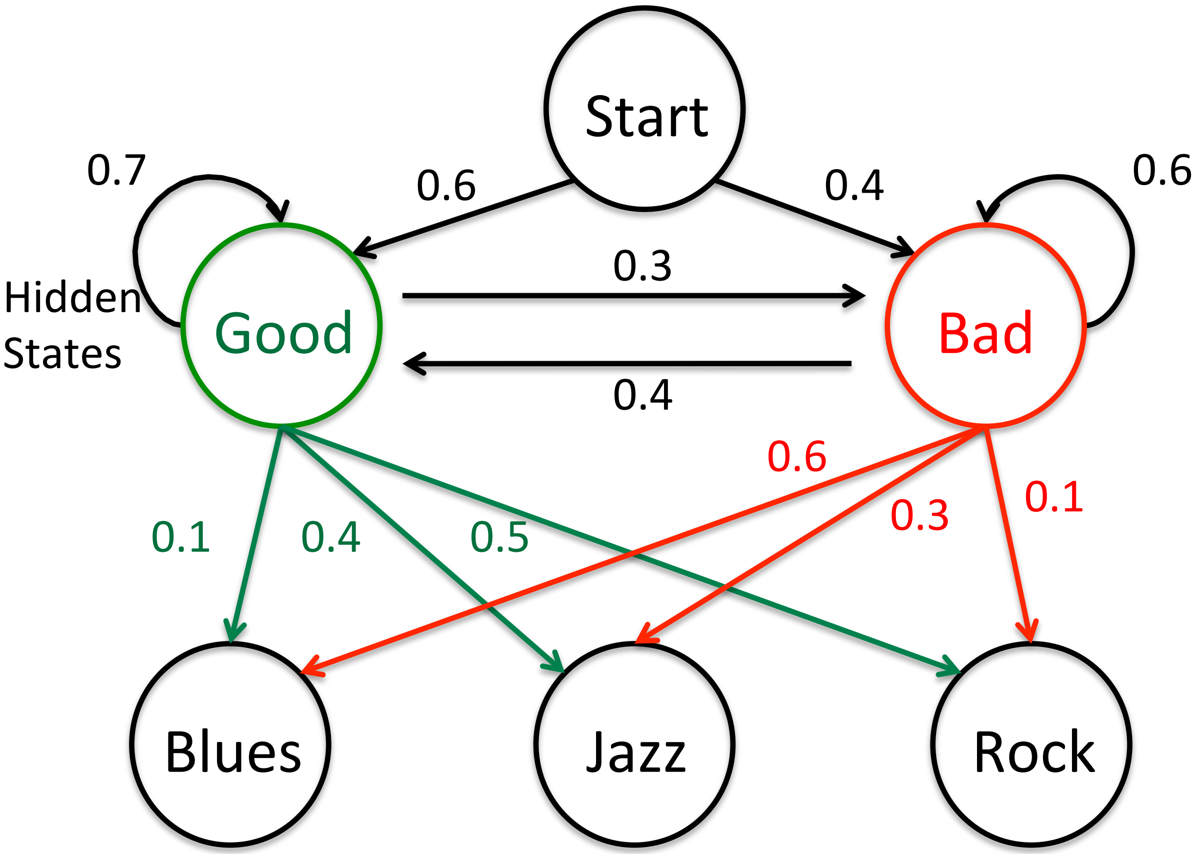
Observe Rock music



- Say today we observe the boss is playing Rock music
- Should we ask for a raise?
- Good mood = 0.6*0.5 = 0.3
- Bad mood = 0.4*0.1 = 0.04
- Most likely a good day to ask!

Bayes theorem can give us the actual probabilities of each hidden state

Observe Rock music



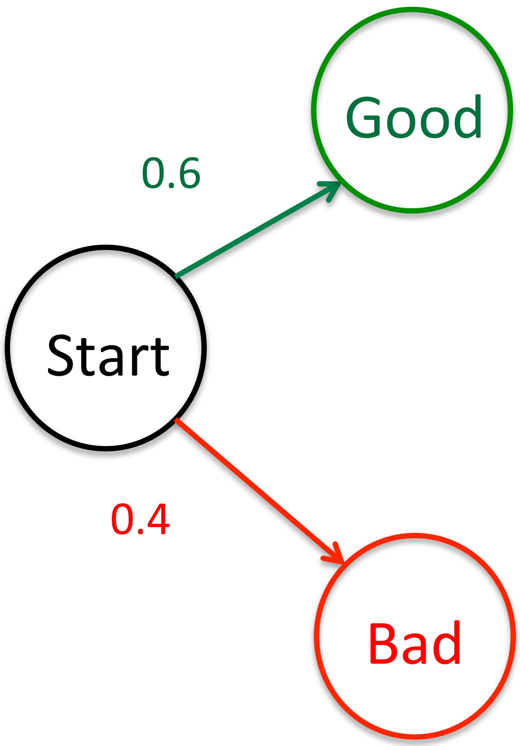
G=Good, B=Bad, R=Rock

- Given the boss is playing Rock music, use Bayes Theorem
- $P(A|B) = \frac{P(B|A)*P(A)}{P(B)}$
- $P(G|R) = \frac{P(R|G)*P(G)}{P(R)}$
- $P(R|G) = 0.5$, $P(G) = 0.6$, $P(R) = 0.34$
- $P(G|R) = 0.5*0.6/0.34 = 0.88$

Agenda

1. Pattern matching vs. recognition
2. From Finite Automata to Hidden Markov Models
3. Decoding: Viterbi algorithm
4. Training

We can estimate the most likely hidden state based on observations

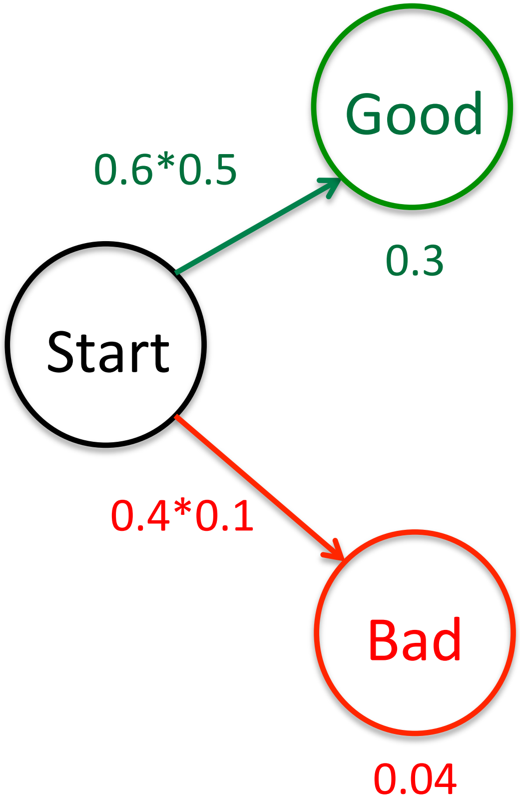


Given no observations, can make a guess at true state

Guess state with highest score

We can estimate the most likely hidden state based on observations

Day 1: Observe Rock



If we make an observation, we might be able to increase our accuracy

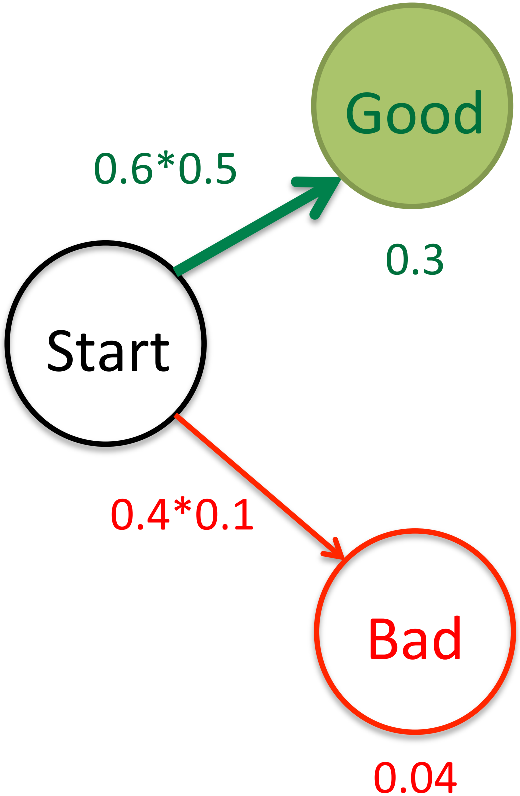
Multiply previous score by likelihood of observation

Most likely in a Good mood (~8X more likely)

Should ask for a raise

We can estimate the most likely hidden state based on observations

Day 1:
Observe
Rock



If we make an observation, we might be able to increase our accuracy

Multiply previous score by likelihood of observation

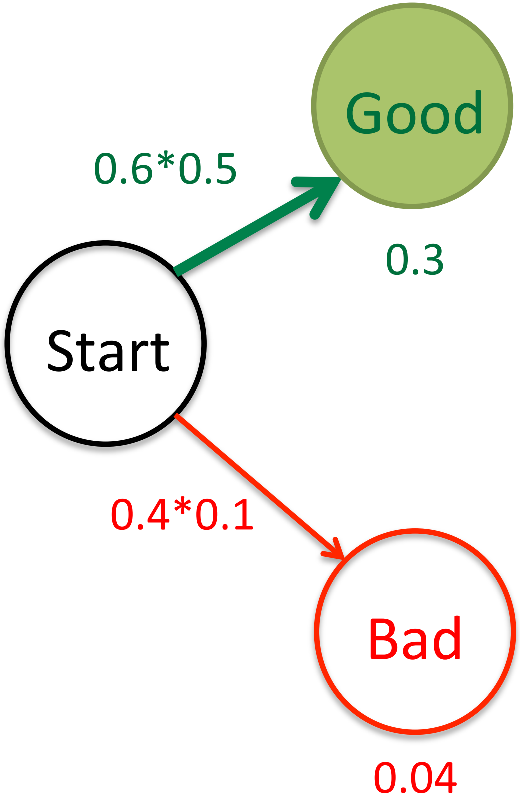
Most likely in a Good mood (~8X more likely)

Should ask for a raise

We can estimate the most likely hidden state based on observations

Day 1:
Observe
Rock

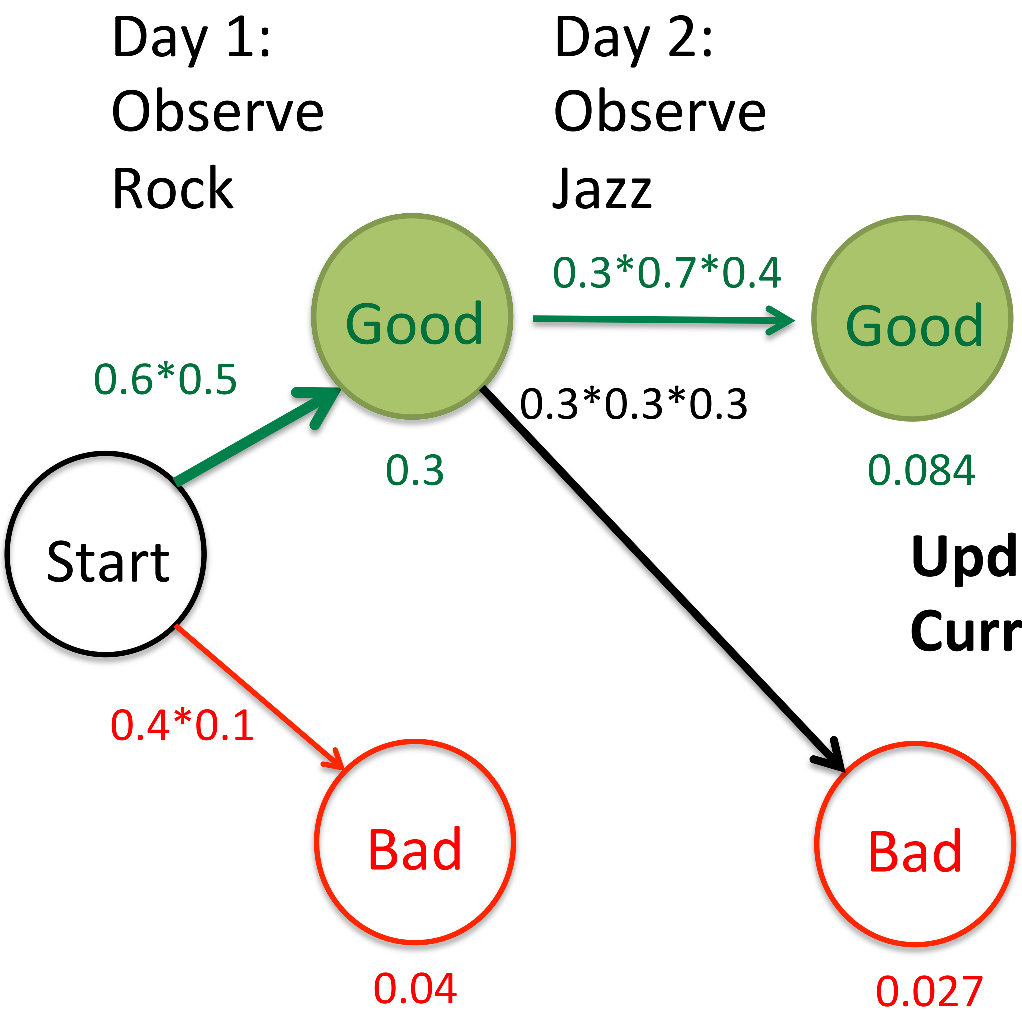
Day 2:
Observe
Jazz



What if you chicken out and don't ask today?

Tomorrow you observe Jazz, should you ask now?

We can estimate the most likely hidden state based on observations

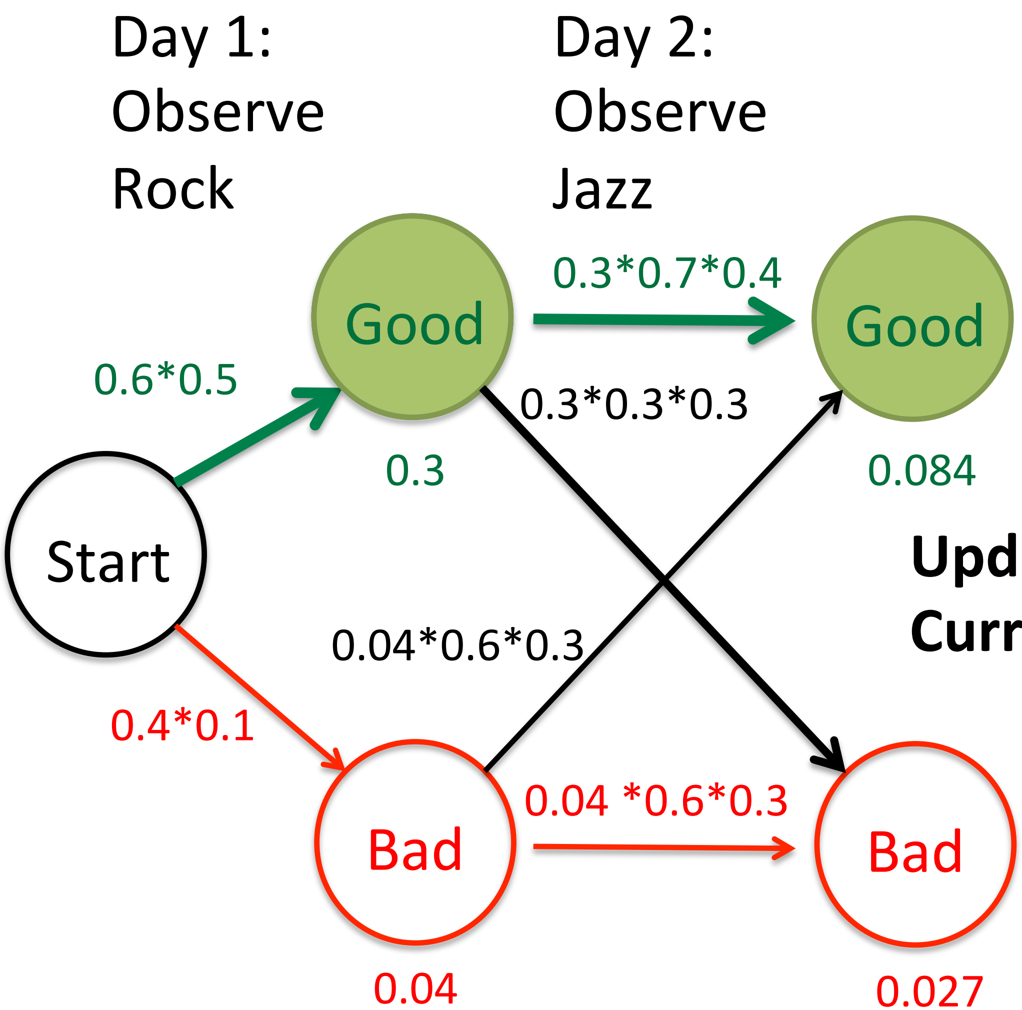


Update rule:
Current* Transition* Observation

Most likely state has highest value

Now only about 3X more likely to be in Good mood

We can estimate the most likely hidden state based on observations

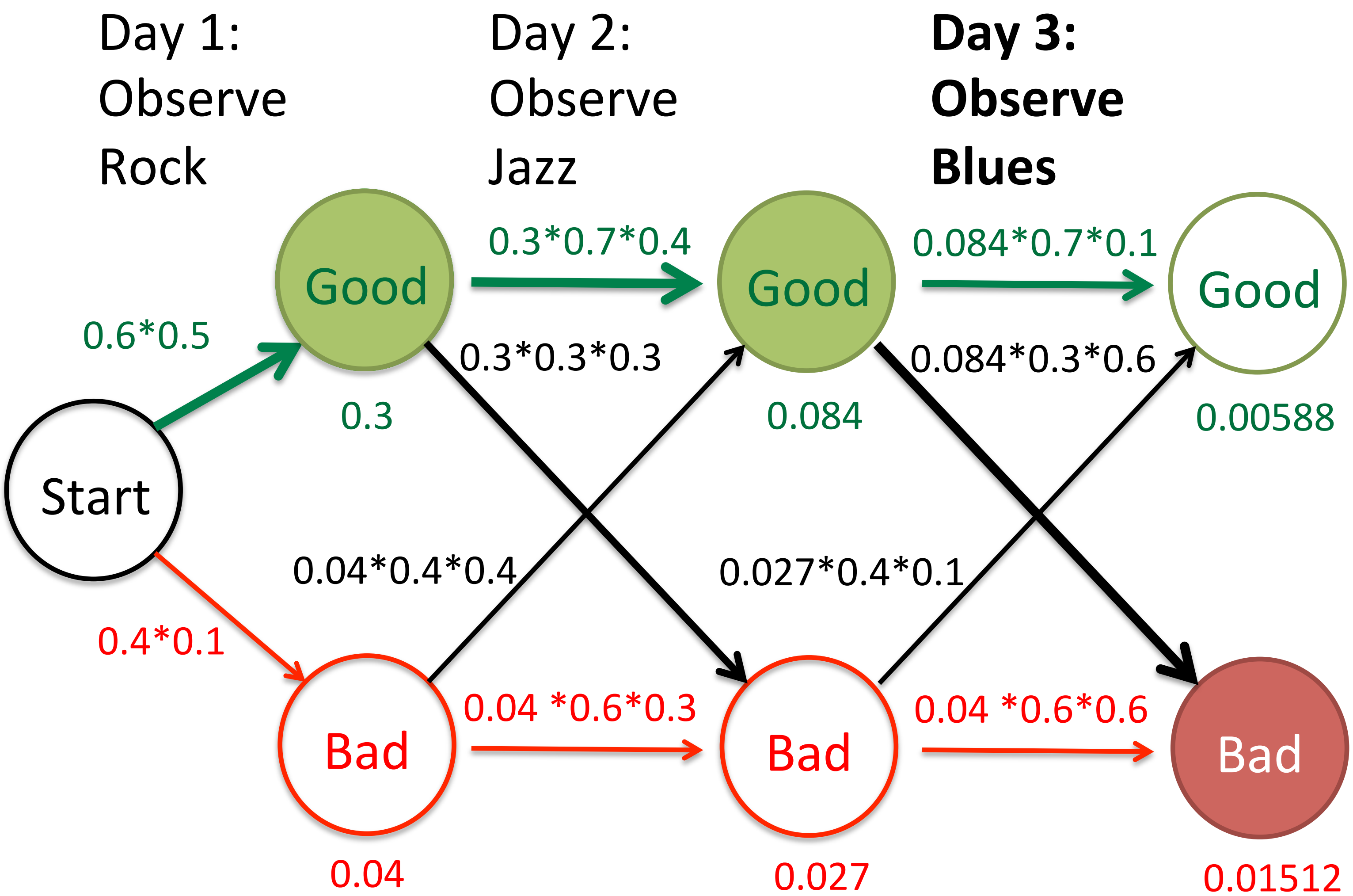


Update rule:
Current* Transition* Observation

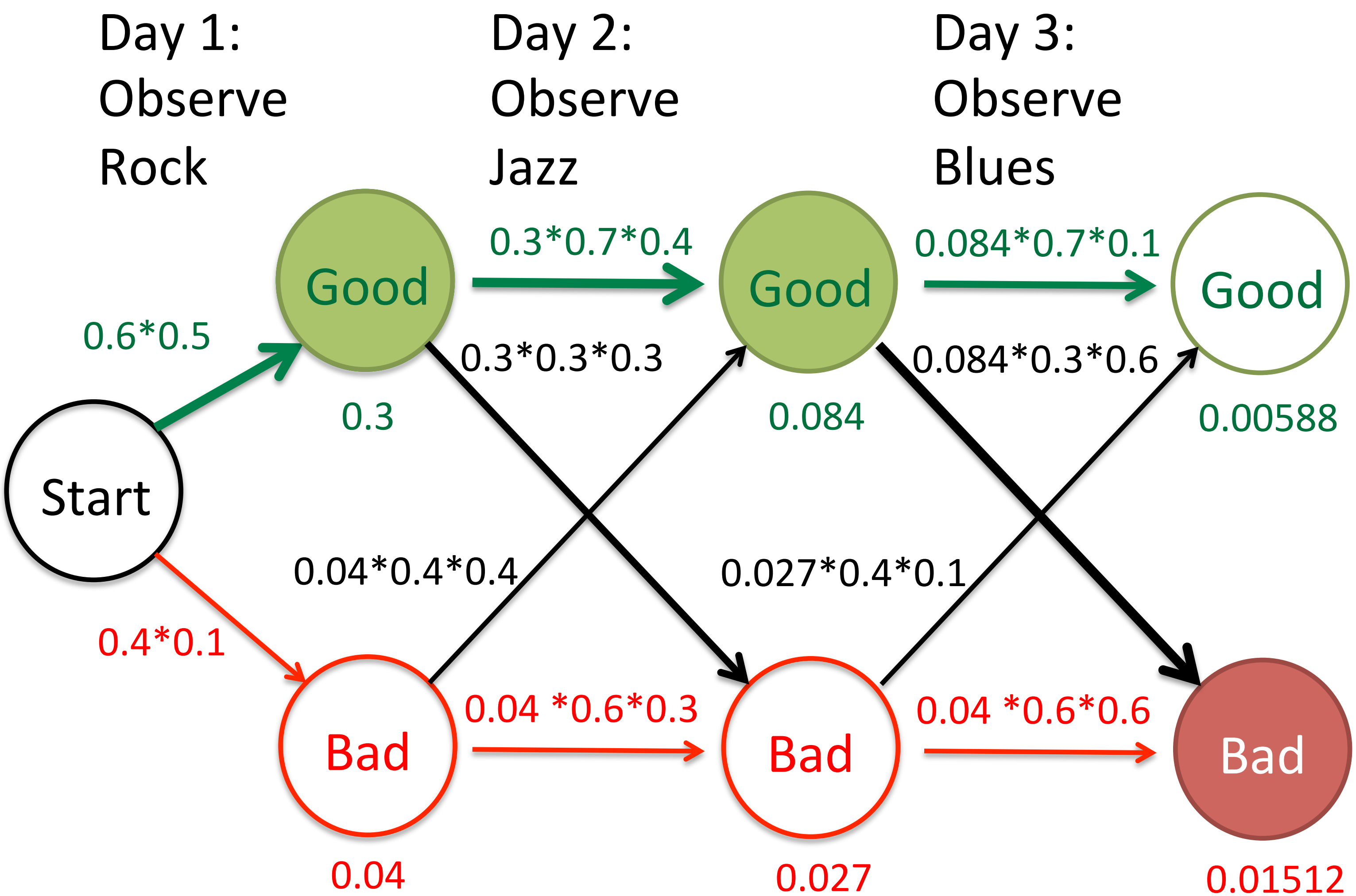
Most likely state has highest value

Now only about 3X more likely to be in Good mood

We can estimate the most likely hidden state based on observations



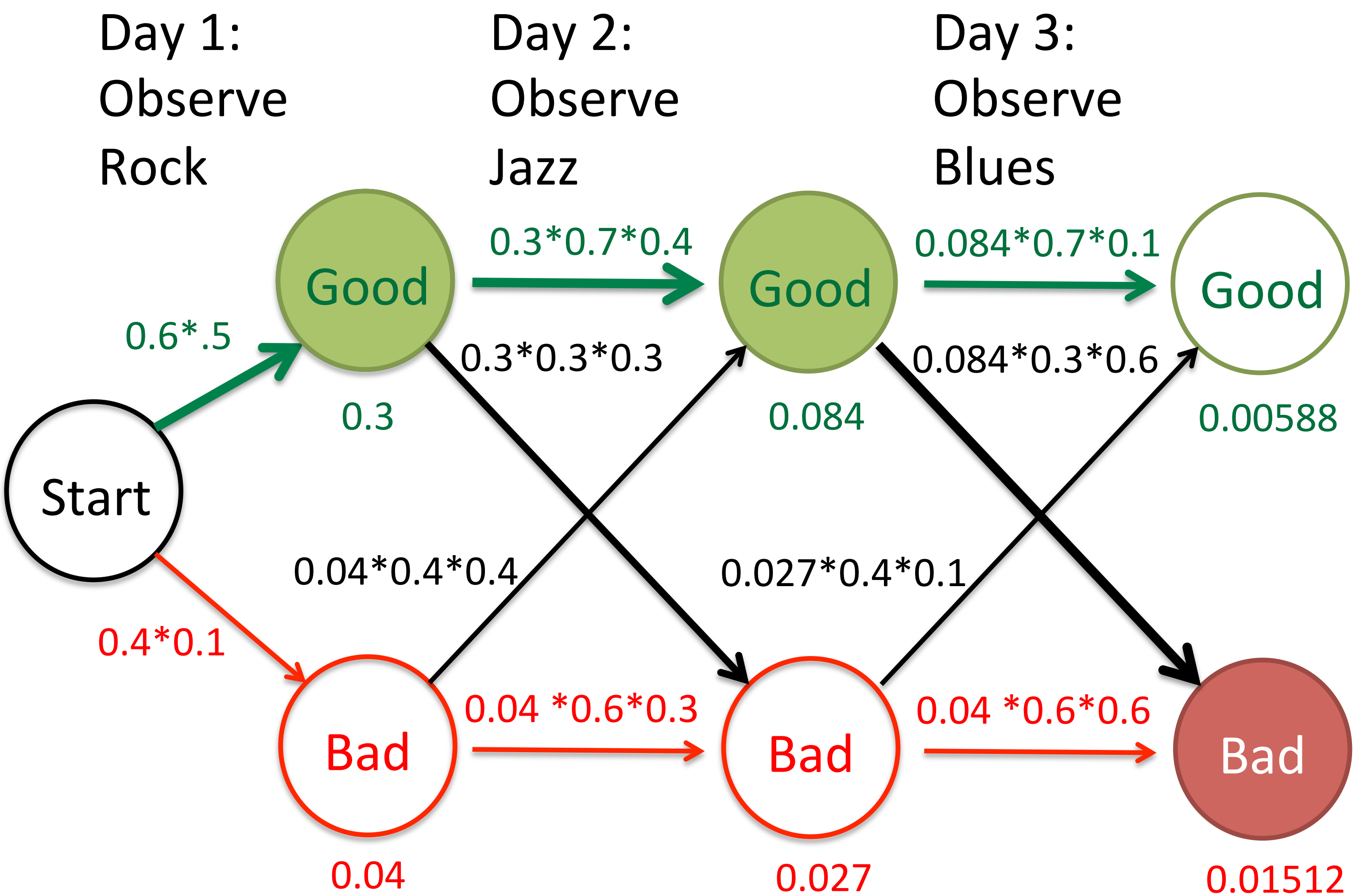
We can estimate the most likely hidden state based on observations



Given observations of {Rock, Jazz, Blues}

The boss's mood mostly likely was {Good, Good, Bad}

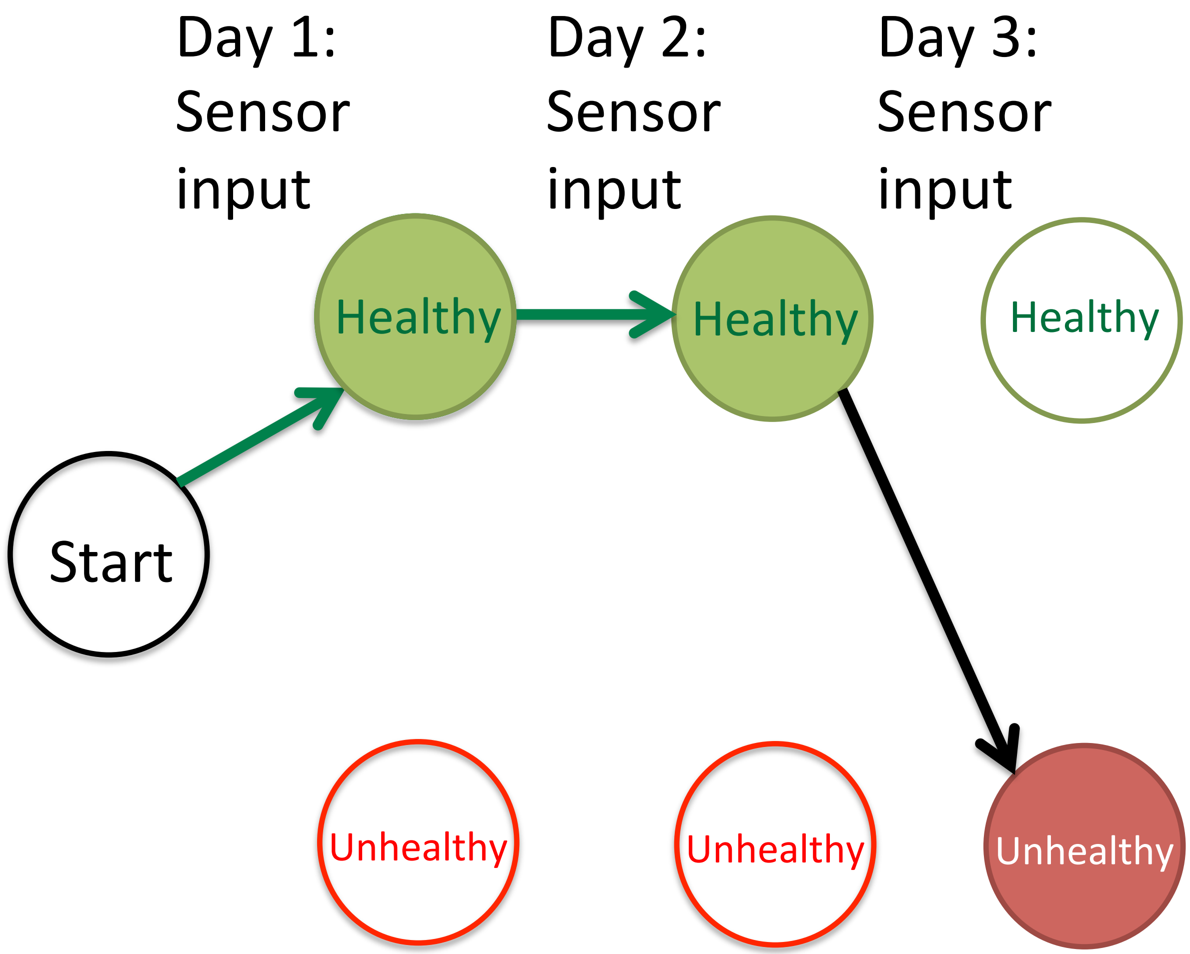
Viterbi algorithm back tracks to find most likely state sequence given observations



Given observations of {Rock, Jazz, Blues}

The boss's mood mostly likely was {Good, Good, Bad}

HMMs and Viterbi algorithm used in a number of fields such a monitoring health



Prof. Campbell's *BeWell* app uses smart phone sensor data and HMM to estimate health behavior of users over time

Given sequence of sensor data, what was the subject's most likely health state on each day

Lane N, Mohammod M, Lin M, Yang X, Lu H, Ali S, et al. BeWell: A smartphone application to monitor, model and promote wellbeing. *International Conference on Pervasive Computing Technologies for Healthcare*; 2011.

HMMs allow us to determine the most likely sequence of state transitions

Key points

We can't directly the hidden state and know the true state with certainty

If there is something we *can* observe, we might be able to *infer* the true state with greater accuracy than guessing

Given a sequence of observations we can determine the most likely state transitions over time

Agenda

1. Pattern matching vs. recognition
2. From Finite Automata to Hidden Markov Models
3. Decoding: Viterbi algorithm
4. Training

To build an HMM we start with previous observations called training data

Annotated training data gives transition probabilities

Situation:
Have a diary with of number of ice cream cones eaten each day (observations)

Want to reconstruct the weather (hidden state) when the cones were eaten

To build an HMM we start with previous observations called training data

Annotated training data gives transition probabilities

Diary entries:

1. Hot day today! I chowed down three whole cones.
2. Hot again. But I only ate two cones; need to run to the store and get more ice cream.
3. Cold today. Still, the ice cream was calling me, and I ate one cone.
4. Cold again. Kind of depressed, so ate a couple cones despite the weather.
5. Still cold. Only in the mood for one cone.
6. Nice hot day. Yay! Was able to eat a cone each for breakfast, lunch, and dinner.
7. Hot but was out all day and only had enough cash on me for one ice cream.
8. Brrrr, the weather turned cold really quickly. Only one cone today.
9. Even colder. Still ate one cone.
10. Defying the continued coldness by eating three cones.

To build an HMM we start with previous observations called training data

Annotated training data gives transition probabilities

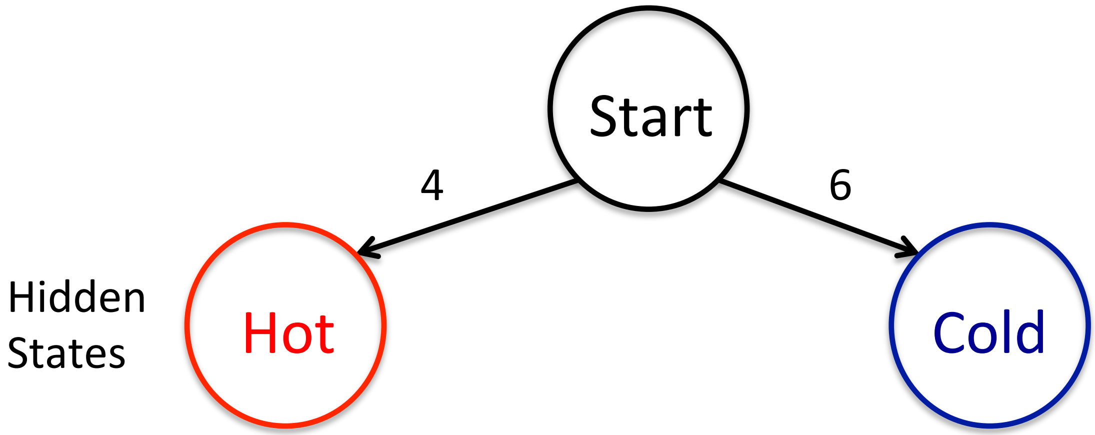
Diary entries:

1. Hot day today! I chowed down three whole cones.
2. Hot again. But I only ate two cones; need to run to the store and get more ice cream.
3. Cold today. Still, the ice cream was calling me, and I ate one cone.
4. Cold again. Kind of depressed, so ate a couple cones despite the weather.
5. Still cold. Only in the mood for one cone.
6. Nice hot day. Yay! Was able to eat a cone each for breakfast, lunch, and dinner.
7. Hot but was out all day and only had enough cash on me for one ice cream.
8. Brrrr, the weather turned cold really quickly. Only one cone today.
9. Even colder. Still ate one cone.
10. Defying the continued coldness by eating three cones.

Hidden states: Hot (4 days) or Cold (6 days)
Observations: 1, 2, or 3 ice cream cones eaten

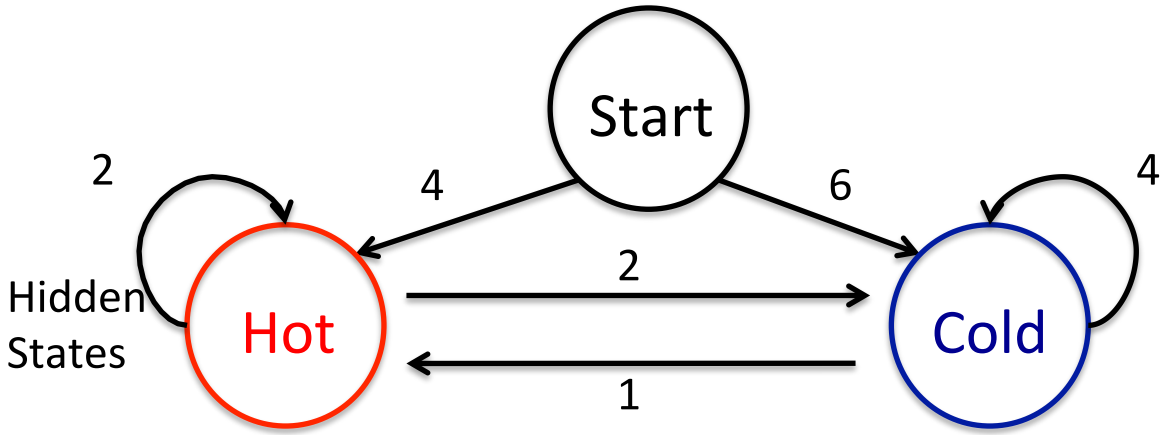
To build an HMM we start with previous observations called training data

Count of observations: 4 hot days, 6 cold days



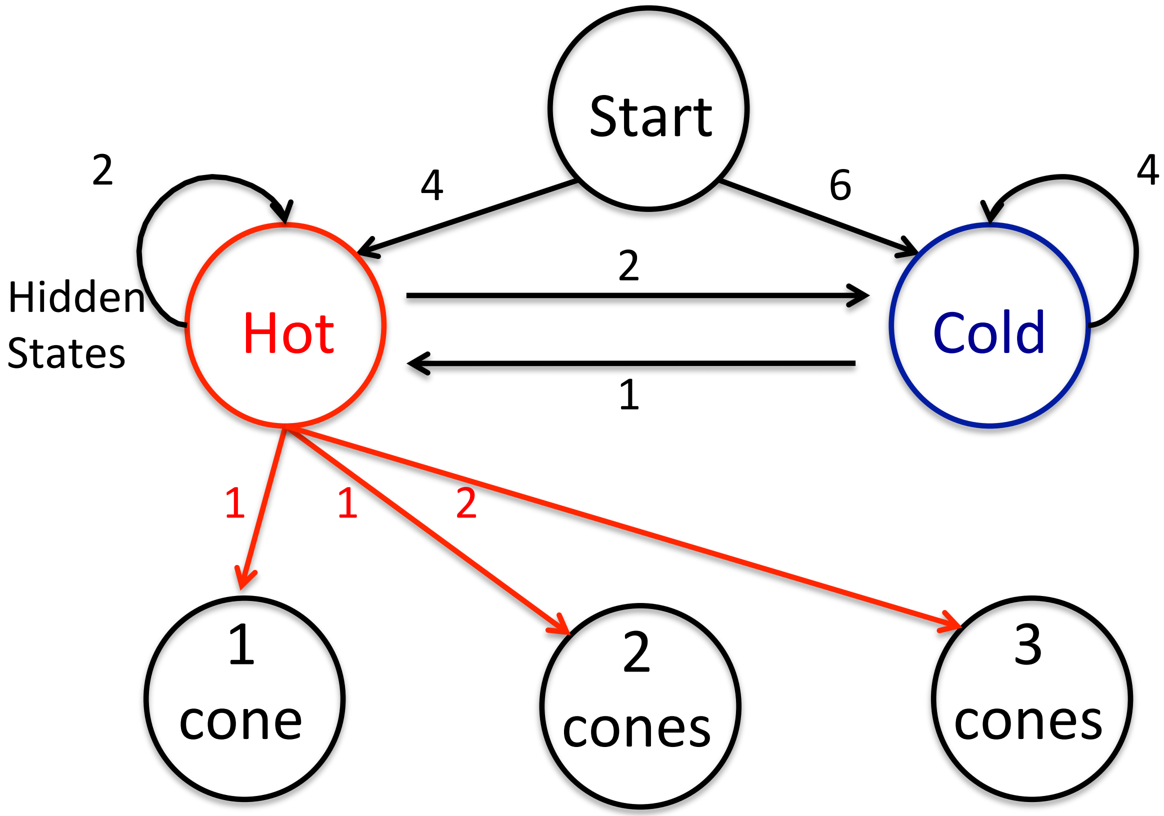
To build an HMM we start with previous observations called training data

Count of observations: transitions between hidden states



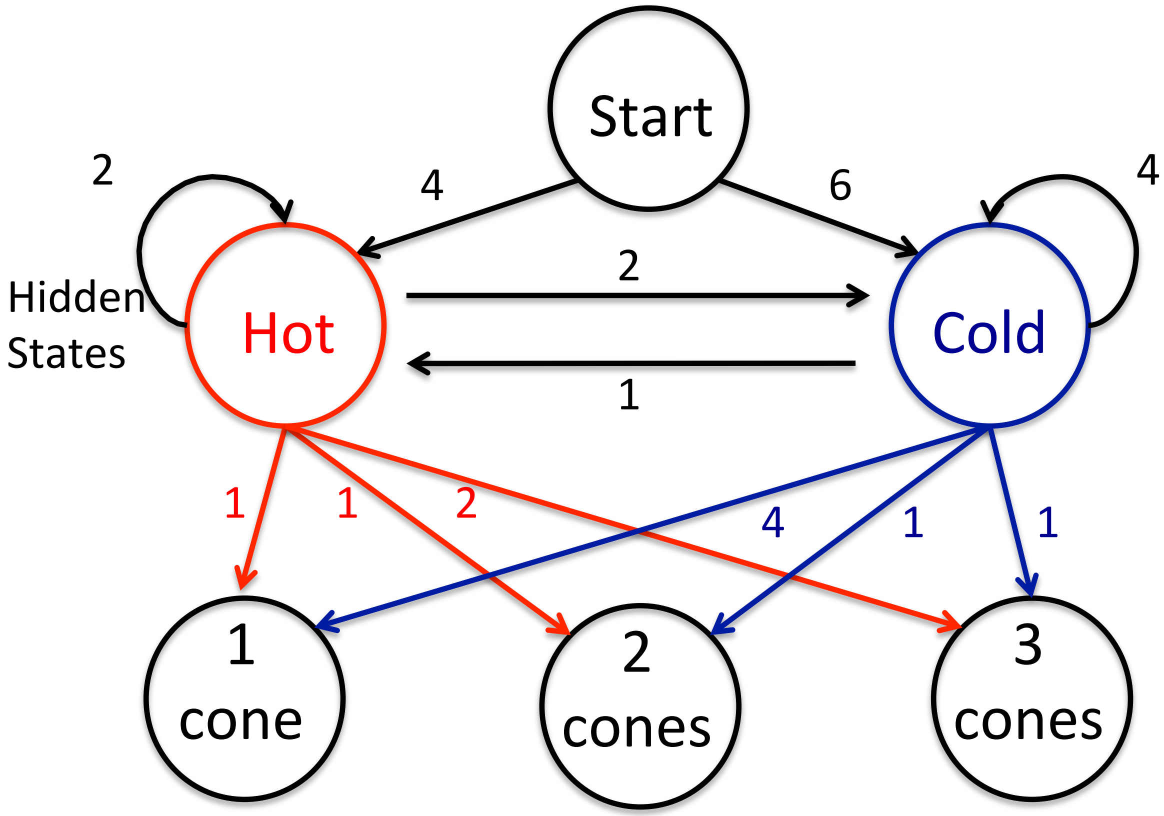
To build an HMM we start with previous observations called training data

Count of observations: cones eaten when hot



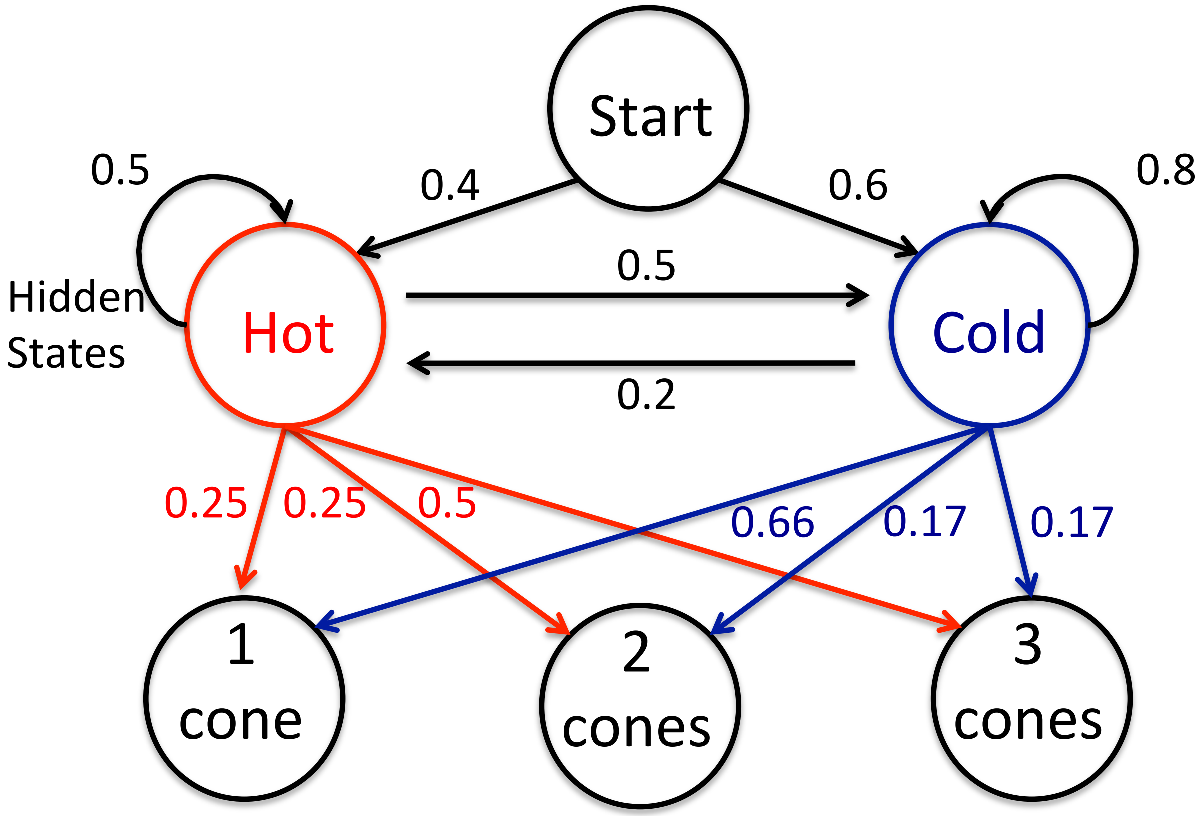
To build an HMM we start with previous observations called training data

Count of observations: cones eaten when cold



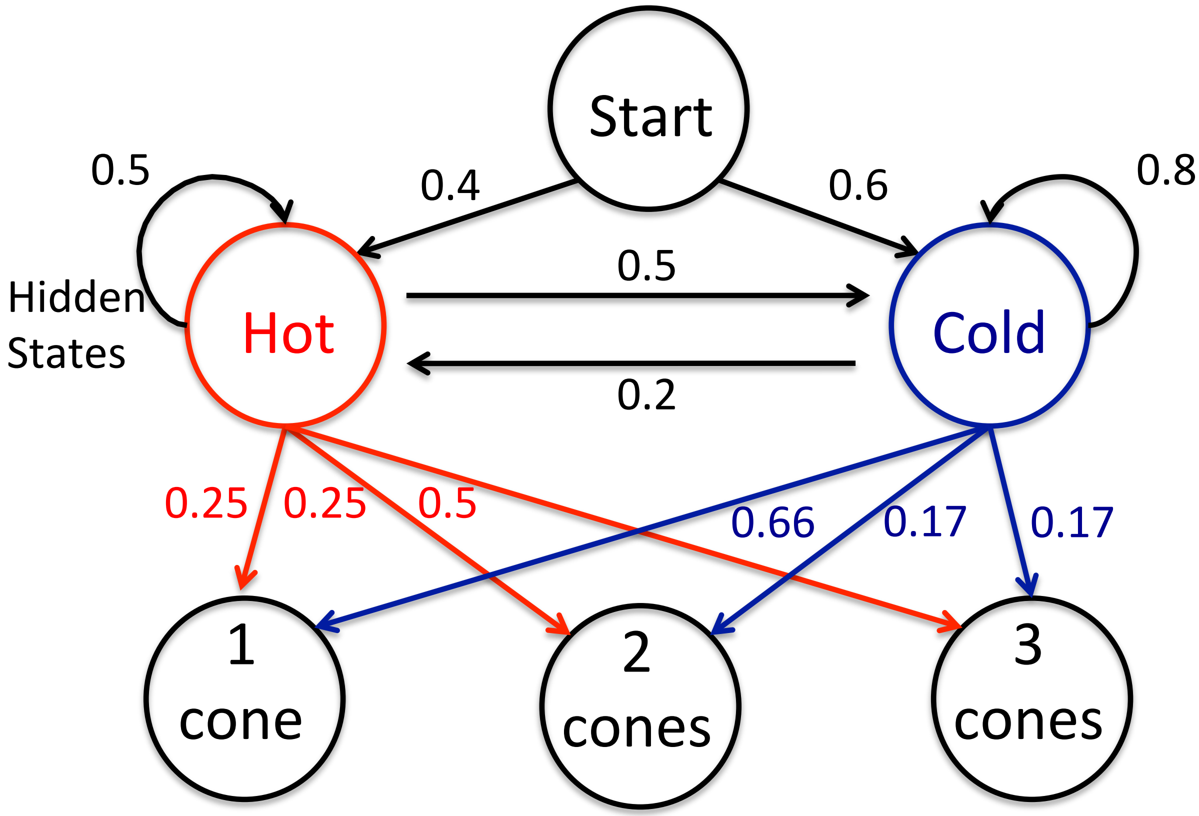
Convert observations into probabilities by dividing count by total count

Probabilities based on observations



Convert observations into probabilities by dividing by total count, then use logs

Probabilities based on observations



Repeatedly multiplying probabilities quickly leads to very small numbers

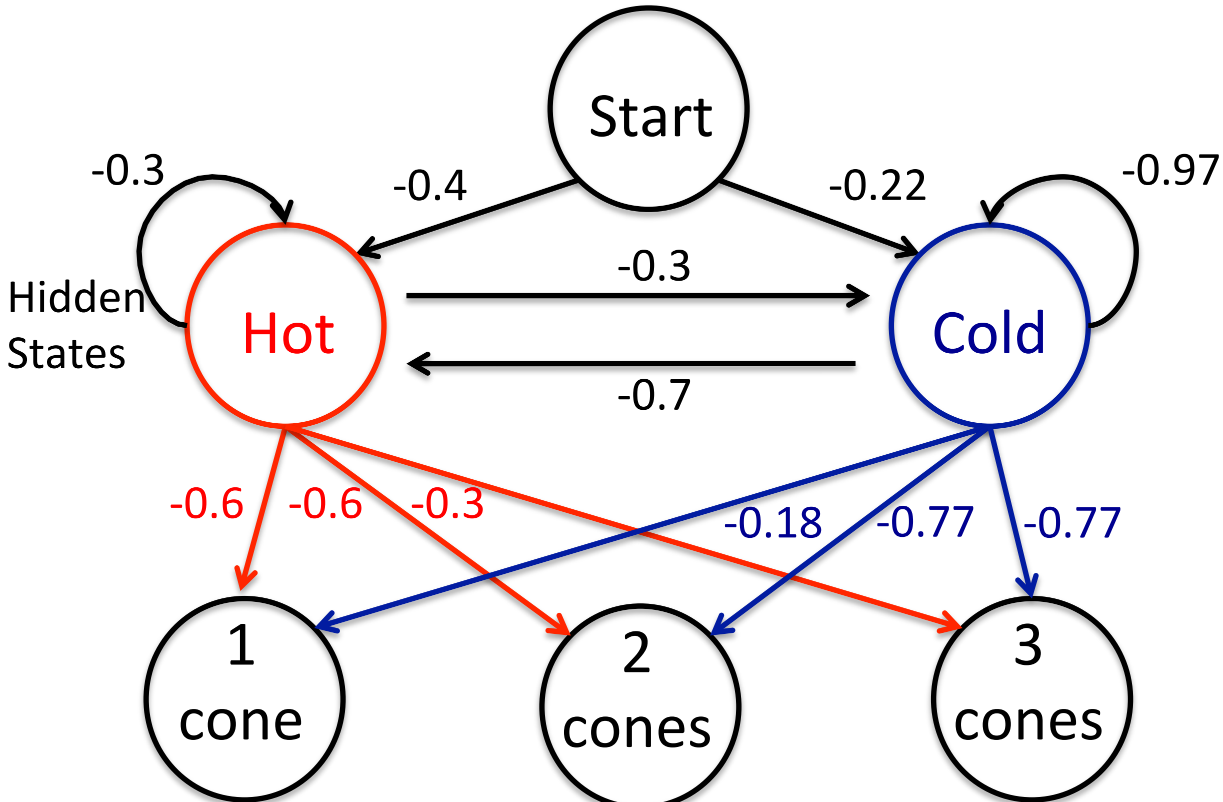
This can cause numerical precision issues

log(mn) = log(m) + log(n)

To calculate Viterbi, add logs of each factor instead of multiplying

Convert observations into probabilities by dividing by total count, then use logs

Log probabilities based on observations



Repeatedly multiplying probabilities quickly leads to very small numbers

This can cause numerical precision issues

log(mn) = log(m) + log(n)

To calculate Viterbi, add logs of each factor instead of multiplying

Example

#	Observation	nextState	currrentState	currScore+transScore +observation	nextScore
Start	n/a	Start	n/a	0	0

Most likely {Hot,Hot,Hot}

Example

#	Observation	nextState	currrentState	currScore+transScore +observation	nextScore
Start	n/a	Start	n/a	0	0
0	Two cones	Cold	Start	0-0.22-0.77	-0.99
		Hot	Start	0-0.4-0.6	-1.0

Most likely {Cold}

Example

#	Observation	nextState	currrentState	currScore+transScore +observation	nextScore
Start	n/a	Start	n/a	0	0
0	Two cones	Cold	Start	0-0.22-0.77	-0.99
		Hot	Start	0-0.4-0.6	-1.0
1	Three cones	Cold	Cold	-0.99-0.97-0.77	~~-2.73~~
		Cold	Hot	-1-0.3-0.77	-2.07
		Hot	Cold	-0.99-0.7-0.3	~~-1.99~~
		Hot	Hot	-1-0.3-0.3	-1.6

Most likely {Hot,Hot}

Example

#	Observation	nextState	currrentState	currScore+transScore +observation	nextScore
Start	n/a	Start	n/a	0	0
0	Two cones	Cold	Start	0-0.22-0.77	-0.99
		Hot	Start	0-0.4-0.6	-1.0
1	Three cones	Cold	Cold	-0.99-0.97-0.77	~~-2.73~~
		Cold	Hot	-1-0.3-0.77	-2.07
		Hot	Cold	-0.99-0.7-0.3	~~-1.99~~
		Hot	Hot	-1-0.3-0.3	-1.6
2	Two cones	Cold	Cold	-2.07-0.97-0.77	~~-3.81~~
		Cold	Hot	-1.6-0.3-0.77	-2.67
		Hot	Cold	-2.07-0.7-0.6	~~-3.37~~
		Hot	Hot	-1.6-0.3-0.6	-2.5

Most likely {Hot,Hot,Hot}

Example

#	Observation	nextState	currrentState	currScore+transScore +observation	nextScore
Start	n/a	Start	n/a	0	0
0	Two cones	Cold	Start	0-0.22-0.77	-0.99
		Hot	Start	0-0.4-0.6	-1.0
1	Three cones	Cold	Cold	-0.99-0.97-0.77	~~-2.73~~
		Cold	Hot	-1-0.3-0.77	-2.07
		Hot	Cold	-0.99-0.7-0.3	~~-1.99~~
		Hot	Hot	-1-0.3-0.3	-1.6
2	Two cones	Cold	Cold	-2.07-0.97-0.77	~~-3.81~~
		Cold	Hot	-1.6-0.3-0.77	-2.67
		Hot	Cold	-2.07-0.7-0.6	~~-3.37~~
		Hot	Hot	-1.6-0.3-0.6	-2.5

Most likely {Hot,Hot,Hot}