

CSCI 5622 Machine Learning

ML Sequence Learning

DATE	TOPIC	DUE
Mon, Dec 7	Sequence Learning	Peer Feedback Due
Wed, Dec 9	Reinforcement Learning	Peer Feedback Grades
Fri, Dec 11	-----	<u>Final Paper Due</u>
W 16 th 4PM	<u>1777 Exposition Dr, Room at Entry</u>	Final Presentations
.. (Dec 16)	<u>Mandatory attendance & participation</u>	<u>Peer Project Questions</u>

www.RodneyNielsen.com/teaching/CSCI5622-F09/

Instructor: Rodney Nielsen

Assistant Professor Adjunct, CU Dept. of Computer Science

Research Assistant Professor, DU, Dept. of Electrical & Computer Engr.

Research Scientist, Boulder Language Technologies

ML Sequence Learning

- Sequences of commands
- Sequences of words in sentences
- Sequences of phonemes in spoken words
- Patterns of weather

ML Deterministic Sequences

- Traffic light
 - Always the same order
 - Easy to analyze once you know the transitions

ML Non-deterministic Patterns

- Weather
 - Sunny, Cloudy, Stormy
 - but not necessarily in that order

ML Markov Assumption

- Markov assumption
 - Assume the state depends only on the previous state
 - Simplifies problems significantly
 - May potentially be an oversimplification
 - E.g., today's weather depends only on the weather of the past few days, and not the wind, barometric pressure, etc.

Markov Process

- State depends only on the previous n states
- Order n Markov model depends on n states
- A first order Markov model depends only on the previous state
 - With n states there are n^2 transitions
- State transitions are made probabilistically
 - State transition probabilities do not vary over time

ML Ex: State Transition Matrix

- Weather state transition matrix

		<i>Today</i>		
		sunny	cloudy	stormy
<i>Yesterday</i>	sunny	0.50	0.375	0.125
	cloudy	0.25	0.125	0.625
	stormy	0.25	0.375	0.375

- The sum of the probability of a row = 1.0
 - E.g., $P(\text{Su}_t | \text{Su}_{t-1}) + P(\text{Cl}_t | \text{Su}_{t-1}) + P(\text{St}_t | \text{Su}_{t-1}) = 1.0$
 - We know yesterday's weather and want to predict tomorrow's weather given that knowledge

ML Ex: State Transition Matrix

- Weather state transition matrix

		<i>Today</i>		
		sunny	cloudy	stormy
<i>Yesterday</i>	sunny	0.50	0.375	0.125
	cloudy	0.25	0.125	0.625
	stormy	0.25	0.375	0.375

- The sum of the probability of a column is not necessarily 1.0
 - E.g., $P(\text{St}_t | \text{Su}_{t-1}) + P(\text{St}_t | \text{Cl}_{t-1}) + P(\text{St}_t | \text{St}_{t-1}) = 1.125$
 - This is not what we are modeling

ML Initialization

- Must initialize the system according to the probable previous state
 - The π vector
 - E.g.,

sunny	cloudy	stormy
0.8	0.1	0.1

- First order Markov process consists of:
 - States: (e.g., sunny, cloudy, stormy)
 - π vector: defines the state at time 0, initial state
 - State transition matrix: the probabilities of transitioning from one state to another
 - Fixed over the life of the model

- Assume you can't observe the weather, but can observe your dog
- Can we devise an algorithm to predict the hidden weather state based only on the state of the dog?

ML Speech Recognition

- Factors in interpreting speech
 - Vocal cords, throat, tongue, etc.
- Assume speech production is a sequence of hidden (internal) states
- The sound we hear is a sequence of observable states approximating the hidden states
 - E.g., there could be noise

ML Hidden vs. Observable States

- The number of observable states might be very different than the number of hidden states
 - E.g., 3 hidden weather states (sunny, cloudy, stormy), but 4 states of the dog (exhausted, spunky, lazy, scared)
 - E.g., say 80 hidden (real) phoneme states but only 40 observable/distinguishable sounds

ML Hidden Markov Model (HMM)

- The hidden and observable states are probabilistically related

ML HMM Confusion Matrix

- Confusion Matrix: Gives the probability of an observable state given a hidden state

		Observable Dog State			
		exhausted	spunky	lazy	scared
Hidden Weather State	sunny	0.60	0.20	0.15	0.05
	cloudy	0.25	0.25	0.25	0.25
	stormy	0.05	0.10	0.35	0.50

- Since we are assuming a given hidden state, the probability summed over a row is 1.0
 - E.g., $P(\text{Ex}_t | \text{St}_{t-1}) + P(\text{Sp}_t | \text{St}_{t-1}) + P(\text{Lz}_t | \text{St}_{t-1}) + P(\text{Sc}_t | \text{St}_{t-1}) = 1.0$

ML Hidden Markov Model (HMM)

- **Hidden states:** the *true* states of the system, the states we care about, are not observable
- **Observable states:** related states that are visible
- **π vector:** the initial probabilities of the hidden states
- **State transition matrix:** Gives the probability of moving from one hidden state to another
- **Confusion matrix:** Gives the probability of an observable state given a hidden state

ML Hidden Markov Model

- An HMM is a standard Markov process, where:
 - The true states are hidden,
 - But there are other observable states,
 - Which are probabilistically related to true states

ML Hidden Markov Model

- A Hidden Markov Model (HMM) is a triple $(\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$
 - $\boldsymbol{\pi} = (\pi_1)$: The vector of initial state probabilities
 - $\mathbf{A} = (a_{i,j})$: The state transition matrix $P(x_{i,t} | x_{j,t-1})$
 - $\mathbf{B} = (b_{i,j})$: The confusion matrix $P(y_i | x_j)$
- The probability matrices do not vary over time
 - Most unrealistic assumption about HMMs

- Evaluation: Compute the probability of a sequence of observed states given an HMM
- Decoding: Compute the probability of a sequence of hidden states given a sequence of observed states and the HMM

ML HMM Evaluation Task

- Suppose you want to determine which of a set of HMMs is responsible for generating a particular sequence of observations
 - E.g., you may have multiple weather models based on the season or multiple phoneme models based on the gender, accent, or other attributes of the speaker

ML HMM Evaluation Task

- Suppose you want to determine which of a set of HMMs is responsible for generating a particular sequence of observations
 - For each HMM, compute the probability of the given sequence of observed states using the forward algorithm
 - Select the HMM that is the most probable

ML HMM Evaluation Task

- Suppose you want to determine which of a set of HMMs is responsible for generating a particular sequence of observations
- E.g., speech recognition
 - Start with several word models (HMMs)
 - Record a sequence of observed sounds
 - Determine word by computing most probable HMM to have generated it

ML HMM Decoding Task

- What sequence of hidden states generated the sequence of observations
- Compute the most probable sequence of hidden states given an HMM and a sequence of observations using the Viterbi algorithm

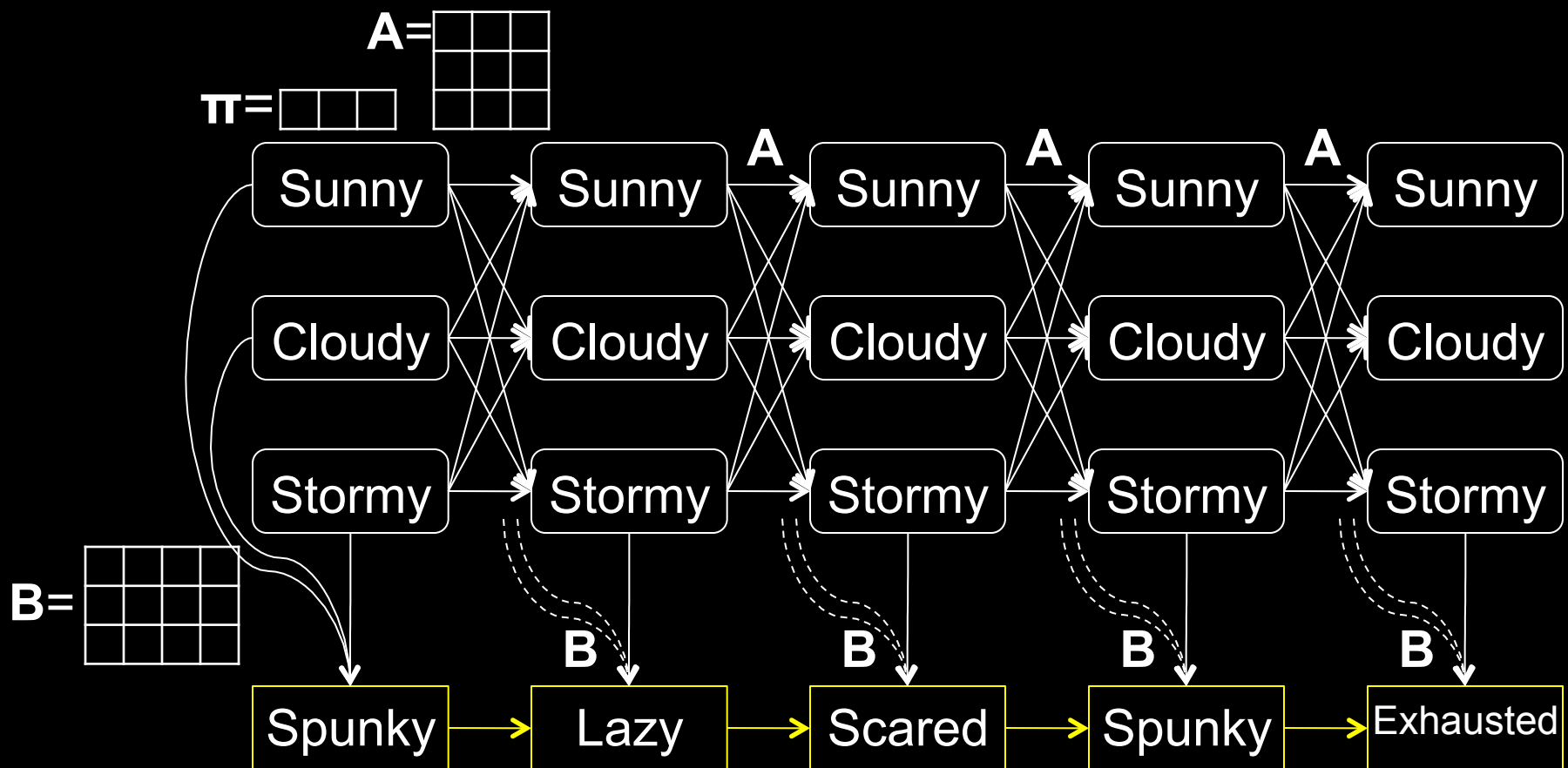
ML Ex: HMM Decoding Task

- What sequence of hidden part-of-speech (POS) states generated the sequence of observed word states
 - She was busily decoding_{verb} the message.
 - The decoding_{common-noun} was “*good luck!*”.
 - Decoding_{proper-noun} Systems, Inc. has all the fun.
 - The decoding_{???} task was fun!

ML Learning an HMM

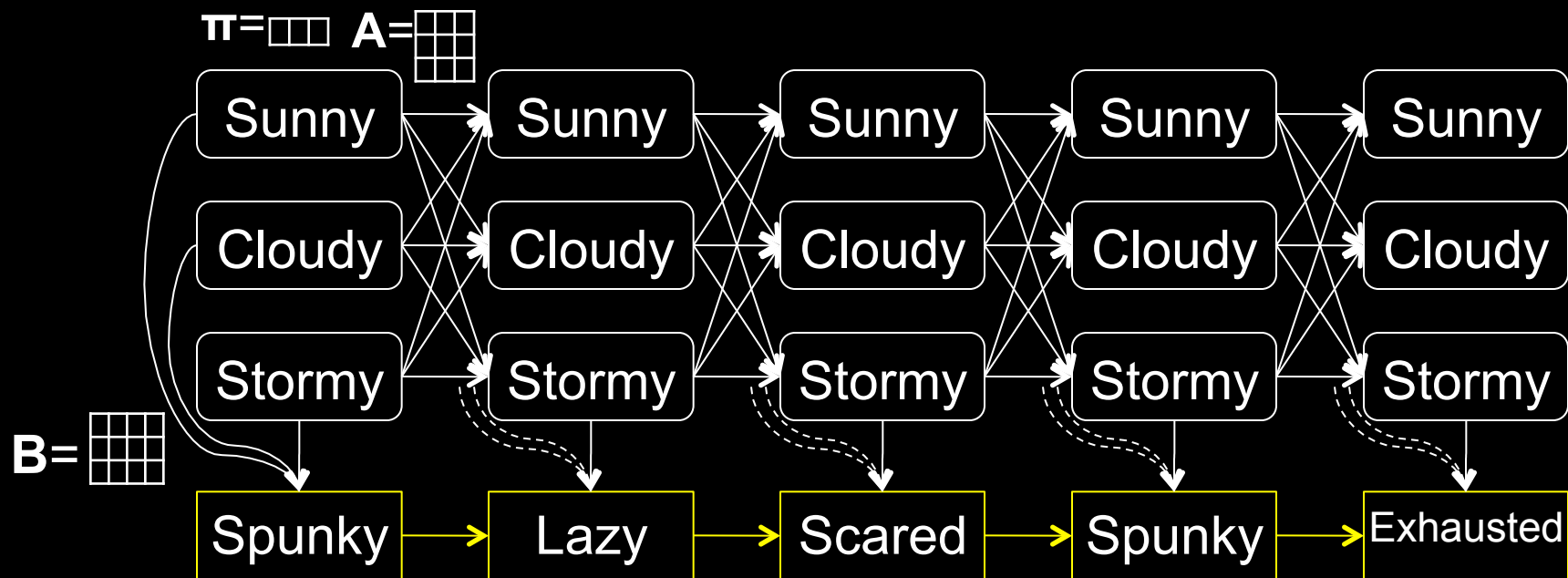
- Learn the most probable HMM, $(\pi, \mathbf{A}, \mathbf{B})$, given a set of observed state sequences associated sequences of underlying hidden states
- Use the forward-backward algorithm to learn the matrices \mathbf{A} and \mathbf{B}
 - These probabilities are often unknown in real-world scenarios

ML Evaluation: $P(\text{Observed Sequence})$



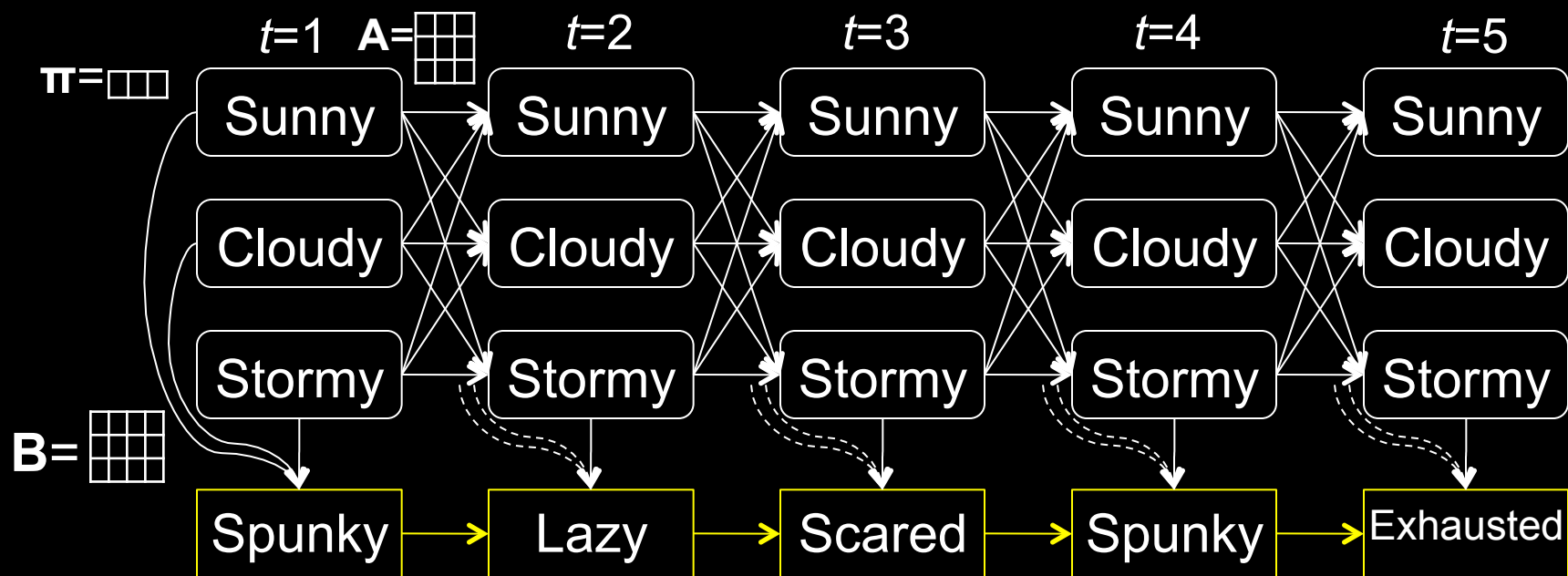
ML Exhaustive Search

- $P(\text{Sp}, \text{Lz}, \text{Sc}, \text{Sp}, \text{Ex}) = P(\text{Sp}, \text{Lz}, \text{Sc}, \text{Sp}, \text{Ex} \mid \text{Su}, \text{Su}, \text{Su}, \text{Su}, \text{Su}) + P(\text{Sp}, \text{Lz}, \text{Sc}, \text{Ex} \mid \text{Su}, \text{Su}, \text{Su}, \text{Su}, \text{Cl}) + \dots + P(\text{Sp}, \text{Lz}, \text{Sc}, \text{Sp}, \text{Ex} \mid \text{St}, \text{St}, \text{St}, \text{St}, \text{St})$
- **Too Expensive**



ML Forward Algorithm

- Instead calculate the probabilities recursively
- Consider the probability of reaching an intermediate state in the trellis, say the probability that it is Stormy at $t=2$

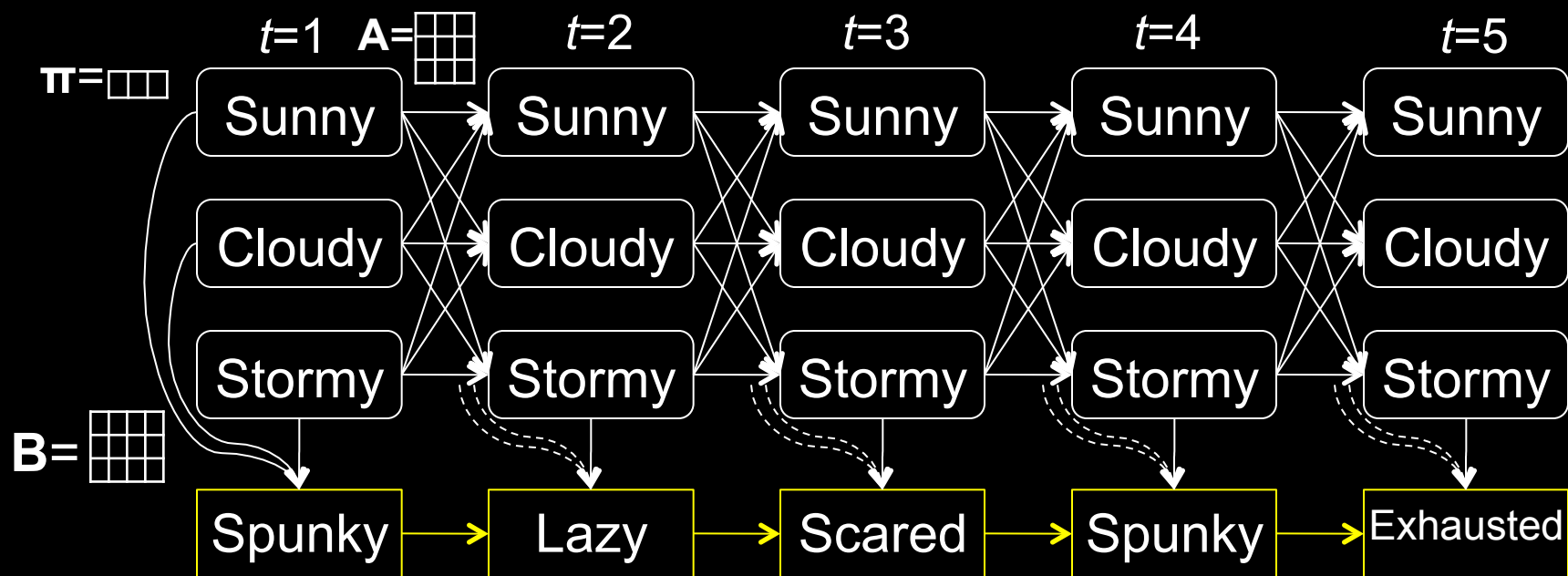


ML Forward Algorithm

- $P(\text{Sp}, \text{Lz}, \text{Sc}, \text{Sp}, \text{Ex})$

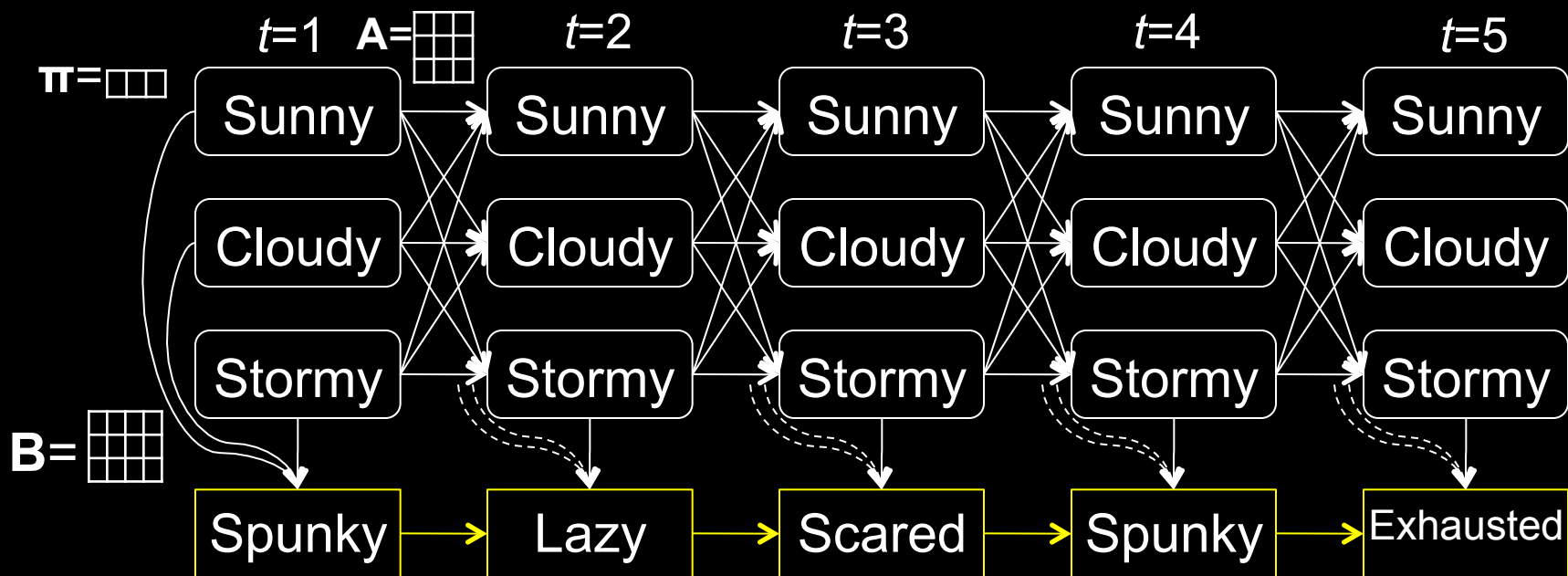
= The sum of the probabilities over all paths

= The sum of the final partial probabilities $\alpha(j_5)$



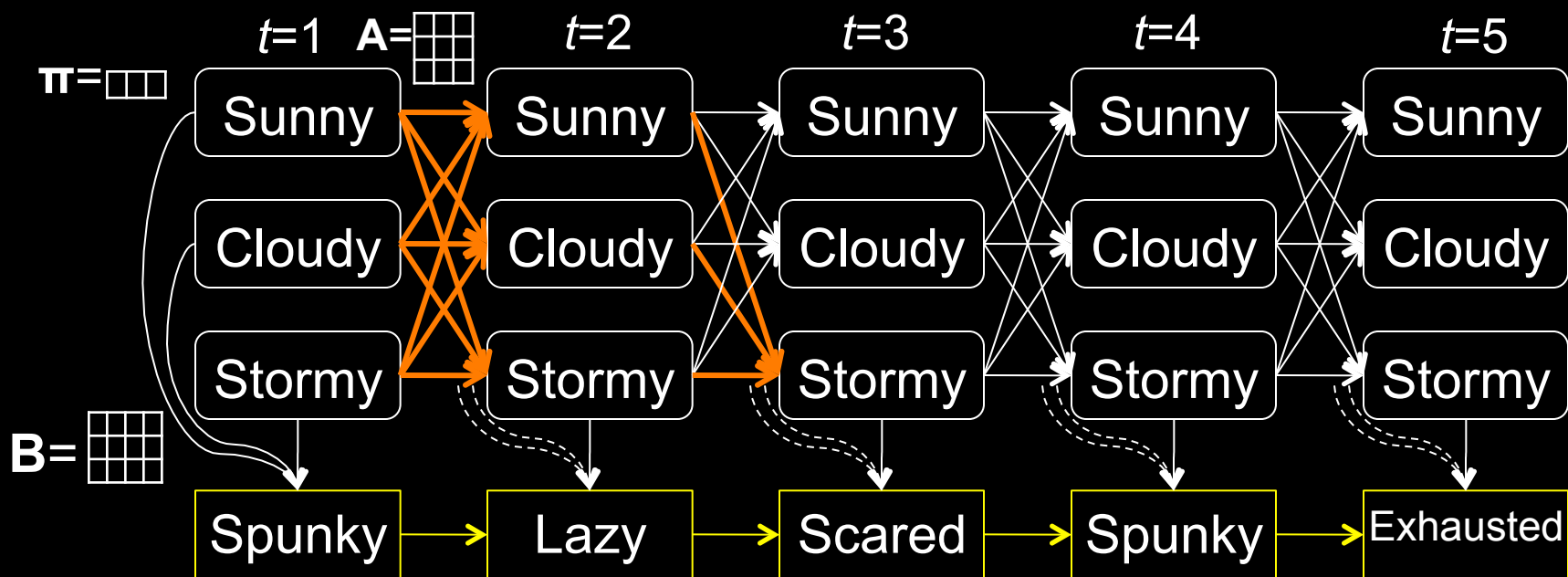
ML Forward Algorithm

- $\alpha(j_t) = P(obs=y_t | j_t) \times \sum_{path\ to\ j_t} P(path); \alpha(j_1) = ?$
- $\alpha(j_1) = P(y_1 | j_1) \times \sum_{path\ to\ j_t} P(path)$
 $= b_{j,y_1} \times \pi(j)$



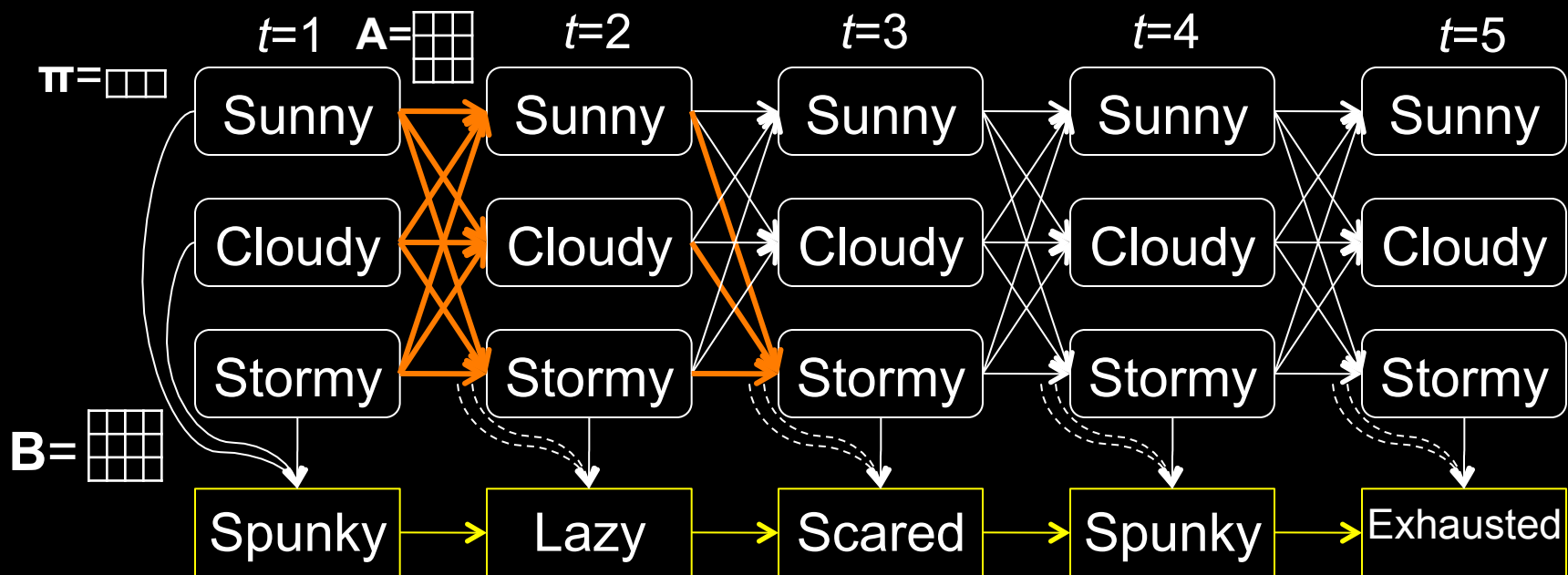
ML Forward Algorithm

- The number of paths grows exponentially with the length of the observation sequence, but...
- We can recursively compute the $\alpha(j_{t-1})$



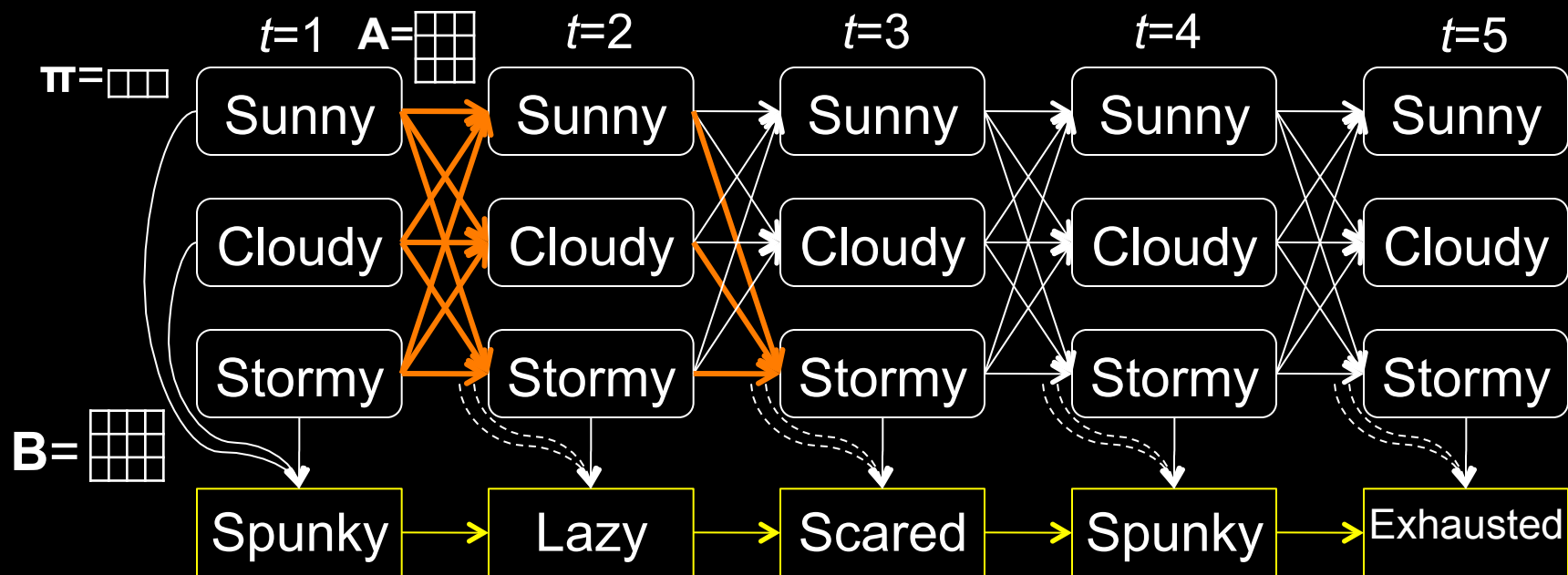
ML Forward Algorithm

- $\alpha(j_t) = b_{y,j} \times \sum_i \alpha(i_{t-1}) a_{i,j}$
- Compute the $\alpha(j_t)$ forward from $t=1$ to $t=T$
- The $\sum_j \alpha(j_T)$ is the probability of the sequence



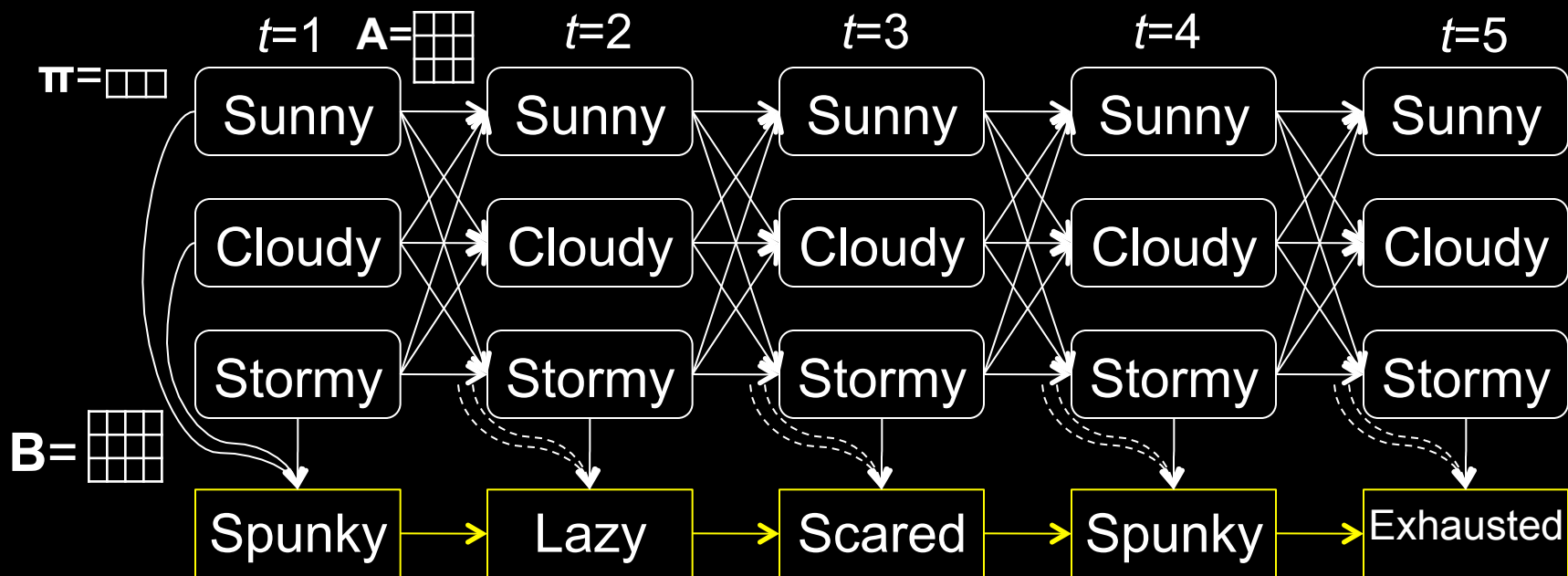
ML Forward Algorithm

- Exhaustive search is $O(n^T)$
- Forward search is $O(nT) \ll O(n^T)$



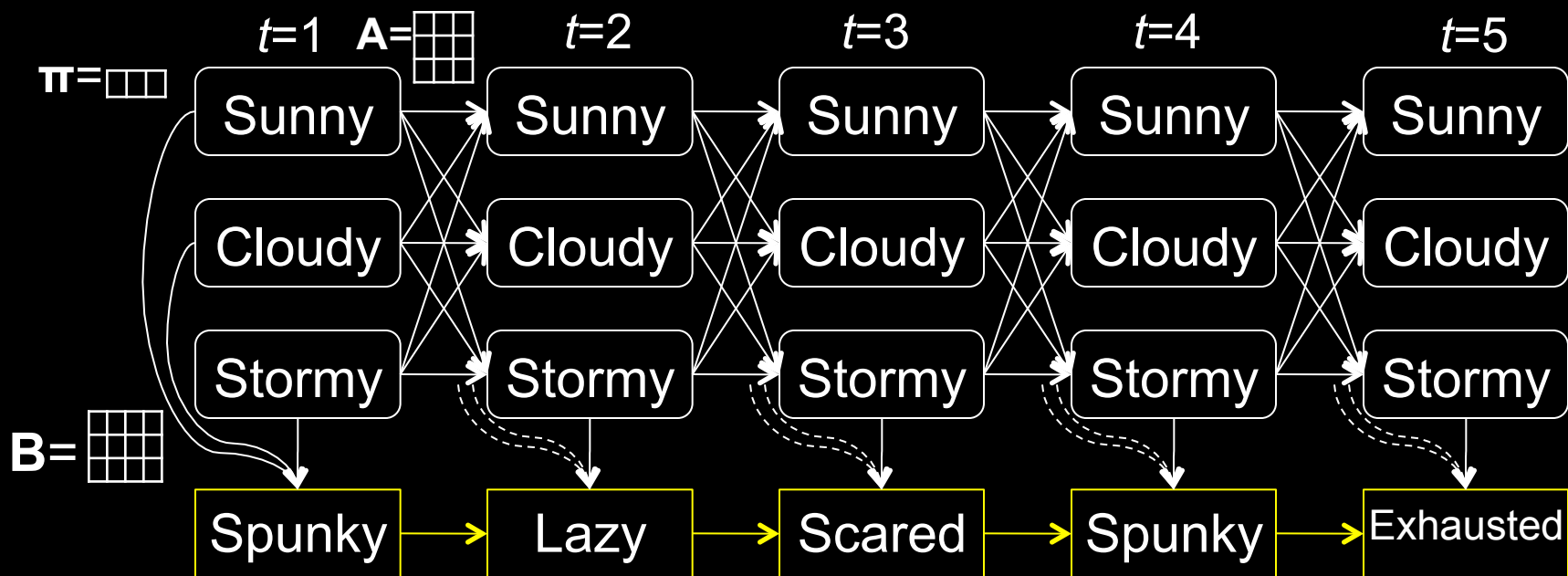
ML The Sequence of Hidden States

- Finding the most probable sequence of hidden states
- Exhaustive search: enumerate all the possible state paths and their probabilities
- Select the path that maximizes the probability



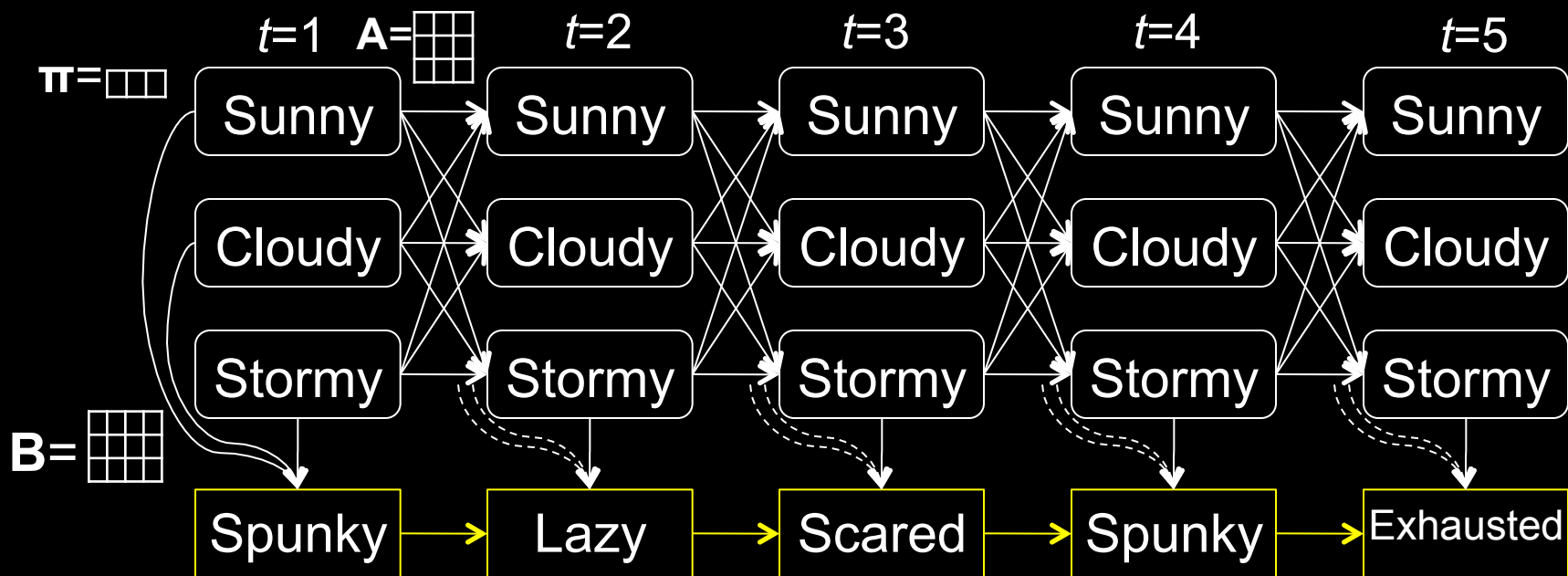
ML The Sequence of Hidden States

- Finding the most probable sequence of hidden states
- Exhaustive search: enumerate all the possible state paths and their probabilities: $O(n^T)$, **too expensive**
- Select the path that maximizes the probability



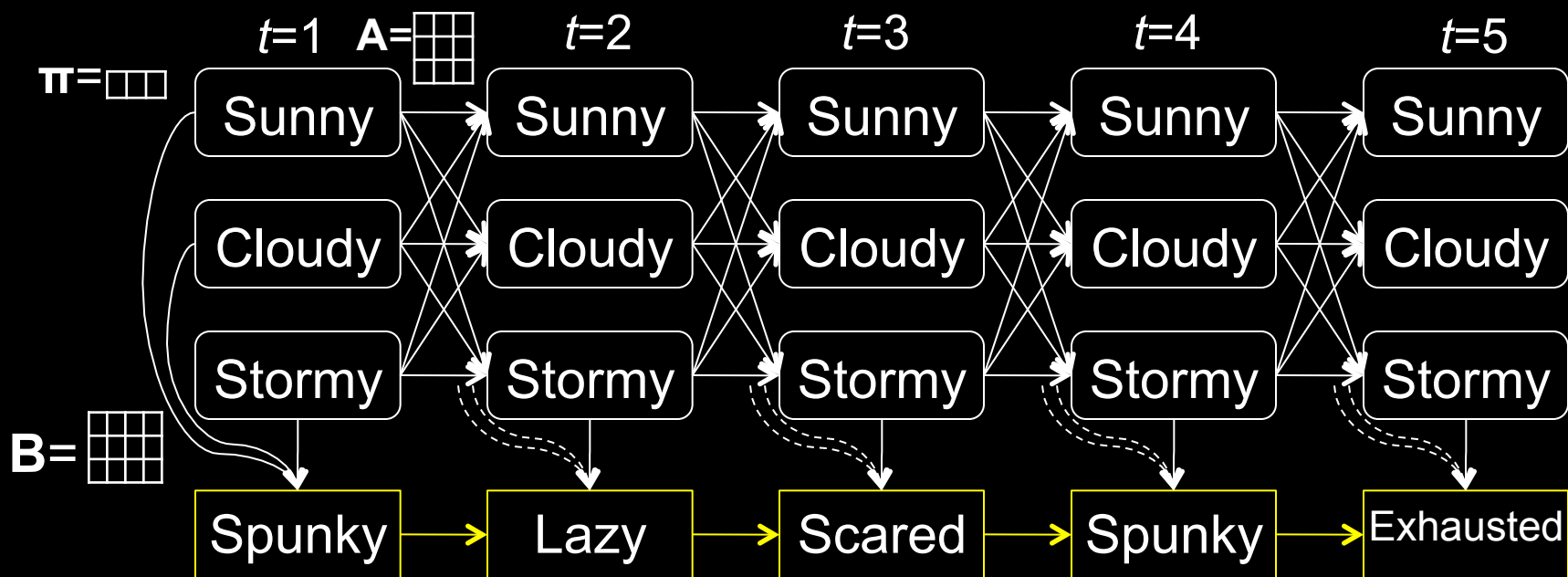
ML Viterbi Algorithm

- Now, we want the highest probability path versus the sum of the probability over all paths
- Each state has a most probable path to it



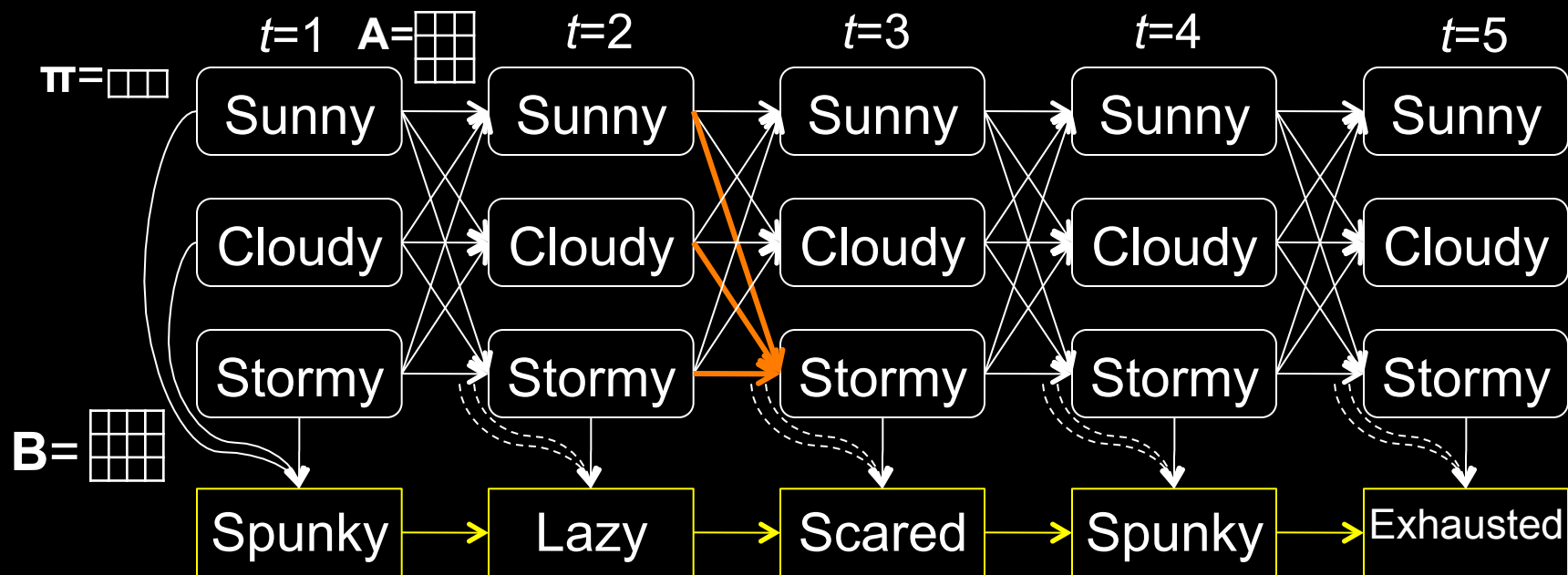
ML Viterbi Algorithm

- Find most probable path to state j_T
- Choose the state with the maximum partial probability and use its partial best path



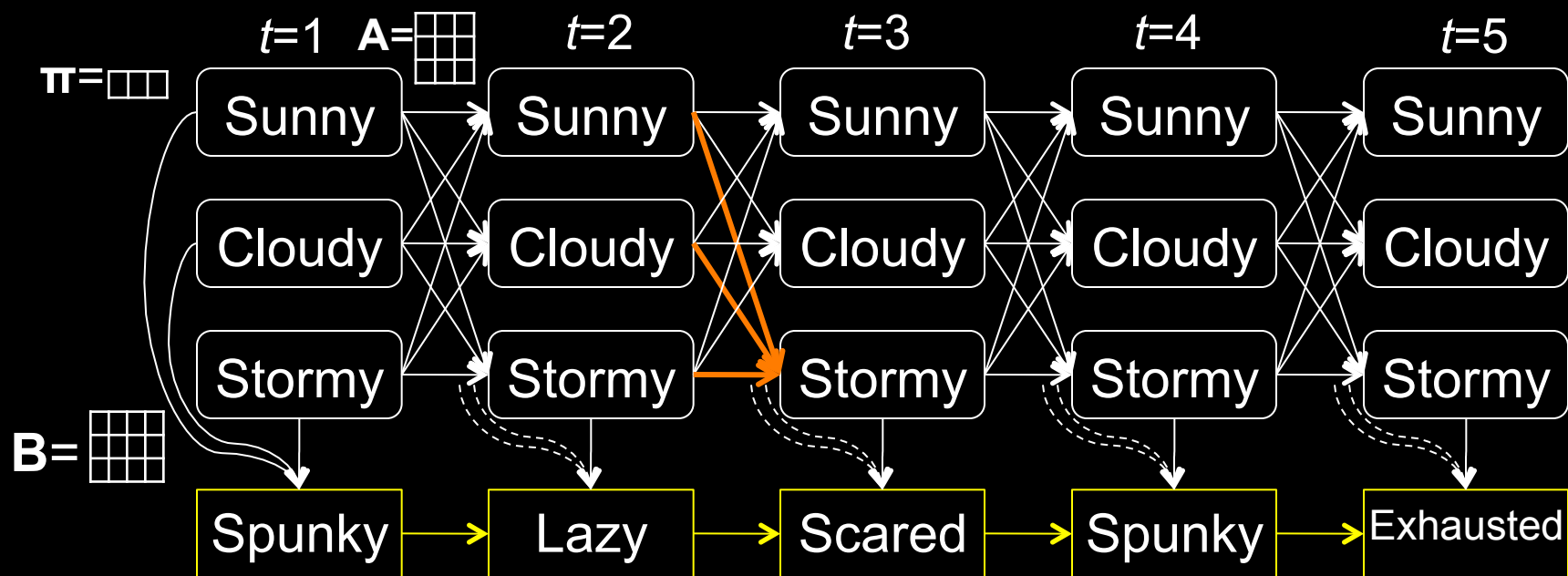
ML Viterbi Algorithm

- $\delta(i_1) = \pi(i)b_{i,k}$
- The most probable path to St_3 is the max of:
- $P(\text{Most probable path to } j_2)P(St | j)$



ML Viterbi Algorithm

- So the probability of the most probable path is:
- $\delta(i_t) = \max_j \delta(j_{t-1}) a_{j,i} b_{i,y_t}$



ML Questions

- Questions???

ML Conference Deadlines

- AAI
 - <http://www.aaai.org/Conferences/AAAI/aaai10.php>
 - January 18, 2010: Electronic abstracts due
 - January 21, 2010: Electronic papers due

ML Presentations

- Mon, Dec 7: Nirav & Kun
- Wed, Dec 9: Tony & Xinyu

ML Projects

- Will stay until all questions are addressed, or
- Please make an appointment if you have any questions at all regarding your project