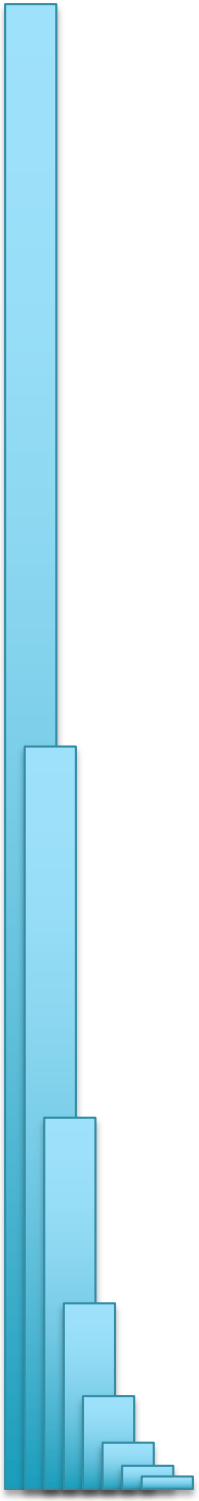
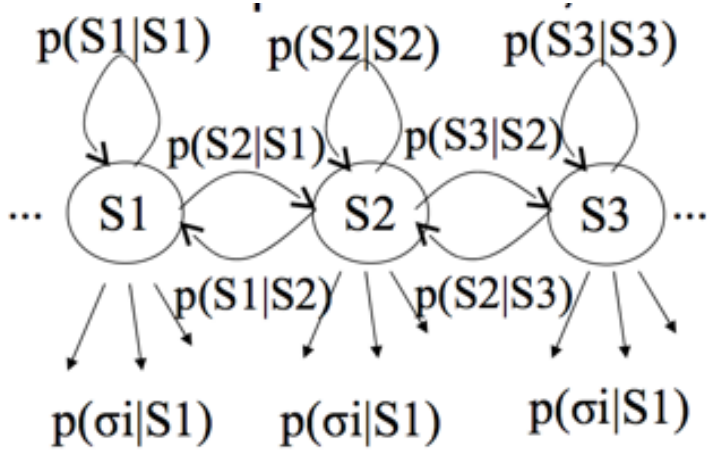


Machine Learning Primer:

Hidden Markov Models

What is an HMM?

- Dynamic Bayesian Network
  - A set of states
    - {Fair, Biased} for coin tossing
    - {Gene, Not Gene} for Bacterial Gene
    - {Intergenic, Exon, Intron} for Eukaryotic Gene
    - {Modern, Neanderthal} for Ancestry



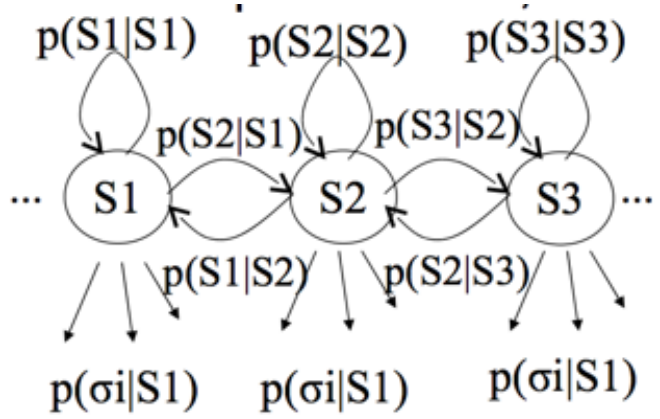
  - A set of emission characters
    - E={H,T} for coin tossing
    - E={1,2,3,4,5,6} for dice tossing
    - E={A,C,G,T} for DNA

  - State-specific emission probabilities
    - P(H | Fair) = .5, P(T | Fair) = .5, P(H | Biased) = .9, P(T | Biased) = .1
    - P(A | Gene) = .9, P(A | Not Gene) = .1 …

  - A probability of taking a transition
    - $P(s_i=Fair|s_{i-1}=Fair) = .9$, $P(s_i=Bias|s_{i-1} = Fair) .1$
    - $P(s_i=Exon | s_{i-1}=Intergenic)$, …

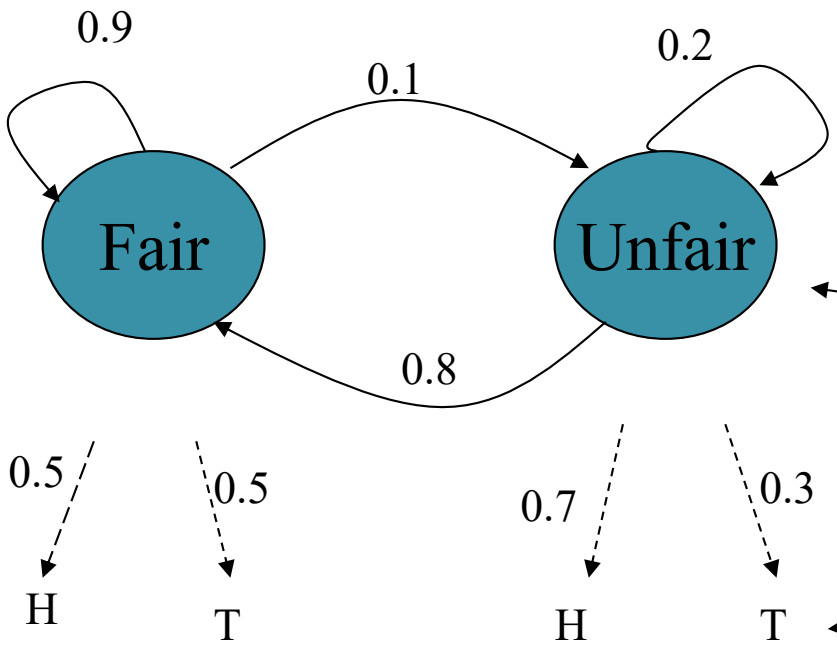
Why Hidden?

- Observers can see the emitted symbols of an HMM (i.e., nucleotides) but have no ability to know which state the HMM is currently in (exon/intron/intergenic/etc).
  - But we can ***infer*** the most likely hidden states of an HMM based on the given sequence of emitted symbols.



AAAGCATGCATTTAACGTGAGCACAATAGATTACA

HMM Example - Casino Coin



State transition probs.

States

Symbol emission probs.

Observation Symbols

Observation Sequence

State Sequence

HTHHTTHHHTHTHTHHTHHHHHHTHTHH

FFFFFFUUUFFFFFFUUUUUUUFFFFFF

Motivation: Given a sequence of H & Ts, can you tell at what times the casino cheated?

Slide credit: Fatih Gelgi, Arizona State U.

Three classic HMM problems

1. **Evaluation**: given a model and an output sequence, what is the probability that the model generated that output?
2. **Decoding**: given a model and an output sequence, what is the most likely state sequence through the model that generated the output?
3. **Learning**: given a model and a set of observed sequences, how do we set the model's parameters so that it has a high probability of generating those sequences?

Three classic HMM problems

1. **Evaluation**: given a model and an output sequence, what is the probability that the model generated that output?

- To answer this, we consider all possible paths through the model

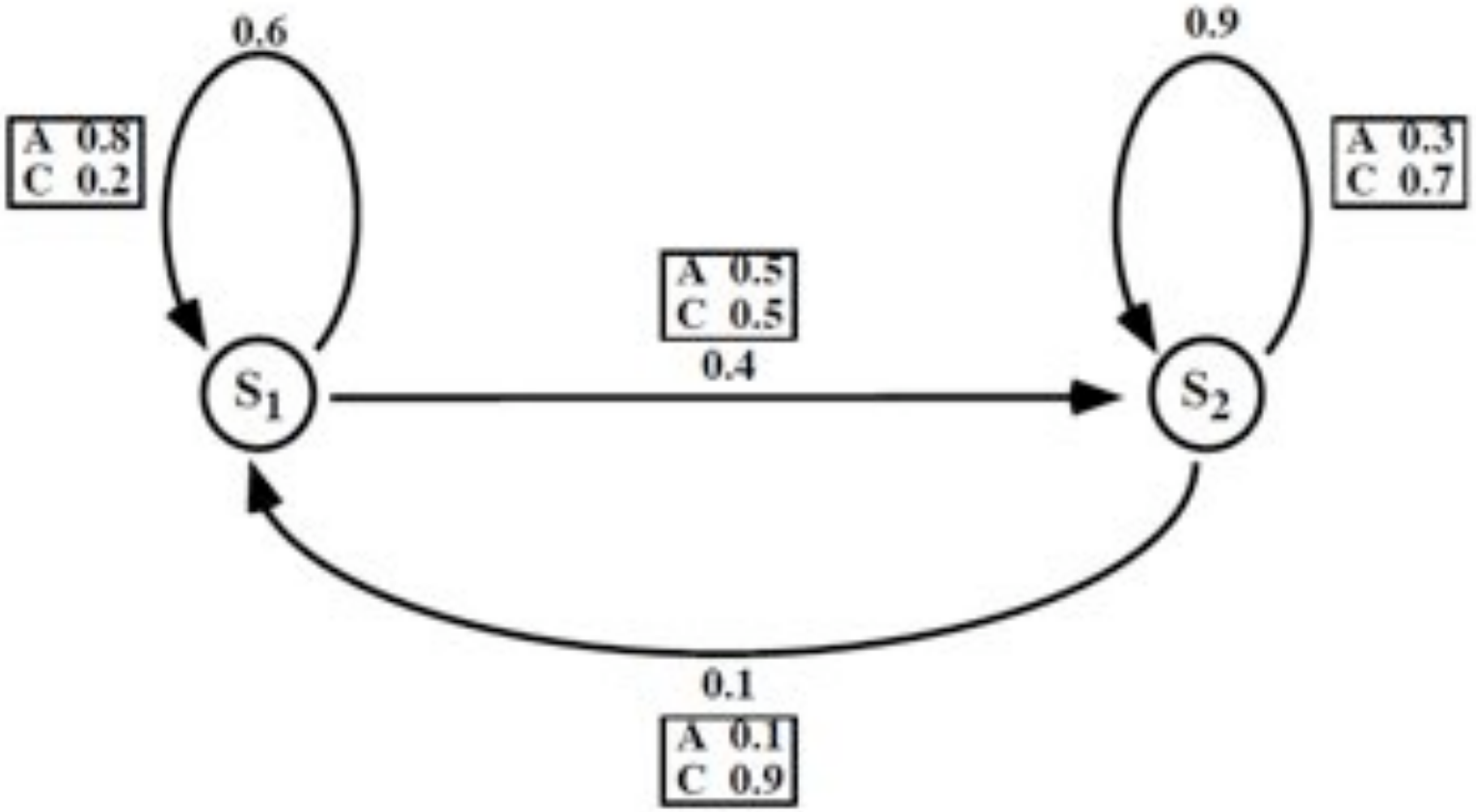
- Example: we might have a set of HMMs representing protein families -> pick the model with the best score

Solving the Evaluation problem: The Forward algorithm

- To solve the Evaluation problem (probability that the model generated the sequence), we use the HMM and the data to build a *trellis*

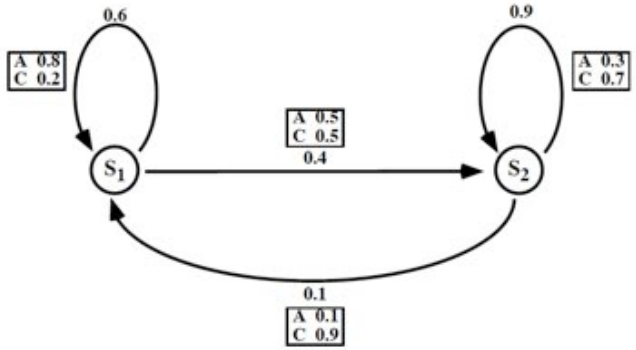
- Filling in the trellis will give tell us the probability that the HMM generated the data by finding all possible paths that could do it
  - Especially useful to evaluate from which models, a given sequence is most likely to have originated

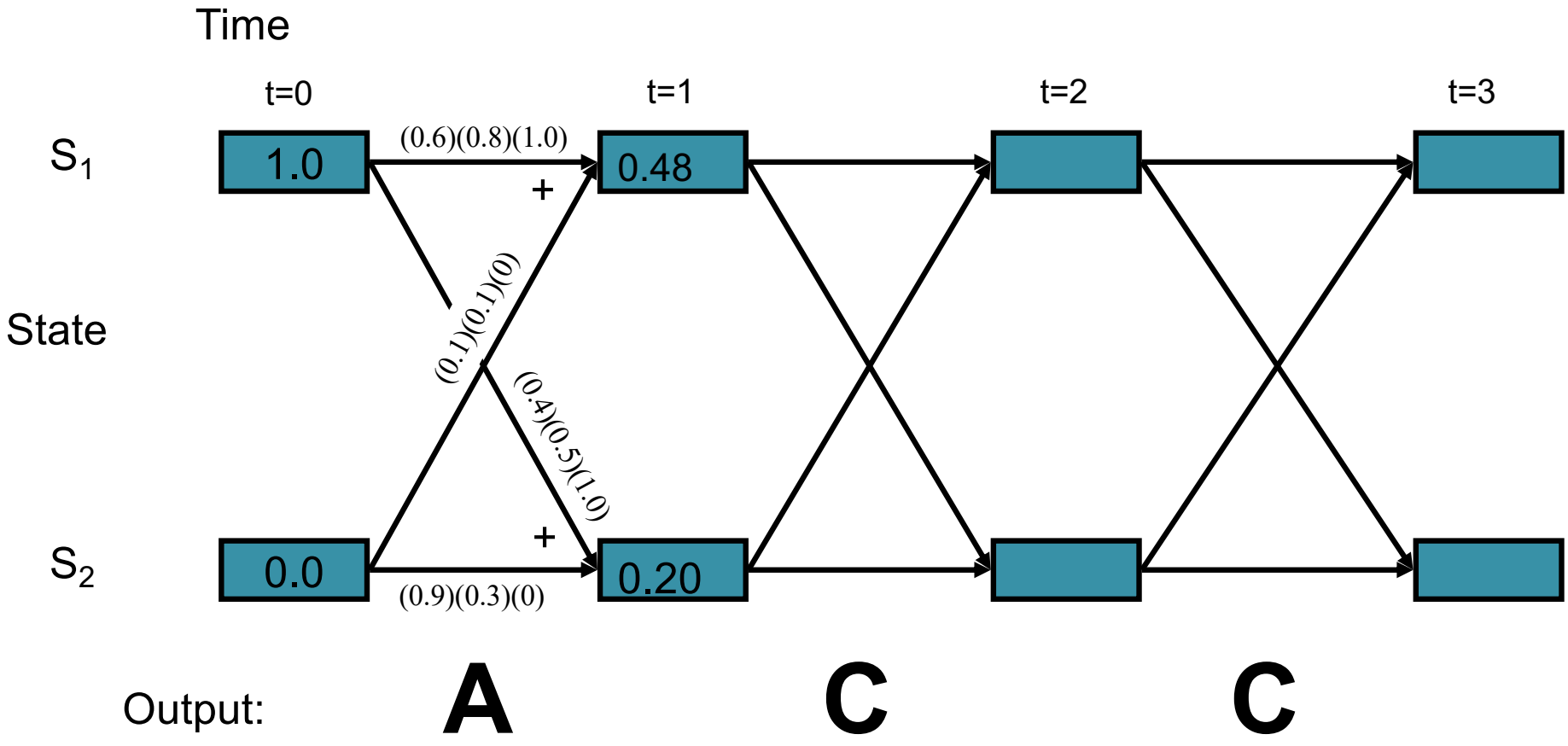
Our sample HMM



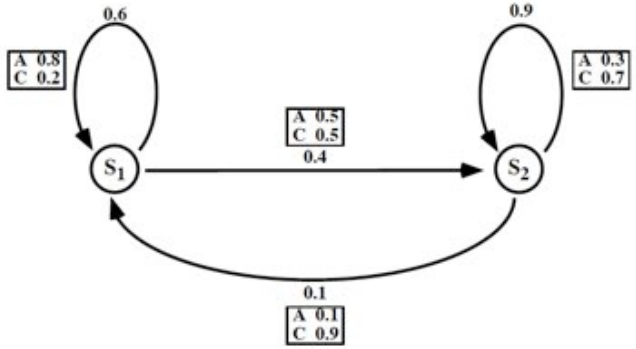
Let S_1 be initial state, S_2 be final state

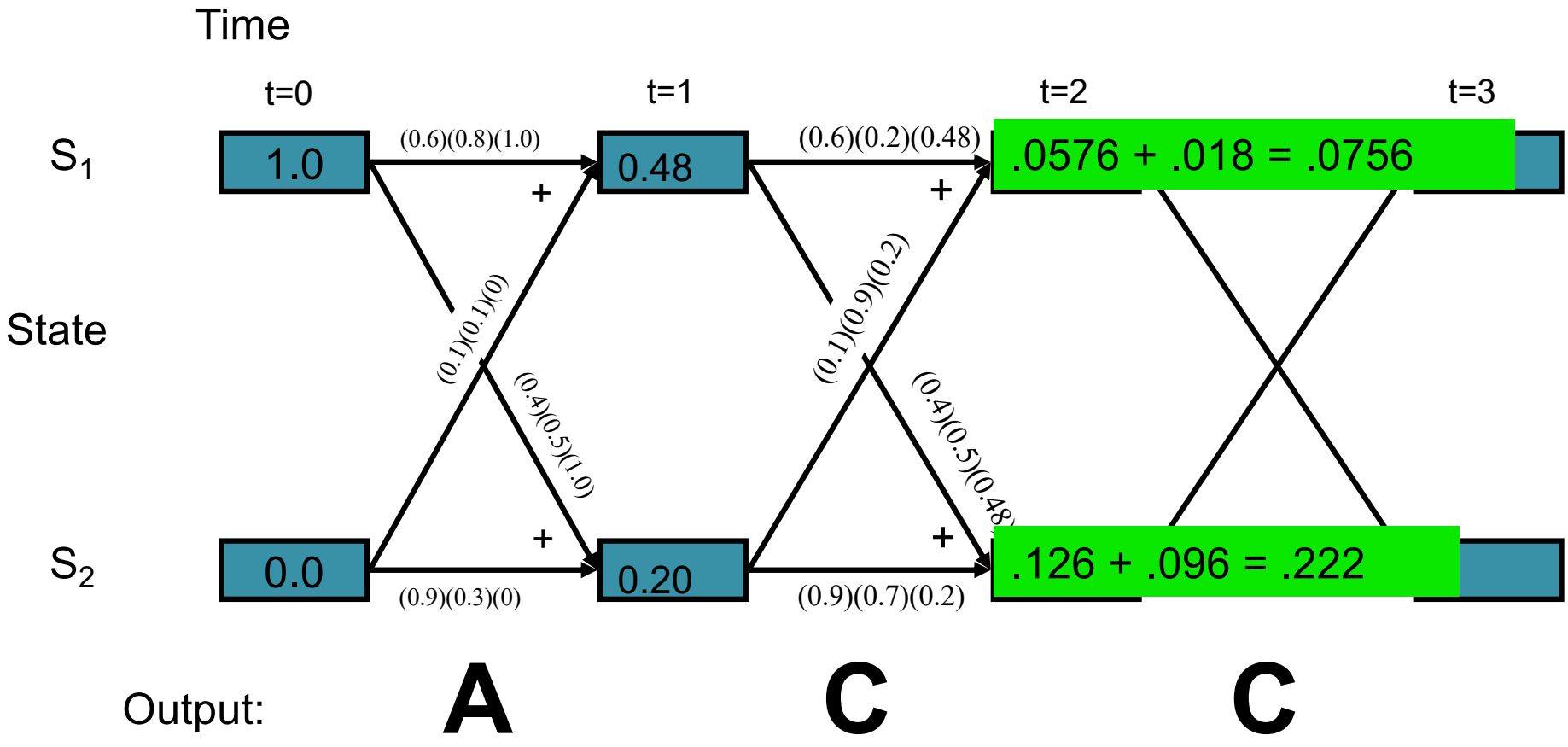
A trellis for the Forward Algorithm



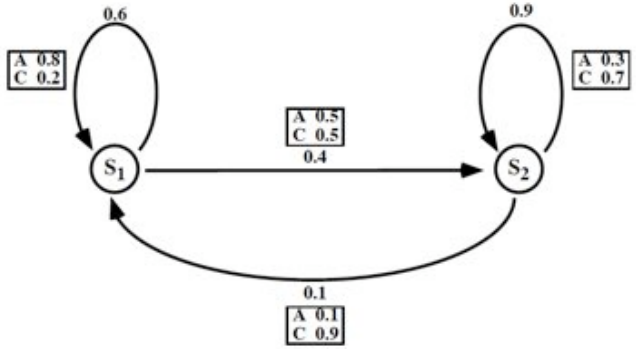


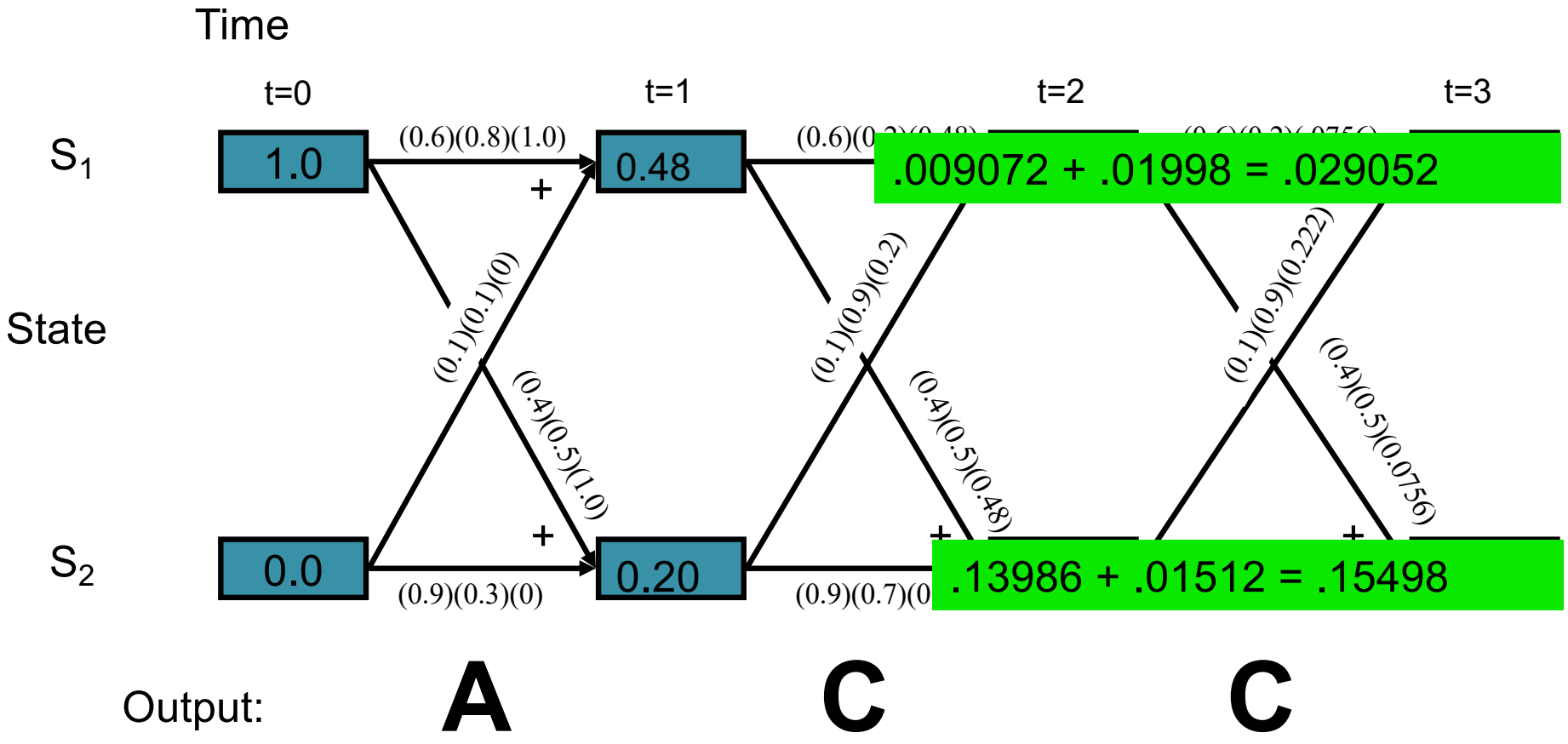
A trellis for the Forward Algorithm



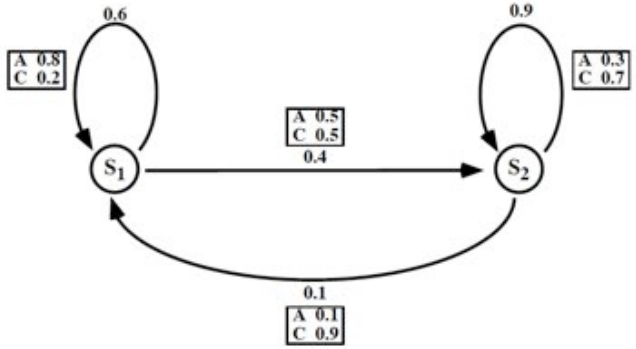


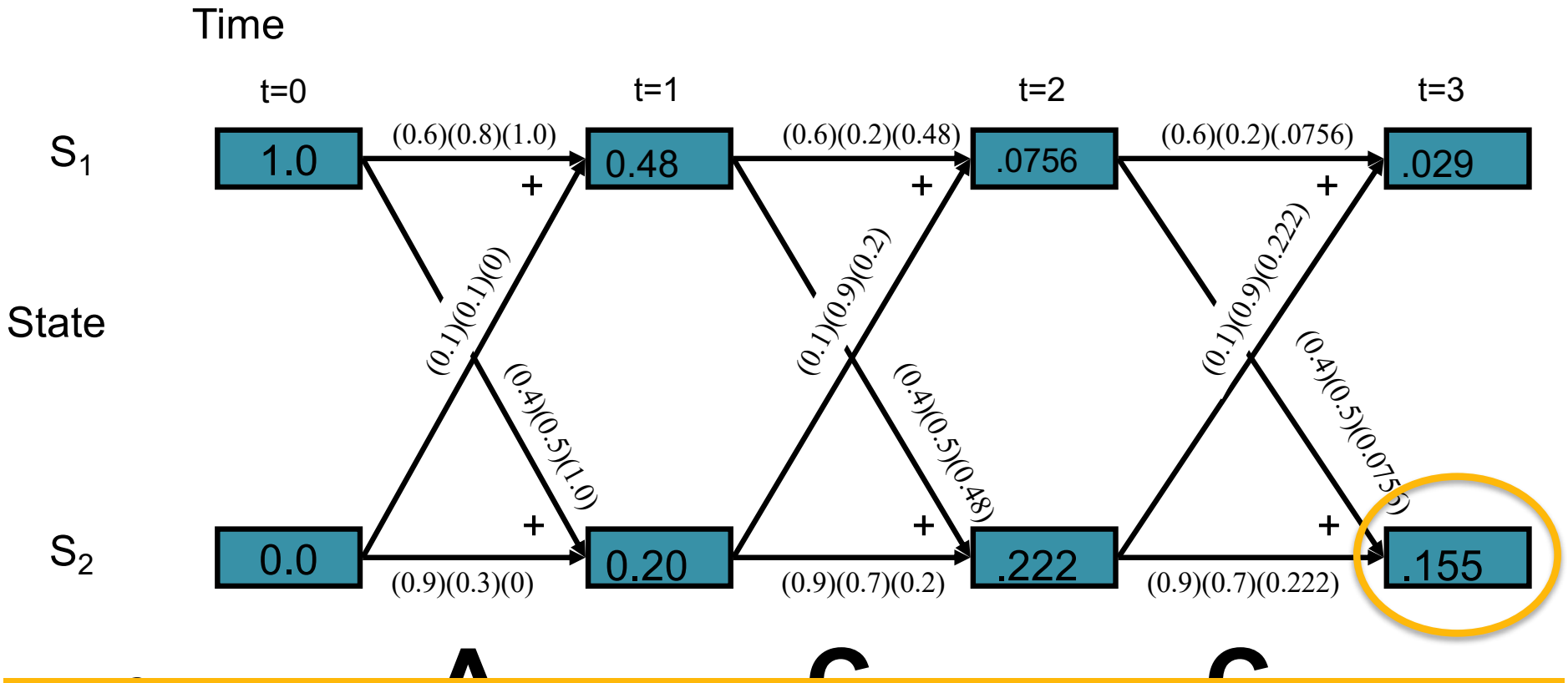
A trellis for the Forward Algorithm





A trellis for the Forward Algorithm





S2 is final state➔ 15.5% probability of this sequence given this model was used

Probability of the model

- The Forward algorithm computes *P(y|M)*

- If we are comparing two or more models, we want the likelihood that each model generated the data: *P(M|y)*

  – Use Bayes' law: $$P(M \mid y) = \frac{P(y \mid M)P(M)}{P(y)}$$

  – Since P(y) is constant for a given input, we just need to maximize *P(y|M)P(M)*

Three classic HMM problems

2. **Decoding**: given a model and an output sequence, what is the most likely state sequence through the model that generated the output?

- A solution to this problem gives us a way to match up an observed sequence and the states in the model.

AAAGCATGCATTTAACGAGAGCACAAGGGCTCTAATGCCG

The sequence of states is an annotation of the generated string – each nucleotide is generated in intergenic, start/stop, coding state

Three classic HMM problems

2. **Decoding**: given a model and an output sequence, what is the most likely state sequence through the model that generated the output?

- A solution to this problem gives us a way to match up an observed sequence and the states in the model.

AAAGC ATG CAT TTA ACG AGA GCA CAA GGG CTC TAA TGCCG

The sequence of states is an annotation of the generated string – each nucleotide is generated in intergenic, start/stop, coding state

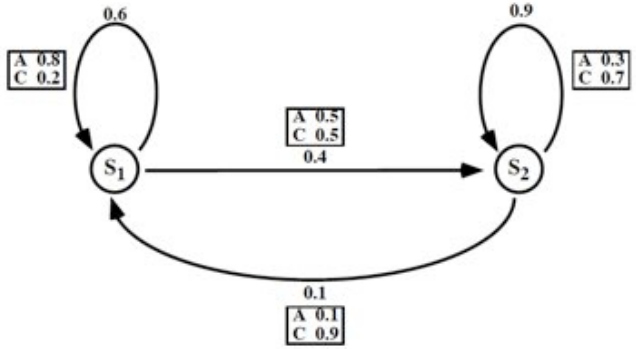
Solving the Decoding Problem: The Viterbi algorithm

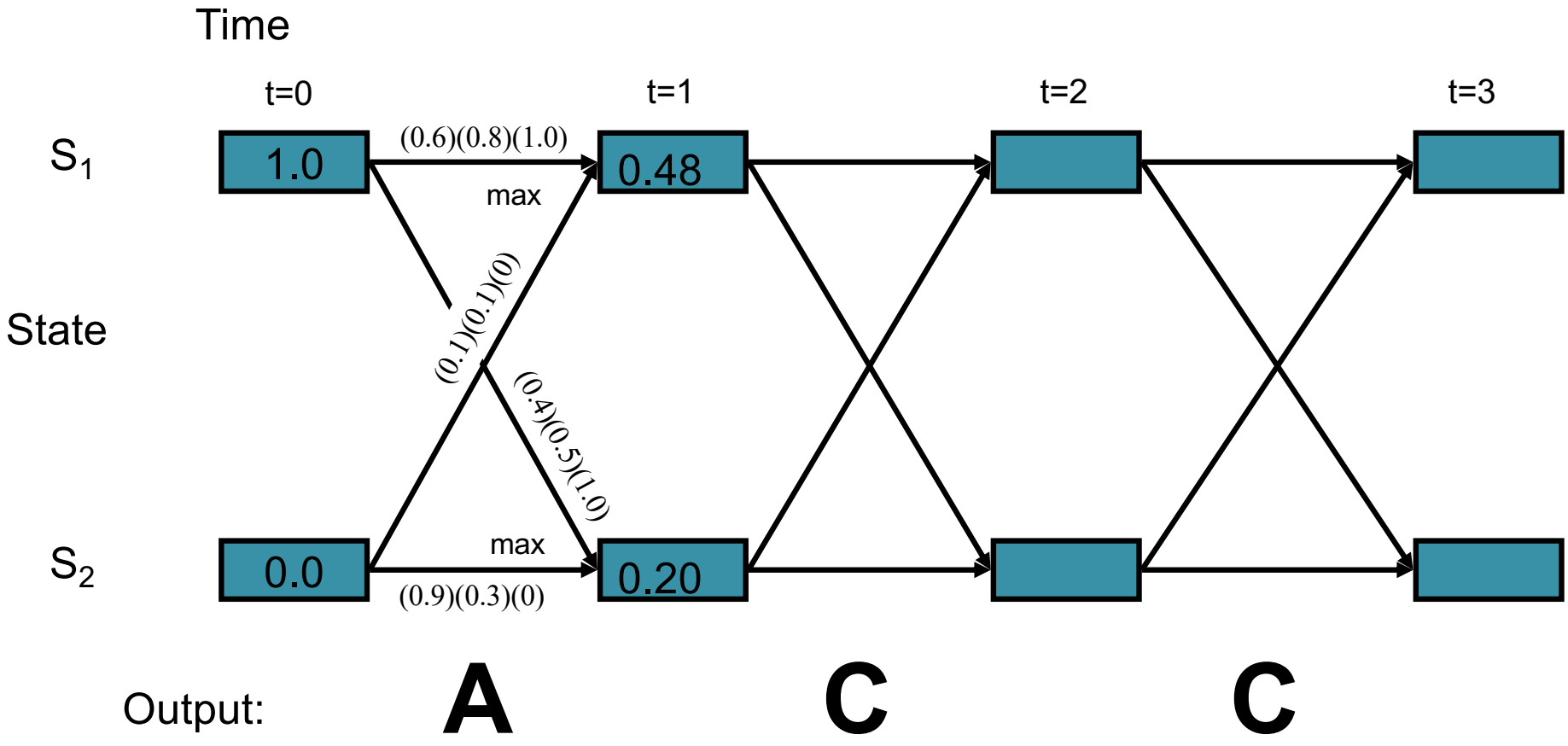
- To solve the decoding problem (find the most likely sequence of states), we evaluate the Viterbi algorithm

$$V_i(t) = \begin{cases} 0 & : \quad t = 0 \wedge i \neq S_I \\ 1 & : \quad t = 0 \wedge i = S_I \\ \max V_j(t-1) a_{ji} b_{ji}(y) & : \quad t > 0 \end{cases}$$

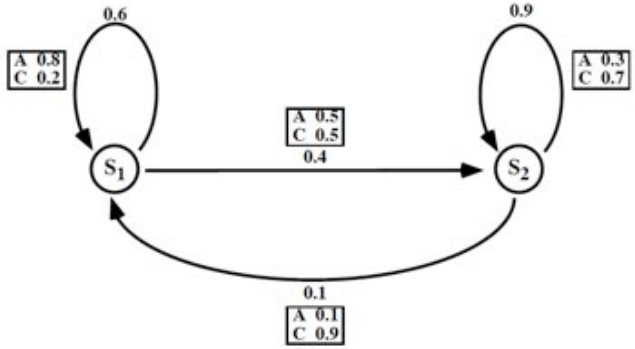
Where $V_i(t)$ is the probability that the HMM is in state i after generating the sequence $y_1, y_2, \ldots, y_t$, following the *most probable path* in the HMM

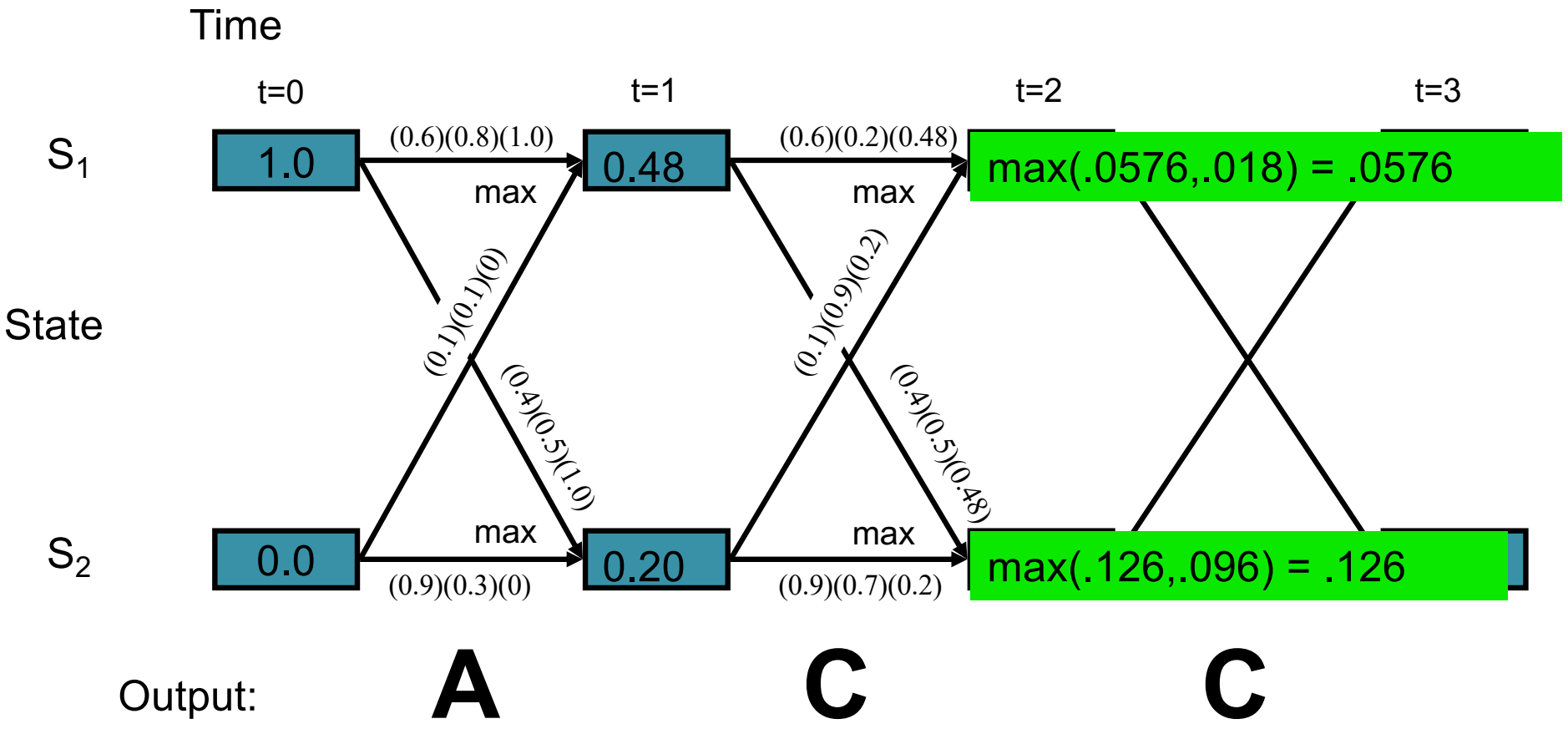
A trellis for the Viterbi Algorithm



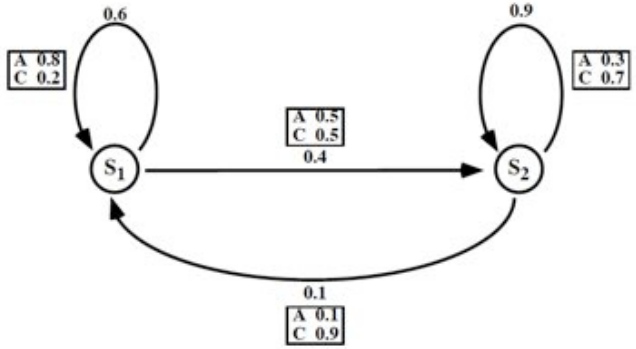


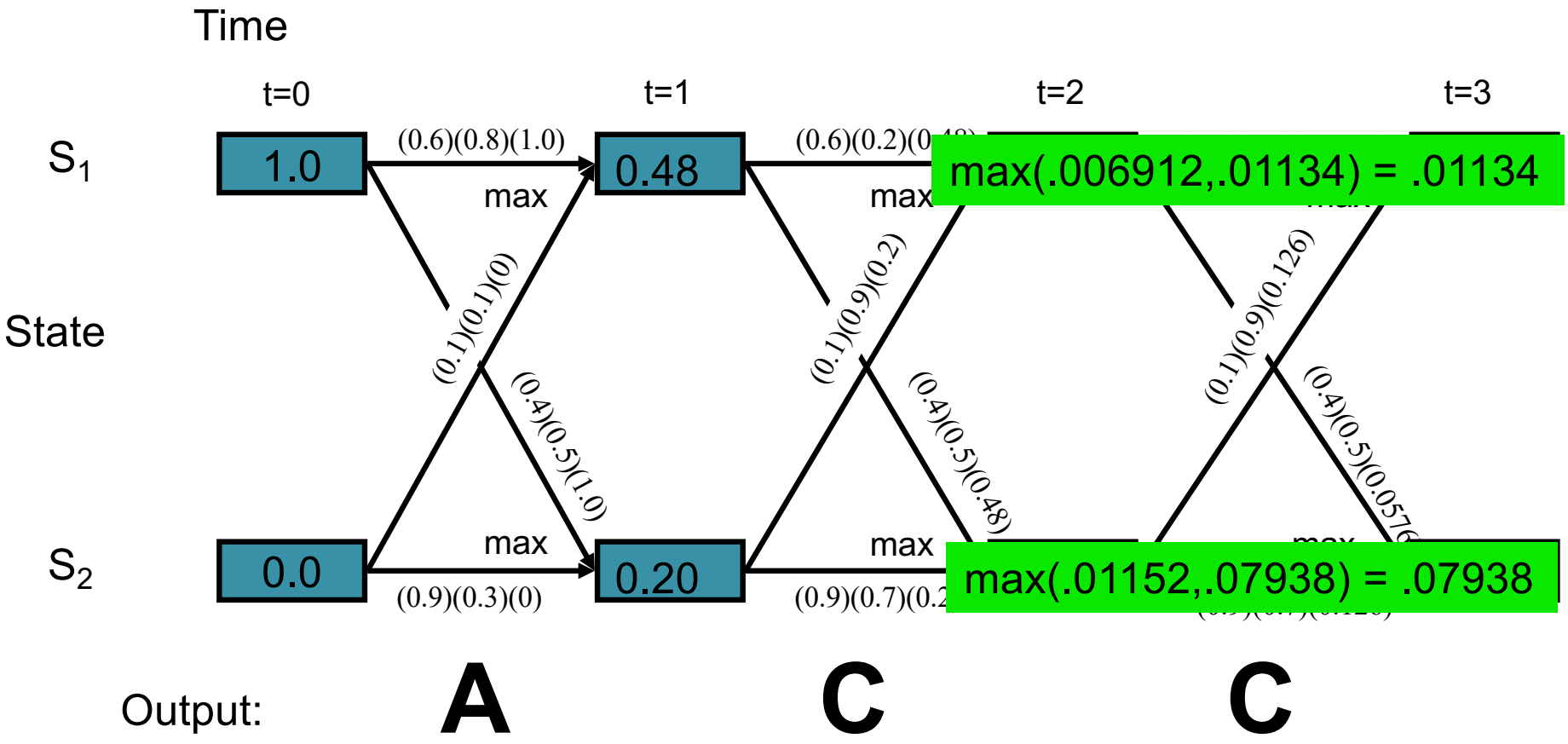
A trellis for the Viterbi Algorithm



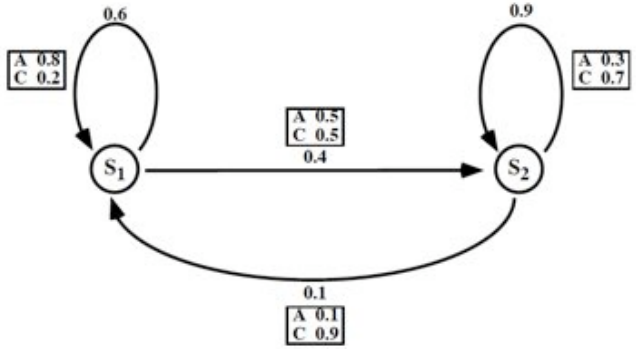


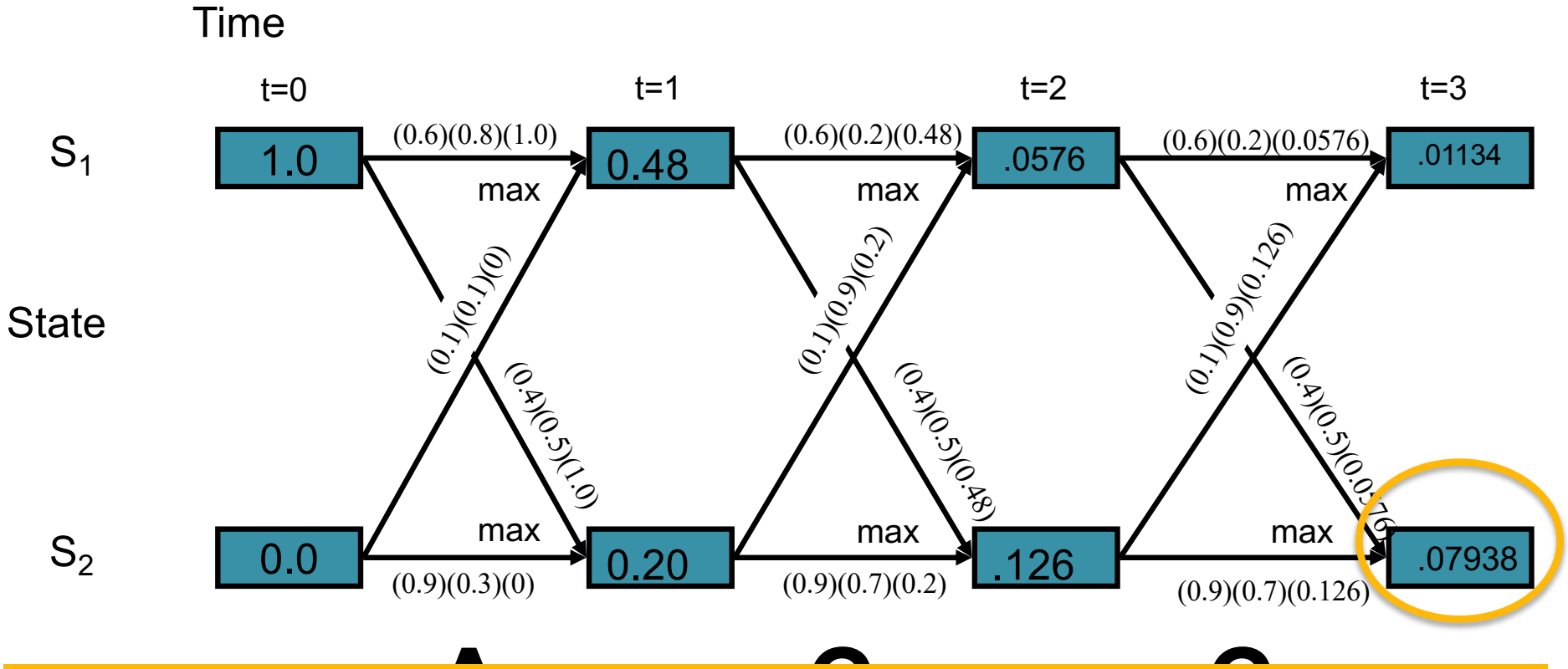
A trellis for the Viterbi Algorithm





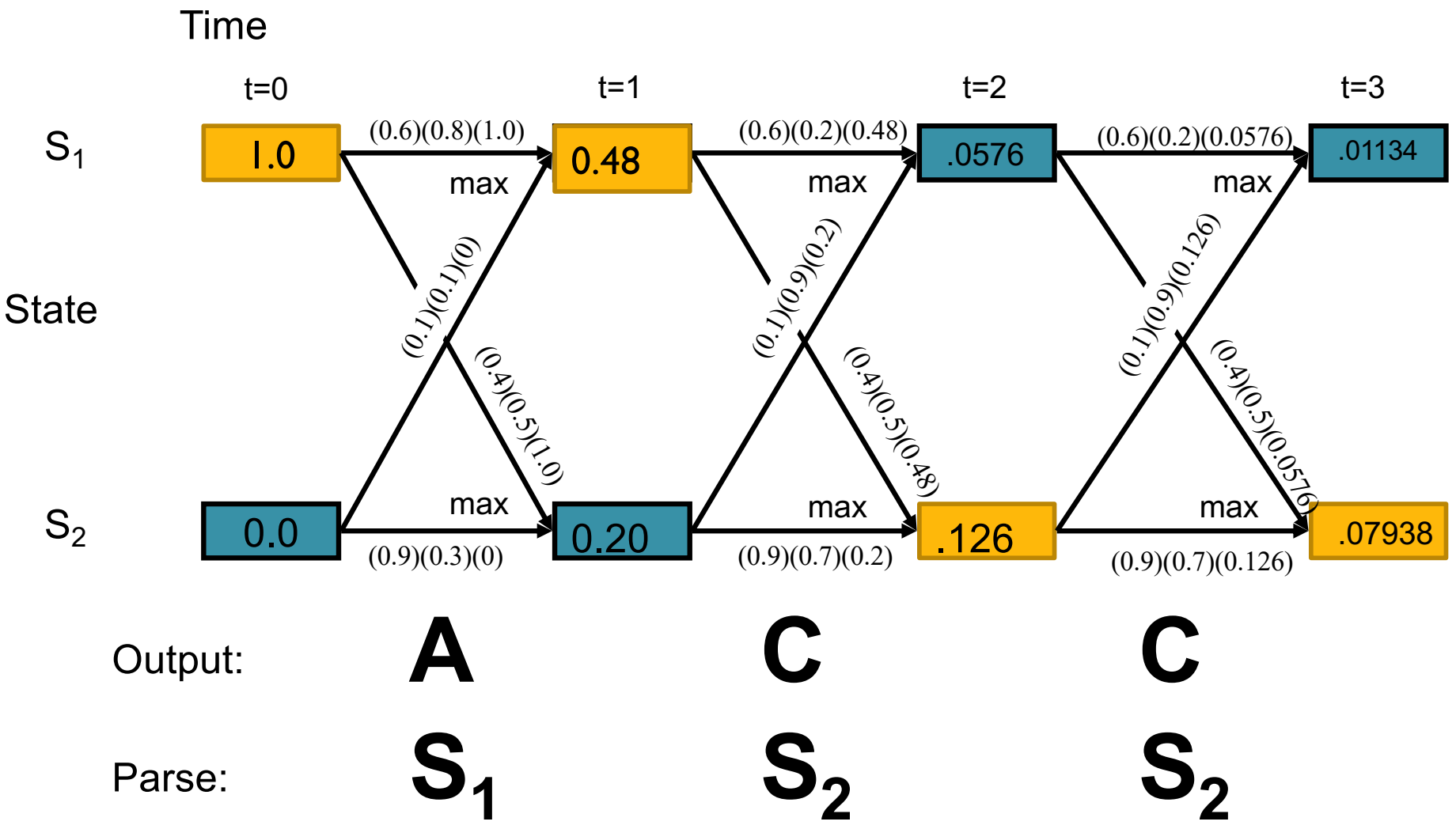
A trellis for the Viterbi Algorithm





S2 is final state➔ the most probable sequence of states has a 7.9% probability

A trellis for the Viterbi Algorithm



Three classic HMM problems

3. **Learning**: given a model and a set of observed sequences, how do we set the model's parameters so that it has a high probability of generating those sequences?

- This is perhaps the most important, and most difficult problem.

- A solution to this problem allows us to determine all the probabilities in an HMMs by using an ensemble of training data

Learning in HMMs:

- The learning algorithm uses Expectation-Maximization (E-M)
  - Also called the Forward-Backward algorithm
  - Also called the Baum-Welch algorithm

- In order to learn the parameters in an "empty" HMM, we need:
  - The topology of the HMM
  - Data - the more the better
  - Start with a random (or naïve) probability, repeat until converges