Comp 555 - BioAlgorithms - Spring 2020

- Problem Set #3 is graded
- Problem Set #5 is On-Line
- What is the most likely sequence to have exhibited a given set of attributes
- States that emit Symbols

Hidden Markov Models

Dinucleotide Frequency

Consider all dimers in a sequence:

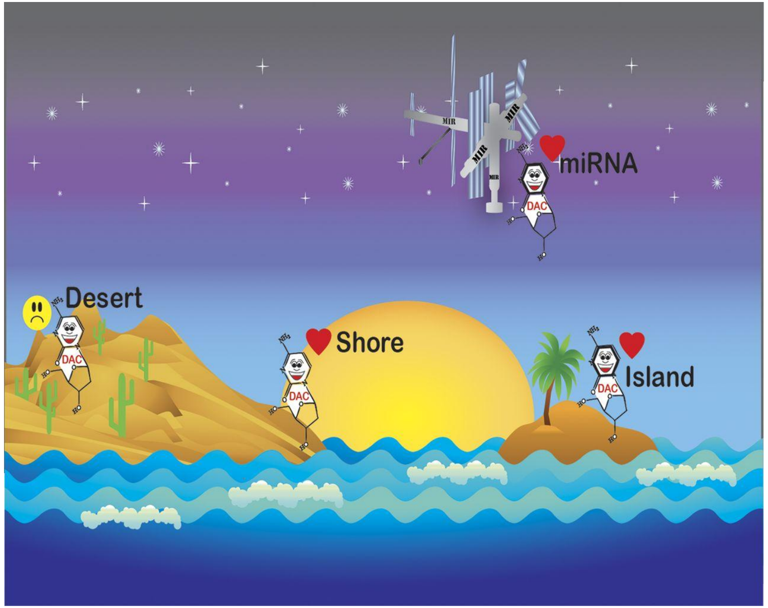
{AA,AC,AG,AT,CA,CC,CG,CT,GA,GC,GG,GT,TA,TC,TG,TT}

Given 4 nucleotides, each with a probability of occurrence of ≈1/4, one would expect that the probability of occurrence of any given dinucleotide is ≈1/16.

However, the frequencies of dinucleotides in DNA sequences vary widely.

In particular, CG is typically underepresented

(frequency of CG is typically << 1/16)



Example

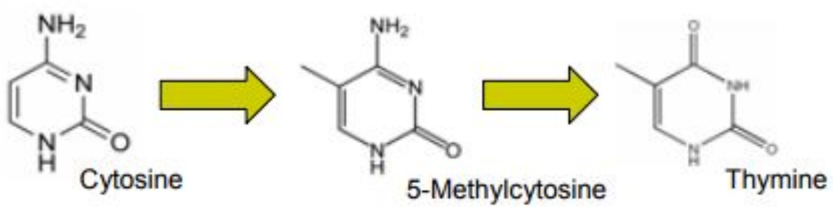
- From a 291829 base sequence

AA	0.120214646984	GA	0.056108392614
AC	0.055409350713	GC	0.037792809463
AG	0.068848773935	GG	0.043357731266
AT	0.083425853585	GT	0.046828954041
CA	0.074369148950	TA	0.077206436668
CC	0.044927148868	TC	0.056207766218
CG	0.008179475581	TG	0.063698479926
CT	0.066857875186	TT	0.096567155996

- Expected value is 0.0625
- CG is 7 times smaller than expected

Why so few CGs?

- CG is the least frequent dinucleotide because C in CG is easily *methylated*.
  And, methylated Cs are easily mutated into Ts.



- However, methylation is suppressed around genes and transcription factor binding sites
- So, CG appears at relatively *higher* frequency in these important areas
- These localized areas of higher CG frequency are called ***CG-islands***
- Finding the CG islands within a genome is among the most reliable gene finding approaches

CG Island Analogy

The CG islands problem can be modeled by a toy problem named *"The Fair Bet Casino"* where the outcome of a casino game is determined by coin flips with two possible outcomes: **H**eads or **T**ails

However, there are two different coins:

A **F**air coin: where **H**eads and **T**ails occur with same probability 1/2.

A **B**iased coin: which lands **H**eads with prob. 3/4,
and **T**ails with prob. 1/4.

The "Fair Bet Casino"

Thus, we define the probabilities:

- *P(H|Fair) = 1/2, P(T|Fair) = 1/2*
- *P(H|Bias) = 3/4, P(T|Bias) = 1/4*

Also, the house doesn't want to get caught switching between coins, so they do so infrequently. Thus, swaps between Fair and Biased coins occur with probability 1/10.

Fair Bet Casino Problem

Input: A sequence $x = x_1, x_2, x_3, \ldots, x_n$ of observed coin tosses made by some combination of the two possible coins (**F** or **B**).

Output: A sequence $\pi = \pi_1, \pi_2, \pi_3, \ldots, \pi_n$, with each π_i being either **F** or **B** indicating that x_i was the result of tossing the Fair or Biased coin respectively.

Problem Subtleties

- Any observed outcome of coin tosses *could* have been generated by *any combination* of coin exchanges
- However, all coin-exchange combinations are not equally likely.

```
Tosses: T H H H H
Coins1: F F F F F    P(Tosses|Coins1) = 1/2 1/2 1/2 1/2 1/2 =  1/32    = 0.03125
Coins2: B B B B B    P(Tosses|Coins2) = 1/4 3/4 3/4 3/4 3/4 = 81/1024 = 0.0791
Coins3: F F B B B    P(Tosses|Coins3) = 1/2 1/2 3/4 3/4 3/4 = 81/256  = 0.3164
```

- We ask, "What ***coin-exchange combination*** has the highest probability of generating the observed series of tosses?"
- The coin tosses are a signal, and figuring out the most likely coin-exchange sequence is a *Decoding Problem*

Let's consider the extreme cases

Suppose that the dealer *never* exchanges coins.

Some definitions:

- *P(x|Fair)*: prob. of generating the *x* using the Fair coin.
- *P(x|Biased)*: prob. of generating *x* using the Biased coin.

We can compute the probability of any set of tosses under these two given conditions, and ask which case was more likely.

$P(x|fair\ coin)$ vs. $P(x|biased\ coin)$

And the expressions look like:

$$P(x|Fair) = P(x_1 \ldots x_n|Fair) = \prod_{i=1,n} p(x_i|Fair) = \left(\frac{1}{2}\right)^n$$

$$P(x|Biased) = P(x_1 \ldots x_n|Biased) = \prod_{i=1,n} p(x_i|Biased) = \left(\frac{3}{4}\right)^k \left(\frac{1}{4}\right)^{n-k} = \frac{3^k}{4^n}$$

Where k is the number of Heads observed in x

P(x|fair coin) = P(x|biased coin)

When is a sequence equally likely from *either* the Fair or Biased coin?

$$P(x|Fair) = P(x|Biased)$$
$$\left(\frac{1}{2}\right)^n = \frac{3^k}{4^n}$$
$$2^n = 3^k$$
$$n = k\log_2 3$$

When $k = \frac{n}{\log_2\ 3}$ $\quad (k \approx 0.63n)$

So, when the number of heads over a contiguous sequence of tosses is greater than 63% the dealer is most likely used the biased coin.

Log-odds Ratio

- We can define a *log-odds ratio* as follows:

$$\log_2\left(\frac{P(x|Fair)}{P(x|Biased)}\right) = \sum_{i=1}^{k} \log_2\left(\frac{P(x_i|Fair)}{P(x_i|Biased)}\right)$$
$$= n - k \log_2 3$$

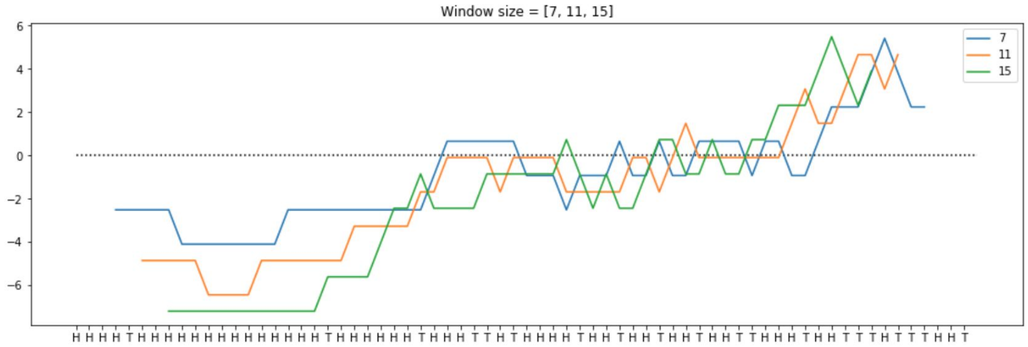
- The log-odds ratio is a means (threshold) for deciding which of two alternative hypotheses is most likely
- *"Zero-crossing"* measure:
  a. If the log-odds ratio is > 0 then the numerator (Fair coin) is more likely
  b. if the log-odds ratio is < 0 then the denominator (Biased coin) is more likely
  c. They are equally likely if the log-odds ratio = 0

Log-odds Ratio over a sliding window

Given a sequence of length n, consider the log-odds ratio of a sliding window of length $w << n$

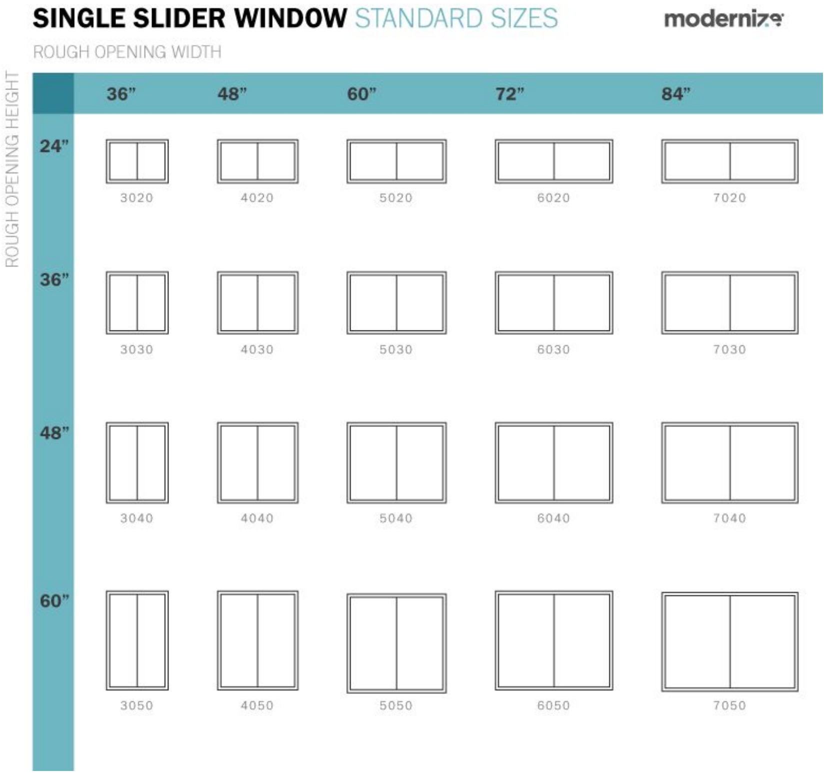
$$x_1, x_2, \underline{x_3, x_4, x_5, x_6, x_7}, x_8, x_9, \ \ldots, \ x_n$$

In [16]:
```python
import numpy
import matplotlib.pyplot as plot
%matplotlib inline

rolls = "HHHHTHHHHHHHHHHHHHHTHHHHHHTHHHTTHTHHHHTHHTHHTHHTHHTTHHHTHHTTTHTTTHHT"

windows = [7, 11, 15]
slope = numpy.log2(3)
yplots = [[w - slope*rolls[i:i+w].count('H') for i in range(len(rolls)-w+1)] for w in windows]

fig, ax = plot.subplots(figsize=(16,5))
plot.title("Window size = %s" % windows)
plot.hlines(0, 0, len(rolls), linestyles='dotted')
for i, y in enumerate(yplots):
    x = range(windows[i]//2, windows[i]//2+len(y))
    plot.plot(x,y, label=str(windows[i]))
plot.legend()
ax.set_xticks([i for i in range(len(rolls))])
result = ax.set_xticklabels([c for c in rolls])
```



Disadvantages of windowed Log-odds

- An appropriate window size (i.e. length of CG-island) is not known in advance.
- Different window sizes may classify the same position differently.
- What about the rule that they don't swap out the coins frequently?

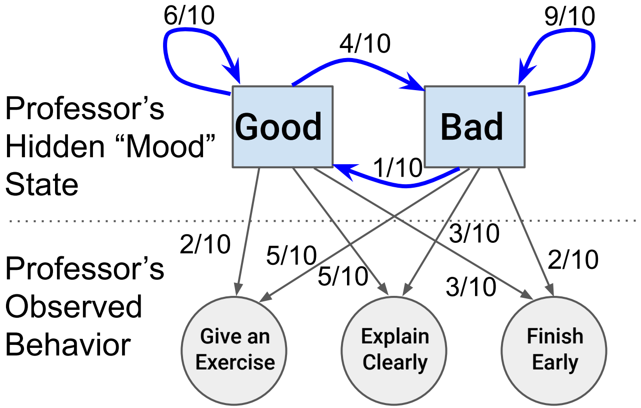


Key Elements of the Problem

- There is an unknown or *hidden* state for each observation (Was the coin the Fair or Biased?)
- Outcomes are modeled probabilistically:
  - $P(H|Fair) = \frac{1}{2}, \quad P(T|Fair) = \frac{1}{2}$
  - $P(H|Bias) = \frac{3}{4}, \quad P(T|Bias) = \frac{1}{4}$
- *Transitions* between states are modeled probabilistically:
  - $P(\pi_i = Bias \mid \pi_{i-1} = Bias) = a_{BB} = 0.9$
  - $P(\pi_i = Bias \mid \pi_{i-1} = Fair) = a_{FB} = 0.1$
  - $P(\pi_i = Fair \mid \pi_{i-1} = Bias) = a_{BF} = 0.1$
  - $P(\pi_i = Fair \mid \pi_{i-1} = Fair) = a_{FF} = 0.9$

Hidden Markov Model (HMM)

- A generalization of this class of problem
- Can be viewed as an abstract machine with *k hidden* states that emits symbols from an alphabet Σ.
- Each state emits outputs with its own probability distribution, and the machine switches between states according to some other probability distribution.
- While in a certain state, the machine makes 2 decisions:
  - What symbol from the alphabet Σ should I emit?
  - What state should I move to next?



Why “Hidden”?

- Observers see the emitted symbols of an HMM, but can’t see which state the HMM is currently in.
- Thus, the goal is to infer the most likely hidden states of an HMM based on the given sequence of emitted symbols.

HHHTHTHHTTTTHTHTHTHHHTHTHTHT
BBBFFFFFFFFFFFFFFFFBBBFFFFFF?

In-class Exercise

Login to your course account and try the given "*Fair Bet Casino*" exercise.

HMM Parameters

- **Σ**: set of emission characters.
  Example: Σ = {0, 1} for coin tossing
  - Σ = {Heads, Tails}
  - Σ = {1, 2, 3, 4, 5, 6} for dice tossing
- **Q:** set of hidden states, emitting symbols from Σ.
  - Q = {Fair,Bias} for coin tossing
  - Q = {Good Mood, Bad Mood}

HMM for Fair Bet Casino

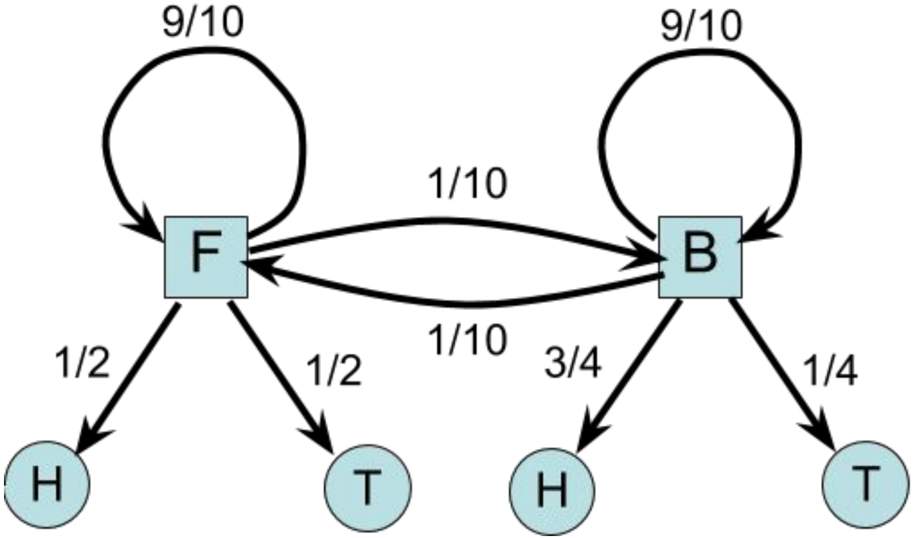
- The Fair Bet Casino in HMM terms:
  - $\Sigma = \{0, 1\}$ (0 for Tails and 1 Heads)
  - $Q = \{F,B\}$ – F for Fair & B for Biased coin

- Transition Probabilities ***A***, *Emission Probabilities* ***E***

A	Fair	Biased
Fair	9/10	1/10
Biased	1/10	9/10

E	Tails(0)	Heads(1)
Fair	½	½
Biased	¼	¾

HMM as a Graphical Model



Directed graph with two types nodes and two types of edges

- hidden states are shown as squares
- emission outputs are shown as circles
- transition edges
- emission edges

Hidden Paths

- A path **π** = $\pi_1 \ldots \pi_n$ in the HMM is defined as a sequence of hidden states.
- Consider
  - path π = FFFBBBBBFFF
  - sequence x = 01011101001

$$
\begin{array}{lccccccccccccc}
x & = & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
\pi & = & F & F & F & B & B & B & B & B & F & F & F \\
P(x_i|\pi_i) & & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{3}{4} & \frac{3}{4} & \frac{3}{4} & \frac{1}{4} & \frac{3}{4} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
P(\pi_i \to \pi_{i+1}) & & \frac{9}{10} & \frac{9}{10} & \frac{1}{10} & \frac{9}{10} & \frac{9}{10} & \frac{9}{10} & \frac{9}{10} & \frac{1}{10} & \frac{9}{10} & \frac{9}{10} &
\end{array}
$$

- What is the probability of the given path (FFFBBBBBFFF)?

P(x|π) calculation

- **P(x|π):** Probability that sequence ***x*** *was generated by the path* ***π***:

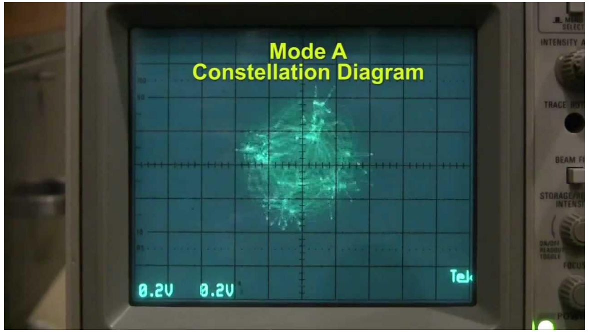
$$P(x|\pi) = \prod_{i=1}^{n} P(x_i|\pi_i) \cdot P(\pi_i \rightarrow \pi_{i+1})$$

$$= \prod_{i=1}^{n} E_{\pi_i,x_i} \cdot A_{\pi_i,\pi_{i+1}}$$

- How many such paths exist? 2^n
- What algorithmic approach would you use to find the best path?
    - Branch and Bound?
    - Divide and Conquer?
    - Dynamic Programming?

Decoding Problem

Finding the optimal path in a graph is equivalent to a classical problem of decoding a message using *constellations*. This is very commonly used when a discrete set of symbols is encoded for transport over an analog medium (e.x. modems, wired internet, wireless internet, digital television).



A simple binary coding does not make good use of the dynamic range of a digital signal, however, if you put the codes too close noise becomes a problem.

- **Goal:** Find an optimal hidden path of state transitions given a set of observations.
- **Input:** Sequence of observations x = x1…xn generated by an HMM M(Σ, Q, A, E)
- **Output:** A path that maximizes P(x|π) over all possible paths π.

How do we solve this?

- Brute Force approach:
  - Enumerate every possible path
  - Compute $P(x_{1..n}|\pi_{1..n})$ for each one
  - Keep track of the most probable path
- A better approach:
  - Break any path in two parts, $P(x_{1..i}|\pi_{1..i}), P(x_{i..n}|\pi_{i..n})$
  - $P(x_{1..n}|\pi_{1..n}) = P(x_{1..i}|\pi_{1..i}) \times P(x_{i..n}|\pi_{i..n})$
  - Will less than the highest $P(x_{1..i}|\pi_{1..i})$ ever improve the total probability?
  - Thus to find the maximum $P(x_{1..n}|\pi_{1..n})$ we need find the maximum of each subproblem $P(x_{1..i}|\pi_{1..i})$, for i from 1 to n
  - What algorithm design approach does this remind you of?

Building Manhattan for Decoding

In 1967, Andrew Viterbi developed a "Manhattan-like grid" (Dynamic program) model to solve the Decoding Problem.

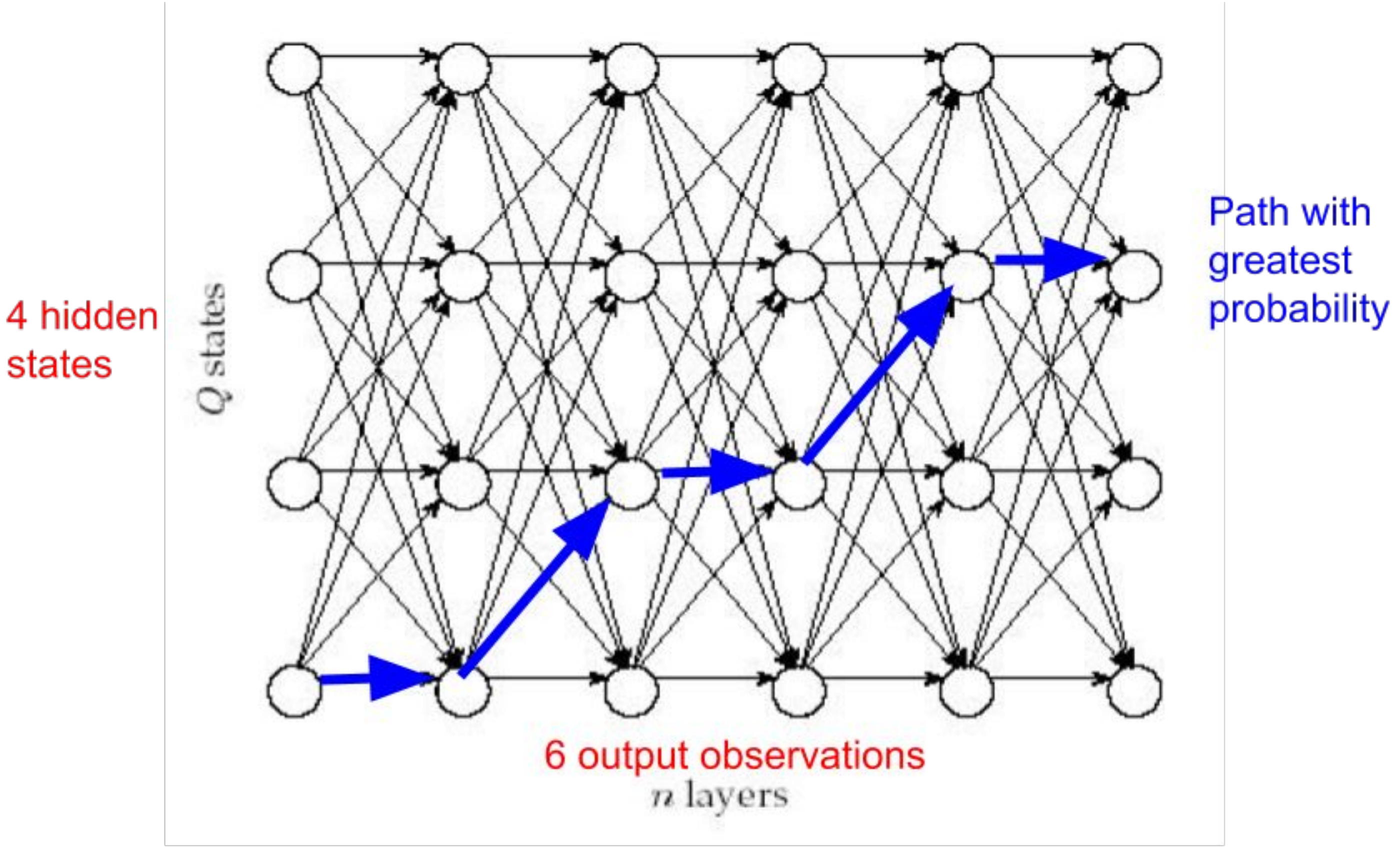
Every choice of $\pi=\pi_1,\pi_2,\ldots\pi_n$ corresponds to a path in the graph.

The only valid direction in the graph is eastward.

This graph has $|Q|^2(n-1)$ edges.

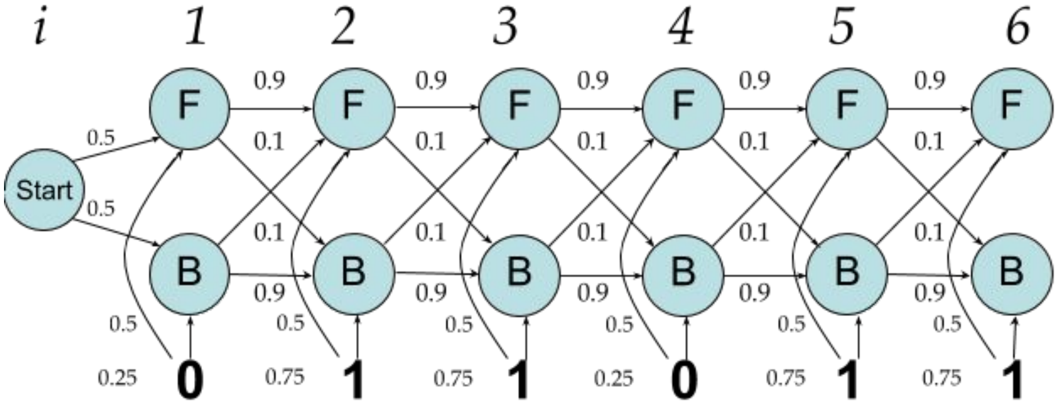
Where Q is the number of states and n is the path length

Graph of Decoding Problem



Viterbi Decoding of Fair-Bet Casino

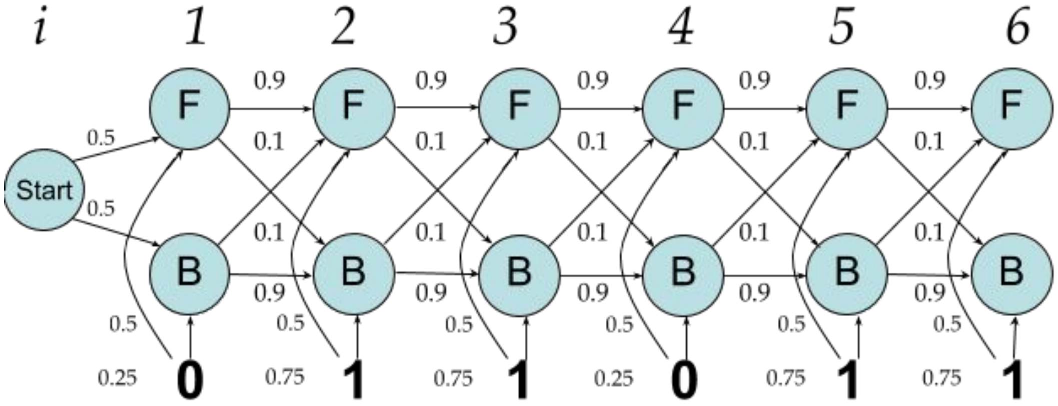
- Each vertex represents a possible state at a given position in the output sequence
- The observed sequence conditions the likelihood of each state
- Dynamic programming reduces search space to:
  $|Q|^2 \times (n-1) = 2^2 \times 5 = 20$ from naïve $2^5 = 32$



- What if n=50? $2^2 \times 50 = 200$ vs. $2^{50} = 1{,}125{,}899{,}906{,}842{,}624$

Decoding Problem Solution

- The *Decoding Problem* is equivalent to finding a longest path in the *directed acyclic graph* (DAG), where "longest" is defined as the maximum product of the probabilities along the path.



Next Time

- We'll find the DP recurrance equations
- See examples and what Viterbi looks like in code
- See how truth and maximum likelihood do not always agree
- Apply to HMMs to problems of biological interest.