Applied Machine Learning

HMM Inference

Hidden Markov Models - Inference

- Inference in HMMs
- Trellis
- Viterbi algorithm for Inference in HMMs

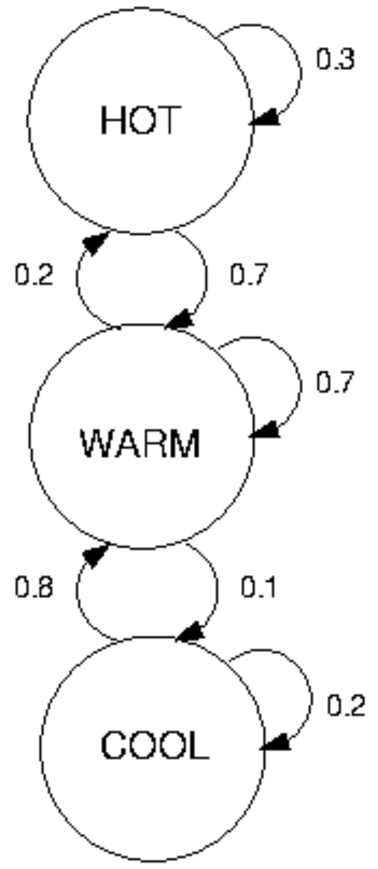
HMMs: Inference

- Inference
  - Estimate sequence of hidden states X_i for known HMM for sequence of observations Y_i
- Maximum a Posteriori Inference or MAP Inference
  - sequence of hidden states $X_1, X_2, \ldots, X_N$

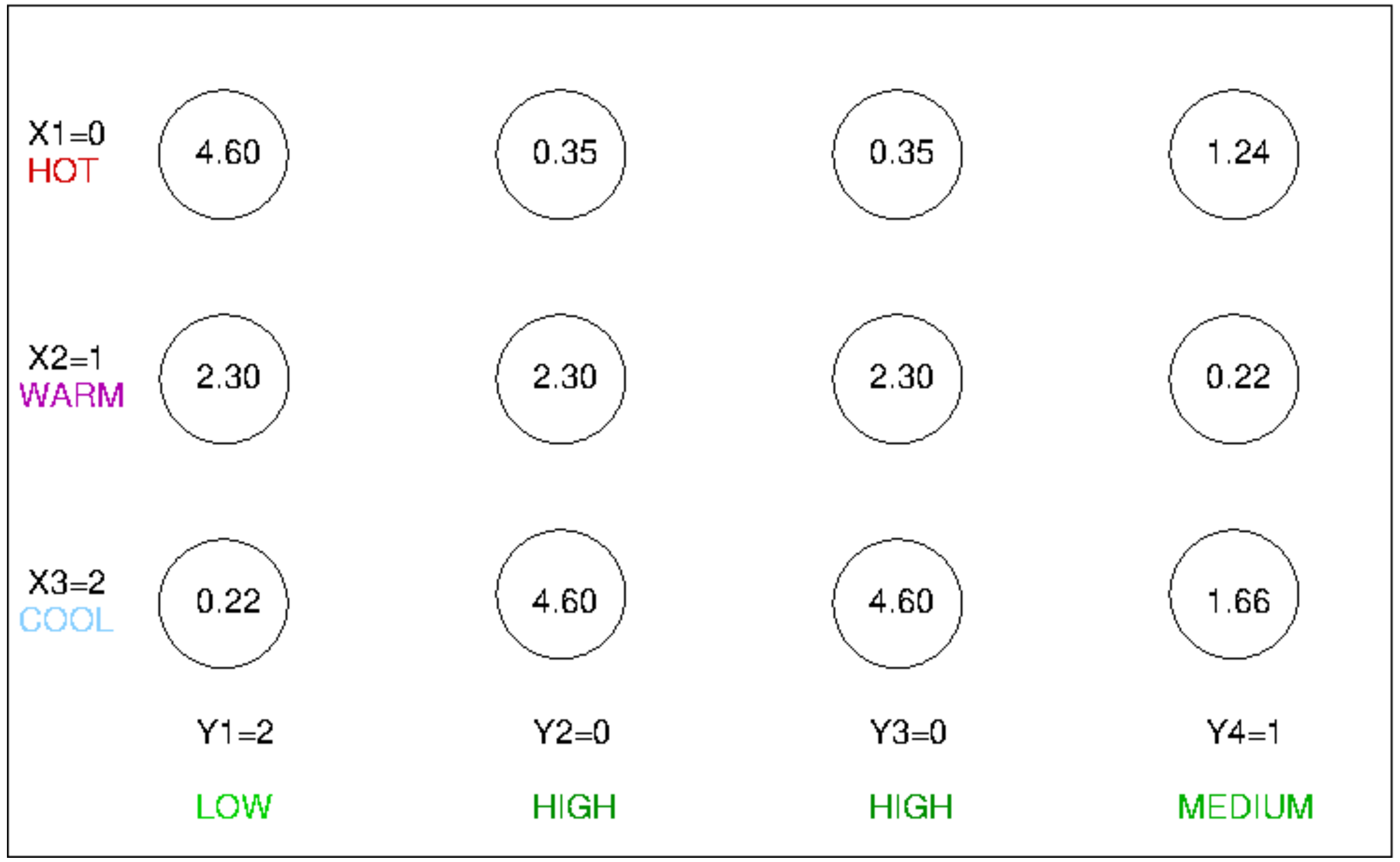
- maximize posterior
  - $P(X_1, X_2, \ldots, X_N | Y_1, Y_2, \ldots, Y_N, P, \mathcal{Q}, \pi)$
- Minimize cost function

$$
\begin{aligned}
-(\quad & [\log P(X_1) + \log P(Y_1 | X_1)] + \\
& [\log P(X_2 | X_1) + \log P(Y_2 | X_2)] + \\
& \ldots + \\
& [\log P(X_N | X_{N-1}) + \log P(Y_N | X_N)] \\
)
\end{aligned}
$$

Trellis

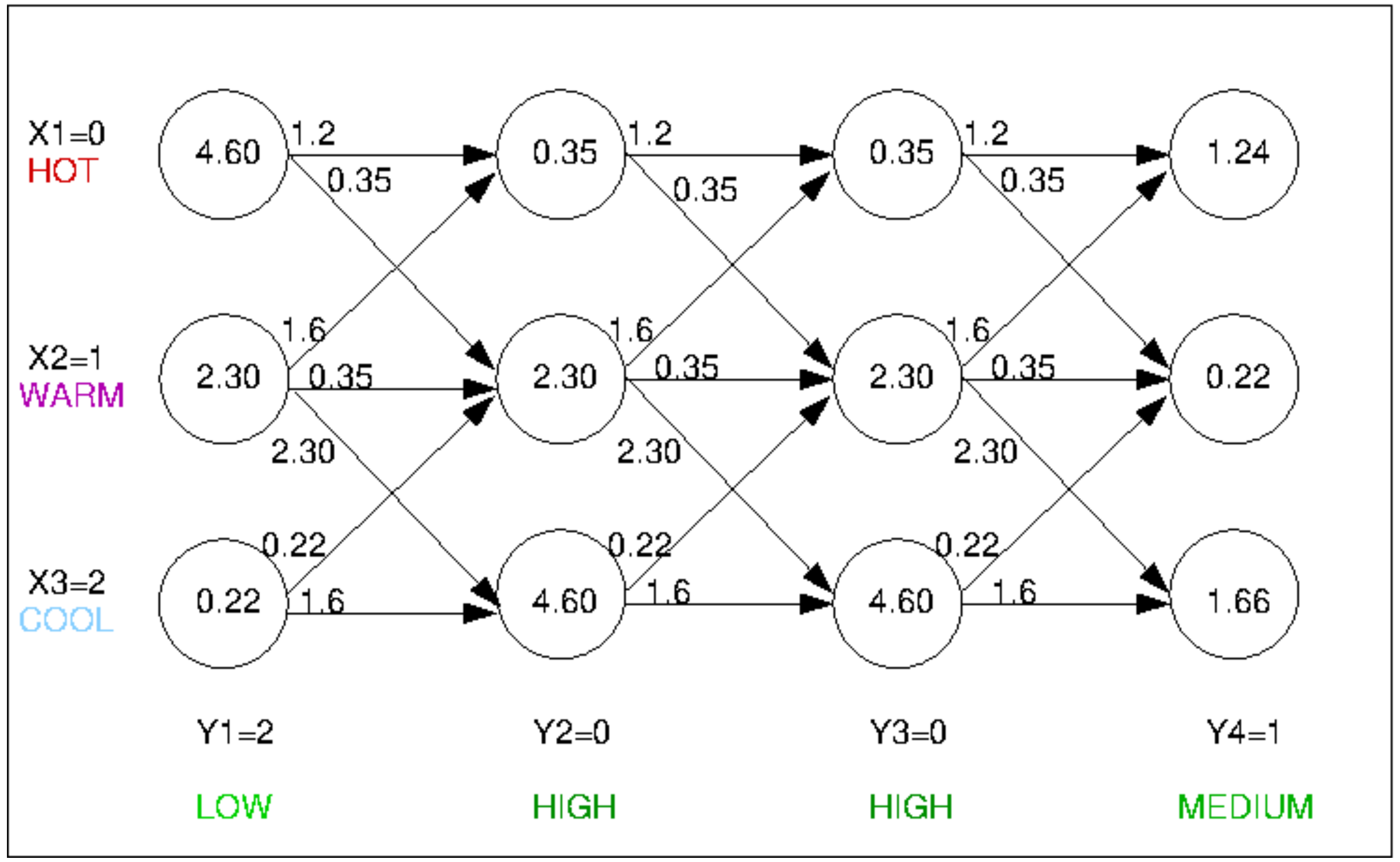


- States X_i: [hot $= 0$, warm $= 1$, cool $= 2$]
- Transition Probability Matrix: $P = \begin{bmatrix} 0.3 & 0.7 & 0 \\ 0.2 & 0.7 & 0.1 \\ 0 & 0.8 & 0.2 \end{bmatrix}$
- Possible outputs $Y_i : O =$ [high $= 0$, medium $= 1$, low $= 2$]
- Emission distribution: $\mathcal{Q} = \begin{bmatrix} 0.7 & 0.29 & 0.01 \\ 0.1 & 0.8 & 0.1 \\ 0.01 & 0.19 & 0.8 \end{bmatrix}$
- Observed data:
  - $Y = [Y_1 =$ low $= 2, Y_2 =$ high $= 0, Y_3 =$ high $= 0, Y_4 =$ medium $= 1]$
- Trellis:
  - Node weights $-\log(P(Y_n | X_n = i)) = -log(q_{i,Y_n})$
  - Edge weights: $-\log(P(X_{n+1} = j | X_n = i)) = -\log(p_{i,j})$



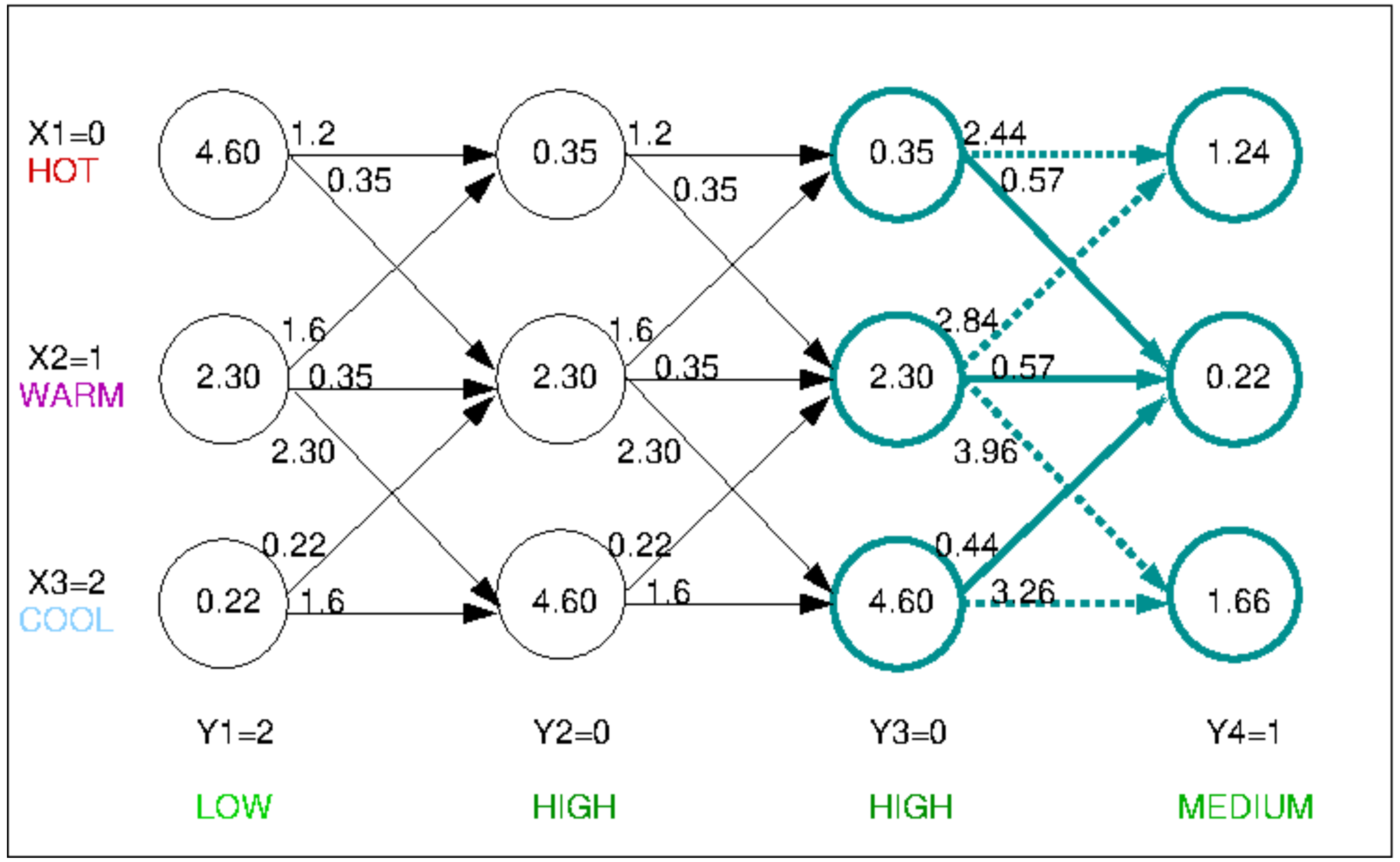
HMM Inference - Viterbi Algorithm

- Viterbi Algorithm
  - Find lowest cost path
  - Dynamic Programming
    - Optimal solution for problem includes optimal solution of subproblems
  - Best path from node in column n is part of best paths between columns $n-1$ and n
  - Iterate from last column backwards



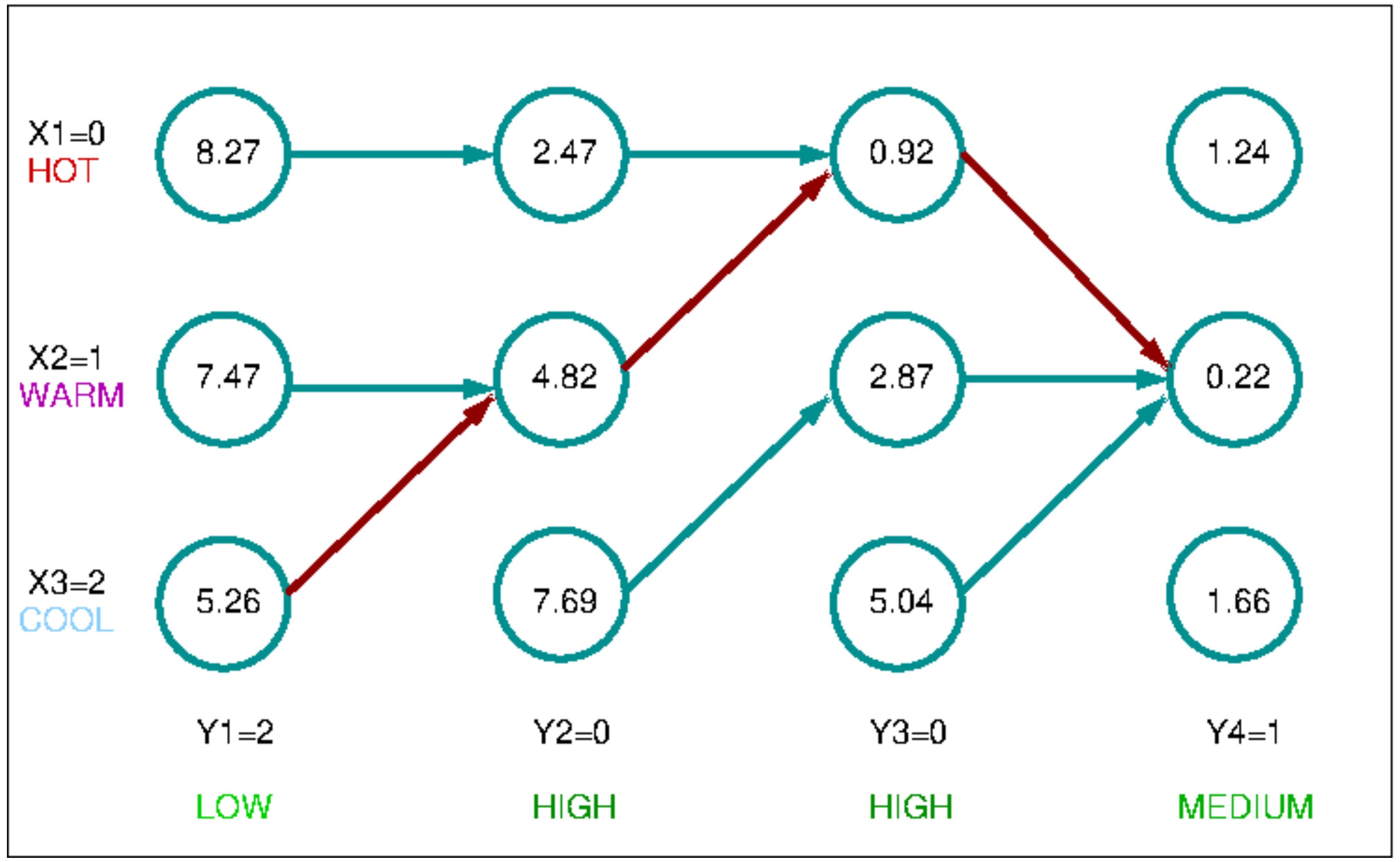
HMM Inference - Viterbi Algorithm

- COST TO GO for node in column n
  - $C_n(i)$: cost of best path segment starting at node i in column n
    - choose cost of minimum edge $(i, B_n(i))$
    - $B_n(i)$: landing node on the best edge
  - Cost of node
- Iterative: for (n = number_of_columns; n > 1; n--)
  - $$\begin{aligned} B_{n-1}(i) &= \text{argmin}_u \left[-\log P(X_n = u \mid X_{n-1} = i) - logP(Y_n \mid X_n = u)\right] \\ C_{n-1}(i) &= \min_u \left[-\log P(X_n = u \mid X_{n-1} = i) - logP(Y_n \mid X_n = u)\right] \end{aligned}$$
- Recursive: for each node i in column $n=1$, compute:
  - $$\begin{aligned} B_n(i) &= \text{argmin}_u \left[-\log P(X_{n+1} = u \mid X_n = i) - logP(Y_{n+1} \mid X_{n+1} = u) - C_{n+1}(u)\right] \\ C_n(i) &= \min_u \left[-\log P(X_{n+1} = u \mid X_n = i) - logP(Y_{n+1} \mid X_{n+1} = u) - C_{n+1}(u)\right] \end{aligned}$$
- Report $B_1((\hat{j}), B_2(B_1(\hat{j})), \ldots)$



HMM Inference - Viterbi Algorithm

- Completion of Text
  - Markov states as n-grams
  - frequencies of n-grams: $P(Y_i \mid X_i)$
  - frequencies of next symbols: $P(X_{n+1} \mid X_n)$
- Other uses:
  - Communication Errors
  - robustness to noise in sequential measurements



Fitting HMMs

- Known: potential observations Y
- Unknown: states X

- Known: states X, potential observations Y
- Unknown: emission distribution $P(Y|X)$

- Expectation Maximization

Hidden Markov Models - Inference

- Inference in HMMs
- Trellis
- Viterbi algorithm for Inference in HMMs

Applied Machine Learning

HMM Inference