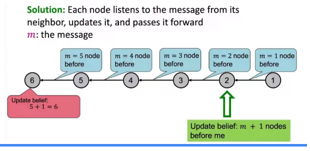
Collective Classification: Belife propagation, Correct & Smooth

Belife propagation

- def
  - dynamic programming
  - answer prob queries in graph
- message passing
  - **Task**: Count the number of nodes in a graph*
  - **Condition:** Each node can only interact (pass message) with its neighbors

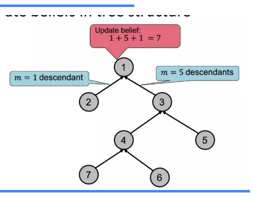
  - algo
    - Define an ordering of nodes (that results in a path)
    - Edge directions are according to order of nodes
      - Edge direction defines the order of message passing
    - For node i from 1 to 6
      - Compute the message from node i to $i+1$ (number of nodes counted so far)
      - Pass the message from node i to $i+1$
    - ordering



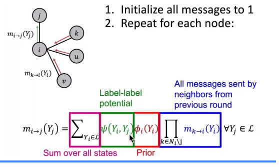
Solution: Each node listens to the message from its neighbor, updates it, and passes it forward

m: the message

- example
  - line graph
  - tree
    - from leave to root



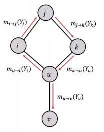
- Loopy BP algo
  - message sent based on what is received i.e. its belif
  - notation
    - ***Label-label potential matrix*** ψ : Dependency between a node and its neighbor. $\psi(Y_i, Y_j)$ is proportional to the probability of a node j being in class Y_j given that it has neighbor i in class Y_i.
    - ***Prior belief*** ϕ : $\phi(Y_i)$ is proportional to the probability of node i being in class Y_i.
    - $m_{i \to j}(Y_j)$ is i's message / estimate of j being in class Y_j.
    - $\mathcal{L}$ is the set of all classes/labels
  - formula



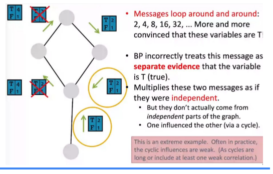
1. Initialize all messages to 1
2. Repeat for each node:

$$m_{i \to j}(Y_j) = \underbrace{\sum_{Y_i \in \mathcal{L}}}_{\text{Sum over all states}} \overbrace{\psi(Y_i, Y_j)}^{\text{Label-label potential}} \underbrace{\phi_i(Y_i)}_{\text{Prior}} \overbrace{\prod_{k \in N_i \setminus j} m_{k \to i}(Y_i)}^{\text{All messages sent by neighbors from previous round}}, \forall Y_j \in \mathcal{L}$$

- graph with cycles
  - no ordering since cycle



  - problem
    - not independent anymore
      - **Beliefs may not converge**
        - Message $m_{u \to i}(Y_i)$ is based on initial belief of i, not a **separate evidence** for i
        - The initial belief of i (which could be incorrect) is reinforced by the cycle $i \to j \to k \to u \to i$



- Messages loop around and around: 2, 4, 8, 16, 32, ... More and more convinced that these variables are T!
- BP incorrectly treats this message as **separate evidence** that the variable is T (true).
- Multiplies these two messages as if they were independent.
  - But they don't actually come from *independent* parts of the graph.
  - One influenced the other (via a cycle).

This is an extreme example. Often in practice, the cyclic influences are weak. (As cycles are long or include at least one weak correlation.)

  - soln
    - follow random order
    - start from arbitary node
  - in practice
    - not a problem since structure more like tree and complex graph
- pros & cons
  - **Advantages**:
    - Easy to program & parallelize
    - General: can apply to any graph model with any form of potentials
      - Potential can be higher order: e.g. $\psi(Y_i, Y_j, Y_k, Y_v \ldots)$
  - **Challenges**:
    - Convergence is not guaranteed (when to stop), especially if many closed loops
  - **Potential functions** (parameters)
    - Require training to estimate