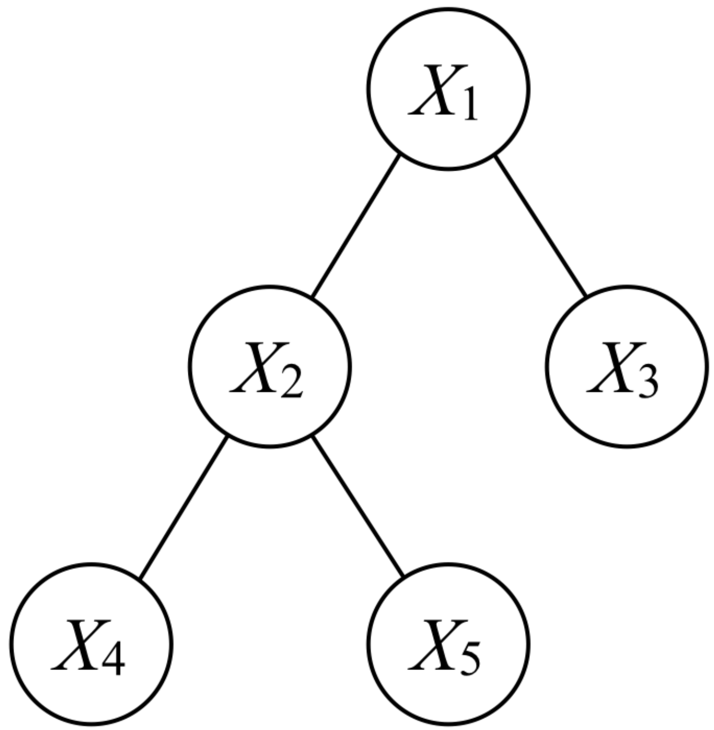
Exercise: The Sum-Product Algorithm - Computational Complexity

Bookmark this page

Exercise: The Sum-Product Algorithm - Computational Complexity

7/7 points (graded)

Let's take another look at our running five node example:



When we computed the marginal distribution p_{X_1} using the sum-product algorithm, note that the ordering in which we pushed summations around matters! We first summed out x_5. This was a good choice because it only depended on two factors. Consider if instead we tried to sum out x_2 first, which depends on factors ϕ_2, ψ_{12}, ψ_{24}, and ψ_{25}:

$$\begin{aligned} p_{X_1}(x_1) &\propto \sum_{x_2}\sum_{x_3}\sum_{x_4}\sum_{x_5}\Big\{\phi_1(x_1)\phi_2(x_2)\phi_3(x_3)\phi_4(x_4)\phi_5(x_5) \\ &\qquad\qquad \cdot \psi_{12}(x_1,x_2)\psi_{13}(x_1,x_3)\psi_{24}(x_2,x_4)\psi_{25}(x_2,x_5)\Big\} \\ &= \sum_{x_3}\sum_{x_4}\sum_{x_5}\Big\{\phi_1(x_1)\phi_3(x_3)\phi_4(x_4)\phi_5(x_5) \\ &\qquad\qquad \cdot \psi_{13}(x_1,x_3)\underbrace{\sum_{x_2}\phi_2(x_2)\psi_{12}(x_1,x_2)\psi_{24}(x_2,x_4)\psi_{25}(x_2,x_5)}_{\text{depends on } x_1,x_4,x_5\text{; let's call this } m_{2\to\{1,4,5\}}(x_1,x_4,x_5)}\Big\}. \end{aligned}$$

- Suppose that every random variable X_i takes on exactly k different values. How many operations (additions, multiplications, table lookups) does it take to compute the table $m_{2\to\{1,4,5\}}$? (Note that this table has one entry for every possible value of x_1, x_4, and x_5.)

  Choose the answer with **smallest** big O bound.

  - $\mathcal{O}(k)$
  - $\mathcal{O}(k^2)$
  - $\mathcal{O}(k^3)$
  - $\mathcal{O}(k^4)$ ✔
  - $\mathcal{O}(k^5)$

- ( ) $\mathcal{O}(2^k)$

- What other summation orders would have been as efficient for computation as the one we chose originally (in computing the marginal for X_1)?

  Select all answers that are as efficient as the one we had chosen in the video/course notes.

  (For each of these answers, you can read it as summing out the variable mentioned left-most first and then working rightward.)

  - [ ] X_5, X_2, X_3, X_4
  - [ ] X_5, X_3, X_2, X_4
  - [x] X_5, X_3, X_4, X_2
  - [ ] X_4, X_2, X_5, X_3
  - [ ] X_4, X_3, X_2, X_5
  - [x] X_4, X_3, X_5, X_2
  - [x] X_4, X_5, X_2, X_3

  ✔

Now let's look at the computational complexity (i.e., how many operations it takes) to run the sum-product algorithm. Let n be the number of nodes in the graph. Assume that every random variable X_i takes on exactly k different values.

As a reminder, the equation for computing a message $m_{i\to j}$ is given by

$$m_{i\to j}(x_j) = \sum_{x_i}\left[\phi_i(x_i)\psi_{i,j}(x_i,x_j)\prod_{k\in\mathcal{N}(i)\text{ such that }k\neq j} m_{k\to i}(x_i)\right]$$

and the equation for computing each node marginal after all messages have been computed is

$$p_{X_i}(x_i) = \frac{1}{Z}\underbrace{\phi_i(x_i)\prod_{j\in\mathcal{N}(i)} m_{j\to i}(x_i)}_{\tilde{p}_{X_i}(x_i)}.$$

In this part of the exercise, we come up with a worst-case running time of the sum-product algorithm as stated so far.

- When running the sum-product algorithm on a tree, how many operations does it take to compute a message table $m_{i\to j}$ for a specific i and j (of course i and j are neighbors, and remember that because of how sum-product computes messages, the messages that $m_{i\to j}$ depends on have already been computed)?

  Choose the answer with **smallest** big O bound in terms of k and n (unless one of these doesn't matter).

Hint: In the worst-case, how many neighbors could node i have?

- $\mathcal{O}(k)$
- $\mathcal{O}(k^2)$
- $\mathcal{O}(k^3)$
- $\mathcal{O}(k^4)$
- $\mathcal{O}(nk)$
- $\mathcal{O}(nk^2)$ ✔
- $\mathcal{O}(nk^3)$
- $\mathcal{O}(nk^4)$
- $\mathcal{O}(n^2k)$
- $\mathcal{O}(n^2k^2)$
- $\mathcal{O}(n^2k^3)$

- $\mathcal{O}(n^2k^4)$

- Exactly how many messages do we compute? Please provide your answer in terms of n, the number of nodes in the tree.

  In this part, please provide your answer as a mathematical formula (and not as Python code). Use ^ for exponentiation, e.g., x^2 denotes x^2. Explicitly include multiplication using *, e.g. x*y is xy.

  2*(n-1)

  $2 \cdot (n-1)$

- Putting together your answers to the previous two parts, how many operations does computing all the messages take?

  Choose the answer with **smallest** big O bound in terms of k and n (unless one of these doesn't matter).

  - $\mathcal{O}(k)$
  - $\mathcal{O}(k^2)$
  - $\mathcal{O}(k^3)$
  - $\mathcal{O}(k^4)$

- $\mathcal{O}(nk)$
- $\mathcal{O}(nk^2)$
- $\mathcal{O}(nk^3)$
- $\mathcal{O}(nk^4)$
- $\mathcal{O}(n^2k)$
- $\mathcal{O}(n^2k^2)$ ✔
- $\mathcal{O}(n^2k^3)$
- $\mathcal{O}(n^2k^4)$
- $\mathcal{O}(n^3k)$
- $\mathcal{O}(n^3k^2)$
- $\mathcal{O}(n^3k^3)$
- $\mathcal{O}(n^3k^4)$

- After computing all the messages, how many operations does it take to compute the marginal distribution p_{X_i} for a specific i?

  Choose the answer with **smallest** big O bound in terms of k and n (unless one of these doesn't matter).

  - ○ $\mathcal{O}(k)$
  - ○ $\mathcal{O}(k^2)$
  - ○ $\mathcal{O}(k^3)$
  - ○ $\mathcal{O}(k^4)$
  - ◉ $\mathcal{O}(nk)$ ✔
  - ○ $\mathcal{O}(nk^2)$
  - ○ $\mathcal{O}(nk^3)$
  - ○ $\mathcal{O}(nk^4)$
  - ○ $\mathcal{O}(n^2k)$

- $\mathcal{O}(n^2 k^2)$
- $\mathcal{O}(n^2 k^3)$
- $\mathcal{O}(n^2 k^4)$

- How many operations does running the sum-product algorithm take?

  Choose the answer with **smallest** big O bound in terms of k and n (unless one of these doesn't matter).

  - $\mathcal{O}(k)$
  - $\mathcal{O}(k^2)$
  - $\mathcal{O}(k^3)$
  - $\mathcal{O}(k^4)$
  - $\mathcal{O}(nk)$
  - $\mathcal{O}(nk^2)$

- $\mathcal{O}(nk^3)$
- $\mathcal{O}(nk^4)$
- $\mathcal{O}(n^2k)$
- $\mathcal{O}(n^2k^2)$ ✔
- $\mathcal{O}(n^2k^3)$
- $\mathcal{O}(n^2k^4)$
- $\mathcal{O}(n^3k)$
- $\mathcal{O}(n^3k^2)$
- $\mathcal{O}(n^3k^3)$
- $\mathcal{O}(n^3k^4)$

It turns out that we can improve the running time of the algorithm by some careful bookkeeping!

Submit You have used 3 of 5 attempts

✔ Correct (7/7 points)

Discussion

Topic: Inference in Graphical Models - Marginalization / Exercise: The Sum-Product Algorithm - Computational Complexity

Show Discussion