The Max-Product Algorithm

Bookmark this page

The Max-Product Algorithm

Start of transcript. Skip to the end.

6.008.1x - The Max-Product Algorithm

0:00 / 10:14 1.0x

In this video, I'm going to talk about how

to compute the most probable configuration

in a tree structured, undirected graphical model.

The resulting algorithm is called

max-product, which closely resembles

the sum-product algorithm.

Video

Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file

These notes cover roughly the same content as the video:

THE MAX-PRODUCT ALGORITHM (COURSE NOTES)

The Key Idea Behind Sum-Product and Max-Product

In deriving sum-product, the key idea was to push in summations. As a simple example, this amounts to the following:

$$\sum_{x_2} f(x_1)g(x_1,x_2) = f(x_1)\sum_{x_2} g(x_1,x_2),$$

for arbitrary functions f and g.

A similar idea applies when we want the maximum:

$$\max_{x_1,x_2}\{f(x_1)g(x_1,x_2)\} = \max_{x_1}\left[\max_{x_2} f(x_1)g(x_1,x_2)\right] = \max_{x_1}\left[f(x_1)\max_{x_2} g(x_1,x_2)\right],$$

where the second equality holds provided that $f(x_1)$ is nonnegative for all x_1. (Why would a negative value kill the equality?)

For the equation above, think of the right-hand side function $f(x_1)\max_{x_2} g(x_1, x_2)$ as some 1-dimensional table with a number associated with each value of x_1. (In particular, suppose we've precomputed what $\max_{x_2} g(x_1, x_2)$ is for every value of x_1.) Then the argument $\hat{x}_1$ achieving the maximum is whichever x_1 has the highest value in the table. In other words, the highest value is

$$f(\hat{x}_1)\max_{x_2} g(\hat{x}_1, x_2),$$

from which we see that to get the optimal value of x_2, we just need to figure out which x_2 maximizes the function $g(\hat{x}_1, x_2)$.

So we first found the optimal value $\hat{x}_1$ for x_1. Then we traced backward and found the best value for $\hat{x}_2$ given $\hat{x}_1$, where we could readily answer this query if when we first compute $\max_{x_2} g(x_1, x_2)$, we also store what value of x_2 achieves the maximum for each value of x_1.

The intuition for this two-variable example carries over to the case when we have many variables represented by a tree-structured undirected graphical model and we want the most probable configuration. The resulting algorithm is the max-product algorithm.

Max-Product

As a reminder of notation, for random variables $X_1, \ldots, X_n$ corresponding to a tree $G = (V, E)$,

$$p_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = \frac{1}{Z}\left(\prod_{i \in V} \phi_i(x_i)\right)\left(\prod_{(i,j) \in E} \psi_{i,j}(x_i, x_j)\right).$$

The max-product algorithm is as follows. First, we pick a root node r and compute messages going from the leaves to the root:

$$m_{i \to j}(x_j) = \max_{x_i} \left[\phi_i(x_i) \psi_{i,j}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \setminus \{j\}} m_{k \to i}(x_i) \right].$$

Note that this is formula is the same as that of sum-product except that the sum has been replaced by a max.

As with our simple two-variable example, we want to keep track of which argument achieves the maximum. Otherwise, all we'd be computing is a *scaled* version of the probability of the most probable configuration. (Remember that in general we don't know the normalization constant Z, in which case we won't actually find out the probability of the most probable configuration! Luckily, the argument achieving the maximum doesn't care what the value of Z is.)

To keep track of which argument achieves the maximum, we compute *traceback messages*:

$$t_{j \to i}(x_j) = \arg\max_{x_i} \left[\phi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \setminus \{j\}} m_{k \to i}(x_i) \right].$$

After computing the messages and traceback messages going from the leaves to the root, we're ready to find the most probable configuration, starting from the root and going to the leaves. This is like how we figured out the optimal value for x_1 in our two-variable example first (i.e., node 1 was the root).

For root node r, we compute the optimal value:

$$\hat{x}_r = \arg\max_{x_r} \left[\phi_r(x_r) \prod_{k \in \mathcal{N}(r)} m_{k \to r}(x_r) \right].$$

We then work backwards to the leaves, using the traceback messages to assign values:

$$\hat{x}_i = t_{j \to i}(\hat{x}_j).$$

Discussion

Topic: Inference with Graphical Models - Most Probable Configuration / The Max-Product Algorithm

Show Discussion