Assignment 4 (NLProc-PGM4NLP-M)

by Yannick Lang

Part 1: order-k linear Markov field MAP inference

We apply the Viterbi algorithm, but instead of one node for each single-node assignment, we introduce one node for each k-gram-node assignment. Therefore, the number of labels grows exponentially. Since normal Viterbi algorithm has time complexity `O(n + r^2)`, this algorithm will have time complexity `O(n + (r^k)^2)`

Part 2: ring-graph MAP inference

We first need to transform the ring-structure into a linear graph, such that we have a place to start with the algorithm. We therefore remove the connection between node `0` and node `n`. To keep all information, we duplicate node `0` and append the new node `0'` to node `n` (via an edge with the same factor as the previously removed edge).

For each possible assignment to node `0`, we now perform the viterbi algorithm. We modify the factor between nodes `n` and `0'` in a way that ensures that node `n` will be assigned the value currently under consideration.

We then have `r` possible paths, from which we can pick the best one.

Part 3: top-k MAP inference

We use a modified Viterbi algorithm, instead of saving only the best (lowest distance or highest factor) path (as back-pointer + value) for each node, we save the top k paths. Each root node starts with one empty path with cost 0 and k-1 empty paths with infinite cost. After going traversing the graph with the Viterbi algorithm, the final node has the top-k paths, which can be reconstructed from the back-pointers.

Because each starting node has only one non-infinite-cost path, these paths are guaranteed to be unique and the Viterbi algorithm ensures that they are optimal paths.

Part 4: sampling from a linear chain Markov field

Apply the forward algorithm to sample the last node. This node has now been sampled an can be considered a constraint on the second-to last node. Therefore, we can remove the last node and incorporate the factor between the last node and the second-to-last node into the factor between the new last and second-to-last nodes.

Repeat until every node has been sampled.

Part 5: marginal probabilities

Marginal Probabilities in a linear Markov field can be calculated via the forward-backward algorithm.

In the forward pass, we start with a vector of initial values and iteratively apply all transition matrices. Those give the probability of ending up in any given state, given all previous states. This represents the information we gain from nodes on the left side.

In the backward pass, the probability of observing the remaining observations from a given starting point are calculated. This represents information we gain from nodes on the right side.

Probabilities from the forward and backward passes can then be multiplied to obtain the marginal probabilities for any given node.