Min-max Approach in Decoding

March 17, 2014

Introduction

Most of the problems in NLP try to solve following problem,

$$y^* = \arg\max_{y \in \mathcal{Y}} F(y)$$

where $\mathcal{Y}$ is the set of all possible structures, and $F(y)$ is the score function. Note that if score function can be written as sum of functions over subsets of y, then problem becomes

$$y^* = \arg\max_{y \in \mathcal{Y}} \sum_i f_i(y_i)$$

When problem have that structure, we will discuss how to solve a different problam which is,

$$y^* = \arg\max_{y \in \mathcal{Y}} \min_i f_i(y_i)$$

POS and Viterbi

Procedure 1 Viterbi for max-min problem

Input: A sentence $x_1, \ldots, x_n$ parameters $q(s|u,v)$ and $e(x|s)$
Definitions: Define $\mathcal{K}$ to be the set of possible tags. Define $\mathcal{K}_{-1} = \mathcal{K}_0 = \{*\}$, and $\mathcal{K}_k = \mathcal{K}$ for $k = 1 \ldots n$.
for $k \leftarrow 1$ to n **do**
  for $u \in \mathcal{K}_{k-1}$ and $v \in \mathcal{K}_k$ **do**

$$\pi(k,u,v) = \max_{w \in \mathcal{K}_{k-2}} \min\{\pi(k-1,w,u), q(v|w,u) + e(x_k|v)\}$$

  end for
end for
return $\max_{u \in \mathcal{K}_{k-1}, v \in \mathcal{K}_k} \min\{\pi(n,u,v), q(\text{STOP}|u,v)\}$

Even though changes to original Viterbi algorithm are minimal, the above algorithm brings out an important uncertanity based on the fact that our problem does not have a unique solution. There can be many different solutions to max-min objective and the above algorithm would pick one randomly. On the other hand if we use the hypergraph representation of the dynamic program, we can prune that graph so that resulting graph would contain only the edges that are in an optimal solution. After constructing the hyper-graph, achieving that goal is actually quiet simple.

- Find
$$w^* = \max_{y \in \mathcal{Y}} \min_i f_i(y_i)$$
- Delete all edges e s.t. $weight(e) < w^*$

Any path from root to end in the remaining graph will be a solution to $y^* = \arg\max_{y \in \mathcal{Y}} \min_i f_i(y_i)$. This is actually quiet easy to observe. Assume there is a path that is not solution to our objective, that would mean it contains an edge e s.t. $weight(e) < w^*$, however, we already prunned those edges, so there cannot be such a path. Also note that there exists at least one path from root to end in the remaining graph, since we found w^* using our modified viterbi, and we did not prune any of the edges cotained in that path.

After that point we can try different ideas to find a solution. A simple idea would be to run generic viterbi algorithm (or shortest path) on remaining hypergraph and get a solution. Another idea would be setting weight of all the edges $weight(e) = w^*$ to infinity, and running our algorithm would give the solutions that maximize lowest, and second lowest weighted edges in the path. We can even keep doing that until we get a single solution.