High Level Overview of Corels

Matthew Scharf

June 2018

1 Introduction

This algorithm uses a branch and bound procedure to prune a trie of rules in order to find an optimal rule list to describe categorical data. The algorithm keeps track of the current optimal rule list and uses a succession of bounds (each of which more computationally difficult than the last) to prune away most of the search space. Once the remaining viable search space is empty, the current optimal rule list becomes certifiably optimal in the search space.

2 Cost function

Where r is a rule list, and x_n is the labeled data,the cost function is

$$C(r, x_n) = b_0(r, x_n) + b_1(r, x_n) + \lambda l(r)$$

where $b_0(r)$ is the mis-classification error of the rules before the final (aka default) rule, $b_1(r)$ is the mis-classification of the default rule, λ is some constant, and $l(r)$ is the length of the rule list (does not include the default rule).

3 Algorithm

The main loop in this algorithm iterates through each new rule list by evaluating (using the aforementioned succession of bounds) how the last rule impacts the rule list to which it is added (which has already been evaluated).

3.1 Classification Bound

The first bound makes sure that the last rule classifies more data than λ-the cost to add one more rule. If it does not, then this last rule must be harmful and so the rule list cannot be optimal.

3.2 Accurate Classification Bound

The algorithm assesses which rule output (the q in $p \rightarrow q$) would classify more data accurately and assigns that output to the rule. Then, the next bound makes sure that the last rule accurately classifies more data than λ. If it does not, then this last rule must be harmful and so the rule list cannot be optimal.

3.3 Objective Lower Bound

The next bound calculates b_0 of the new rule list and makes sure that it is less than $C*$ (the current optimal rule list cost). If it is not, then the rule list cannot be optimal. Through this process,

3.4 Final Evaluation

The algorithm then assesses which default rule output would classify more data accurately and assigns that output to the default rule. Then, the algorithm calculates C of the new rule list and compares it to $C*$. If $C < C*$, then the new rule list replaces the optimal rule list and prunes the search space of rule lists with higher costs.

3.5 Note

Note: this is a BROAD overview of the main components of the algorithm. A combination of data structures and other bounds are used to optimize the algorithm.

4 Discussion about clause-length of individual rules

The algorithm uses individual rules which potentially depend on the intersections of multiple clauses. However, it does not incorporate the clause-length of rules into the optimization algorithm. So, the dimension of the feature space evaluated by this algorithm increases very quickly as the maximum clause-length of rules increases (the highest maximum clause length demonstrated in this paper is 2).