Random Intersection Trees for finding interactions in large datasets

Rajen Shah, University of Cambridge
joint work with Nicolai Meinshausen, ETH Zurich

October 10, 2015

Finding interactions between variables in large and high-dimensional datasets is often a serious computational challenge. Because of the huge number of possible interactions, most approaches build up interaction sets incrementally, adding variables in a greedy fashion. In order for this to work, higher order interactions must contain informative lower order interactions. Important examples of the greedy approach include decision trees, and related methods. One drawback of such approaches is that potentially informative deep tree splits can be overlooked due to the greedy nature of the search.

Here, we look at an alternative approach for finding potentially high-order interactions in classification problems with binary predictor variables. Our proposal, which we call *Random Intersection Trees*, differs from most existing approaches in two important respects. Firstly, instead of directly searching over variables for the best split-point, we effectively search over observations to find those that are most informative about important interactions. We exploit sparsity in the data by arranging the search in a tree structure, thereby improving computational efficiency. Secondly, rather than building up interactions greedily starting from the empty set, we start with a maximal interaction that includes all variables. From this maximal set, variables are gradually removed if they fail to appear in randomly chosen observations of a class of interest. Repeating over many such random draws, we can show that very informative interactions are retained with high probability, even if their lower order interactions are not informative.

Thus *Random Intersection Trees* functions as a sort of efficient importance sampling from the set of all possible interactions. We show that the computational complexity of our procedure is of order p^κ for a value of κ that can reach values as low as 1 for very sparse data; in many more general settings, it will still beat the exponent s in brute force search if looking for interactions of order s.

In addition, by using some new ideas based on min-wise hashing schemes, we are able to further reduce the computational cost. Interactions found by our algorithm can be used for predictive modelling in various forms, but they are also often of interest in their own right as useful characterisations of what makes a certain class different from others.

An implementation of the methodology is available in the R package *FSInteract*.