Monte Carlo Decision Tree Search for clinical recommendation

C. Nunes[1], O. Camara[1], B. Bijnens[1], M. De Craene[2], A. Jonsson[1]

[1] Universitat Pompeu Fabra Tànger, Spain, cecilia.nunes@upf.edu

[2] Philips Research Medisys, Paris, France

1. Introduction

Decision trees (DTs) are an interpretable knowledge-representation, where each node contains a test that divides the population between the branches, and each leaf provides an estimate of the target variable. They have potential for clinical integration, as clinical guidelines are often presented as DTs [4]. Although recent learning algorithms offer higher predictive accuracy, their output is often a black box. A study proposing a deep learning approach to predict cardiovascular events highlights the challenge in understanding the predictions, and stresses the importance of supporting visualizations [8]. European regulation has been enacted to secure the right to an explanation of algorithmic decisions [2].

In the clinic, we wish to provide the best prediction with the smallest possible number of measurements. Learning the best DT for a decision problem is NP-complete [3]. Locally-optimal approaches offer a good trade-off between computational complexity and prediction accuracy, they do not output a DT which minimizes the cost of the decision. A DT is a static model targeted at a set of measurements, although clinical reality is highly dynamic. An ideal decision support tool is able to adapt to context changes, such as unavailable measurements or tight budgets. We propose an algorithm for non-greedy DT learning, which outputs a space of DTs. The clinician can interact with this output, by selecting the best model for the context.

2. Learning decision trees by greedy search

Greedy methods such as Iterative Dichotomiser 3 (ID3) [5] and the C4.5 [6] remain the foundation for recent approaches [1, 5]. Consider the input variable X and the categorical output variable Y . Each DT node is a partition of the input space, and is assigned a test function $t(x)$. We want to choose $t(x)$ to split the training data with maximum stratification of Y. Given a node with training data, a locally-optimal $t(x)$ is sought. The data is divided between the child nodes, where the search proceeds recursively [7].

3. Monte Carlo Tree Search

MCTS is a combinatorial optimization method that learns optimal actions by taking guided samples of the decision space. It is successful in applications with large search spaces. MCTS builds a search tree of states and actions of a Markov decision process (MPD), with the aim of selecting the sequence of actions that maximizes the expected reward. It does so by alternating a tree policy for selecting the next states to explore, with the default policy that simulates experience and estimates the reward. The tree policy models the choice of the next node to explore as a multi-armed bandit problem, and using the Upper Confidence Bound for Trees (UCT) algorithm.

4. Proposed approach

We propose a MCTS algorithm to learn DTs non-greedily, where each sample is a patient, and each variable is a assessment made by the physician. Furthermore, we modify the MCTS output strategy to allow adaptation of the resulting search tree to the specific clinical context. To use MCTS to DT learning, we specify the MPD as:

- The state s is a set of univariate functions of X that partition the data forming a directed rooted tree, aiming at stratification with respect to Y . In other words, the state s defines a DT.5
- The action a = l, t(x) is a univariate test at DT leaf l, that splits its population. When action a is added to the DT, the leaf becomes an internal node.
- The reward r(s, a) is the expected accuracy, or other relevant metric, of a prediction made by the DT for an unseen patient.

We propose a heuristic method to generate actions for expanding nodes during the tree policy phase, and use C4.5 as reward simulator during the default policy phase. Preliminary results are presented, as well as an application to interact with the output search tree.

References

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Wadsworth International Group, Belmont, CA, 1984.

[2] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016, and repealing Directive 95/46/EC, 2016 O.J. L 119, 4.5.:1–88.

[3] L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is NP-complete. Information Processing Letters, 5(1):15–17, 1976.

[4] J. H. Keffer. Guidelines and algorithms: Perceptions of why and when they are successful and how to improve them. Clinical Chemistry, 47(8):1563–1572, 2001.

[5] J. R. Quinlan. Induction of decision trees. Machine Learning, 1(1):81–106, 1986.

[6] R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[7] L. Rokach and O. Maimon. Top-down induction of decision trees classifiers - A survey. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 35(4):476–487, 2005.

[8] S. F. Weng, J. Reps, J. Kai, J. M. Garibaldi, and N. Qureshi. Can machine-learning improve cardiovascular risk prediction using routine clinical data? PLOS ONE, 12(4), 2017.