EE TITLE

Bryan Deng

Contents

1 Background Information

1.1 Machine Learning and its Applications

Machine learning (ML) is a branch of artificial intelligence that uses large datasets and algorithms to mimic the way humans learn and improve accuracy over time [3]. Since its debut in 1952, it has been steadily gaining popularity for its abilities in recognizing patterns and continuous learning. Machine learning powers many of the applications we use on a daily basis, including chatbots, language translation tools, and social media feeds [2].

Where machine learning shines is in its ability to solve problems that would typically be either impossible for impractical for traditional algorithms. Furthermore, machine learning models are able to generalize these solutions and apply them to additional problems it has never encountered before.

In short, machine learning is a combination of computer science, statistics, and optimization. It uses knowledge from different fields to "teach" computers to complete tasks. As the model looks at more and more data, it starts to recognize patterns among it and optimizes itself.

1.2 Decision Trees

When most think about machine learning, the first thing that comes to mind are neural networks. Neural networks, which are abstractly complex yet powerful algorithms, only make up one subfield of machine learning itself, called *deep learning*. However, there exist several other branches of machine learning, such as *supervised learning*, the main focus of this paper. The most well known model within supervised learning has to be the decision tree. They are binary trees which employ a straightforward *if-else* flow to classify data.

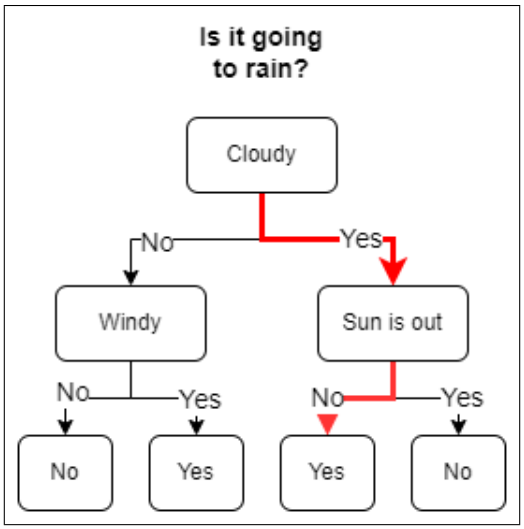

Figure 1: Decision Tree Diagram

Figure 1 shows a simple decision tree to predict whether it will rain based on other weather conditions. Given a cloudy day with no sun, the model will predict that it will rain, as outlined by the red lines on the diagram. The anatomy of a decision tree consists of several parts [6]:

- **Node**: cells that contain data. A tree is made of several nodes connected by edges.
  - **Root node**: the single node at the very top of the tree.
  - **Split node**: a split node splits into two other child nodes based on a feature and split value.
  - **Leaf node**: a node at the end of a tree; it does not split into further nodes.
  - **Child node**: the nodes that follow split nodes.
- **Feature**: an individual characteristic of a dataset [1], the ith feature is represented as x_i.
- **Split Value**: a value that classifies any data that the node encounters (e.g. $x_i \leq 2.7$)

The main computational problem with decision trees is how first construct, then optimize them. If we were to generate a random decision tree structure, it could potentially become

very large and unnecessarily complex, taking up extra resources in computation. And if we were to simply assign each node a random feature and split value, it is very unlikely the model will perform well in predictions.

The approach used in vanilla decision trees is a greedy top-down algorithm that builds nodes, assigning it features and split values as it moves down the tree [5]. Other well-established algorithms for constructing decision trees include random forests and gradient boosting. This paper aims to investigate a new decision tree construction and optimization algorithm by employing the genetic algorithm.

1.3 Genetic Algorithm

Taking a page straight out of Darwin's theory of evolution, genetic algorithms employ the principle of *survival of the fittest*. It generates populations of algorithms or models that evolve and reproduce over time based on a set of criteria, improving on performance. Each *individual* of the population is represented by a form of data structure. Each piece of data can be paralleled to a gene, which when all combined describes the behavior of the individual [4]. The steps of the genetic algorithm are as follows:

- **Initialization**: an initial population is generated with random genes using a set of preset hyperparameters.
- **Fitness evaluation**: a score given to each individual in the population based on how well it performs for the problem.
- **Selection**: a process to select the individuals that will carry forward or reproduce for the next generation, usually based on fitness.
- **Crossover**: a process to either sexually or asexually reproduce individuals for the next generation based on the previous generation.
- **Mutation**: randomly altering genes of an individual to maintain diversity and encourage further exploration of the solution space.

- **Termination**: a set of criterion to determine when reproduction for new generations should stop.

The species in real life which benefit the most from evolution are those that are able to maintain diversity. With diversity, they can overcome external problems like diseases or natural disasters. Similarly, the genetic algorithm puts a strong emphasis on diversity, so it can explore a large solution space and escape falling into a local minimum. Diversity is achieved with several stages of randomness introduced in each generation, in selection, crossover and mutation.

1.4 Evolutionary Decision Tree

An evolutionary decision tree (EDT) uses a genetic algorithm to optimize a decision tree.

2 Experimental Methodology

3 Data Analysis

4 Error Analysis

5 Conclusion

6 Appendix

References

[1] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[2] Sara Brown. Machine learning, explained, Apr 2021.

[3] IBM Cloud Education. What is machine learning?, Jul 2020.

[4] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1998.

[5] J.R. Quinlan. Induction of decision trees. *Machine Learning 1*, 1985.

[6] scikit-learn developers. 1.10. decision trees, Dec 2021.