technische universität dortmund

computer science 12

Bachelor/Master Thesis

Meta Machine Learning Optimization: Cache-aware Organization of Decision Trees

Christian Hakert
Dr. Ing. Kuan-Hsun Chen
Otto-Hahn Str. 16
Technische Universität Dortmund
Email: christian.hakert@tu-dortmund.de
06.01.2020

Decision trees are a fairly simple machine learning algorithm, which operate on labeled input data. The input data is interpreted as multi dimensional data object. At each node of the decision tree, one dimension of the input is compared to a fixed threshold value. Depending on the outcome of the comparison, either the left or the right sub-tree is traversed, which again contains tests on other dimensions. The leafs of the tree contain outputs of the machine learning model, which can be either classification or regression outputs.

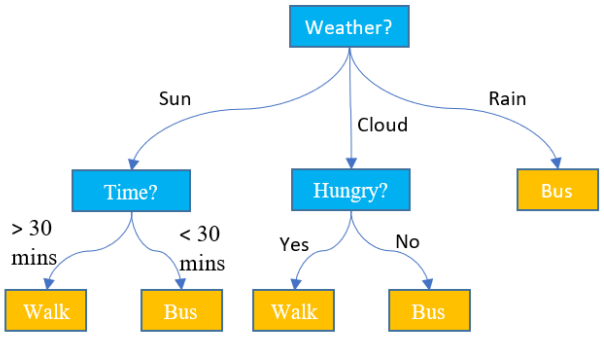

Figure 1: A simple decision tree

However, in the age of big data and massive use of machine learning algorithms decision trees are used to analyze massive high dimensional data, which leads to megabyte or even gigabyte sized decision trees. These trees are usually stored in the main memory and processed directly out of the memory. For modern computers, the speed of such a memory intensive application mainly is determined by the use of the various caches of the CPU. Hence, the memory layout of decision trees was leveraged to keep the most often accesses paths of the tree in a cache-aware manner.

As an alternative to an analytical analysis of the access probabilities of the tree, a simple machine learning approach could be used to improve the cache usage of the decision tree execution. For this purpose, the cache behavior of a specific memory layout of a decision tree has to be analyzed. This can be done by a simple cache simulator, implementing the cache size and the preemption strategy. The output of this simulator is used as a fitness function for a genetic algorithm, which maintains and optimizes different memory layouts as the population.

In this thesis, students first should get familiar with the implementation of decision trees and understand how the memory layout can be changed, without changing the execution behavior. The aforementioned simulation of the cache behavior should be implemented and validated on execution time measurements on different machines. Afterwards, the genetic algorithm approach should be implemented. Modifications of the memory layout can be achived by swapping the memory location of two nodes randomly in a simple version. For an advanced modification step, the tree structure itself can be changed randomly, changing the order of evaluation of dimensions of the input data.

Other suggestions and related topics are also welcome. Please do not hesitate to make an appointment.

Required Skills:

- Knowledge of C and C++ programming
- Basic knowledge of machine learning and optimization

Acquired Skills after the thesis:

- Knowledge about modern computer architectures, the organization of cache hierarchies and the ability to design cache-aware algorithms
- Deep understanding of decision trees and possible ways to implement them