EXERCISE X:
A Mini Competition

Abstract

For this exercise, you are given an unsolved real world problem. You may use whatever method you like. You have until April 14th and should work in groups between 4 and 5 people. Your group is supposed to hand in three things: First, your source code. Second, predictions for all datapoints in MLiP_test. And third, a report describing your method(s), why chose them and your results. You can get up to 20 points for this exercise.

The best group(s) will be invited to write a paper for the PAAR workshop and compete in the annual world championship for automated theorem provers.

1 The Problem

Automated theorem provers (ATPs) are programs that, given a set of axioms and a conjecture, try to *automatically* find a proof from the axioms for the conjecture. Their main application areas are formal mathematics (i.e. prove that a human mathematical proof is correct) or software verification (i.e. prove that code follows the specifications).

Automated theorem proving is a search problem. One popular ATP, E, has over 10^{27} different search strategies. Which search strategy is used can have a huge impact in the performance: with the right search strategy, a proof is often found within milliseconds, with a suboptimal strategy it can take years. The first version of the problem is: given a new ATP problem, i.e. a set of axioms and a conjecture, predict which search strategy to use.

Because no accurate prediction method is known, the usual approach is to not use a single search strategy, but instead a sequence of search strategies. Such a sequence is also called a *strategy schedule*. The problem you are tasked to solve is defined as follows: Given a 300 second time limit, a set of axioms, and a conjecture, predict a strategy schedule for problem.

2 The Data

You are given two data files: *MLiP_train* and *MLiP_test_features*. *MLiP_train* contains data for 900 ATP problems. The first line shows the names of the features (f0,..,f21) and the names of the strategies. Each following line consists of the name of a problem, its 22 features, and for the 488 search strategies, the time it took to solve this problem. All times are given in seconds. A -1 indicates that the strategy could not solve the problem in the 300 second time limit. You may use this data to train your models. *MLiP_test_features* has the names and features of the 213 test problems. Your algorithm should create a strategy schedule for each test problem. The file *MLiP_train_example_schedule* is a very simple example schedules for the training problems that were created by *MLiP_template.py*.

3 The Score Function

Your strategy schedules will be evaluated by two measures. First: How many problems were solved by your solutions. Second: How fast did you solve the problems. *MLiP_eval.py* and *eval_schedule.m* contain implementations of the score function that you can use during cross validation.

4 The Competition

At any point during the next two weeks, you may submit schedule for the test problems. Every schedule may use at most 300 seconds. A public list of the scores of the different teams will be shown online.