Incremental feedback in competitive learning

José Paulo Leal and Fernando Silva
{zp,fds}@dcc.fc.up.pt

Competitiveness is an important factor to the success of automatic evaluation systems in computer programming classes. By competing with colleagues most students feel motivated to work harder and improve their scores. However, the opposite may also happen to students that lag behind on the scoreboard. Constantly receiving a plain negative answer is a source of frustration and discouragement, and some students simply give up on successfully completing their assignments.

To cope with this problem we introduced automated feedback in Mooshak. Mooshak is a contest management system with automatic evaluation developed for ICPC competitions, that is increasingly used also as a pedagogical tool. The goal of giving feedback in Mooshak is to provide enough information to students, so that they become more encouraged in finding an adequate solution by themselves.

The feedback given to students is incremental in the sense that repeated information is avoided within certain limits. On one hand, the system gradually increments the amount of information, until a different (hopefully better) evaluation is produced. On the other hand, the system is parametrized with a minimum delay between incremental messages, to avoid spurious submissions intended only to collect all feedback at once.

Another important feature of the automated feedback in Mooshak is its simplicity. Previous attempts to use feedback in Mooshak relied on special evaluators that had to be implemented for each programming problem. The proposed feedback approach can be used with any programming problem and with a minimum effort.

Incremental feedback messages are a list with several items. Currently, the system supports two types of feedback messages items: evaluation and content. Evaluation message items provide detailed information on the evaluation process. For instance, instead of just reporting a Wrong Answer the system reports how many test cases contributed to that classification. Thus, if number of test cases marked with Wrong Answer changes between submissions, then the student will understand if the submission is converging to a solution. Content message items require extra information associated with test cases. Hints are an example of this type of feedback. If an hint is assigned to a test case then the hint is shown to students failing it. Showing the actual test case is the ultimate feedback. Obviously, showing all test cases is not pedagogical as it would tempt students to cheat. Moreover, some test cases are not adequate to be shown to students, typically due to their size. Hence, only those test cases explicitly set as available for feedback are actually shown to students.

A version of Mooshak with automated feedback was used on a course on data structures, with programming assignments given to computer science undergraduates. The students could make an indefinite number of submissions and only the last one was counted. On each submission students received an automatic evaluation with feedback. Each test case wad associated a mark and the final grade was computed by summing marks of accepted test cases. After receiving their grades students were asked to fill-in a questionnaire.

In this paper we give a comprehensive description of the feedback system added to Mooshak and provide details of its implementation. We also present the evaluation of this feature based on empirical results and on the analysis of student's responses, ending with a discussion of the obtained results