

Red Agent Continued

Francois Plamandon

This summer I worked with Philipp Keller (also an NSERC summer student) on an entry to the Trading Agent Competition, Supply Chain Management scenario (TAC SCM). I think it would be best for me to talk right after Philipp, since our presentations are closely related. Philipp will give a quick introduction to the TAC SCM. In this competition, several software agents compete to obtain components from suppliers, manufacture PCs and sell them to customers. I will talk about the inventory strategy used by our trading agent.

The agent schedules the production of more units of a particular PC as long as the expected increase in revenue exceeds the marginal production cost. Determining the utility of producing more units involves predicting the future market prices of PCs.

We used a memory-based learning approach to the price prediction problem. Relevant data is extracted from previous games to produce a set of k -dimensional "reference" data points, where each of the k dimensions corresponds to a feature influencing the market price. Such features include the daily customer demand, the number of units to sell and the number of days elapsed since the start of the game. The relative importance of the different dimensions is decided by a simple gradient descent algorithm. At run time, the points that most closely match the game situation are used for prediction. The nearest neighbour search is performed using the standard search algorithm on kd-trees with incremental distance updates.

Some of the data points are used as queries in order to evaluate the performance of the price predictions. The behavior of the players lead to irregularities in the games themselves, and it seems somewhat difficult to obtain very accurate predictions from game data. The performance improves with the number of data points used. The performance also depends on the query because the data set contains more points for certain value ranges.