In *AAAI-08 Workshop on Trading Agent Design and Analysis* (TADA 08), Chicago, IL, July 2008.

The 2007 TAC SCM Prediction Challenge

David Pardoe and Peter Stone
Department of Computer Sciences
The University of Texas at Austin
{dpardoe, pstone}@cs.utexas.edu

Abstract

The TAC SCM Prediction Challenge presents an opportunity for agents designed for the full TAC SCM game to compete solely on their ability to make predictions. Participants are presented with situations from actual TAC SCM games and are evaluated on their prediction accuracy in four categories: current and future computer prices, and current and future component prices. This paper introduces the Prediction Challenge and presents the results from 2007 along with an analysis of how the predictions of the participants compare to each other.

Introduction

The Trading Agent Competition Supply Chain Management scenario (TAC SCM) (Collins et al. 2006) provides a unique testbed for studying and prototyping supply chain management agents by providing a competitive environment in which independently created agents can be tested against each other in an open academic setting. In order to be competitive, an agent must be able to successfully perform a number of interrelated tasks. While this fact contributes to the complexity and realism of the scenario, it can also make it difficult to determine the relative effectiveness of agent components in isolation. To address this issue, in 2007 two challenges were designed to be run in addition to the full SCM game, each designed to measure an agent's performance on one specific task: a Procurement Challenge, and a Prediction Challenge. This paper focuses on the Prediction Challenge. The contributions of this paper are the specification of this new challenge (designed by the authors), the presentation of the 2007 results, and an analysis of how the predictions of the challenge participants compare to each other. In addition, a brief description of the prediction methods used by the challenge participants and others is provided.

The Prediction Challenge [1]

As the Prediction Challenge is closely tied to the full TAC SCM game, we begin by providing a short summary of the full game. Full details may be found in the official specifications (Collins et al. 2006). In a TAC SCM game, six agents compete as computer manufacturers. Agents must purchase components (CPUs, motherboards, hard drives, and memory, each coming in multiple varieties) from suppliers and sell the assembled computers (coming in 16 configurations) to customers. Both component and computer sales take place through a process involving requests for quotes (RFQs): the buyer sends the seller an RFQ with details such as quantity and due date, the seller responds with a price, and the buyer accepts or declines the offer. Each type of component and computer is assigned a base price that serves as a point of reference, but prices can fluctuate significantly during a game due to factors such as variable customer demand, supplier capacity, and the actions of the agents themselves. Agents are unable to see the prices at which other agents are buying components and selling computers. A game lasts for 220 simulated days, and in each round of the competition, a group of agents competes in a number of games, usually 16.

While the methods used by different SCM agents to manage the supply chain vary considerably, many of these agents share a similar design at a high level - they divide the full problem into a number of smaller tasks and then solve these tasks using decision theoretic approaches based on maximizing utility given various predictions about the economy. The success of an agent thus depends on both the accuracy of the many kinds of predictions it makes and the manner in which these predictions are used, making it difficult to assign credit to individual agent components. To give a concrete example, suppose that based on available statistics from past games and the current one, agent A predicts that it will be able to sell one type of computer for \$2000 on day 45, and agent B predicts that it will be able to sell that computer for \$1900. They then make component purchases, plan manufacturing, and commit to customer orders based on these and other predictions. Ultimately, agent A wins. Is it safe to draw conclusions about the accuracy of these predictions based on this outcome? No.

The goal of the Prediction Challenge is to allow a head to head comparison of agents' prediction accuracy without concern for how these predictions are used. In the example above, if we had recorded the predictions and then observed on day 45 that the specific type of computer sold for an average price of \$1870, we could say that agent B made a more

[1] Software, results, and the complete specifications are available from the Prediction Challenge website: `http://www.cs.utexas.edu/~TacTex/PredictionChallenge`

accurate prediction. This is exactly what takes place in the Prediction Challenge. There are many quantities for which agents may make predictions, such as customer demand, the probability that a particular offer to a customer will be accepted, and supplier capacities. However, the Prediction Challenge focuses only on those predictions that can be expressed in the form of a price, namely component prices and computer prices. As agents need to be able to make predictions about future prices as well as current prices in order to plan effectively, the accuracy of predictions for both current and future prices is measured. There are thus four prediction categories in the Prediction Challenge: current and future computer prices, and current and future component prices.

Instead of making predictions about live TAC SCM games in which they are participating, participants in the challenge make predictions on behalf of another agent called the SCM-PredictionAgent (or PAgent for short). (For clarity, we will refer to the manufacturing agents that participate in SCM games as *agents*, and the prediction agents participating in the Prediction Challenge as *participants*.) Before the competition, the organizers of the challenge run a number of games in which PAgent competes against other agents. The identities of these other agents and the resulting game logs are not made available to participants until after the competition. During the competition, participants connect to a game server which re-plays these games from the game logs. For each day of each game, participants receive the exact messages sent to PAgent (incoming messages), as well as the messages it sent to the game server in response (outgoing messages) - exactly the same information that would be available to an agent during a live game. In addition to these incoming and outgoing messages, each participant is also given a set of predictions that must be made before the information for the following day will be sent.

There are a number of benefits to running the competition using logs from completed games instead of using live games. First, there is no restriction on the number of participants that may compete head to head at one time. Second, each participant will receive exactly the same information about the state of each game and will be asked to make the same predictions. Finally, in live games there would be an incentive for participants to behave differently than in normal TAC SCM games, such as by manipulating prices in order to make past predictions come true.

Although predictions could be made on behalf of any agent from a completed game, the use of a single agent (PAgent) for which source code is available simplifies the task of participants by helping them to understand exactly what behavior to expect from the agent. PAgent was designed to be as simple as possible and to behave in a consistent and predictable manner while still exhibiting reasonable behavior. (PAgent was developed by the authors and is an extension of their TacTex Starter Agent[2] , which is in turn a simplified version of their TacTex agent (Pardoe and Stone 2008) made available for educational purposes.)

The exact predictions that are made by each participant are as follows:

- **Current computer prices:** The price at which each RFQ sent from customers on the current day will be ordered (i.e., the lowest price that will be offered by any manufacturer for that RFQ). These predictions are required on all but the first day and the last two days of each game, when few or no computers are sold. If the RFQ does not result in an order, the prediction will be ignored when accuracy is evaluated. Therefore, participants do not need to be concerned with whether an order will result, only what the price will be if there is an order.
- **Future computer prices:** For each of the 16 types of computers, the median price at which it will sell 20 days in the future. These predictions are required on all but the last 22 days of each game (thus the last day *on* which current computer price predictions is required is the last day *for* which future computer price predictions are required). If no computers of a certain type are sold, the prediction for that type will be ignored when accuracy is evaluated.
- **Current component prices:** The price that will be offered for each RFQ sent by the PAgent to a supplier on the current day. The PAgent sends RFQs to suppliers on all but the last 10 days of each game. If an RFQ results in no offer (due to the reserve price) or an offer (or offers) with modified quantity or due date, the prediction for that RFQ will be ignored when accuracy is evaluated.
- **Future component prices:** The price that will be offered for each of a number of provided RFQs that will be sent by the PAgent to suppliers in 20 days. For each of the 16 pairs of a supplier and a component that it supplies, a zero-quantity RFQ is provided that will be sent by the PAgent in 20 days with a due date chosen at random between 5 and 30 (or the number of days remaining, if less than 30) days after the date the RFQ is sent. Because the PAgent sends no RFQs during the last 10 days of a game, predictions for future RFQs do not need to be made during the last 30 days of the game.

To test the ability of participants to make predictions for games with various competitors, each participant is required to make predictions for 3 sets of games. In each set, the PAgent will have run against a different group of five competitors chosen at random from the TAC agent repository.[3] Each set contains 16 games, meaning that participants have a chance to improve their predictions through repeated experience with the same group of competitors. Participants make predictions for one game at a time, and must complete the predictions for one game day before receiving information for the next day. Unlike the standard SCM game, participants do not need to compete simultaneously, so they may connect to the game server at any time and make predictions at their own pace. There is, however, an eight hour time limit.

Performance is evaluated separately for each of the four prediction categories. Root mean squared error is used as the scoring metric, and all errors are measured as a fraction

[2] `http://www.cs.utexas.edu/~TacTex/starterAgent`

[3] `https://www.sics.se/tac/showagents.php`

of the base price of the computer/component. Participants are ranked in each category, and the overall winner is the agent with the highest average rank over all four categories.

Prediction Methods

Four participants competed in the 2007 Prediction Challenge: Botticelli (Brown University), DeepMaize (University of Michigan), Kshitij (Indian Institute of Technology Kharagpur), and TacTex (The University of Texas at Austin). TacTex and DeepMaize finished second and third, respectively, in the full 2007 SCM competition, and Botticelli was one of 12 semifinalists. This section provides brief descriptions of the prediction methods used by the top two participants, which have been published in full elsewhere, along with an overview of other prediction methods that have been used by TAC SCM agents. The methods used by Botticelli and Kshitij have not been published or made known to the authors.

DeepMaize (Kiekintveld et al. 2008) makes predictions for current and future computer prices using a k-nearest neighbors algorithm. For each prediction to be made, similar situations from a data set of previous games are identified, and the prediction is based on the prices observed in those situations. Predictions can be made about both the probability of winning an order at a given price and the expected winning price. Situations are chosen and weighted using Euclidean distance between a set of state features such as the date, estimated levels of supplier capacity and customer demand, and observed computer prices. Each neighbor is chosen from a different past game to provide sufficient diversity. DeepMaize uses two separate data sets, one from past TAC SCM tournament data and one from self-play, and updates the weighting of each set online based on past accuracy.

TacTex (Pardoe and Stone 2008) tracks computer prices using a particle filter. For each of the 16 types of computer, TacTex maintains a filter that represents a distribution over possible sales prices (to be precise, the lowest price that will be offered by *another* agent in response to an RFQ for that type of computer). Each particle represents a Gaussian with a certain mean and variance and has a weight indicating its relative likelihood. The distribution over sales prices represented by the filter is the weighted sum of these Gaussians. Each day, a new set of particles is generated from the old. For each new particle to be generated, an old particle is selected at random based on weight, and the new particle's estimate of mean and variance are set to those of the old particle plus small changes, drawn randomly from the distribution of day-to-day changes seen in a data set of past games. The new particles are then reweighted, with the weight of each particle set to the probability of the previous day's price-related observations occurring according to the distribution represented. As with DeepMaize, TacTex uses the distributions generated by these filters during the full TAC SCM game to estimate the probability of winning an order given a certain offer price. In the Prediction Challenge, for each computer RFQ TacTex predicts that the sales price will be the mean of the distribution for that computer, or the price offered by the PAgent if that is lower.

To make predictions for future computer prices, TacTex uses the additive regression algorithm from the WEKA machine learning package (Witten and Frank 1999). Additive regression is an iterative method in which at each step a decision stump is fit to the residual of the previous step, and the sum of the output of the stumps is taken as the output of the model. Using a large number of games including a variety of agent groups from the TAC agent repository (and including the PAgent in each game), TacTex creates a training data set in which each instance represents a future computer price prediction that would have been made and is labeled with the difference between the actual median price for a computer and the price that would have been predicted by the particle filter 20 days previously. Each instance consists of 31 features that represent data available to the agent during the game and are similar to those used by DeepMaize in its k-nearest neighbors approach. During the Prediction Challenge, TacTex makes predictions for each type of computer's future price by adding the change predicted by its learned additive regression model to its prediction of current prices for that type of computer.

DeepMaize tracks component prices by recording the prices offered by each supplier over a number of recent days (five days in the full SCM competition, but only one day in the Prediction Challenge). The price for a component request with a given due date can then be predicted by taking the recorded price for that due date, if one exists, or by linearly interpolating between prices offered on different due dates if not. To improve the resulting predictions, DeepMaize also uses the reduced error pruning tree from WEKA, a form of decision tree, to learn the difference between actual observed prices in a data set of past games and the predictions of the linear interpolation method. This use of regression is similar to the method used by TacTex to predict changes in future component prices; however, instead of only learning to make predictions for the change in prices over 20 days, DeepMaize also includes features that allow it to specify the number of days in the future for which the change should be predicted. As a result, DeepMaize can used its learned model to predict both the corrections needed to the linear interpolation method for the current component price predictions, and the changes in prices expected for the future component price predictions.

TacTex makes predictions about current component prices by attempting to directly estimate the available production capacity of each supplier on each future day. The prices offered by suppliers are determined entirely by the fraction of their capacity that is free before the requested due date, so each offer can be used to determine the free capacity over a certain range. If two offers with different due dates are available, the fraction of the supplier's capacity that is committed in the period between the first and second date can be determined by subtracting the total capacity committed before the first date from that committed before the second. With enough offers over many days, TacTex can maintain a reasonable estimate of the fraction of capacity committed by a supplier on any single day, and use this estimate to make price predictions.

TacTex makes future component price predictions using

Name	*Error*
1. TacTex	0.0455
2. DeepMaize	0.0468
3. Botticelli	0.0471
4. Kshitij	0.0487

Table 1: Current computer prices

Name	*Error*
1. TacTex	0.0916
2. DeepMaize	0.0959
3. Botticelli	0.1024
4. Kshitij	0.1109

Table 2: Future computer prices

Name	*Error*
1. DeepMaize	0.0392
2. Botticelli	0.0417
3. TacTex	0.0428
4. Kshitij	0.1333

Table 3: Current component prices

Name	*Error*
1. DeepMaize	0.0943
2. Botticelli	0.0970
3. TacTex	0.1034
4. Kshitij	0.1389

Table 4: Future component prices

the same method it uses for future computer price predictions. Additive regression is used to learn a model that can predict the difference between current predictions and the prices that will exist in 20 days.

In addition to the prediction methods used by these participants, a number of techniques used in previous TAC SCM agents have been documented, primarily for predicting current computer prices. A previous version of DeepMaize used equilibrium analysis to make predictions about the future state of the market, from which information such as future prices could be extracted (Kiekintveld et al. 2004). CMieux (Benisch et al. 2006) makes predictions about computer prices using a form of modified regression tree called a distribution tree that learns to predict a distribution over winning prices using data from past games. For current component prices, CMieux predicts the price that will be offered for an RFQ with a given due date by using a nearest neighbors approach that considers recent offers with similar due dates. Foreseer (Burke et al. 2006) uses a form of online learning to learn multipliers indicating the impact of various RFQ properties on current computer prices. A previous version of Botticelli (Benisch et al. 2004) used a heuristic in which linear regression is performed on recent computer prices to predict a distribution over winning prices.

Results and Analysis

In this section, we present the results of the Prediction Challenge and then analyze the data in a number of ways.

Results

Tables 1-4 show the prediction accuracy of each participant in each prediction category in terms of RMS error. Table 5 shows the overall place and average rank of each participant. Table 6 shows the five agents against which the PAgent competed in each of the three sets of 16 games. The winning participant, DeepMaize, had the lowest error on both current and future component price predictions, while TacTex had the lowest error on both current and future computer price predictions. For each category, the difference between the top agent and other agents is statistically significant with at least 98% confidence according to paired t-tests comparing the RMS errors for each of the 48 games. A few observations can be made from these results.

First, in each prediction category, the difference between the best and third best RMS error was fairly small, at most 12%. This fact suggests that the prediction methods used by the top three participants are all reasonably effective, and that there may be limited room for improvement. At the same time, the magnitudes of these errors are significant, suggesting that making predictions in TAC SCM is inherently difficult. To give perspective to these results, the agents in the final round of the 2007 TAC SCM competition had average profit margins between 1% and 7.5%, so prediction errors of these (similar) magnitudes could conceivably have a significant impact on agent performance.

Place	*Name*	*Avg. rank*
1	DeepMaize	1.5
2	TacTex	2
3	Botticelli	2.5
4	Kshitij	4

Table 5: Overall placing and average rank of each participant

Set	*Agents*
A	Maxon06, MinneTAC05, DeepMaize05, Foreseer05, PhantAgent06
B	GoBlueOval05, GeminiJK05, RationalSCM05, PhantAgent05, TacTex06
C	PhantAgent06, Maxon06, RationalSCM05, Tiancalli06, PhantAgent05

Table 6: Agents in each of the three sets of games

Also, for both computers and component prices, the ranking of participants is the same for both current and future predictions. This is perhaps not surprising, as it seems reasonable that a participant able to make better short term predictions would have an advantage in making long term predictions. As expected, errors for future price predictions are much higher than errors for current price predictions, roughly by a factor of two.

Average daily errors

We begin our analysis by looking at how prediction errors vary across time. Figures 1-4 show the average RMS errors in each prediction category over all 48 games for each game day. (To improve visibility only the top three participants are shown; Kshitij's errors are consistently higher without displaying notably different patterns.) The most obvious feature of these graphs is that errors are usually very high at the beginning and end of games. Making predictions at the beginning of games can be difficult because there is little or no information about previous prices, and because prices can change rapidly as agents place large component orders (driving component prices up) and begin selling computers as components arrive (driving computer prices down). Computer prices are often unpredictable at the ends of games when agents are trying to sell off their remaining inventory – for each computer type, prices may suddenly become very high or low depending on inventory levels and thus competition. TacTex appears to suffer the most from errors at the start and end of games, especially when predicting component prices, while DeepMaize has particularly low errors in initial component price predictions and is roughly the same as Botticelli elsewhere. Occasional large errors such as these can be very damaging to a participant's overall performance

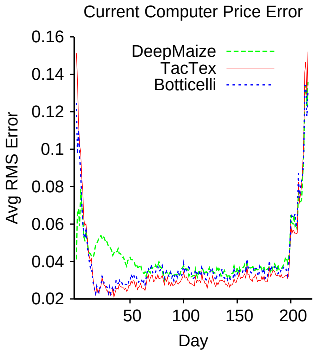

Figure 1: Current computer prices (avg. over all games)

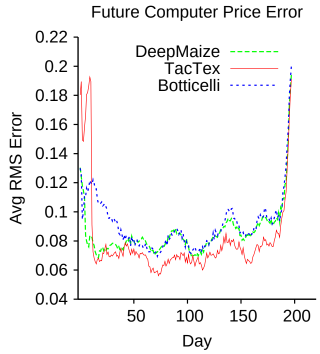

Figure 2: Future computer prices (avg. over all games)

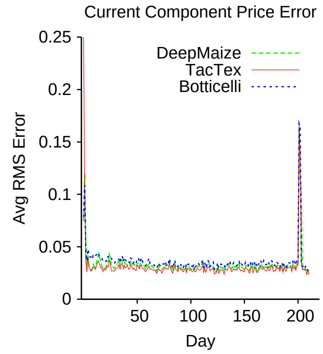

Figure 3: Current component prices (avg. over all games)

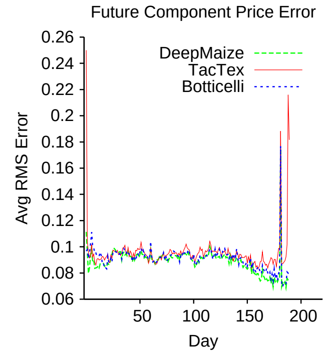

Figure 4: Future component prices (avg. over all games)

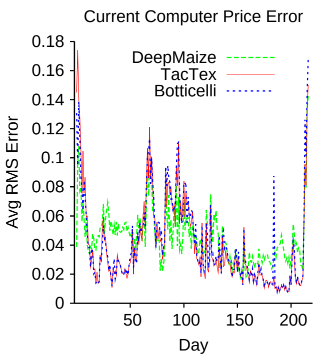

Figure 5: Current computer prices (game A-3)

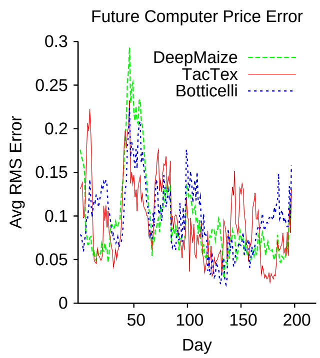

Figure 6: Future computer prices (game A-3)

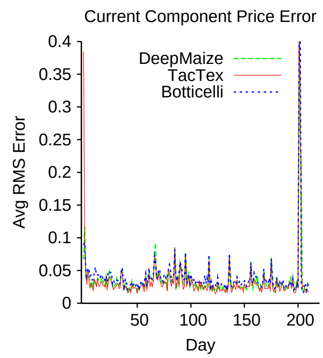

Figure 7: Current component prices (game A-3)

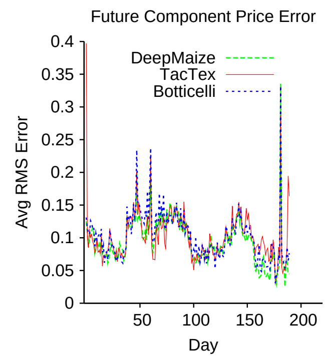

Figure 8: Future component prices (game A-3)

due to the fact that RMS error, and not mean absolute error, is used in scoring.

In some cases, sudden error spikes can be attributed to the behavior of a specific agent. The spike in current component price errors on day 201 (and thus in future component price errors on day 181) occurs only in the games in Set A and is caused by MinneTAC05 sending large requests for components on that date but not accepting the resulting offers, presumably with the goal of driving up prices for other agents. It is interesting to note that Botticelli and TacTex recovered completely (returned to the previous low error level) in two days, while DeepMaize recovered in three days, suggesting that such spikes will only confuse agents for a short period of time.

The timing of the distinct jumps in late-game current computer price prediction errors observable in Figure 1 can also be traced to specific agents. The jump at day 202 occurs only in games from set C and is caused by Tiancalli06 suddenly dropping the prices it offers, while the jump at day 209 occurs only in games from set B and is caused by GeminiJK05 doing the same. The final rise over the last few days appears to be caused by widely varying (often very high) prices resulting from reduced competition to sell certain types of computers.

Compared to the starting and ending errors, average prediction errors during the middle of games tend to be much lower, and they are more consistent both over time and between participants. Still, there are some notable patterns. TacTex consistently has slightly lower errors for current computer price and current component price predictions and significantly lower errors for future computer price predictions, but errors for future component price predictions are generally a little higher than those of Botticelli or DeepMaize. DeepMaize suffers early on from higher errors for current computer price predictions, while Botticelli likewise has higher errors for future computer price predictions over the first portion of games, but otherwise the two participants have extremely similar patterns of errors.

The level of errors for current component prices remains nearly constant throughout games, while errors for future computer prices undergo notable swings for reasons that are unclear. These swings appear to some degree when each of the three sets of games is considered alone, although the swings occur at different times and scales for each set. While somewhat consistent, errors for current computer prices tend to be lower in the early parts of games for TacTex and Botticelli (probably due to the fact that competition tends to remain strong across all computer types while agents work through the components ordered at the start of each game), and errors for future component prices drop near the ends of games for Botticelli and DeepMaize (probably due to the fact that component orders, and thus changes in supplier prices, tend to dwindle during this period).

It is important to note that these observations do not necessarily hold when individual games are analyzed. Figures 5-8 show the daily RMS errors for a single representative

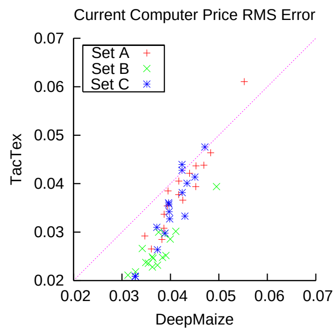

Figure 9: Current computer prices (each game, TT / DM, r=.92)

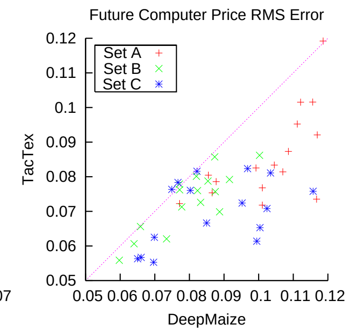

Figure 10: Future computer prices (each game, TT / DM, r=.70)

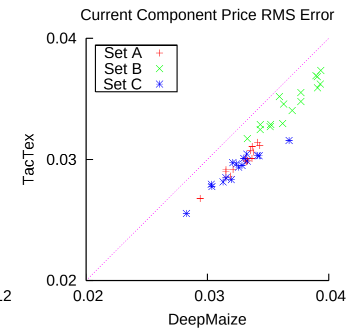

Figure 11: Current component prices (each game, TT / DM, r=.97)

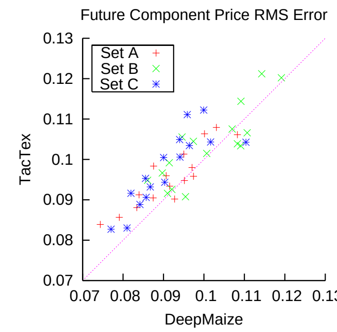

Figure 12: Future component prices (each game, TT / DM, r=.87)

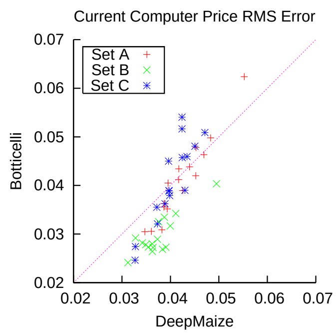

Figure 13: Current computer prices (each game, Bot. / DM, r=.87)

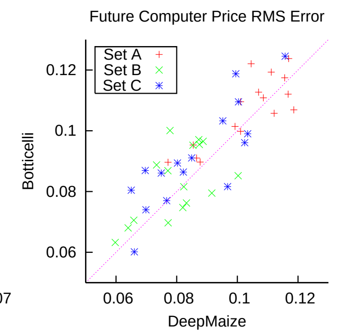

Figure 14: Future computer prices (each game, Bot. / DM, r=.85)

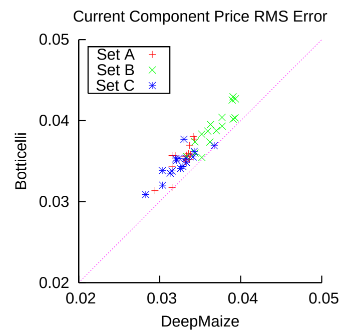

Figure 15: Current component prices (each game, Bot. / DM, r=.93)

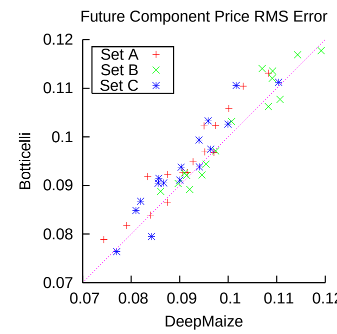

Figure 16: Future component prices (each game, Bot. / DM, r=.95)

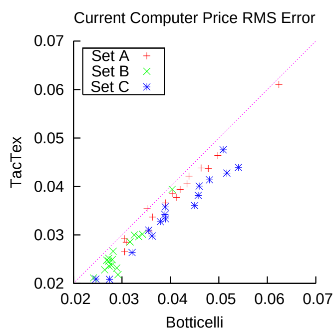

Figure 17: Current computer prices (each game, TT / Bot., r=.97)

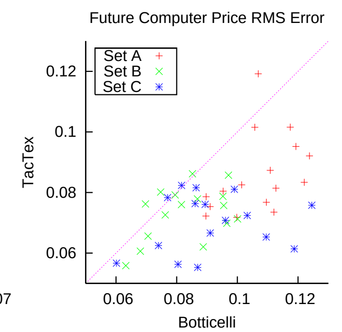

Figure 18: Future computer prices (each game, TT / Bot., r=.49)

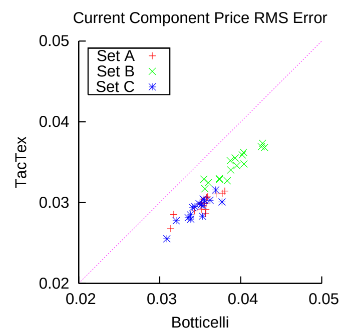

Figure 19: Current component prices (each game, TT / Bot., r=.94)

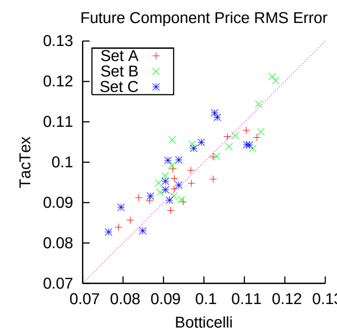

Figure 20: Future component prices (each game, TT / Bot., r=.87)

game, game 3 from Set A. The most striking difference is the fact that for both current and future computer price predictions, errors vary considerably between the participants. Errors also show more variance across time, except for current computer prices, where there are only occasional spikes that are likely caused by unusually heavy component requests.

In the remainder of this paper, we will ignore errors over the first and last 20 days for which predictions are required for each prediction category. Doing so removes the highly variable effects (start and end game conditions, and the spike in component price errors caused by MinneTAC05) that can obscure patterns that would otherwise be visible. Table 7 shows how the elimination of these errors affects the results.

Differences between participants across games

To get a better view of how prediction error varies across games, we now compare the performance of participants on each game individually. Figures 9-12 show the errors for TacTex in each game plotted against those for DeepMaize. Each figure shows a different prediction category. Figures 13-16 show the same information for Botticelli and DeepMaize, and Figures 17-20 compare TacTex to Botticelli. For each figure, the correlation coefficient r is given. The dotted line in each figure is the line $y = x$, meaning that a point below the line represents a game for which the participant on the y-axis had lower error.

We begin by looking at the current computer price predictions. Figure 17 shows that the errors of TaxTex and Botticelli are highly correlated, with Botticelli's errors be-

Name	Current computer prices			Future computer prices			Current component prices			Future component prices		
	start	*mid*	*end*	*start*	*mid*	*end*	*start*	*mid*	*end*	*start*	*mid*	*end*
DeepMaize	**0.0562**	0.0403	**0.0947**	**0.0965**	0.0913	0.1356	**0.0484**	0.0341	0.0810	**0.0920**	**0.0951**	**0.0936**
TacTex	0.0771	**0.0342**	0.1026	0.1473	**0.0774**	**0.1262**	0.0868	**0.0313**	**0.0797**	0.1219	0.0992	0.1210
Botticelli	0.0665	0.0381	0.0984	0.1240	0.0952	0.1408	0.0505	0.0365	0.0858	0.0965	0.0975	0.0969

Table 7: RMS errors over the first 20 days, last 20 days, and middle portion of the prediction interval for each category (lowest error in bold)

ing higher than TacTex's by a similar amount in each game. Comparing either participant to DeepMaize (Figures 9 and 13) paints a different picture. While there is still a strong correlation between errors, it appears that as the difficulty of making predictions in a game increases, the performance of DeepMaize increases relative to the performance of the others, to the point that DeepMaize has the lowest errors of any participant on the most difficult games. One possible explanation for this result is that the prediction methods of other participants (the particle filter in the case of TacTex) are highly tuned for "typical" games and thus suffer as computer prices behave more atypically, while DeepMaize's use of a kNN-based predictor allows it to better handle unusual situations by matching them with similar situations from its data set. This prediction category is the only one in which such a phenomenon occurs, and this fact is particularly interesting because it makes it difficult to state that one participant's method of prediction is best (in expectation) under all circumstances. An agent with access to the prediction methods of all participants might choose to use TacTex's method in most cases but to use DeepMaize's method in certain games where prediction appeared particularly difficult.

Errors for future computer prices (Figures 10, 14, and 18) exhibit a different pattern. Here there is some correlation between the errors of DeepMaize and Botticelli, but very little between the errors of either of these two participants and those of TacTex. In fact, for Set C, the errors of TacTex appear completely unrelated to those of the other two participants. This low correlation suggests that the difficulties experienced by DeepMaize and Botticelli are not related to a particular set of games or common to all games, but have to do with particular situations that can occur in all three sets of games and that TacTex is able to handle correctly. In the case of DeepMaize, these situations may be different from those encountered in the data set used by the kNN-based predictor, or the distance metric used by the predictor may be unable to distinguish these situations from unrelated ones in the data set. The fact that DeepMaize and Botticelli have a higher degree of correlation suggests that they may have difficulties under some of the same circumstances.

The pattern of errors for current component prices (Figures 11, 15, and 19) is much clearer. Here there is a high degree of correlation between the errors of different participants, with the errors of one participant differing from the errors of another by a fairly consistent amount. It should be noted that the reason why TacTex has the lowest errors in these figures, but the third lowest error in Table 3, is the exclusion of the beginning of each game, where TacTex had very high errors.

While not as highly correlated as the errors in current component prices, the errors for future component price predictions (Figures 12, 16, and 20) show a somewhat similar pattern.

In addition to making comparisons between the participants, we can also compare the difficulty of making predictions for each of the three sets of games. For computer prices, it appears to be easier to make predictions for Set B, especially current predictions, while Set A tends to have higher future prediction errors. On the other hand, predicting component prices appears to be more difficult for Set B, especially current component prices. The reasons for these differences between sets are not clear, unlike the error spikes in Figures 1 and 3 that could be traced to specific agent behaviors. Better understanding these differences would likely be useful in designing improved predictors that can handle a wider variety of agent behaviors.

It is interesting to note that the patterns observed above (such as correlations between errors and which participant had the lowest errors) generally appear to hold equally well for all three sets of games. Given that the prediction methods used often require the user to choose a data set composed of past game results, it would not be surprising for a participant to make particularly accurate predictions on games that are most similar to the games in the chosen data set, and for certain participants to favor certain sets of games as a result, but this does not appear to have happened.

Differences between participants across days

To make comparisons at a finer level of detail, we can also plot the errors of each agent on a daily basis, rather than for each game. Figures 21-26 show a subset of the comparisons from Figures 9-20 at this level. Again, RMS error is measured, and the first and last 20 days of errors are omitted. These figures largely serve to shed further light on the observations that have been made previously. Unlike the previous set of figures, no indication is given about the set of games from which each point plotted came, but plotting each set separately reveals very similarly shaped distributions for each set.

Figure 23 shows that the daily errors for the current computer predictions of TacTex and Botticelli are highly correlated, as would be expected from Figure 17. The correlation between TacTex and DeepMaize is much weaker, as seen in Figure 21 (the plot of Botticelli and DeepMaize is nearly the same). As noted before, Figures 9 and 13 show that the performance of DeepMaize tends to improve relative to the other participants as the predictions become more challenging. While a distribution of the same shape (high correlation but a high slope) in Figure 23 would cause this outcome, instead it appears that there are some predictions for which DeepMaize has similar errors (those along the line $y = x$), along with a cluster of predictions (along the bottom-left)

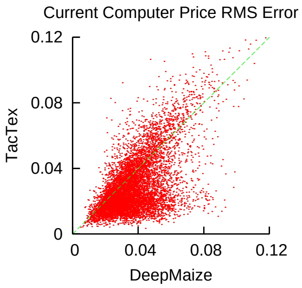

Figure 21: Current computer prices (each day, TT / DM, r=.60)

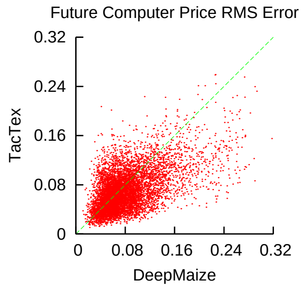

Figure 22: Future computer prices (each day, TT / DM, r=.55)

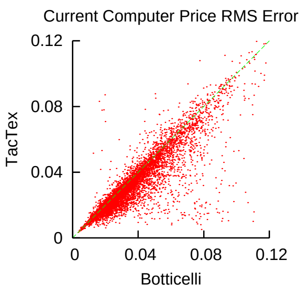

Figure 23: Current computer prices (each day, TT / Bot., r=.89)

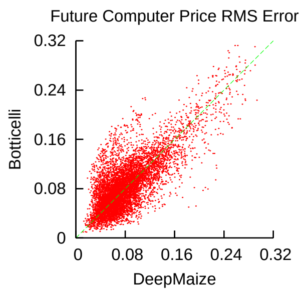

Figure 24: Future computer prices (each day, Bot. / DM, r=.77)

for which DeepMaize has higher errors. It may be the case that there are certain relatively easy predictions with which DeepMaize has difficulty, and that these easy predictions occur less often in the more challenging games. Many of the points in this cluster are from the early parts of games where DeepMaize has higher errors (see Figure 1), but not all – even with the first 70 days omitted from the plot, the cluster is still visible.

Based on Figures 10 and 14, we would expect DeepMaize's daily future computer price prediction errors to be weakly correlated with those of TacTex and somewhat correlated with those of Botticelli, and Figures 22 and 24 confirm this expectation (the plot of TacTex and Botticelli is similar to Figure 22). Looking at Figure 2, in which Botticelli and DeepMaize have nearly identical average daily errors, it is perhaps surprising that their correlation here is not higher.

Figures 25 and 26 show the daily errors for the current and future component price predictions of DeepMaize and TacTex (plots for Botticelli are similar). These errors are highly correlated, as they were in Figures 11 and 12. Figure 25 illustrates that the pattern observed for a single game in Figure 7 (mostly low errors around 0.03, with occasional spikes that affect all participants similarly) is true in general. Similarly, the pattern seen in one game in Figure 8 (errors more evenly distributed over a wide range but still highly correlated between participants) appears in Figure 26.

One additional observation that can be made is that while a participant may show consistently lower errors at the full-game level (for instance, TacTex in Figures 10 and 11), there may still be a large number of days on which it has higher errors (Figures 22 and 25). This observation may indicate that there is still room for the participant to improve, or it may

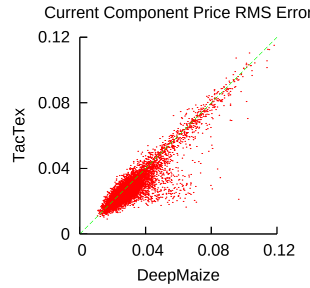

Figure 25: Current component prices (each day, TT / DM, r=.90)

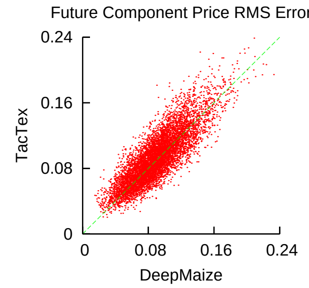

Figure 26: Future component prices (each day, TT / DM, r=.85)

simply be a result of the stochastic nature of the game (that is, for each situation in which predictions are made, there may actually be a wide distribution over possible outcomes depending on random game factors such as demand fluctuations). Both possibilities are likely true to some degree.

An alternative way of presenting this data is to look at the *differences* between the errors of participants and to plot these differences for different pairs of participants. For example, in Figure 30 the x-axis shows the daily future component price prediction errors of DeepMaize minus those of TacTex, while the y-axis shows the errors of Botticelli minus those of TacTex. In this case, the points show a moderately high degree of correlation ($r = .72$), indicating that the situations in which TaxTex differs in accuracy from DeepMaize tend to be the same as those in which it differs from Botticelli. In most cases, these plots are fairly uniform clusters with low correlation, but there are some exceptions, as shown in Figures 27- 30. Figure 27 shows that for current computer price predictions, DeepMaize tends to vary more widely from the accuracy of TacTex than Botticelli does. Interestingly, for those situations in which DeepMaize has considerably higher errors than TacTex (those extending to the right here, and seen on the bottom-left of Figure 21), TacTex and Botticelli have nearly identical errors. For current component price predictions, Figure 29 shows that TacTex and Botticelli usually differ only slightly from DeepMaize and in a weakly correlated way, but that there are certain situations in which DeepMaize has much higher errors than them both. Figures 28 and 30 show that DeepMaize and Botticelli differ from TacTex in similar situations for future computer and component price predictions. The degree of correlation for these two categories drops considerably when comparing differences with a participant other than TacTex, suggesting that Botticelli's methods of making predictions of future prices have more in common with those of DeepMaize than TacTex.

Error persistence

Another question we can address is whether errors tend to persist. That is, if a participant has a high prediction error on one day, will it also have a high error on the next day? We answer this question by plotting daily errors against the next day's errors. In most cases, the answer is yes; for current and future computer price predictions and future component price predictions, errors are highly correlated across days,

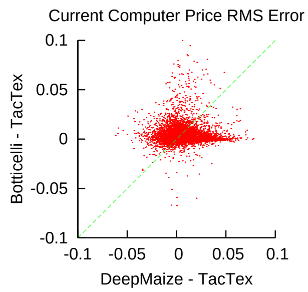

Figure 27: Current computer prices (daily differences with TT, r=-.05)

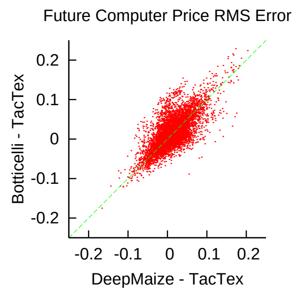

Figure 28: Future computer prices (daily differences with TT, r=.71)

with correlation coefficients above .73 for current computer price predictions and above 0.82 for both future price predictions for each participant. The fact that correlation is highest for the future price predictions makes sense given that game conditions can change significantly over 20 days while the information available to make predictions about these conditions changes little from day to day. In the case of current component price predictions, however, there is little correlation between daily errors. Figure 31 shows the plot for daily errors of TacTex on consecutive days, and the plots for other participants are similar. There appear to be a large number of cases in which the error jumps greatly for a single day (the vertical arm) and then decreases to a more normal level on the following day (the horizontal arm). Even on days with more moderate errors, the degree of correlation between days is fairly low. These observations make sense given the error pattern seen in Figure 7.

As before, we can also consider the differences between the error rates of different participants. When daily differences in errors are plotted for consecutive days, the differences in future computer price prediction errors turn out to be highly correlated across days ($r > 0.85$ for each pair of participants), and the differences in future component price prediction errors turn out to be moderately correlated (r of about 0.6 for each pair). For current computer price predictions, the results depend on the pair of participants considered. While the differences between the errors of DeepMaize and TacTex show a moderately high degree of correlation across days ($r = 0.77$), the differences between Botticelli and TacTex are not only uncorrelated across days, but as shown in Figure 32, there are a number of cases in which the error of Botticelli jumps from being nearly the same as TacTex to being much higher for a single day. For current component price predictions, a similar pattern emerges when comparing DeepMaize to either TacTex (Figure 33) or Botticelli, while the differences between the errors of TacTex and Botticelli are similarly uncorrelated across days without showing such jumps.

Individual error distribution

Finally, it is possible to look beyond the RMS error of a set of predictions covering a whole game or day and consider the errors of individual predictions. As this level of detail is not available in the competition logs, we give here only a brief look at the data we were able to generate for TacTex

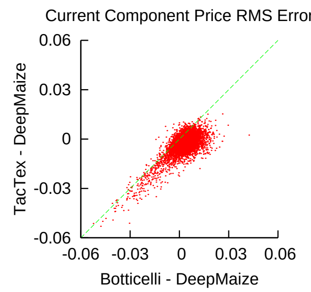

Figure 29: Current component prices (daily differences with DM, r=.70)

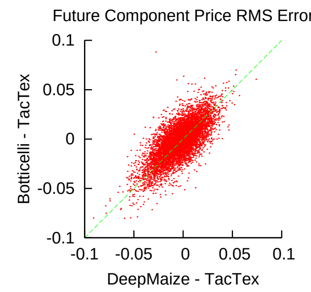

Figure 30: Future component prices (daily differences with TT, r=.72)

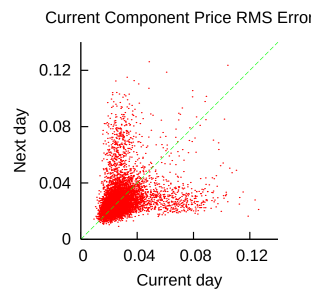

Figure 31: Current component prices (errors on consecutive days, TT, r=.23)

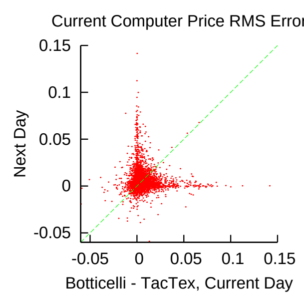

Figure 32: Current computer prices (daily differences, Bot-TT, r=.03)

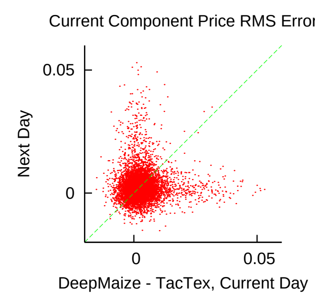

Figure 33: Current component prices (daily differences, DM-TT, r=.02)

after the competition. Instead of considering RMS error, we look at the actual errors to determine their distribution. For current price predictions, errors are grouped into 40 bins of width 0.005 spanning the range [-0.1, 0.1], and for future price predictions, errors are grouped into 40 bins of width 0.015 spanning the range [-0.3, 0.3]. Errors falling above or below these ranges are grouped into an additional bin. Figures 34-37 show the resulting histograms. In each category, errors appear to be nearly normally distributed with a mean near zero. This result is not necessarily unexpected, as we would expect a normal distribution if the errors were due to a large number of uncorrelated factors. However, given the nature of the TAC SCM game, it would also not have been surprising if a more interesting pattern had emerged. For instance, it is fairly simple for a single agent to drive computer prices down by offering lower prices, or to drive component prices up by requesting a large number of components. It is less likely for these prices to suddenly move in the opposite direction, and so the distributions could conceivably have shown a tendency toward larger errors in one direction.

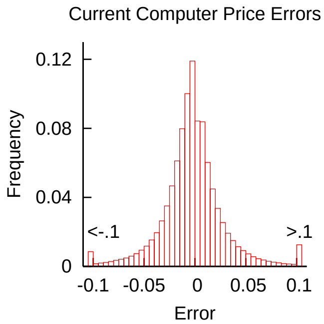

Figure 34: Current computer price error distribution for TacTex

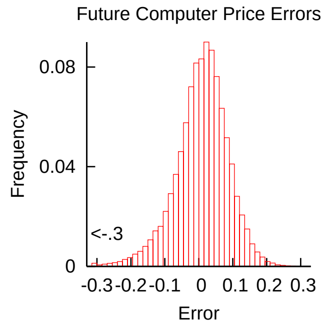

Figure 35: Future computer price error distribution for TacTex

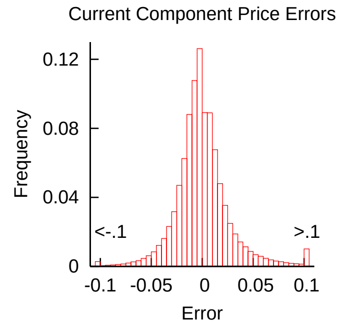

Figure 36: Current component price error distribution for TacTex

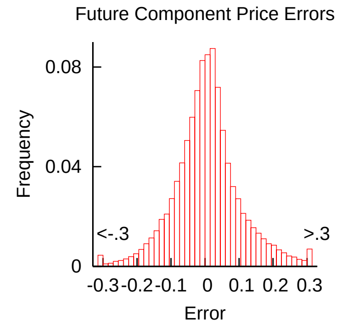

Figure 37: Future component price error distribution for TacTex

The fact that this tendency was not observed suggests that the prediction methods used may already account for such factors.

Conclusion

We have introduced the TAC SCM Prediction Challenge and analyzed the results of the 2007 competition. Our analysis showed that different prediction methods can achieve similar prediction accuracy and that errors are frequently, but not always, correlated. At the same time, some participants are clearly stronger than others in certain areas. TacTex regularly had lower errors in three of the four prediction categories during the middle of each game, but suffered from high errors at the start and end of games. The winner, DeepMaize, was fairly effective in all aspects of the challenge.

There are many additional ways in which the results of the competition could be analyzed. We have focused on giving a high-level comparison of the prediction accuracy of the participants, but it would also be possible to continue this analysis at a finer level, such as by comparing accuracy on predictions for individual RFQs or by trying to identify the specific conditions under which one agent outperformed another. Such analysis could be useful in helping participants to identify the shortcomings of their prediction methods and to make future improvements.

Acknowledgments

We would like to thank the SICS team for developing the TAC SCM game server, all teams that have contributed to the agent repository, and the participants in the Prediction Challenge. This research was supported in part by NSF CAREER award IIS-0237699.

References

Benisch, M.; Greenwald, A.; Grypari, I.; Lederman, R.; Naroditskiy, V.; and Tschantz, M. 2004. Botticelli: A supply chain management agent. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 3, 1174–1181.

Benisch, M.; Sardinha, A.; Andrews, J.; and Sadeh, N. 2006. Cmieux: Adaptive strategies for competitive supply chain trading. In *Eighth International Conference on Electronic Commerce*.

Burke, D. A.; Brown, K. N.; Hnich, B.; and Tarim, A. 2006. Learning market prices for a real-time supply chain management trading agent. In *AAMAS 2006 Workshop on Trading Agent Design and Analysis / Agent Mediated Electronic Commerce*.

Collins, J.; Arunachalam, R.; Sadeh, N.; Eriksson, J.; Finne, N.; and Janson, S. 2006. The Supply Chain Management game for the 2007 Trading Agent Competition. Technical report. Available from https://www.sics.se/tac/tac07scmspec.pdf.

Kiekintveld, C.; Wellman, M.; Singh, S.; Estelle, J.; Vorobeychik, Y.; Soni, V.; and Rudary, M. 2004. Distributed feedback control for decision making on supply chains. In *Fourteenth International Conference on Automated Planning and Scheduling*.

Kiekintveld, C.; Miller, J.; Jordan, P. R.; Callender, L. F.; and Wellman, M. P. 2008. Forecasting market prices in a supply chain game. *Submitted to Electronic Commerce Research Applications*.

Pardoe, D., and Stone, P. 2008. An autonomous agent for supply chain management. In Adomavicius, G., and Gupta, A., eds., *Handbooks in Information Systems Series: Business Computing*. Elsevier.

Witten, I. H., and Frank, E. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.