Forecasting Market Prices in a Supply Chain Game

Christopher Kiekintveld, Jason Miller, Patrick R. Jordan, and Michael P. Wellman

University of Michigan
Computer Science and Engineering
Ann Arbor, MI 48109-2121 USA
{ckiekint,mijason,prjordan,wellman}@umich.edu

ABSTRACT

Future market conditions can be a pivotal factor in making business decisions. We present and evaluate methods used by our agent, Deep Maize, to forecast market prices in the Trading Agent Competition Supply Chain Management Game. As a guiding principle we seek to exploit as many sources of available information as possible to inform predictions. Since information comes in several different forms, we integrate well-known methods in a novel way to make predictions. The core of our predictor is a nearest-neighbors machine learning algorithm that identifies historical instances with similar economic indicators. We augment this with an online learning procedure that transforms the predictions by optimizing a scoring rule. This allows us to select more relevant historical contexts using additional information available during individual games. We also explore the advantages of two different representations for predicting price distributions. One uses absolute prices, and the other uses statistics of price time series to exploit local stability. We evaluate these methods using both data from the 2005 tournament final round and additional simulations. We compare several variations of our predictor to one another and a baseline predictor similar to those used by many other tournament agents. We show substantial improvements over the baseline predictor, and demonstrate that each element of our predictor contributes to improved performance.

General Terms

Economics, Experimentation, Measurement

Keywords

Forecasting, Markets, Price prediction, Trading agent competition, Supply chain management, Machine learning

1. INTRODUCTION

Future market conditions can be a pivotal factor in economic decisions. We present and evaluate methods of forecasting market conditions used by our agent, Deep Maize in the 2005 Trading Agent Competition Supply Chain Management game. There are several advantages to studying forecasting in this domain. The SCM game offers a challenging scenario, but in a relatively accessible and controlled environment amenable to extensive experimentation. Participation also requires a fully automated agent that makes end-to-end decisions. This offers opportunities for evaluating predictions based on how they affect decisions in addition to raw error metrics. Finally, TAC attracts a diverse pool of participants. These provide both performance benchmarks and opponents. Predictions in multi-player games are strongly influenced by the full profile of player strategies, so the option to test prediction methods based on a diverse and unanticipated set of opponents is a key advantage of TAC.

Real prediction environments often provide a variety of sources of information that may inform forecasts. The guiding principle of our forecasting design for Deep Maize is to support exploitation of all available sources. Our contribution is to integrate several well-known ideas in a novel way that allows us to make effective use of the various types of information available. We first discuss the different forms of information available for making forecasts in TAC SCM and highlight some strengths and weaknesses of each. Many of these forms and the associated tradeoffs are likely to arise in other forecasting problems. We then present our system for forecasting prices, describing three distinct elements. The first is a nearest-neighbors algorithm that we use to induce the relationship between features and market conditions from historical games. Next, we describe two different representations we use for predictions; an absolute representation and a relative representation that exploits local price stability. The final element is an online adaptive procedure that combines and improves predictions by computing a transformation that optimizes a logarithmic scoring rule.

We evaluate these techniques using both prediction error metrics and simulation performance. As a point of reference we use a baseline predictor that is similar to the methods used by many of the other competitors. Our predictors show substantial improvement over the baseline on both error and simulation metrics. The technique we use for online learning shows particularly strong performance across a range of conditions.

2. THE TAC SCM SALES MARKET

Agents in the TAC Supply Chain Management game play the role of personal computer (PC) manufacturers competing against five other participants to maximize profits over a simulated year [11, 10].[1] Each game has 220 simulated days, lasting 15 seconds each. Agents assemble 16 different types of PCs for sale, defined by compatible combinations of CPU, motherboard, hard disk, and memory components. Each day agents negotiate with suppliers to purchase components, negotiate with customers to sell finished

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
AAMAS'07 May 14–18 2007, Honolulu, Hawai'i, USA.
Copyright 2007 IFAAMAS .

[1]The full specification and general information about TAC is available at `http://www.sics.se/tac`.

products, and create manufacturing and shipping schedules for the next day. Each agent has an identical factory with limited capacity for production.

Here we focus on the task of predicting prices in the market for finished goods (PCs).[2] Negotiations for PC sales use a simultaneous sealed-bid auction mechanism. Each day, the customer generates a set of requests (short for request-for-quote, or RFQ) representing total demand. A request specifies a PC type, due date, reserve price, and penalty for late delivery. Agents respond with bids, specifying price offers for a subset of the RFQs. Each RFQ is then awarded as an order to the lowest bidder, who is then obligated to deliver the product at the offered price.

There is a new round of auctions for each day's batch of RFQs. The total demand (number of RFQs) varies according to a random walk with a variable trend; the underlying parameters of this walk are not observable. The PC types are divided into three market segments with separate demand processes. Supplier capacities also vary according to a mean-reverting random walk, and this in turn affects the prices for purchasing components. Given the demand and supply patterns and overlapping component requirements of PC types, there is a strong interaction among auctions both within and across days.

3. RELATED WORK

Forecasting has been studied across several disciplines, and a wide variety of approaches have been developed. These include exponential smoothing, GARCH, ARIMA, spectral analysis, neural networks, Bayesian networks, Kalman filters, and others [5, 7, 13]. The best method for any particular problem depends on many factors, including the type and quantity of information available and the properties of the time series itself. Much of the applied work on forecasting focuses on a single prediction method or compares several methods to determine the "best" one for a given application [1, 19]. Predictions are typically evaluated on the basis of prediction error; it is somewhat rare to find forecasting integrated with a system for decision-making. One element of our approach is an online algorithm that combines predictions to optimize a logarithmic scoring rule. A variety of approaches exist for combining predictions [2, 9]. Logarithmic scoring rules in particular have been used to support information aggregation in the context of prediction markets [12].

Price prediction was identified as an important element of agent behavior in the original TAC Travel game. In this travel shopping scenario, agents have applied a variety of approaches for predicting the closing prices of hotel auctions, including both machine learning and model-based approaches [8, 23]. One striking observation from a study of prediction methods applied in the TAC-02 tournament [25] was that the common element among best predictors was not the specific method employed, but rather the information taken into account—in this example the initial flight prices in individual game instances.

Researchers have developed a variety of methods for predicting prices in the TAC SCM game. Many of these approaches use only price information from the current game instance to predict prices, and do not attempt to predict changes in prices from the current levels [3, 4, 6, 14]. Pardoe and Stone [20] explored a variety of machine learning techniques to estimating the probability of winning the current customer auctions. This predictor was not used in the 2005 version of TacTex, which used a simple heuristic method coupled with an online adaptive procedure [22]. The 2006 version of TacTex does use a more advanced predictor based on machine learning methods, and attempts to predict changes in future price levels [21]. Ketter et. al. [15] developed methods for predicting changes in market levels based on estimating discrete market "regimes." This method has not yet been integrated into an agent, so the value of the approach is not yet clear.

[2]There is also an interesting problem in predicting the cost of supplies. Our approach for this problem is less developed, partially due to changes in the specification of this market.

4. SOURCES OF INFORMATION

There are various sources of evidence about future market prices in TAC SCM. Each is imperfect, but provides unique insights into market activity. We start with information available during a game instance. Agents observe all of the requests issued each day, and get low fidelity information about overall market prices. Each day they receive high and low selling prices for each PC type from the previous day. Every 20 simulation days they receive a report containing the average selling price and total quantity sold during the preceding period. Agents are also informed of whether or not they won each auction they bid in. The details of individual market transactions and opponents' bids are not revealed. In addition, there is information available about the supply market that may be indirectly related to customer prices. The primary advantage of evidence from the current game is that it exactly matches the conditions of interest. In particular, it reflects the exact profile of agent strategies being used. However, this information is of low fidelity and says little about how conditions are likely to evolve - especially if the current conditions have not been observed earlier in the game.

A complementary source of available evidence is historical game instances. After each game finishes, logs are available with detailed information about all transactions and game state. This precise information shows the evolution of actual market prices and game state, which is potentially very useful for identifying leading indicators of price changes. There are two important limitations of this data. One is that the exact state (inventory, capacities, price history, etc.) will not be identical to the state in the current game, and even if it were, agents could not tell. Differences in the profile of strategies playing the game are also critical, since market prices are determined by aggregate agent behaviors. Even having an agent with the same name is not a reliable indicator of an identical strategy, since agents are continually updated over the course of tournament play.

There are difficult questions for any method that reasons from prior games regarding how to select training instances based on strategy profiles that are likely to be representative of the target environment (e.g., the tournament finals). One issue is simply identifying "similar" strategies, based on some prediction of the likely strategic environment. Even given a similarity metric, there is a tradeoff between having similar training instances and having a large data set.

5. FORECASTING METHODS

Deep Maize uses price forecasts extensively for making decisions. A full description of the decision-making architecture is available in [16]. For each day, the agent forecasts the price curves it faces in the customer market. On the current day the set of requests is known so this amounts to estimating the probability of winning each RFQ auction for any bid, denoted $\Pr(win|bid)$. For future days, the agent combines a prediction of the distribution of selling prices with a prediction of the requests that will be generated to form an *effective demand curve*. This curve encodes the price the agent expects to receive for selling each additional PC on each possible day. These forecasts are used to make bids in the customer auctions and project a production schedule, both using approximate optimization.

To predict the requests that will be generated in the future, the agent uses a Bayesian network model to estimate the hidden mean and trend parameters of the demand generation process. This model was introduced in [18].[3] The properties of individual RFQs (e.g., quantity, due date) are drawn from known static distributions. In the sequel, we describe our methods for predicting the current and future distributions of selling prices. This is a much more challenging task because it depends on the behaviors of other agents. We note that these methods have been tuned in various ways since the 2005 tournament, but the basic structure remains the same and was used for the simulation experiments presented here and our 2006 tournament entry.

5.1 Learning From Historical Data

We use a k-nearest neighbors algorithm to induce the relationship between indicators of market conditions (features) and likely distributions of current and future prices. The idea is simple: given a set of games, find situations that most closely resemble the current situation and predict based on the observed outcomes. While this method may not provide much advantage when prices are stable, if has benefits when prices are changing or the agent encounters novel situations within a game instance.

We use weighted Euclidean distance between state features to define similarity. Each feature has both a weight and bound; if any individual feature distance is greater than the bound the overall distance is infinite. We normalize each feature by its standard deviation and constrain the k neighbors to come from different games to ensure diversity. We use different features and weightings to predict the current day's price distribution and future sales distributions. We experimented with a range of features and weightings, selecting a measure that performed well after extensive but ad-hoc exploration. The potential features included the current simulation day, statistics about current prices, and estimates of underlying supply, demand, and inventory conditions. Many of these were motivated by analysis showing that related measures were strongly correlated with customer market prices in previous competitions [17].

The results presented here use the following equally weighted features for the current day lookup: average high and average low selling price from the previous 5 days, estimated mean customer demand, the simulation day, and the linear trend of selling prices from the previous 10 days. The features for the long-term lookup are: average of high selling price from the past 5 days, estimated mean customer demand, estimated customer demand trend, simulation day, and the linear trend of selling prices from the previous 10 day period. When extracting data from historical games we adjust for the number of opponent agents by randomly selecting one agent and removing that agent's bid, if it exists. We also smooth the observed distributions by considering all requests in the range $d-10,\ldots,d+10$.

5.1.1 Data Sets

We employ two distinct data sets for making predictions: one based on recent tournament games and one based on designed simulations. The tournament data is a recent set of games selected for competitiveness (all agents playing and no obvious failures). During tournament play, this set was automatically updated with games from the current round. The second data set is based on self-play games. We played a sequence of games, continually updating the training data to include the most recent games (the initial data set contained tournament data). The self-play data set contains the last 45 games of this sequence. This motivation for this setup was that it should converge towards a configuration resembling a competitive equilibrium. This in turn might be more reflective of the strategic environment in the final round where six very competitive agents face off directly. This situation is relatively rare in earlier rounds. Our analysis indicates that the predictions based on this data set were not very useful, but we are optimistic that a better equilibrium model can be achieved in future research.

5.2 Representation

We experiment with two different representations for the distribution of selling prices. One is based on absolute price levels, and the other is relative to recent price observations. The advantage of absolute prices is that they translate readily across games and can easily represent changes in the overall level of prices. The relative distribution is centered on moving averages of the high and low selling price reports for the previous 5 days. Between these two averages we use 60 uniformly distributed points, above the average of the highs and below the average of the lows we use 20 points distributed uniformly. The strength of this representation is in its ability to exploit the local stability of prices. It effectively shifts observations from historical games into the relevant range of prices for the current game, which is often small.

To compute the estimated price distribution we take the average of the distributions stored for the k nearest neighbors. After averaging, we enforce monotonicity in the distribution. To make a forecast i days into the future, we consider what happened i days in the future in each of the historical games. [4] The absolute distributions can be averaged directly. For the relative repesentation, we account for changes in the overall prices over time by modifying the pricing statistics. These are adjusted based on the average price changes over the horizon i in the historical data.

5.3 Online Learning

We employ an online learning procedure that optimizes predictions according to a logarithmic scoring rule.[5] This serves two purposes. First, it computes a linear combination of the predictions based on the two data sets described in Section 4 (tournament and self-play). This allows us to combine these sources of information based on their predictive properties in the current environment, essentially selecting online the data this is most representative of the set of agents we are currently facing. Second, it computes an affine transformation of the combined predictions to correct for systematic errors. The possible transformations are scored using the recent history of bids won and lost, a valuable but otherwise unused source of information. The scoring procedure optimizes a function of the form:

$$a(b \cdot P_{\text{tournament}} + (1-b) \cdot P_{\text{self-play}}) + c, \tag{1}$$

where $P_{\text{tournament}}$ and $P_{\text{self-play}}$ are the predictions based on the two data sets and a, b, and c are constants optimized using a brute-force algorithm. The optimization minimizes $-\sum_{\text{bids}} \log(\alpha_{\text{bid}})$ where

[3] The code for this estimation was released to the TAC community, and can be found at `http://www.eecs.umich.edu/~ckiekint/downloads/DeepMaize_CustomerDemand_Release.tar.gz`

[4] To allow this, we require the starting points to have exactly the same simulation day.

[5] We use a logarithmic scoring rule because it is a strictly proper scoring rule, so the optimal score is obtained only for the true probability. Scoring according to other types of rules can lead to biased estimators.

α_{bid} is the magnitude of the difference between the predicted probability of winning and the actual event (1 or 0). This optimization is performed using the *actual* bids submitted by the agent in recent days, and only modifies the distribution for the range where data is available. To prevent extreme predictions we set (loose) bounds on the values of the constants.

6. EVALUATION

We evaluate our forecasting techniques using both prediction error and simulation performance. We evaluate several variations of the predictor that employ subsets of the functionality described in Section 5 to assess the contributions of each. As a baseline we use a heuristic predictor described in [20] (labeled "TacTex-04" in figures). The baseline predictor uses only information contained in the current price level, and relies completely on local price stability for predictive power. This is similar to what our predictor might look like using the relative representation without any historical data or online transformations. This predictor produces surprisingly accurate predictions despite its simplicity. The reason is that prices in the TAC SCM game tend to be quite stable for long periods, especially if the start and end game conditions are excluded. [6] One of the primary reasons we use this as a baseline is that we believe it is reflective of the "default" heuristic methods used by many of the agents to make decisions (especially in the early years of the competition). In particular, it only makes use of local information about prices in the current game, and does not attempt to project changes in price levels over time.

The baseline method predicts the distribution of selling prices using a weighted average of uniform densities between the low and high prices from the previous 5 days.[7] We project this into the future by assuming that the distribution of selling prices does not change and adjusting for the predicted change in the number of requests generated, as given by the Bayesian model of customer demand.

We present results for the baseline and four variations of our kNN-based predictor. These variations use either the relative or absolute representation, and either apply online affine transformations or not. All results use both the tournament and self-play data sets, which contain 95 and 45 game instances respectively. Regardless of whether overall affine transformations are applied we use the online scoring method to weight the predictions from these two sources. At the end of this section we briefly discuss the effects of varying the data sets. All error results are derived from the 14 clean finals games.[8]

6.1 Prediction Error

We discuss two different types of prediction error. The first is for predictions made about the probability of winning the current day's auctions, and the second is for predictions about the effective demand curve on future days.

As we discuss the prediction error results, it is important to keep in mind that there is a substantial amount of uncertainty inherent in the domain model; we do not expect *any* prediction method to achieve anywhere near perfect predictions. To demonstrate this and provide some calibration, we consider the process that generates the overall level of customer demand in the SCM game. The level of customer demand (number of requests generated each day) is one of the most important factors influencing market prices. Recall that we use a Bayesian model to track the mean customer demand and trend; this model is the theoretically optimal predictor of the process. We ran tests comparing this predictor to a naive predictor that predicts that the mean demand on all future days will remain exactly the demand observed today. The optimal Bayesian model improved on the naive model, but the difference was relatively small. The naive model had and RMS error of approximately 35, while the Bayesian model achieved an RMS error or approximately 30 - a reduction on the order of 15%. While this may seem a relatively small improvement in prediction quality, these small improvements can translate into substantial differences in overall agent performance.

6.1.1 *Current Day Prediction Error*

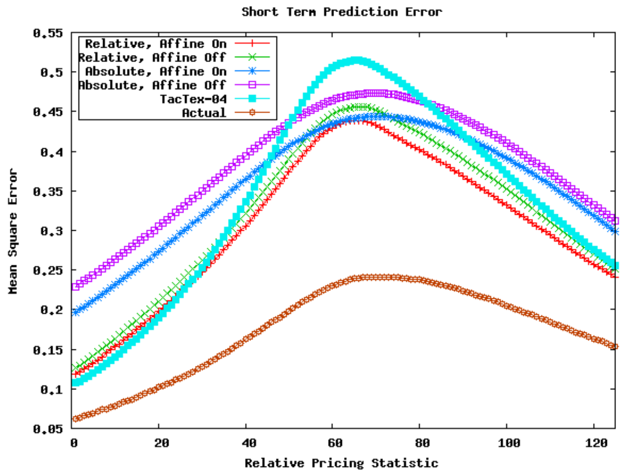

Figure 1: RMS prediction error for current day predictions in the TAC SCM final round games. Error is plotted relative to a uniform distribution around current market prices.

Predicting conditional distribution $\Pr(win|bid)$ for the current day's RFQ set is an important special case because these estimates are used directly for making bidding decisions. It is also distinct because is the only day for which we have the actual set of requests available; for future days, there is substantial additional uncertainty since the quantities, PC types, due dates, penalties, and reserve prices are all determined randomly.

Figure 1 shows root mean squared (RMS) error over the conditional probability distribution for each of the five predictions. The error depends on the bid level; for instance, a predictor may predict a 70% chance of winning a bid of \$100 when the actual probability is 80%. The same predictor may correctly give a 90% chance of winning a bid of \$80. The errors are averaged over all of the finals games from 2005, restricted to the middle part of the game (days 20–200).[9] Showing error relative to possible bids allows us to present a more detailed picture of how errors are distributed. The possible bid levels are determined by 125 relative pricing statistics that account for the differences in absolute prices across games. These 125 points are distributed uniformly between 10% above and

[6] A linear regression on the average selling prices from one day to the next results in an R^2 value on the order or 0.99, so current prices are extremely predictive of prices in the near future.

[7] We use weights of 0.3 for the most recent two days, 0.2 for the middle day, and 0.1 for the oldest two days.

[8] In two games (3718 and 4254) the University of Michigan network lost connectivity and this team's agent was unable to connect to the game server for a large part of the game. This distorts the results for uninteresting reasons, so we omit these games.

[9] We restrict to the middle of the game because the baseline predictor is not designed to have reasonable behavior in these border cases (for example, when no initial price levels have been observed).

below the average high and low selling prices from the previous 5 simulation days. We note that bids towards the ends of this distribution (much higher or lower than the recent price levels) are very rare. Deep Maize's bids are centered around 62 on this scale, with a large majority falling between 50 and 80.

The "Actual" predictor in Figure 1 represents the best possible distribution-based predictor, derived from the actual selling prices in these game instances. Of the candidate methods, the the relative predictor with affine updating has the lowest overall prediction error. Affine updating reduces prediction error across the entire distribution for both the absolute and relative representations. The relative predictors generally perform better than the absolute predictors, though the absolute affine predictor has lower error than the relative predictor without affine updating in the center of the price distribution where the affine updates have the greatest amount of data. All four of our predictors have lower error than the baseline in the center part of the distribution, though the absolute predictors show greater error towards the extremes. Conceptually, errors towards the center of the distribution are likely to be much more important because bids are centered there. However, is is not clear how errors should be weighted, since bid densities may vary in different situations (and certainly for different agents). These predictions are also used for making non-bidding decisions, which may use the information in different ways. Using simulations is one way to resolve this dilemma, since we can test the predictors as they are used by a real decision-making procedure.

6.1.2 *Future Forecasting Error*

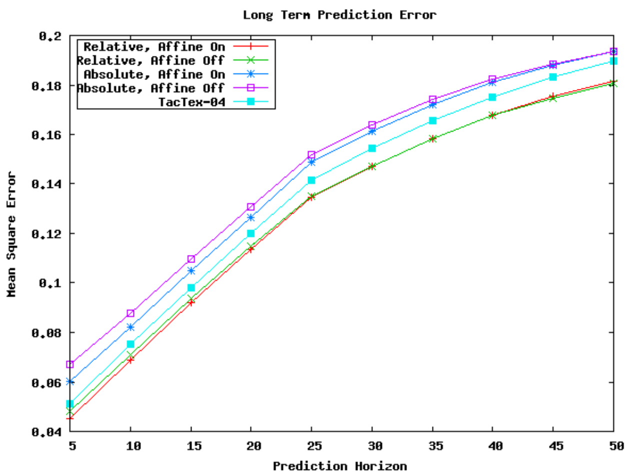

Figure 2: RMS error between predicted and actual effective demand curves out to a horizon of 50 days, computed on the 2005 finals.

A distinctive feature of Deep Maize is that it explicitly forecasts changes in future market conditions. Figure 2 compares the five predictors on an aggregate measure of forecasting error out to a horizon of 50 days. We compute the RMS error between the predicted and observed effective demand curves, which is essentially a summation of the pairwise differences between two sorted price lists. Demand is grouped in 5-day buckets to smooth noise in customer demand levels. We compute error over the top 200 values, which is approximately twice the number of PCs the agent would typically sell.

The pattern of results is similar to the results for the current day. The two relative predictors score the best over all horizons, followed by the baseline and then the two absolute predictors. The online affine transformations are beneficial regardless of the representation used. We note that this measure averages error over the distribution (each data point represents a compression of the full distribution shown in Figure 1). We present information in this way partially for easy visualization, but also because future planning may be more dependent on errors over the entire distribution than bidding decisions. However, we should note that the absolute predictor shows higher overall error on this measure primarily because of errors at the extreme parts of the distribution, as in the short-term error results.

6.2 Simulation Performance

Since our forecasting methods were specifically designed to support decision-making in a fully implemented and automated agent, we have the opportunity to assess how forecasts impact agent performance. This is particularly interesting because of the complex interactions between prediction error and decisions. As discussed earlier, it can be difficult to determine how to weight different types prediction errors. Another issue with online predictions is that the decisions made affect the information available for making future predictions.

We ran simulations using a profile of two MinneTAC agents, two TacTex-05 agents, one static version of Deep Maize, and one modified version. MinneTAC and TacTex-05 were finalists in 2005 that were among the first to release agent binaries. We ran four sets of games with a minimum of 35 samples. The static version of Deep Maize used the relative affine predictor. We present results as differences between the static and variable versions of Deep Maize. This pairing reduces variance but introduces bias to the extend that varying the sixth agent affects the performance of the static version. Another important caveat is that these results are for a single profile of strategies. Alternative approaches to selecting test environments are considered in [24], but we make do with a single convenient but ad-hoc profile here.

In addition to scores we give three measures of customer sales performance: average selling price (ASP), "selling efficiency," and "timing efficiency." Selling efficiency is the fraction of achieved revenue to the maximum possible revenue for identical daily sales, given perfect information about opponents' bids and the option to partially fill orders. Timing efficiency measures how well the agent distributed sales over time. Let the t-optimal same day selling price be the highest ASP the agent could achieve by selling each PC at most t days after it was actually delivered and no earlier than it was actually produced. PCs are labeled according to a FIFO queuing policy. Timing efficiency is the ratio of the t-optimal same day selling price to the selling efficiency. This factors out the effects of bidding policy and leaves the effects of deciding which days to sell on. The appeal of these additional metrics is that they allow us to compare a specific element of decision performance against an optimal value, though we must remain cognizant of the additional constraints when interpreting the results.

The simulation results are in Table 1. The most striking thing is that all of our kNN-based predictors comfortably outperform the same agent using the baseline predictor. To calibrate, the difference between the top and bottom scores in the 2005 finals was approximately 6.5 M, less than the advantage of any of our predictors over the baseline. Clearly, effective prediction can have a sizable impact on agent performance when the decision-making architecture is capable of exploiting the information. The affine transformations again showed benefits for both representations.

The absolute predictors scored better than the relative predictors, albeit by fairly small margins. This is somewhat unexpected,

Predictor	Score	ASP	Selling Efficiency	20-day Timing
Relative, No Affine	-2.3M	0.004	-0.007	-0.015
Absolute, Affine	0.3M	0.001	-0.008	0.000
Absolute, No Affine	-1.2M	0.007	-0.014	-0.009
Baseline	-9.1M	-0.032	0.002	-0.032

Table 1: Performance measures in simulated games. The numbers represent the difference between two versions of Deep Maize using the relative affine predictor and the indicated predictor.

given the error results. Earlier results using a smaller data set for the predictors actually showed a slight advantage for the relative predictor. We expect a larger data set to benefit the absolute representation disproportionately because some of the features are based on current prices. With more data more similar instances are available, and the absolute representation should predict more like the relative representation. We conjecture that the relative representation may have greater advantages for smaller data sets, but have not verified this experimentally. Another likely explanation for this discrepancy is that the absolute representation better reflects the actual uncertainty present because it is able to spread predictions over a broader range of prices. This may cause the agent to delay making purchasing decisions until more information is available, potentially improving performance. We intend to explore these possibilities more carefully in future analysis.

In general, the differences between the agent variations on the sales metrics are relatively small, but we note some intersting features. The timing measures correlates quite well with the overall scores. The baseline predictor performs very poorly on this measure as well as having a low ASP. The selling efficiency numbers do not show a strong pattern, and the ASP numbers for the kNN variations are also ambiguous. This suggest that one of the more consistent benefits of better forecasts for Deep Maize is the usefulness of these forecasts in timing sales activity.

6.3 Tournament and Self-Play Data

We also generated error scores for the four predictor variants using only the tournament data set and only the self-play data set (a total of 12 different variations). We do not present full data here, but we can describe the pattern of results fairly easily. Using only the tournament data resulted in almost exactly equivalent performance to using the combined data sets. The results using only the self-play data were almost always worse than either the tournament only or combined settings. We draw two conclusions from this. The first is that the process we used to generate the self-play data did not result in a very useful configuration for making predictions. The second is that the online scoring procedure was effective at detecting this and preventing these bad predictions from affecting the overall performance of the combined predictor.

6.4 Tournament Performance

Tournament performance is an important indicator of the overall competence of TAC SCM agents. Each TAC SCM tournament comprises a sequence of rounds. Qualifying and seeding rounds are held early on, each lasting for two weeks. These rounds are used by agent developers for testing and debugging, though some agents may be eliminated for consistent inactivity or poor performance. The main 2005 tournament was held in conjunction with IJCAI-05, and the 2006 tournament was held in conjunction with AAMAS-06. Three rounds are played on successive days. Out of approximately 30 agents that started in the qualifying round each year, approximately 24 were selected to play in the quarter-final round on the first day. Agents are assigned to heats of 6, from which the bottom three are eliminated each day. The semi-final round on the second day retains the 12 top-scoring agents (3 from each quarter-final heat). The six top-scoring agents from the semi-finals heats proceed to the final round. The quarter-final heats play 8 games each, while the heats in the final two round play 16 games.

Deep Maize has performed very well in each of the four TAC SCM tournaments to date. Results from the 2005 and 2006 tournaments are shown in Tables 2 and 3. Deep Maize made it to the final round each year, placing fourth and third overall. Unofficially, if we disregard the games with network problems, Deep Maize placed second in 2005. Deep Maize also demonstrated strong performance in the quarter- and semi-final rounds, placing first in each heat. This record of overall performance provides evidence that Deep Maize is effective in a wide variety of competitive environments. The version of the customer prediction module presented here is very similar to the modules that played in both the 2005 and 2006 tournaments. This aspect of the agent was largely unmodified between 2005 and 2006, though other aspect of the agent underwent significant revisions.

7. CONCLUSIONS

We document and analyze a successful approach for forecasting prices in a challenging market domain. The general principle of our design is to support exploitation of all possible sources of information. To this end, we integrate several well-known ideas in an novel way, resulting in better overall predictions.

We evaluate our ideas against a baseline predictor similar to those used by many competing agents, applying both error measures and simulation performance as benchmarks. We show that improved predictions can dramatically improve agent performance in TAC SCM. This is particularly interesting in light of the substantial uncertainties inherent in the domain and challenges of predicting the behavior of other agents.

The technique we present for applying transformations to predictions using online scoring is particularly interesting, and should be applicable in many domains. This method provides a way to perform principled combinations of predictions based on different sources of data, as well as a way to reduce systematic errors. It does these by making use of additional information derived from online performance. This method showed consistent benefits in our experiments across a range of conditions on both prediction error and simulation performance metrics.

We also investigated the use of absolute and relative representations for predictions. It appears that each of these representations has advantages under certain conditions, particularly related to how the information is used in decision-making.

Acknowledgments

We would like to thank the participants and organizers of the TAC SCM competition for providing an interesting opportunity for experimentation in a multi-agent market environment. We would also like to thank Kevin O'Malley for his contributions to the Deep

Table 2: Scores and rankings from the rounds Deep Maize played in during the 2005 SCM tournament. All scores are in millions of dollars. The adjusted column eliminates two game instances (3718 and 4259) where the University of Michigan network lost connectivity during the game, distorting the results substantially.

Quarter-Finals D		Semi-Finals 2		Finals		Finals (adjusted)	
Deep Maize	17.49	Deep Maize	3.68	TacTex-05	4.74	TacTex-05	5.18
CMieux	15.03	TacTex-05	3.57	SouthamptonSCM	1.60	Deep Maize	2.06
RationalSCM	14.61	MinneTAC	2.27	Mertacor	0.55	SouthamptonSCM	1.55
CrocodileAgent	11.64	RationalSCM	-2.28	Deep Maize	-0.22	Mertacor	0.53
Cylinder	5.79	CMieux	-2.33	MinneTAC	-0.31	MinneTAC	0.33
optimiSCM	0	PhantAgent	-6.64	Maxon	-1.99	Maxon	-1.74

Table 3: Scores and rankings from the rounds Deep Maize played in during the 2006 SCM tournament. All scores are in millions of dollars.

Quarter-Finals A		Semi-Finals 2		Finals	
Deep Maize	9.61	Deep Maize	6.45	TacTex-06	5.85
Botticelli	0.83	Maxon	4.08	PhantAgent	4.15
SouthamptonSCM	0.01	Botticelli	1.95	Deep Maize	3.58
Foreseer	-1.42	CMieux	1.81	Maxon	1.75
kshitij	-3.45	SouthamptonSCM	1.63	Botticelli	0.48
Filler	-5.51	Mertacor	1.44	MinneTAC	-2.70

Maize infrastructure. This work was supported in part by NSF grant IIS-0205435, and the STIET program under NSF IGERT grant 0114368.

8. REFERENCES

[1] I. Alon, M. Qi, and R. J. Sandowski. Forecasting aggregate retail sales: A comparison of artificial neural networks and traditional methods. *Retailing and Consumer Services*, 8:147–156, 2001.

[2] J. M. Bates and C. W. J. Granger. The combination of forecasts. *Operations Research*, 20(4):451–468, 1969.

[3] M. Benisch, A. Greenwald, V. Naroditskiy, and M. Tschantz. A stochastic programming approach to scheduling in TAC SCM. In *Fifth ACM Conference on Electronic Commerce*, pages 152–159, New York, 2004.

[4] M. Benisch, A. Sardinha, J. Andrews, and N. Sadeh. CMieux: adaptive strategies for competitive supply chain trading. In *Eighth International Conference on Electronic Commerce*, 2006.

[5] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*. Springer-Verlag, 1996.

[6] D. A. Burke, K. N. Brown, B. Hnich, and A. Tarim. Learning market prices for a real-time supply chain management trading agent. In *AAMAS-06 Workshop on Trading Agent Design and Analysis*, 2006.

[7] C. Chatfield. *Time-Series Forecasting*. Chapman & Hall/CRC, 2001.

[8] S.-F. Cheng, E. Leung, K. M. Lochner, K. O'Malley, D. M. Reeves, and M. P. Wellman. Walverine: A Walrasian trading agent. *Decision Support Systems*, 39:169–184, 2005.

[9] R. T. Clemen and R. L. Winkler. Combining probability distributions from experts in risk analysis. *Risk Analysis*, 19(2):187–203, 1999.

[10] J. Collins, R. Arunachalam, N. Sadeh, J. Eriksson, N. Finne, and S. Janson. The supply chain management game for the 2005 trading agent competition. Technical Report CMU-ISRI-04-139, Carnegie Mellon University, 2004.

[11] J. Eriksson, N. Finne, and S. Janson. Evolution of a supply chain management game for the Trading Agent Competition. *AI Communications*, 9:1–12, 2006.

[12] R. Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1):107–119, 2003.

[13] T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning*. Springer-Verlag, 2001.

[14] M. He, A. Rogers, X. Luo, and N. R. Jennings. Designing a successful trading agent for supply chain management. In *Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Hakodate, 2006.

[15] W. Ketter, J. Collins, M. Gini, A. Gupta, and P. Schrater. Identifying and forecasting economic regimes in TAC SCM. In *IJCAI-05 workshop on Trading Agent Design and Analysis*, 2005.

[16] C. Kiekintveld, J. Miller, P. R. Jordan, and M. P. Wellman. Controlling a supply chain agent using value-based decomposition. In *Seventh ACM Conference on Electronic Commerce*, pages 208–217, 2006.

[17] C. Kiekintveld, Y. Vorobeychik, and M. P. Wellman. An analysis of the 2004 supply chain management trading agent competition. In *IJCAI 2005 Workshop on Trading Agent Design and Analysis*, 2005.

[18] C. Kiekintveld, M. P. Wellman, S. Singh, J. Estelle, Y. Vorobeychik, V. Soni, and M. Rudary. Distributed feedback control for decision making on supply chains. In *Fourteenth International Conference on Automated Planning and Scheduling*, pages 384–392, 2004.

[19] F. J. Nogales, J. Contreras, A. J. Conejo, and R. Espinola. Forecasting next-day electricity prices by time series models. *IEEE Transactions on Power Systems*, 17(2):342–348, 2002.

[20] D. Pardoe and P. Stone. Bidding for customer orders in TAC SCM. In *Agent Mediated Electronic Commerce VI: Theories for and Engineering of Distributed Mechanisms and Systems (AMEC 2004)*, 2005.

[21] D. Pardoe and P. Stone. An autonomous agent for supply chain mangement. Technical report, University of Texas, 2006.

[22] D. Pardoe and P. Stone. Predictive planning for supply chain

management. In *Proceedings of the International Conference on Automated Planning and Scheduling*, June 2006.

[23] P. Stone, R. E. Schapire, M. L. Littman, J. A. Csirik, and D. McAllester. Decision-theoretic bidding based on learned density models in simultaneous, interacting auctions. *Journal of Artificial Intelligence Research*, 19:209–242, 2003.

[24] M. P. Wellman, P. R. Jordan, C. Kiekintveld, J. Miller, and D. M. Reeves. Empirical game-theoretic analysis of the TAC market games. In *AAMAS-06 Workshop on Game-Theoretic and Decision-Theoretic Agents*, 2006.

[25] M. P. Wellman, D. M. Reeves, K. M. Lochner, and Y. Vorobeychik. Price prediction in a trading agent competition. *Journal of Artificial Intelligence Research*, 21:19–36, 2004.