

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/221155639>

Data Mining–Driven Analysis and Decomposition in Agent Supply Chain Management Networks

Conference Paper · December 2008

DOI: 10.1109/WIIAT.2008.395 · Source: DBLP

CITATIONS

0

READS

45

3 authors:



[Kyriakos Chatzidimitriou](#)

Aristotle University of Thessaloniki

25 PUBLICATIONS 118 CITATIONS

[SEE PROFILE](#)



[Andreas Symeonidis](#)

Aristotle University of Thessaloniki

95 PUBLICATIONS 413 CITATIONS

[SEE PROFILE](#)



[Pericles A. Mitkas](#)

Aristotle University of Thessaloniki

272 PUBLICATIONS 1,635 CITATIONS

[SEE PROFILE](#)

All content following this page was uploaded by [Andreas Symeonidis](#) on 23 January 2017.

The user has requested enhancement of the downloaded file. All in-text references [underlined in blue](#) are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

Data Mining-Driven Analysis and Decomposition in Agent Supply Chain Management Networks

Kyriakos C. Chatzidimitriou¹, Andreas L. Symeonidis^{1,2} and Pericles A. Mitkas^{1,2}

¹Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki, GR541 24, Thessaloniki, Greece

²Intelligent Systems and Software Engineering Laboratory
Informatics and Telematics Institute/CERTH, 57001, Thessaloniki, Greece
{kyrcha,asymeon}@issel.ee.auth.gr, mitkas@auth.gr

Abstract

In complex and dynamic environments where interdependencies cannot monotonously determine causality, data mining techniques may be employed in order to analyze the problem, extract key features and identify pivotal factors. Typical cases of such complexity and dynamicity are supply chain networks, where a number of involved stakeholders struggle towards their own benefit. These stakeholders may be agents with varying degrees of autonomy and intelligence, in a constant effort to establish beneficiary contracts and maximize own revenue. In this paper, we illustrate the benefits of data mining analysis on a well-established agent supply chain management network. We apply data mining techniques, both at a macro and micro level, analyze the results and discuss them in the context of agent performance improvement.

1. Introduction

As agent technology matures in time, autonomous agents gain applicability and trust in trading and auctioning goods in real-world electronic markets, as well as in managing more complex environments like supply chain networks [4, 8]. Various approaches are followed in order to determine the optimal agent strategy with respect to the challenges they (agents) come up against. The plethora of data generated by these highly dynamic markets can be exploited in various contexts, either for online, or for a posteriori analysis. Our work attempts to evaluate data mining (DM) methodologies for analyzing and improving agent behavior based on market data analysis for an agent supply chain testbed, the Trading Agent Competition, Supply Chain Management game (TAC SCM) [1]. TAC SCM is a dynamic, stochastic, partially observable and competitive

multi-agent environment that simulates an instantiation of a realistic supply chain management scenario. It allows for a huge strategy space for agents to apply, while it is very challenging in the sense that decisions taken on any given day may affect the state of the agent many days later. Due to the time constraints and the huge dimensionality of the game, analytic strategies and algorithms are much more difficult to apply and succeed than in simpler auction settings. All the above make the TAC SCM game a very good testbed for applying DM techniques and establishing their worth as intelligent agent modules against various conditions and opponents.

In general, DM analysis of a domain may identify opportunities and help the agent designer gain a predictive edge over his/her opponents. In this context, we have applied DM in order to: a) provide a DM-driven performance evaluation of the agents over the past competitions, derive conclusions as to what makes an agent successful, and subsequently guide the design of SCM agents and, b) deduce models of market behavior in order to build a robust and high-performing agent that will be able to be ahead of its competition. The resulting agent was used as a strawman against other methodologies that involve on-line learning and/or heuristic approaches. In this work, we focus our DM methodology on the bidding component of the agent, which is the one that comes in direct competition with its counterparts.

The rest of the paper is organized as follows: Section 2 provides an overview of the TAC SCM and references related work on the domain. Section 3 describes performance evaluation models, while Section 4 presents the bidding algorithm. Finally, Section 5 summarizes work conducted.

2. Background & Related Work

Within the scenario of the TAC SCM game, each agent participating represents a PC manufacturer with limited production capacity. Six such agents compete in selling 16 different types of PCs to potential customers. The agents' tasks are to negotiate on supply contracts, bid for customer offers, manage daily assembly activities and ship completed orders to customers. Every day, customers send Request-For-Quote (RFQ) messages and agents bid on them, depending on their ability to satisfy delivery dates and prices. The bid price should not exceed the reserve price the customer requires, which is between 75% – 125% of base price of PC components. The next day, the customer sends the order to the agent that has made the winning offer. To get paid, the agent must deliver the products on time, otherwise it is charged with a penalty. Winner is declared the agent with the greater bank balance at the end of the game. Game length is 220 simulated days, with each day lasting 15 seconds.

Several approaches have been employed for analyzing the behavior of agents in the competition and providing bidding strategies based on supervised learning algorithms from past game log analysis. As far as bidding strategies and DM methods are concerned, agent TacTex [6] has employed models to predict the probability of acceptance of a bid based on RFQ details, current game conditions and offered price, while a more recent design of the same agent [7] has attempted to forecast future changes in price, in order to predict more profitable days of selling. On the other hand, DeepMaize [5], another highly performing agent, uses a k-nearest neighbors algorithm to find the relationship between current market conditions and distributions of current and future prices. From what the TAC SCM literature has to offer, our analysis differs in the fact that, through DM we ultimately result in a simple generalized rule to guide SCM agent design. Additionally, our DM-enabled bidding algorithm builds upon the analysis provided and tries to combine the price, probability, opportunities and future conditions predictions in a unified bidding schema.

3. DM-Enabled Performance Evaluation

DM analysis was performed on game logs from 2005 to 2007 (3 years). Previous years' (2003 and 2004) logs were not considered, since in 2005 there was a major change in game rules. Our goal was to identify the qualities that a high (or low) performing agent should (should not) exhibit during a game. Agent performance is rated against its bank balance at the end of a game, with respect to 21 metrics that are calculated as percentages or averages during game time. These metrics are: 1) market share (%), 2-4) three due date shares (%), 5-7) three reserve price shares (%), 8-10)

three range shares (%), 11-13) three quantity shares (%), 14) delivery performance (%), 15) average (AVG) selling price, 16) AVG purchase price, 17) AVG lead time of supplies, 18) AVG factory utilization, 19) AVG storage costs, 20) AVG PC inventory level and 21) AVG component inventory level.

Two filtering approaches were applied: a) a correlation-based feature subset evaluation algorithm using hill climbing, forward, backward and genetic search and b) a ReliefF feature ranking, with a wrapper approach for attribute selection with linear regression and forward search, all using 10 cross-fold validation (10-CV), separately for each one of the three years. This way, the best and most frequent appearing features were extracted: AVG selling price (positive), AVG purchase price (negative), AVG storage costs (negative), delivery performance (positive) and AVG factory utilization (positive). The correlation-based feature subset evaluation algorithm, regardless of the search technique used would always output these features. Next to each selected feature, one may find the positive or negative sign of the coefficient when linear regression is performed. The results for root mean square error (RMSE) in millions (M) and the correlation coefficient (CC) are presented in Table 1. On the left side of the arrow is the training set and on the right the test set used for the analysis.

Table 1. Predicting the performance of an agent over a game.

	Full set of attributes			
	2005 → 2006		2006 → 2007	
Model	CC	RMSE	CC	RMSE
Linear	0.86	7.050M	0.81	7.999M
M5'	0.89	6.336M	0.92	5.292M
	Best subset of attributes			
	2005 → 2006		2006 → 2007	
Model	CC	RMSE	CC	RMSE
Linear	0.87	6.872M	0.807	8.210M
M5'	0.905	5.878M	0.91	5.568M

The results from Table 1 indicate that more than 80% of bank balance variance can be explained by monitoring only these 5 features. The following rule can be defined:

“An agent should sell as high as possible and buy as low as possible, while maintaining the highest possible throughput in the factory and inventory (and by throughput we mean to have a high factory utilization, but also sell the produced PCs and not just store them) and not default on deliveries”

4. A DM-Enabled Bidding Mechanism

We have focused on the bidding task of the agent, since it is the one that interacts directly with the other competitors and, thus, requires precise modeling of market conditions for every period during the game. Algorithm 1 summarizes the major points of the mechanism. The auctions are first-price, sealed bid auctions, with the customer setting up a reserve price. The winning agent is the one that has the lowest bid below that reserve price.

Algorithm 1 The DM enabled bidding algorithm.

```

RFQs  $\leftarrow$  current day RFQs
current  $\leftarrow$  current game state
for  $i = 1$  to  $i = d$  do
    demand  $\leftarrow$  predict demand in RFQs for day  $i$ 
    RFQs.add(generateRFQs(demand))
end for
for all RFQs do
    state  $\leftarrow$  dm.predictState(current,rfq.creationDate)
    if dm.isUnfruitful(rfq,state) then
        rfq.bidPrice = rfq.reservePrice
    else
        rfq.bidPrice = dm.predictPrice(rfq,state)
    end if
    rfq.prob = dm.calculateProb(rfq.bidPrice, rfq, state)
    rfq.utility = rfq.prob  $\times$  rfq.bidPrice
end for
sort in descending utility order
while resources > 0 do
    agent.makeOffer(rfq)
    resources -= rfq.prob  $\times$  rfq.resources
end while

```

Every day the agent receives RFQs from the customers, while future RFQs are also predicted. The importance of predicting future RFQs is that one should not spend all its factory cycles on today’s RFQs, since it is likely that a percentage of forthcoming RFQs will be more profitable. After having calculated its free capacity with respect to pending orders in order to choose the most profitable current and future RFQs to bid on, the agent attempts to predict whether each auction established will be fruitful or not. In the latter case, there will be no competition and thus a bid equal to the reserve price can be placed. In the former case, the RFQ details and the state of the game the given date are fed to a DM model responsible for predicting the closing price of the auction. Utility is calculated for all offers as they are sorted in descending utility order. For each offer, if resources are adequate the agent makes a bid. Resources (factory cycles and components) are discounted based on the probability of acceptance of any given offer.

Table 2. Offer price prediction performance: Mean Absolute Error.

Product ID	Linear	M5'
1	63	54
8	118	86
9	66	55
16	113	83

4.1. Predicting Offer Prices

One of the most important tasks an agent has to uptake during the competition is to be able to predict the closing price of an RFQ. Our work presented in [2] introduces a way to predict the offer price given 20 attributes: RFQ details (reserve price, due date, quantity, penalty) and the current state of the game (current simulation date, demand, minimum and maximum prices the five previous days). Numerous experiments have been conducted like performing multiple linear regression with backward elimination using the F-statistic, performing correlation feature subset evaluation [3] and ReliefF ranking evaluation, in order to identify good features that are able to generalize well. Based on this analysis, we have identify a subset of important features: due date (negative), reserve price (positive), minimum and maximum prices for the last 2 days (positive) and current demand (positive). Tags in parentheses represent the sign of the coefficient when linear regression is applied. From all algorithms and methods tested, like Classification Algorithms, Neural Networks and Support Vector Machines, M5' was selected due to better prediction accuracy from other methods. Table 2 the RMSE for product IDs 1, 8, 9 and 16. The results are 10-CV on 10% of the entire 2007 dataset, equaling around 170000 instances. Again we can see that above 80% of the variance of the offer price can be explained by these 7 variables.

4.2. Predicting Unfruitful Auctions

Several auctions during a TAC SCM game are not met by the manufacturers. This could be either because demand is higher than supply, or due to start- and end- effects, where the agents try to build or minimize inventories. Another potential factor could be the shortage of a certain component, resulting to the inability of manufacturers to produce a certain product. In any case, identifying such opportunities could be beneficial for the agent, since it would set the price of that offer equal to the reserve price of the RFQ in order to gain more profit and increase revenues. Again, one could deal with the problem of predicting unfruitful auctions us-

ing DM. By employing the same initial set of features as in the previous section, we have performed classification, trying to identify whether an auction will be met (1) or not (0).

Initially we employed attribute selection methodologies, in order to find the most prominent features. Experiments were performed using the datasets from the finals (more competitive) and second finals rounds (less competitive) in 2005 and three datasets were generated: tiny (6245 and 5806 instances), medium (124223 and 118831 instances) and huge (621577 and 592084 instances, respectively). Using correlation-based feature subset evaluation and Relief ranking on the tiny datasets we created several potential candidate subsets of features (14 of them). By applying several algorithms in order to measure accuracy (J48, REPTrees, DecTables, NaiveBayes, RBFNets, IBk, JRip, BayesNet and PART) on the tiny datasets, we decided on the ones that performed best in accuracy for a 10-CV evaluation: current simulation date, reserve price, minimum price of product the previous day, maximum price of product the previous day, current demand and previous day demand. Accuracy may not be the best evaluation measure, nevertheless it gives a qualitative overview of the problem at hand. More elaborate testing was performed using the features selected, in order to evaluate whether they are related to start and end effects (current date), demand (customer demand the last two days) and whether or not there is strong competition (minimum and maximum prices distance from the reserve price) between agents.

Subsequently, we focused on the optimal configuration of the selected algorithm (C4.5), in order to improve *Recall* and *Precision*, since in the dataset most of the auctions were met, thus the extracted model was biased towards met auctions. We re-adjusted the cost matrix for the tiny dataset and then tested it on the medium and huge datasets. The C4.5 was applied, pruned, with confidence 0.0005 and a cost matrix of (0,2,3,0) for (TN, FP, FN, TP) respectively. Higher cost was assigned to false negatives (FN), since by predicting an auction is unfruitful and by setting the offer price as the reserve price, it is very likely for the offer not to be accepted due to the rest of competition. On the other hand, the cost of predicting a auction will be met by the manufacturers when in fact it is not, will make the agent lose some profit but not the entire order. The results for the two datasets for the finals are provided in Table 3.

5. Conclusions & Future Work

Within the context of this work we have provided a data mining analysis of the TAC SCM game. We were able to identify the key features of success for the agents and subsequently we have built a bidding mechanism that employs this knowledge, in order to enhance agent performance. Fu-

Table 3. Predicting unfruitful auctions in the finals of TAC SCM 2005

Measure	Medium		Huge	
	Finals	2nd Fin.	Finals	2nd Fin.
Accuracy	97.2	88.1	96.9	89.5
Precision	0.89	0.78	0.93	0.83
Recall	0.26	0.48	0.59	0.53
F-measure	0.4	0.59	0.72	0.65
MAE	0.04	0.16	0.02	0.15
ROC	0.89	0.86	0.94	0.88

ture work is focused on the analysis of the supplier side, in order to improve agent procurement and agent reputation.

Acknowledgements

We would like to thank Ioannis Kontogounis (BSc), Jun Hu (MSc), and the graduate students of the 2007-2008 class of “Databases and Knowledge Extraction” course, for conducting some of the experiments, whose results are contained in this work.

References

- [1] R. Arunachalam and N. M. Sadeh. The Supply Chain Trading Agent Competition. *Electronic Commerce Research and Applications*, 4:6684, 2005.
- [2] K. C. Chatzidimitriou, A. L. Symeonidis, I. Kontogounis, and P. A. Mitkas. Agent Mertacor: A Robust Design for Dealing with Uncertainty and Variation in SCM Environments. *Expert Systems with Applications*, Article in Press, 2007.
- [3] M. A. Hall. *Correlation-based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, 1999.
- [4] M. He, N. R. Jennings, and H.-F. Leung. On Agent-Mediated Electronic Commerce. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):985–1003, Jul/Aug 2003.
- [5] C. Kiekintveld, J. Miller, P. R. Jordan, and M. P. Wellman. Forecasting Market Prices in a Supply Chain Game. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1318–1325, Honolulu, Hawaii, 2007.
- [6] D. Pardoe and P. Stone. Bidding for Customer Orders in TAC SCM: A Learning Approach. In *AAMAS-04 Workshop on Trading Agents Design and Analysis*, 2004.
- [7] D. Pardoe and P. Stone. An Autonomous Agent for Supply Chain Management. In G. Adomavicius and A. Gupta, editors, *Handbooks in Information Systems Series: Business Computing*. Elsevier, 2007.
- [8] P. Stone. Learning and Multiagent Reasoning for Autonomous Agents. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 13–30, January 2007. IJCAI-07 Computers and Thought Paper.