UNIVERSITY OF MINNESOTA

This is to certify that I have examined this bound copy of a doctoral thesis by

Wolfgang Ketter

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Maria Gini

Name of Faculty Advisor

Signature of Faculty Advisor

Date

GRADUATE SCHOOL

Identification and Prediction of Economic Regimes to Guide Decision Making in Multi-Agent Marketplaces

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Wolfgang Ketter

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Maria Gini, Advisor

January 2007

Dedication

To my parents

Acknowledgments

There are many people who I am deeply indebted to, and without whom this work would never have been finished. First and foremost I would like to thank my advisor Maria Gini. It was a real apprenticeship, whenever I wanted to rush away with my grandiose visions she channeled my energy to what matters most, and guided me to think very deep about our current research problems. She has the remarkable ability to see and articulate the core problem in a maze of confusion. She also has taught me to think about the big picture and present my work accordingly. Maria was always willing to take time out of her busy schedule when important or usual issues came up.

I like to send my gratitude to the other members of my committee John Collins, Alok Gupta, and Paul Schrater. Few people are fortunate enough to have such a fantastic committee. John Collins, who has helped me and the whole MinneTAC team with his expert supervision, great design skills, hands-on analysis, and so much more was indispensable. Especially in this last semester he devoted a lot of his time to discuss urgent research problems. John repeatedly reminded me that the most important thing about a dissertation was having finished it, and while teaching with him I learned how to design good class projects and exams.

Alok Gupta was my window into the business and economics community and helped me in more ways than he possibly can imagine. He gave me multiple times a forum for testing some of my ideas with his students in his courses at the Carlson School of Management, as well as at the INFORMS 2005 conference in San Francisco. He taught me the principle of analytical eyeballing, meaning to test out an idea quickly before embarking totally with possibly no output. Another very useful principal I learned from him is the theoretical lens. Everybody has a theoretical background that shapes his or her view on a problem. Develop alternative views on the same problem and your insight will be magnified.

Paul Schrater deepened my knowledge of the beautiful subject of machine learning. Over the course of many research meetings we had great discussions and he taught me how to take a loose idea or result and push it to its limits by discovering variations and generalizations, implications and applications.

I'm grateful to all former and current members of the MAGNET and MinneTAC research team, especially: Alexander Babanov, John Collins, Steve Damer, Maria Gini, Roman Kulik, Elena Kryzhnyaya, and Eric Sodomka. I really enjoyed working with Alex the first two years on the evolutionary framework, feel honored that he came to my thesis defense all the way from Seattle. I'm looking forward to attend his Ph.D. defense whenever it will be.

I also would like to thank the members of the machine learning group for stimulating discussions, and the opportunity to present my work to this critical audience. Thanks to Bonnie Bennett who was so nice to offer me to give a thesis presentation at Adventium Labs two days before my thesis defense.

I would like to thank the organizers of the TAC SCM game for a stimulating research problem and the whole TAC SCM community for great presentations and discussions at the workshops for trading agent design and analysis. I also would like to acknowledge the help from TAC SCM researchers who made this work possible by making their agents available for others to use. Thanks to the American Association for Artificial Intelligence for selecting me to present my work at the 2005 Doctoral Consortium in Pittsburgh. It was great to present my work and to get feedback from senior researchers and other doctoral students in the field.

Thanks to the many in the wider research community who have taken the time to discuss my work and explain their work, including, Gedas Adomavicius, Alexander Babanov, Ravi Bapna, Arun Batchu, Gary Berosik, Anindya Ghose, Otto Koppius, Steve Miller, David Pardoe, Michael Smith, Katia Sycara, Eric van Heck, Harini Veeraraghavan. I would also like to extend my gratitude to the many anonymous reviewers of our papers. Their knowledge and scientific rigor were crucial in building and shaping this work.

The majority of our work is done using open-source software, including gnu/Linux, LaTeX, ant, log4j, and other tools from the Apache project, emacs, and many others. Thanks to the talented folks who make all this possible. A big thanks to our systems staff who were always there when their help was most needed.

Fortunately, my years in St. Paul and Minneapolis were not all work and no play. I especially want to thank Ulyana and Alexander Babanov, Rima Parikh and Arun Batchu, Heda and Matt Batalden, Barb and Gary Berosik, Saladin Cerimagic, Nicole Grewling, Ascan and Reka Körner, Anna and Helen Kryzhnyaya, Vladimir Kryzhniy, Thomas Jabusch, Ashutosh Jaiswal, Easwar Lakshminarayanan, Dennis Ley, Lidia Ley, Mel McManus, Jan Makowski, Tim Meier, Cheryl Quellette, Henning Sjurseth, Michele Tertilt, Amy Thoren, Nikolas Trawny, Harini Veeraraghavan, Armando Vilchez, Janel and Jon Wittmayer, and Arno Wünschmann, for being such good friends here for many years. I will also miss everybody from the German Stammtisch, whose organizer I was for seven years, but I am comforted to know that new talented folks have taken over.

I would not be who I am today without my parents, Werner and Irene. They have been supportive at every step, always encouraging me to pursue what I am capable of. They provided me a lifeline to Germany with all the wonderful care packages they sent to me.

Finally, and most importantly, I would like to thank my wonderful girlfriend Elena. Without her support, encouragement, confidence in me from the day I started this research until now I would not have finished this dissertation.

Imagination is more important than knowledge.
Albert Einstein

If not now? - When?
Zen Philosophy

Do not regret your choice in the valley of Emyn Muil, nor call it a vain pursuit. You chose amid doubts the path that seemed right: the choice was just, and it has been rewarded.
Gandalf to Aragorn
The Two Towers by J.R.R. Tolkien

Keywords: Agent-Mediated Electronic Marketplaces, Artificial Intelligence, Auctions and Exchanges, Dynamic Pricing, Information and Decision Theory, Intelligent Agents, Electronic Commerce, Machine Learning, Market Forecasting, Multi-Agent Systems, Rational Decision Making, Software Architecture, Software Engineering, Software Patterns, and Supply-Chain Management.

Abstract

Supply chain management is commonly employed by businesses to improve organizational processes by optimizing the transfer of goods, information, and services between buyers and suppliers. Traditionally, supply chains have been created and maintained through the interactions of human representatives of the various companies involved. However, the recent advent of autonomous software agents opens new possibilities for automating and coordinating the decision making processes between the various parties involved.

Autonomous agents participating in supply chain management must typically make their decisions in environments of high complexity, high variability, and high uncertainty since only limited information is visible.

We present an approach whereby an autonomous agent is able to make tactical decisions, such as product pricing, as well as strategic decisions, such as product mix and production planning, in order to maximize its profit despite the uncertainties in the market. The agent predicts future market conditions and adapts its decisions on procurement, production, and sales accordingly.

Using a combination of machine learning and optimization techniques, the agent first characterizes the microeconomic conditions, such as over-supply or scarcity, of the market. These conditions are distinguishable statistical patterns that we call *economic regimes*. They are learned from historical data by using a Gaussian Mixture Model to model the price density of the different products and by clustering price distributions that recur across days.

In real-time the agent identifies the current dominant market condition and forecasts market changes over a planning horizon. Methods for the identification of regimes are explored in detail, and three different algorithms are presented. One is based on exponential smoothing, the second on a Markov prediction process, and the third on a Markov correction-prediction process. We examine a wide range of tuning options for these algorithms, and show how they can be used to predict prices, price trends, and the probability of receiving a customer order.

We validate our methods by presenting experimental results from the Trading Agent Competition for Supply Chain Management, an international competition of software

agents that has provided inspiration for this work. We also show how the same approach can be applied to the stock market.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Business organizations have an increasing need for software that can assist decision makers by gathering and analyzing information, making recommendations, and supporting business decisions. Advanced decision support systems and autonomous software agents promise to address this need by acting rationally on behalf of humans in numerous application domains. Examples include procurement [Sandholm, 2006, CombineNet, 2006], scheduling and resource management [I2, 2006, Collins *et al.*, 2001], and personal information management [Berry *et al.*, 2006, Mark and Perrault, 2006].

1.1 Objective

In this thesis, we show how machine learning techniques can be used to support rational decision making by an autonomous agent that operates in a market for durable goods to sell products. We are particularly interested in environments that are constrained by capacity and materials availability. We demonstrate our approach in the context of an autonomous agent that is designed to compete in the Trading Agent Competition for Supply Chain Management (TAC SCM) [Collins *et al.*, 2005].

Our method characterizes market conditions by distinguishable statistical patterns, that we call *economic regimes*. We show how such patterns can be learned from historical data and subsequently identified in real time from observable data. We describe how to identify regimes and to forecast regime transitions. This prediction, in turn, can be used by the agent to allocate resources to current and future sales in a way that maximizes resource value. While this type of prediction about the economic environment is commonly used at the macro economic level [Osborn and Sensier, 2002], such predictions are rarely done for micro-economic environments.

1.2 Agents for Electronic Marketplaces

What is an agent? The term "Agent" has been used in a rather vague way in industry and academic literature. According to Webster's Third New International Dictionary, an agent is "one that acts or exerts power... a means or instrument by which a guiding intelligence achieves a result... one that acts for or in the place of another by authority from him." Russell and Norvig [Russell and Norvig, 2002] say that "An agent is just something that acts (agent comes from the Latin *agere*, to do)". Bradshaw [Bradshaw, 1997] reviews in detail the various meanings of the term as it has been used in the research community.

The meaning we use focuses primarily on the "agency" and "intelligence" dimensions as used by Bradshaw. By "agency", we refer to the notion that agents have persistent existence and identity within an environment in which they can perceive, act, and observe the effects of their actions. By "intelligence", we mean that the agents are *rational* to the limits of their computational capabilities. We use the term "rational" in the decision-theoretic sense to mean that an agent acts to maximize its own *utility*. As an example, a TAC SCM agent exists in a market environment where it competes with other agents, and its utility is measured in economic terms.

TAC SCM agents can be characterized as autonomous, self-interested, and heterogeneous. They are "autonomous", in the sense that once a game starts there is no human intervention. This means that they are not directed by commands coming from a user (or another agent), but by a set of utility and learning functions, which can take the form of individual goals to be achieved. They are "self-interested" in that they are expected to behave in a way that maximizes their own utility, without regard to the utilities of other agents or of the society as a whole. They are "heterogeneous" in the sense that agents differ in their capacities, and in general must find other agents to supply the resources they need to satisfy their own goals. A second type of heterogeneity is that agents of the same type are often implemented in different ways. In TAC SCM each team provides its own agent which complies with the rules of the market, but internally runs different algorithms to achieve the external goal as its competitors.

1.3 Motivating examples

To make the ideas more clear, here are several example scenarios from different industries where our proposed approach could be applied if the vision behind economic regimes were to be further developed and commercialized.

Supply-chain management –

In TAC SCM [Collins *et al.*, 2005], each of the competing agents plays the part of a manufacturer of personal computers. Agents compete with each other in the procurement market where they buy computer components, and in the sales market where they sell finished computers. Each game runs for 220 simulated days, which take approximatively an hour of real time. Each agent starts with no inventory and an empty bank account, and so must borrow money (and pay interest on it) to build up an initial parts inventory before it can begin assembling and selling computers. The agent with the largest bank account at the end of the game is the winner.

Other examples where agents are used in supply-chain management include procurement [Sandholm, 2006, CombineNet, 2006], scheduling and resource management [I2, 2006, Collins *et al.*, 2001], and personal information management [Berry *et al.*, 2006, Mark and Perrault, 2006].

Financial markets – The Penn-Lehman Automated Trading Project [Kearns and Ortiz, 2003] is a broad investigation of algorithms and strategies for automated trading in financial markets and related environments. The project makes use of the Penn Exchange Simulator, a simulator for automated trading that uses real-world, real-time stock market data.

Auction-based contracting – The MAGNET [Collins *et al.*, 2002a] automated contracting environment is designed to support negotiation among multiple, heterogeneous, self-interested agents over the distributed execution of complex tasks that have time and precedence constraints. MAGNET is highly configurable and extensible, and has been used for several statistical studies aimed at understanding the decision processes for a Customer agent.

Travel arrangements – In the TAC classic game [Wellman *et al.*, 2001], each agent is a travel agent, with the goal of assembling travel packages from TACtown to Tampa, during a notional 5-day period for its clients. Each agent acts on behalf of eight clients, who express their preferences for various aspects (hotel, entertainment, etc) of the trip. The objective of the travel agent is to maximize the total satisfaction of its clients (i.e. the sum of the client utilities).

1.4 Contributions

The key contributions of this dissertation are:

1. We present a mathematical formulation of an economic regime. Economic regimes are microeconomic conditions, such as over-supply or scarcity, of a market. Regimes are learned from historical data by using a Gaussian Mixture Model to model the price density of the different products and by clustering price distributions that recur across days. An innovative aspect of our work is that we treat economic regimes at the micro-economic level. So far the existence of regimes has only been reported at the macro-economic level.

2. We develop methods for dynamic identification of regimes and for prediction of regime distribution over a planning horizon. We present three different algorithms (plus an additional one in the Appendix). The first is based on exponential smoothing, the second on a Markov prediction process, and the third on a Markov correction-prediction process. We examine a wide range of tuning options for these algorithms, and show how they can be used to predict prices, price trends, and the probability of receiving a customer order.

3. We present principles and algorithms for tactical decision making, such as calculation of customer offer prices, for strategic decision making, such as allocation of products over the planning horizon, and for market manipulation. We show how knowledge of the current and future regime distribution facilitates tactical and strategical decision making by the agent.

4. We demonstrate the value of prediction within supply-chain environments, and how predictions impact all aspects of the supply-chain, such as procurement and production. We also demonstrate that the identification and prediction of

economic regimes is valuable outside of supply-chain management domain, such as for financial markets.

Our proposed method has the advantage that it works in any market for durable goods, since the computational process is completely data driven and no classification of the market structure (monopoly vs competitive, etc) is needed. A regime encapsulates a whole set of market parameters, with their appropriate range tailored to a specific market condition, i.e. the dimensionality of the parameter space is decreased.

Economic regimes provide more degrees of freedom than ordinary regression based approaches, since the full price distribution is available for decision making. Classical time-series models assume a stationary environment, which is not true for micro-economic environments. Economic regimes are a tool specially suited to make predictions in non stationary environments. Economic regimes also provide an opportunity of niche learning, i.e. an agent is able to apply different approaches and actions when specific regimes are dominant.

1.5 Guide to the thesis

Here we outline the main contents of the chapters that follow.

Chapter 2 - Literature Review – We present a categorization of related work. In particular, we examine work in multi-agent marketplaces, identification and prediction of regimes, price prediction, order probability prediction, opponent modeling, and agent design.

Chapter 3 - Simulation of a Multi-agent Supply-Chain Environment – We describe the TAC SCM game, and we outline the architecture and design of our MinneTAC agent.

Chapter 4 - Tactical and Strategic Sales Decision – We describe what an agent needs to know to be able to make strategic and tactical sales decisions.

Chapter 5 - Economic Regimes – We introduce the concept of "economic regime" and its representation based on learned probability density functions.

Chapter 6 - Performance Evaluation – We show how our method is used in an automated trading agent in TAC SCM and in financial markets, and we analyze experimentally the performance of the method.

Chapter 7 - Conclusions – We conclude with a review of our contributions along with considerations of future work in the area of economic regimes.

Appendix – For reader's convenience, we present a summary of our notation and details of some algorithms.

Figure 1.1 shows graphically the different parts of the decision processes of the agent and maps them to chapters in this thesis.

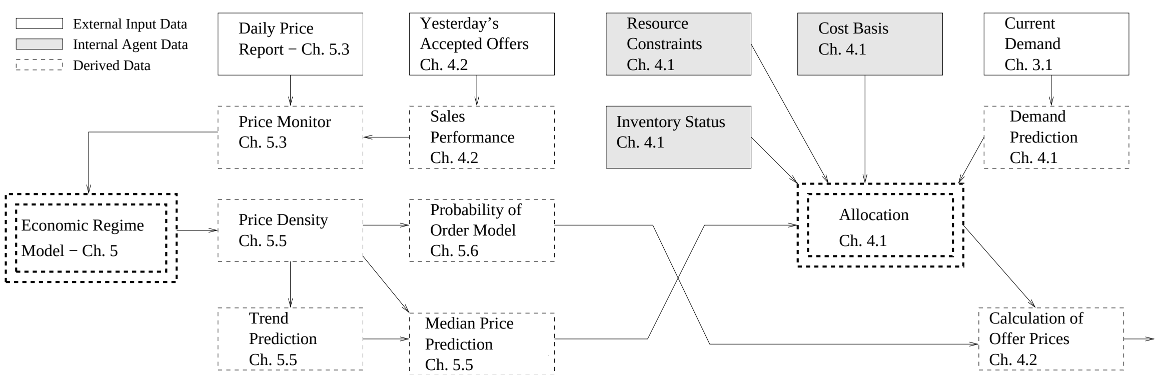


Figure 1.1: Pricing Chain. Allocation and regime modeling (double borders) are the focus of the thesis. These are tools for strategic decision making. The calculation of the offer prices is a tactical decision.

Chapter 2

Literature Review

This work draws from several fields. In Computer Science, it is related to Artificial Intelligence and autonomous agents, especially machine learning, planning, and reasoning under uncertainty. In Economics and Information Decision Science, it draws from the framework of auction theory, probability theory, decision theory, and game theory. From Operations Research, it draws from work in combinatorial optimization and supply-chain management.

In this Chapter we explore previous work on world modeling, regime formulation, model selection, opponent modeling, machine learning, and predictions in dynamic multi-agent environment, especially electronic market places. We also review the work on agent limitations, focusing primarily on handling limitations within the context of decision making, and the connecting perception problem.

2.1 Electronic Commerce

2.1.1 Architectures and Auctions

Markets play an essential role in the economy, and market-based architectures are a popular choice for multiple agents (see, for instance, [Chavez and Maes, 1996, Sycara and Pannu, 1998, Wellman and Wurman, 1998, Collins *et al.*, 2002a, Karacapilidis and Moraïtis, 2001, Choi and Liu, 2001]). Traditional economic approaches based on common knowledge often do not work in electronic marketplaces, since the agents usually lack large background information and the sophisticated reasoning ability of their human counterparts. In our research we focus on the study of economies of manufacturer and customer agents, where physical goods are the objects of exchange. Specifically, we are interested in how an agent manufacturer of physical goods can efficiently learn about changing market conditions and use this knowledge to steer its internal operations and adapt to the preferences of the customer population.

2.1.2 Issues

Electronic commerce has brought many opportunities and even more challenges. [Gupta *et al.*, 1996] discusses the economic challenges that electronic commerce will present. The article deals with a wide range of issues from Internet traffic pricing to information pricing to online micro payment to competitive markets. [Gupta *et al.*, 1997] discusses the problems that congestion will create and showed that it may be detrimental for both customers and access providers to not have volume based traffic pricing.

With the increasing number of companies doing business in the Internet, security and trust in multi agent system to protect against fraud have gained an important role, for instance in [Jaiswal *et al.*, 2003] the authors identify the security vulnerabilities of MAGNET and present a solution that overcomes these weaknesses.

2.2 Agents and Domain Modeling

2.2.1 Characterization of an Agent

All agents have three key attributes: perception, reasoning, and action. These three components operate within the context of some environment. The percepts an agent receives depends on the environment, and the actions an agents performs affects the environment. We will characterize different types of environments an agent, or multiple agents, can be in and the coupling between them. With general but careful assumptions about the environment, agents can effectively reason about appropriate actions to select.

2.2.2 Characterization of the Domain

In [Sutton and Barto, 1998] the authors define a model of the environment as anything that an agent can use to predict how the environment will respond to its actions.

Given a state and an action, a model produces a prediction of the resultant next state and the next reward. If the model it stochastic then there are several different possible next states and rewards and each has a attached a probability of occurrence.

Our special domain of interest are automated and mixed-initiative multi-agent systems in electronic markets. The characteristics of such an environment are listed below:

- complex (many governing variables, spatio-temporal patterns)
- heterogeneous (different types of agents, different strategies)
- dynamic (structure is changing, interests are changing, patterns of communication/interaction are changing)
- limited resources (economic realities) and limitations on reasoning capability
- competitive environment (many direct or indirect opponents)
- not centralized and not centralizable, because their participants are self-interested, rational, and economic agents
- unlimited time frame (no closing date)
- open market/system (influenced by other markets, external parameters)
- high degree of uncertainty (the participating agents have only limited knowledge of the state of the world)
- strategic (agents need to implement many different strategies to compete in the market) and strategic behaviors of agents (perform market manipulation to exploit the environment to gain an advantage)

In Table 2.1 we characterize multi-agent systems [1] and classify TAC SCM in the given schema.

[1] Table format taken from [Weiss, 1999].

	Attribute	Range	TAC SCM
Agents	number	from two upward	6
	uniformity	homogeneous ... heterogeneous	heterogeneous
	goals	contradicting ... complementary	contradicting
	architecture	reactive ... deliberative	deliberative
	abilities (sensors, effectors, cognition)	simple ... advanced	medium
Interaction	frequency	low ... high	depends on type
	persistence	short-term ... long-term	long-term
	level	signal-passing ... knowledge passing	signal passing
	pattern (flow of control)	decentralized ... centralized	decentralized
	variability	fixed ... changeable	changeable
	purpose	competitive ... cooperative	competitive
Environment	predictability	foreseeable ... unforeseeable	partial foreseeable
	accessibility and knowability	unlimited ... limited	limited
	dynamics	fixed ... variable	variable
	diversity	poor ... rich	rich
	availability of resources	restricted ... ample	restricted

Table 2.1: Characterization of multi-agent systems and relationships with TAC SCM.

2.3 Regime Formulation

The analysis in [Massey and Wu, 2005] shows that the ability of decision makers to correctly identify the onset of a new regime can mean the difference between success and failure. Furthermore they found strong evidence that individuals pay inordinate attention to the signal (price in our case), and neglect diagnosticity (regime dynamics) and transition probability (Markov matrix), the aspects of the system that generates the signal. Individuals who do not pay enough attention to regime identification and prediction have the tendency to over- or underreact to market conditions. The degree to which used products cannibalize new product sales for books on Amazon.com is analyzed empirically in [Ghose *et al.*, 2006]. In their study they show that product prices go through different regimes over time.

Marketing research methods have been developed to understand the conditions for growth in performance and the role that marketing actions can play to improve sales. For instance, [Pauwels and Hanssens, 2002, Pauwels and Hanssens, 2004] provide marketing research methods to gain a dynamic understanding of the conditions for performance growth and of the role marketing actions play to improve sales performance in mature markets. The authors analyze how strategic windows of performance change alternate with long periods of performance stability in mature economics markets.

All these methods fail to take into account market conditions that are not directly observable. They are essentially regression models, and do not represent qualitative differences in market conditions. Our method, in contrast, is able to detect and forecast a broader range of market conditions. Regression based approaches (including non-parametric variations) assume that the functional form of the relationship between dependent and independent variables has the same structure. An approach like ours that models variability and does not assume a functional relationship provides more flexibility and detects changes in relationship between prices and sales over time.

An analysis of the TAC SCM 2004 competition ([Kiekintveld *et al.*, 2005]) shows that supply and demand (expressed as regimes in our method) are key factors in determining market prices, and that agents which were able to detect and exploit these conditions had an advantage.

2.4 Model Selection

Model selection is the task of choosing a model of optimal complexity for the given (finite) data. A good overview about concept, theory and different methods of model selection is given in [Cherkassky and Mulier, 1998]. Brooks et al. [Brooks *et al.*, 2002] addresses the problem of a monopolist producer agent selecting a model of a population of customer agents to learn when it must account for the cost of learning.

2.5 Opponent Modeling

Since the success of a TAC SCM agent is likely to depend greatly on the strategies of other agents, it is reasonable to assume that learning the behaviors of other agents may be vital to good performance. The problem is that an agent in the TAC SCM domain does not have direct contact with its opponents, e.g., while bidding for a customer RFQ it only knows if it received an offer or not, but not which other agents bid and at what price. Our suggest method of estimating and using regimes for price prediction and other agent goals is a way of indirectly modeling opponents. Opponent modeling is a hard problem in multi-agent systems and is successful only is smaller settings so far.

Urszula Chajewska, Daphne Koller and Dirk Ormoneit [Chajewska *et al.*, 2001] show a method for predicting the future decisions of an agent based on it past decisions. They are learning the agent's utility functions by observing its behavior. Their approach is based on the assumption that the agent is a rational decision maker. According to decision theory, rational decision making amounts to the maximization of the expected utility [von Neumann and Morgenstern, 1947]. In TAC SCM we cannot assume that all the agents are rational.

Michael Littman [Littman, 1994] describes reinforcement learning approach to solving two-player zero-sum games in which the "max" operator in the update step of standard Q-learning algorithm is replaced by a "minimax" operator that can be evaluated by solving a linear program. He demonstrates the performance of his minimax-Q learning algorithm on a grid-soccer scenario, where his algorithm plays against a random player and a player of its own kind. The algorithm minimax-Q correctly extends Q-learning to find optimal probabilistic policies. We could use the idea of grid-soccer to implement our own learning algorithms and test them in this scenario.

Carmel and Markovitch [Carmel and Markovitch, 1993] describe a game-player that tries to analyze and learn the strategy of its opponent. They discuss the benefits of using a model of the opponent s strategy, and give an algorithm called M* that attempts to exploit the opponent s strategy.

2.6 Prediction Methods

Andrew Ng and Stuart Russel [Ng and Russell, 2000] show that the agent's decisions can be viewed as a set of linear constraints on the space of possible utility (reward) functions. The simple reward structure they used in their experiments will not scale to our future needs, i.e. offer price prediction in TAC SCM, we will need to extend this.

2.6.1 Price Prediction

Predicting prices is an important part of the decision process of agents or human decision makers. [Kephart *et al.*, 2000] explored several dynamic pricing algorithms for information goods, where shopbots look for the best price, and pricebots adapt their prices to attract business. [Wellman *et al.*, 2003] analyzed and developed metrics for price prediction algorithms in the TAC Classic game, similar to what we have done for TAC SCM.

TAC participants recognized early on that price prediction in connection with overall agent performance is important [Stone and Greenwald, 2005]. [Stone *et al.*, 2003] lists a diversity of price prediction methods among TAC-Classic 2001 agents.

Brooks et al. [Brooks *et al.*, 2002] designed pricing strategies for agents which exchange information goods, they assessed their performance, and how well they adapt to changing environmental situations. In our research we focus on agents which deal with the exchange of physical goods, we intend to look into the implications when dealing with information goods.

The University of Michigan team demonstrate a method [Kiekintveld *et al.*, 2004] for predicting future customer demand in the TAC SCM game environment, and use the predicted future demand to inform agent behavior. Their approach is specific to the TAC SCM situation, since it depends on knowing the formula by which customer demand is computed. Note that customer demand is only one of the factors for characterizing the multi-dimensional regime parameter space.

The problem of allocating finite resources to producing a set of products in a way that maximizes some measure of utility is the well-known “product-mix” problem ([Hillier and Lieberman, 1990]).

Similar techniques have been used outside TAC SCM to predict offer prices in first price sealed bid reverse auctions for IBM PCs ([Lawrence, 2003]), or PDA’s on eBay ([Ghani, 2005]).

In [Schapire *et al.*, 2002] the problem of predicting prices of goods in auctions is solved via a machine-learning approach. There method is based on logistic regression and boosting which is explained in detail in [Collins *et al.*, 2002b, Witten and Frank, 2000].

[Gupta *et al.*, 2000] proposes alternative approaches for pricing Internet access dynamically. We are thinking to adapt parts of their non-parametric statistical technique to predict the daily order price range for different computer types.

2.6.2 Probability of Order Prediction

The problem of predicting the probability of order in sealed bid auctions is commonly approached through statistical methods as those surveyed in [Papaioannou and Cassaigne, 2000]. These kinds of methods require large amounts of observed data in terms of opponents bidding behavior and a static environment. TAC-SCM on the other side is a highly dynamic and uncertain environment and therefore nearly all agents in the TAC SCM competition use some dynamic way of modeling the probability of receiving an order. Botticelli [Benisch *et al.*, 2004] uses a linear CDF to determine the relationship between offer price and order probability. We use a reverse CDF and take other factors into account, such as quantity and due date. TacTex [Pardoe and Stone, 2004] uses the lowest and highest offer price, which are provided for each product every day by the game server, and determines the probability of an order by linear interpolation. Their estimates depend only on the type of computer requested and the reserve price, whereas we use more parameters in our previous work [Ketter *et al.*, 2004c, Ketter *et al.*, 2004a, Ketter *et al.*, 2004b] (6 parameters for the MaxEProfit strategy and 5 parameters for the DemandDriven strategy). RedAgent [Keller *et al.*, 2004], the winner of last year TAC SCM , uses an internal marketplace structure with competing bidders to set offer prices. PackaTAC [Dahlgren and Wurman, 2004] lets other agents set the price and tries to follow. The Jackaroo team [Zhang *et al.*, 2004] applied a game theoretic approach to set offer prices, using a variation of the Cournot game for modeling the product market. PSUTAC [Sun *et al.*, 2004] employs an expert system for decision making. They are able to express market strategies and knowledge in a human-understandable form. Pindyck et al. [Pindyck and Rubinfeld, 1998] give a good overview of the science and art of building and using forecast models.

Since we estimated the bottleneck was going to be in the supply and not in the production [McMillen, 2003], we did not worry, as other teams [Benisch *et al.*, 2004, Pardoe and Stone, 2004], about optimizing the production of our agent.

2.6.3 Evaluation of Predictive Quality

If competitions such as TAC-Classic and TAC-SCM are to be successful in facilitating research, it will be necessary to separately evaluate methods that have been applied to individual tasks [Stone, 2003], such as procurement and sales. Many of these interesting tasks in such complex environments are not strictly separable, which makes the evaluation of those tasks harder.

The University of Michigan team analyzed post-competition performance of the TAC SCM winning agents and explored relationships between total profit and other measurements of performance [Kiekintveld *et al.*, 2005, Jordan *et al.*, 2006]. The same team [Wellman *et al.*, 2006] translates end of game profit into a new metric, demand-adjusted profit (DAP), which attempts to factor out profit variations caused by differing amounts of game demand. This use of control variates greatly reduces the amount of variance in profit, and it is quite useful as a benchmark for agent performance. DAP effectively controls the most influential market parameter, but there are undoubtedly other market factors affecting profit, and if these were also accounted for, variance could be reduced by an even greater margin.

Attempts have been made to control the profile space as well. For instance, [Wellman *et al.*, 2005] introduced a variation of the TAC games, called SCM$\downarrow_3$, which uses half as many distinct agents per game and significantly reduces the profile space.

To facilitate analysis of TAC SCM games, the Supply Chain Trading Analysis and Instrumentation Toolkit (AIT) has been developed [Benisch *et al.*, 2005] and made available to the community. The tool simplifies downloading and parsing game data, and provides support for analysis of agent performance, prices, market shares, average daily prices, etc.

In [Babanov *et al.*, 2003b], [Babanov *et al.*, 2003a] and [Ketter *et al.*, 2003] we describe how an evolutionary framework could be used as a platform for systematic testing of agent strategies and illustrate the idea with results from a simple supply-demand model.

2.7 Agent Design

Most agent design efforts have focused on either the autonomous behavior aspects of agency, or on interaction among agents. Shoham's Agent-Oriented Programming [Shoham, 1997] examines a cognitive and societal view of computation. Bradshaw's KAoS agents [Bradshaw *et al.*, 1997] are BDI agents in a CORBA environment. Agents have *capabilities* based on existing document management applications. Norman *et al.* [Norman *et al.*, 1997] describe agent societies that model organizational structures and automate business processes. These ADEPT agents negotiate over service agreements that can involve many parties and many dimensions. JADE [Moraitis *et al.*, 2003] is an agent framework that has been used to build trading agents, and could have been used for MinneTAC. However, its primary emphasis is on building multi-agent systems that comply with FIPA specifications for inter-agent communications, and with flexible deployment in a network environment. This is not a requirement for the TAC SCM domain. The MinneTAC design is *compositional* in the sense of Brazier *et al.* [Brazier *et al.*, 2002], but not hierarchically so. The DESIRE method from Brazier *et al.* does not seem applicable to the MinneTAC situation, since we are dealing with a single agent in an existing environment, and the blackboard approach used in MinneTAC is not easily modeled with DESIRE. RETSINA [Sycara and Pannu, 1998] suggests both a multi-agent architecture with a variety of agent roles, and an architecture for individual agents that provides communications, planning, scheduling, and execution monitoring. This architecture could probably be adapted to the TAC SCM domain, but its planning and communication capabilities would not be especially useful. Vetsikas and Selman [Vetsikas and Selman, 2003] show a method for studying design tradeoffs in a trading agent. This approach could be likely be used effectively in MinneTAC.

A few of the participants in TAC SCM have described their agent designs. He *et al.* [He *et al.*, 2006] have adopted a design consisting of three internal "agents" to handle Sales, Procurement, and Production/Shipping. Sales decisions use a fuzzy logic module. Some algorithmic detail is given, but there is little further detail on the architecture of the agent. TacTex05, the winner of the 2005 competition [Pardoe and Stone, 2006] is based on two major modules, a Supply Manager that handles procurement, and a Demand Manager that handles sales, production, and shipping. These modules are supported by a supplier model, a customer demand model, and a pricing model that estimates sales order probability.

Ultimately, the TAC SCM problem domain does not require the sort of flexible cognitive and social elements of these more "traditional" agent designs. Instead, our focus has been on separating the decision tasks and supporting research needs, and we have found the component-oriented model to be ideal.

Chapter 3

Simulation of a Multi-Agent Supply-Chain Environment

Electronic commerce is one of the more compelling application areas for autonomous agents. In most electronic commerce applications, decisions can be relatively clear-cut (buy or sell, set a price, submit a bid, award bids, etc.), and communications among agents and between agents and their environments can be constrained and highly scripted.

One way to drive development and understanding of decision making processes by autonomous agents in complex domains is to hold competitions. An example of such a competition is the Supply-Chain Management Trading Agent Competition (TAC SCM) [Collins *et al.*, 2005], an international tournament which engages agents in simultaneous buying, selling, production scheduling, and inventory management problems.

This Chapter provides an overview of TAC SCM and outlines the design of the MinneTAC trading agent, which has competed effectively in TAC SCM for several years. The design has attempted to respond both to the challenges of the game scenario as well as to the need to support multiple relatively independent research efforts that are focused on meeting one or more of those challenges. We evaluate the success of our design both in terms of the competitiveness of the agents that have been implemented with it, and in terms of its ability to support our research agenda.

3.1 TAC SCM Game Description

In a TAC SCM game, each of the competing agents plays the part of a manufacturer of personal computers. In an instance of a TAC SCM game six autonomous agents compete with each other in the procurement market, where they buy computer parts, and in the sales market, where they sell computers to customers, as shown in Figure 3.1.

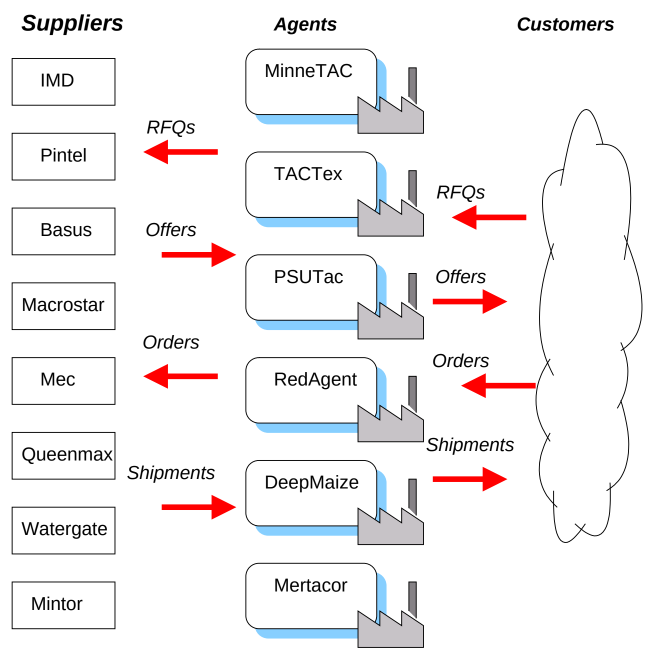


Figure 3.1: Schematic overview of a typical TAC SCM game scenario. Six autonomous agents compete to buy parts from suppliers and to sell finished computers to customers.

Availability of parts and demand for computers varies randomly through the game and across market segments (low, medium, and high computer price). The market is affected not only by variations in supply and demand, but also by the actions of the other agents. The small number of agents and their ability to adapt and to manipulate the market makes the game highly dynamic and uncertain. Each agent is self-interested and tries to maximize its profit, while competing with the other agents for raw materials (parts) and customer orders.

The simulation takes place over 220 virtual days, each lasting fifteen seconds of real time. Each agent starts with no inventory and an empty bank account, and must borrow money (and pay interest) to build up an initial parts inventory before it can begin assembling and selling computers. Agents pay for the parts they buy, pay interest on the money they borrow, and pay storage costs for their inventory. If they ship computers late, they pay late penalty. Agents earn money by selling computers. Any inventory left at the end of the game has no residual value. The agent with the largest bank balance at the end of the game wins.

A Component Catalog, see Table 3.1, and Bill of Materials, see Table 3.2, are sent to each agent at the beginning of the game. The Component Catalog lists each part, along with its base price and the list of suppliers who can produce it. Each part is produced by one or two suppliers; each supplier provides two different types of parts. The Bill of Materials lists 16 different combinations of parts that can be assembled into personal computers (PC). Each of these computer types is identified uniquely by a stock keeping unit number. Each computer type is assigned a number of processing cycles that specifies how much time it takes to assemble that type of computer from raw materials. These PCs are the finished goods of the TAC SCM supply chain.

To obtain parts, an agent must send a *request for quotes* (RFQ) to an appropriate *supplier*. Each RFQ specifies a part type, a quantity, and a due date. The next day, the agent receives a response to each of its requests.

Suppliers respond by evaluating each RFQ to determine how many parts they can deliver on the requested due date and how long it would take to produce all the parts requested, considering the outstanding orders they have committed to and the RFQs they have already responded to in this turn. If the supplier can produce the desired quantity on time, it responds with an offer that contains the price of the parts supplied. If not, the supplier responds with two offers: (1) an earliest complete offer with a revised due date and a price, and (2) a partial offer with a revised quantity and a price. The agent can accept either of these alternative offers, or reject both.

Suppliers may deliver late, due to randomness in their production capacities. If a supplier has excess capacity, the price offered for its parts will be discounted; discounted prices may be as low as 50% of the base price.

Once an agent has parts to assemble computers, it must schedule the assembly tasks in its production facility. Each computer model requires a specified number of assembly cycles, and the assembly capacity of each agent is limited. Assembled computers are added to the agent's finished-goods inventory, and may be shipped to customers to satisfy outstanding orders.

Every day each agent receives a set of RFQs from potential *customers*. Each customer RFQ specifies the type of computers requested, along with quantity, due date, reserve price, and late penalty. Each agent may choose to bid on some or all of the day's RFQs. Customers accept the lowest bid that is at or below their reserve price, and notify the winning agent the following day.

The agent must ship customer orders on time, or pay a penalty for each day an order is late. If a product is not shipped within five days of the due date the order is canceled, the agent receives no payment, and no further penalties accrue.

Table 3.1 shows the part catalog for a typical game, with information about each part, its base price, and the suppliers that produce it.

Part	**Base price**	**Supplier**	**Description**
100	1000	Pintel	Pintel CPU, 2.0 GHz
101	1500	Pintel	Pintel CPU, 5.0 GHz
110	1000	IMD	IMD CPU, 2.0 GHz
111	1500	IMD	IMD CPU, 5.0 GHz
200	250	Basus, Macrostar	Pintel motherboard
210	250	Basus, Macrostar	IMD motherboard
300	100	MEC, Queenmax	Memory, 1 GB
301	200	MEC, Queenmax	Memory, 2 GB
400	300	Watergate, Mintor	Hard disk, 300 GB
401	400	Watergate, Mintor	Hard disk, 500 GB

Table 3.1: Part Catalog

There is a total of 10 different parts, which can be combined to build 16 different PC configurations, as described in the Bill of Materials given in Table 3.2.

Computer types are classified into three market segments: High range, Mid range, and Low range. For each of these market segments, at the start of each day, d, customers exhibit their demand by issuing N customer RFQs, according to the following distribution (described in [Collins *et al.*, 2005]):

$$N = \text{poisson}(Q_d) \tag{3.1}$$

where Q_d is the "target average" number of customer RFQs for day d issued in each

SKU	Parts	Cycles	Market segment
1	100, 200, 300, 400	4	Low range
2	100, 200, 300, 401	5	Low range
3	100, 200, 301, 400	5	Mid range
4	100, 200, 301, 401	6	Mid range
5	101, 200, 300, 400	5	Mid range
6	101, 200, 300, 401	6	High range
7	101, 200, 301, 400	6	High range
8	101, 200, 301, 401	7	High range
9	110, 210, 300, 400	4	Low range
10	110, 210, 300, 401	5	Low range
11	110, 210, 301, 400	5	Low range
12	110, 210, 301, 401	6	Mid range
13	111, 210, 300, 400	5	Mid range
14	111, 210, 300, 401	6	Mid range
15	111, 210, 301, 400	6	High range
16	111, 210, 301, 401	7	High range

Table 3.2: Bill of Materials. Each row shows the components needed to build each computer type and the corresponding market segment.

market segment. Q_d is varied using a trend τ that is updated by a random walk:

$$Q_{d+1} = \min(Q_{max}, \max(Q_{min}, \tau_d Q_d)) \tag{3.2}$$

$$\tau_{d+1} = \max(\tau_{min}, \min(\tau_{max}, \tau_d + \text{random}(-0.01, 0.01)) \tag{3.3}$$

Q_0, the start value of Q, is chosen uniformly in the interval $[Q_{min}, Q_{max}]$ (see Table 3.3), and τ_0, the start value of the τ, is 1.0. The trend τ is reset to 1.0 when the random walk exceeds the minimum or maximum boundaries. In other words, if $\tau_d Q_d < Q_{min}$ or $\tau_d Q_d > Q_{max}$ then $tau_{d+1} = 1.0$. This reduces the bimodal tendency of the random walk.

3.1.1 Game Parameters

Table 3.3 gives the parameter settings for the standard TAC SCM competition games. Values for most of these parameters are sent to the agents at the start of every game.

Table 3.4 and Table 3.5 specify the visibility of the game parameters during the games.

Parameter	**Standard Game Setting**
Length of game	220 days
Agent assembly cell capacity	2000 assembly cycles / day
Nominal capacity of supplier assembly lines C^{nom}	500 parts / day
Start capacity of the suppliers assembly lines	$C^{nom} \pm 35\%$
Supplier price discount factor δ	0.5
Down payment due on placement of supplier order	10%
Acceptable purchase ratio for single-source suppliers	0.9
Acceptable purchase ratio for two-source suppliers	0.45
Initial reputation endownment	2000
Reputation recovery rate	100 units/day
Computer types in the low range market	1, 2, 9, 10 and 11.
Computer types in the mid range market	3, 4, 5, 12, 13 and 14.
Computer types in the high range market	6, 7, 8, 15, and 16.
Average number of customer RFQs $[Q_{min}, Q_{max}]$ in the High and Low range markets	25 – 100 per day
Average number of customer RFQs $[Q_{min}, Q_{max}]$ in the Mid range market	30 – 120 per day
Interval between Market Reports	20 days
RFQ volume trend for customers $[\tau_{min}, \tau_{max}]$ (all market segments)	$[0.95, 1/0.95]$
Range of quantities for individual customer RFQs $[q_{min}, q_{max}]$	$[1, 20]$
Range of lead time (due date) for customer RFQs $[\mathtt{due}_{min}, \mathtt{due}_{max}]$	3 to 12 days from the day the RFQ is received
Range of penalties for customer RFQs $[\Psi_{min}, \Psi_{max}]$	5% to 15% of the customer reserve price per day
Customer Reserve Price	75 – 125% of nominal price of the PC parts
Annual bank debt interest rate $[\alpha_{min}, \alpha_{max}]$	6.0 – 12.0%
Annual bank deposit interest rate $[\alpha'_{min}, \alpha'_{max}]$	0.5α
Annual storage cost rate $[S_{min}, S_{max}]$	25% – 50% of nominal price
Short-term horizon for supplier commitments T_{short}	20 days
Down payment due on supplier order	10%
Daily reduction in supplier available capacity for long-term commitments z	0.25%

Table 3.3: Parameters used in the TAC SCM game.

The parameter average unit sales price, which is listed in Table 3.5, is computed every 20 days for the previous 20 days in the following manner:

$$AvergeUnitSalesPrice = \frac{\sum_{i \in \mathcal{I}} OrderQuantity(i) \times OrderUnitPrice(i)}{\sum_{i \in \mathcal{I}} OrderQuantity(i)} \tag{3.4}$$

Parameter	**Visibility**
Parts inventory for each agent.	None
Total quantity per part produced by the suppliers since the last market report.	Every 20 days in market report
Total quantity per part delivered by the suppliers since the last market report.	Every 20 days in market report

Table 3.4: Parameter visibility in the raw-material market of the TAC SCM game.

Parameter	**Visibility**
Customer RFQs (RFQid, computer type, due date, penalty, reserve price)	Full
Total number of bids per customer RFQ.	None
Total number of computers offered per customer RFQ.	None
Unit offer price per customer RFQ per agent.	None
Inventory by computer type of each agent.	None
Total quantity per computer type ordered by the customers since the last market report.	Every 20 days in market report
Average unit sales price per computer type ordered by the customers since the last market report (see Equation 3.4).	Every 20 days in market report
Lowest and highest unit order price per computer type.	One day delayed in price report.

Table 3.5: Parameter visibility in the customer market of the TAC SCM game.

3.2 The design of the MinneTAC trading agent

To address the design challenges of the MinneTAC agent, we follow a component-oriented approach [Szyperski, 1998]. The idea is to provide an infrastructure that manages data and interactions with the game server, allowing individual researchers to encapsulate agent decision problems within the bounds of individual components that have minimal dependencies among themselves. Two pieces of software form the foundation of MinneTAC: the Apache Excalibur component framework [Foundation, 2006], and the "agentware" package distributed by the TAC SCM game organizers. Excalibur provides the standards and tools to build components and configure working agents from collections of individual components, and the agentware package handles interaction with the game server.

The MinneTAC agent is a set of components layered on the Excalibur container, as shown in Figure 3.2. Four of these components are responsible for the major decision processes: Sales, Procurement, Production, and Shipping. All data that must be shared among components is kept in the Repository, which acts as a blackboard [Buschmann *et al.*, 1996]. The Oracle is host to a large number of smaller components that maintain the market and inventory models, and do analysis and prediction. The Communications component handles all interaction with the game server. The components themselves are identified by their roles; in several cases multiple components have been built to fill those roles. It is an explicit goal of this architecture to minimize couplings between the components. Ideally, each component depends only on Excalibur and the Repository.

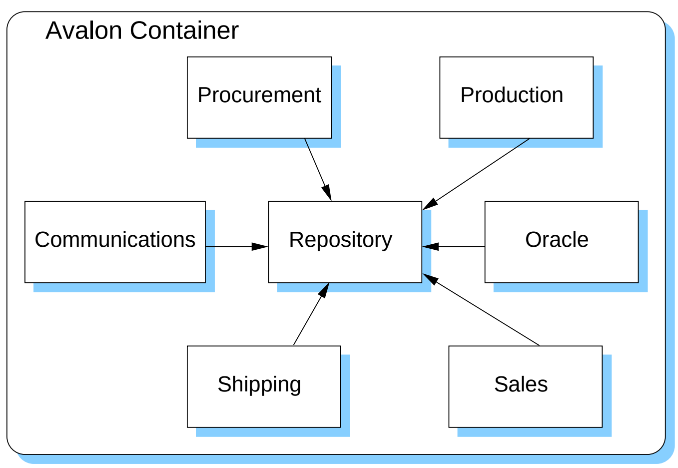


Figure 3.2: MinneTAC Architecture. Arrows indicate API dependencies.

The agent opens three configuration files when it starts. The system configuration file specifies the set of roles that make up the system. The component configuration file specifies runtime configuration options for each component. For example, the Sales component may have a parameter that controls the maximum level of overcommitment of its existing inventory or capacity when it makes customer offers. The log configuration file controls the names and locations of log files that are produced by the running agent, the general format of log entries, and for each component, the level of detail to be logged.

3.2.1 MinneTAC components

The MinneTAC agent consists of seven components. We describe these components and their responsibilities briefly here. More details on the architecture and implementation are in [Collins *et al.*, 2007].

Repository is the unifying element of the MinneTAC design, the one component that is visible to the other components. It serves as an internal database, maintains the state of the system, and notifies other components of changes in state. All other activity is driven by these state changes. Ii also provides the core elements of the Evaluation subsystem.

Communications handles communication with the game server. This includes joining games, acquiring initial game parameters, importing start-of-game and daily data into the Repository, and retrieving agent decisions from the Repository for communication back to the game server.

Procurement procures parts. It may build and maintain target inventory levels, it may attempt to procure parts to meet customer orders, or it may use some other decision process. It must issue RFQs to suppliers and decide whether to accept offers that are returned.

Production schedules the manufacturing facility. It may build and maintain target finished goods inventory levels, or it may build only to meet existing customer orders.

Sales makes offers in response to customer RFQs. It must decide, for each RFQ, whether to bid and what price to quote, based on available and predicted inventories and current market conditions. A sophisticated Sales component might attempt to predict the probability of order acceptance in order to maximize profits.

Shipping ships product to customers. In general, there is a benefit in shipping product as late as possible, because this gives the agent an opportunity to minimize penalties for late deliveries. Late deliveries can happen, for example, if predicted inventories do not materialize due to late supplier shipments.

Oracle maintains market models and predicts future demand and availability. This is done primarily through a set of Evaluators.

To minimize coupling between the various components we use *evaluations* that are accessible through the various data elements in the Repository. The general idea is that when a component needs to make a decision, it will inspect the available data and run some utility-maximizing function. The available data consists of any data it maintains internally, and the data in the repository. Any data reductions or analyses that are performed on Repository data can be encapsulated in the form of Evaluations, and made available to other components. These analyses are implemented by the Oracle component through a configurable set of *evaluators*.

Chapter 4

Tactical and Strategic Sales Decision

We are primarily interested in competitive market environments that are constrained by resources and/or production capacity. In such an environment, a manager who wants to maximize the value of available resources should be concerned about both strategic and tactical decisions.

The basic strategic decision is to allocate the available resources (financial, capacity, inventory, etc.) over some time horizon in a way that is expected to return the maximum yield. For example, in a market that has a strong seasonal variation, one might want to build up an inventory of finished goods during the off season, when demand is low and prices are weak, in order to prepare for an expected period of strong demand and high prices.

For the purpose of this work, tactical decisions are concerned with setting prices to maximize profits, within the parameters set by the strategic decisions. So, for instance, if the forecast sales volume for the current week is 100,000 units, we would want to find the highest sales price that would move that volume.

We will show how our technique of modeling the economic regimes in a market can be used to inform both the strategic and tactical decision processes. In Figure 4.1 we show this process in a schematic way. In our formulation, a regime is essentially a distribution of prices over sales volume. We characterize the market in terms of

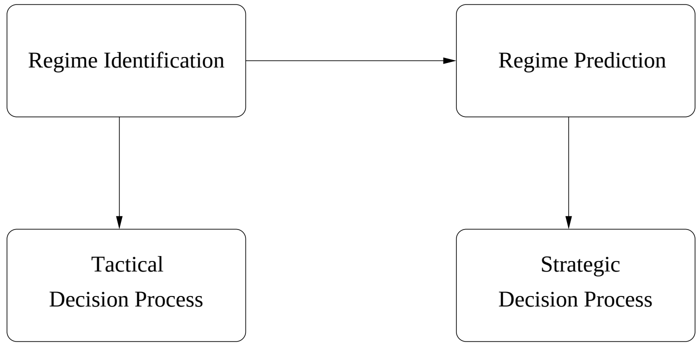


Figure 4.1: Process chart – Regime identification is a tool for tactical decision making and regime prediction is a tool for strategic decision making.

regimes. For tactical decisions, we approximate the probability of selling a product at a given asking price. This combined with demand information leads directly to (nearly) optimal pricing decisions.

To make strategic decisions, we need to forecast regime shifts in the market. If our forecast shows an upcoming period of low demand and weak prices, we may want to sell more aggressively in the short term, and we may want to limit procurement and production to prevent driving an oversupply into the market. On the other hand, if our forecast shows an upcoming period of high demand and strong prices, we may want to increase procurement and production, and raise short-term prices, in order to be well-positioned for the future.

We wish to maximize the profit the agent can expect to earn over some reasonable period in the future. Our approach is to treat procurement, production, and sales as separate components each with its own decision process, and to keep interactions among the components to a minimum. This is common in industries where procurement, and often production as well, are driven by relatively long-term forecasts, while sales is expected to move the products it has available to sell (and expects to have in the future) at the best possible price.

In the MinneTAC agent, sales interact with procurement primarily through current and expected inventories. Both sales and procurement have access to projections of future customer demand (see [Kiekintveld *et al.*, 2004]) and to customer pricing models. Production is primarily to order, except that small inventories of finished goods are maintained to support short lead-time demand.

Sales decisions must allocate the agent's resources over two dimensions, product mix and time, in a way that maximizes value. In the TAC SCM environment, the small number of competitors means that individual agents have capacity to supply a significant fraction of the total market, and therefore the power to depress prices by offering too much product. Production is primarily driven by sales, which also determine what is produced. Because procurement is at least partly driven by projected inventories and by predicted customer demand, sales activity also influences procurement. But procurement typically operates over a longer time horizon, and sales must be focused on getting the highest possible prices for the products it has available. If inventory is sold out during a period when prices are low, then there may be nothing available to sell when prices recover.

4.1 Strategic decision – resource allocation[1]

Sales decisions can be informed both by experience in the past and by current observations. We first focus on the information that is visible to the agent during the game. In TAC SCM, game data include the following:

- $\mathcal{C}$ is the set of all available component types.
- $\mathcal{G}$ is the set of all goods (product types) that can be built and sold. Each good is made up of a set $\mathcal{C}_g$ of components. This means that in turn, each component c is a part of some set of products $\mathcal{G}_c$.
- On each day d, customer demand is represented by a set $\mathcal{R}_d$ of customer RFQs received by the agent. Each RFQ $r \in \mathcal{R}_d$ specifies a product type g_r, a lead time of i_r days, a quantity q_r, and a reserve price ρ_r. Reserve price is uniformly distributed between $\rho^{\min}$ and $\rho^{\max}$. Details and semantics are given in [Collins *et al.*, 2005].
- Customer demand is projected into the future over some planning horizon h. In the TAC SCM scenario, we model customer demand following the method given in [Kiekintveld *et al.*, 2004]. For each market segment m, and for each future day over some planning horizon h, this produces expected values for mean demand $Q^m_{d,i}$, $i = 0 \ldots h$, and for demand trend $\tau^m_{d,i}$.

[1] We are indebted to John Collins for the development of the material in this Section.

- At the beginning of each day d, the agent has an inventory of raw materials consisting of $I_{d,c}$ for each component type $c \in \mathcal{C}$, and an inventory of finished goods consisting of $I_{d,g}$ for each type of good $g \in \mathcal{G}$.
- On each day d, there is a set of outstanding customer orders $\mathcal{O}_d^{cust}$ that have not yet been shipped or canceled, and a set of outstanding supplier orders $\mathcal{O}_d^{supp}$ that have not yet been received.

From this data, we would like to find a way to set prices and respond to customers' RFQ to maximize the agent's overall profits. On any given day d, the total demand $D_{d,g}$ for a given good g among $\mathcal{R}_d^{cust}$ is the total of the of the requested quantities among requests for good g, given by

$$D_{d,g} = \sum_{r \in \mathcal{R}_{d,g}^{cust}} q_r \tag{4.1}$$

We assume that the price $price_{d,g} = f(D_{d,g}, A_{d,g})$ sustainable by the market for a given product p on a particular day d is a function of the demand $D_{d,g}$ and the quantity of product the agent wishes to sell represented by the *allocation* or sales quota $A_{d,g}$ for good g on day d.

The profit per unit for product p to be sold on day d at price $price_{d,g}$ is given by

$$\Phi_{d,g} = discount(d)(price_{d,g} - cost(\mathcal{C}_g)) \tag{4.2}$$

We include the discount term as a rough approximation of inventory holding cost. It can also be used to encourage early selling, as a hedge against the uncertainty of the game.

For any given day d, there is an unsold inventory I'_g of good g, and an expected uncommitted inventory $I'_{d,c}$ of parts of type c. This includes parts in current inventory, and parts that are expected to be delivered by day d, and excludes parts that are committed to producing goods for outstanding customer orders.

The *effective demand* function $D_{d,g}^{eff} = f(D_{d,g}, price_{d,g})$ for our goods will be some function of the prices $price_{d,g}$ we wish to charge. In the TAC SCM environment, there is a linear distribution of reserve prices among customer RFQs. The effective demand, then, is the portion of total demand with reserve prices $\rho \geq price_{d,g}$ at or above the price we want to sell at:

$$D^{\mathit{eff}}_{d,g} = \frac{\rho^{\max}_g - \mathit{price}_{d,g}}{\rho^{\max}_g - \rho^{\min}_g} D_{d,g} \tag{4.3}$$

where $\rho^{\max}_g$ is the maximum reserve price for good g. This assumes that actual demand is uniformly distributed across the range of reserve prices, which is only approximately correct.

The total profit Φ over a planning horizon of h days for the set of goods $\mathcal{G}$ is then

$$\Phi = \sum_{d=0}^{h} \sum_{g \in \mathcal{G}} \Phi_{d,g} A_{d,g} \tag{4.4}$$

This is what we wish to maximize, by computing values for $A_{d,g}$, subject to the following constraints:

1. We can't sell more of any product than the effective demand at the price we wish to charge:
$$\forall d, \forall g, A_{d,g} < D^{\mathit{eff}}_{d,g} \tag{4.5}$$

2. For any given period of time from now until the planning horizon h, we can sell goods that we have in inventory, and goods for which we have the necessary parts in inventory. Note that this is unnecessarily conservative, since we are asking for goods or their parts to be available at the time we propose to *sell* them, not when we expect to *ship* them. This means that we cannot easily discriminate on lead time.
$$\forall m \in 1..h, \forall c \in \mathcal{C}, \sum_{d=1}^{m} \sum_{g \in \mathcal{G}_c} A_{d,g} \leq I'_{m,c} + \sum_{g \in \mathcal{G}_c} I'_g \tag{4.6}$$
Note that this constraint limits commitments of the *sets* of goods that share a given component. If we don't carry any uncommitted finished goods inventory, in other words if $\forall g \in \mathcal{G}_c, I'_g = 0$, then this is a sufficient expression of inventory constraint. Otherwise, imbalances in the finished-goods inventories of individual goods sharing a component could lead to overcommitment. This is easy to see if for some component c, $I'_{m,c} = 0$. Then the sum of individual product inventories constrain the whole set of products. In this case, it is also necessary to constrain

every subset of product types that can share some component. This requires that we replace Equation 4.6 with

$$\forall m \in 1..h, \forall c \in \mathcal{C}, \forall \mathcal{G}'_c \subseteq \mathcal{G}_c, \sum_{d=1}^{m} \sum_{g \in \mathcal{G}_c} A_{d,g} \leq I'_{m,c} + \sum_{g \in \mathcal{G}'_c} I'_g \tag{4.7}$$

3. The agent's factory has limited daily capacity F. If each unit of good g requires y_g production cycles, then

$$\forall m \in 1..h, \sum_{g \in \mathcal{G}} y_g \left(\sum_{d=1}^{m} A_{d,g} - I'_g \right) \leq mF - F_m^{commit} \tag{4.8}$$

   where F_m^{commit} is the factory capacity that is committed to manufacture all outstanding customer orders that are due on or before day m and are not satisfiable by existing finished goods inventory.

The outcome of our objective function (Eq. 4.4) is daily sales quotas $A_{d,g}$ for each good. The next step is to set prices so that we sell what we intend to sell, in a competitive market. Assume we have a formula for probability of a customer placing an order as a function of price $P(order|price)$, produced by some learning process (see Sect. 5.6). But the quantity we sell is just the effective demand multiplied by the probability of order at the price we set. So to make our sales quota, we need

$$A_{d,g} = P(order|price_{d,g}) D_{d,g}^{eff} \tag{4.9}$$

In the TAC SCM environment, with its linear distribution of reserve prices, this gives

$$A_{d,g} = P(order|price_{d,g}) \frac{\rho_g^{\max} - price_{d,g}}{\rho_g^{\max} - \rho_g^{\min}} D_{d,g} \tag{4.10}$$

which is quadratic in $price_{d,g}$, assuming that $P(order|price_{d,g})$ is linear. Combining Equation 4.4 with Equations 4.2 and 4.10, we have

$$\Phi = \sum_{d=1}^{n} \sum_{g \in \mathcal{G}} discount(d) \left(price_{d,g} - cost(\mathcal{C}_g) \right) P(order|price_{d,g}) \frac{\rho_g^{\max} - price_{d,g}}{\rho_g^{\max} - \rho_g^{\min}} D_{d,g} \tag{4.11}$$

which is at least cubic in $price_{d,g}$.

Because the formula for sales quota allocations above is probably unsolvable given the time constraints of the TAC SCM game environment, there is a need for heuristics and simplifications. An obvious simplification is to assume that the partial derivative of the order probability function with respect to price is very steep. This is equivalent to saying that (most) sales occur at a "market clearing price," or alternatively that the probability of order is much more sensitive to price than is profit. Then the per-unit profit and the effective demand can be computed separately, by substituting an estimated clearing price $price^{est}_{d,g}$ for the actual sales price into Eq. 4.2. We will explore a way to compute $price^{est}_{d,g}$ in the next section.

4.2 Tactical decision – sales offer pricing

Once the strategic sales process has determined daily sales quotas, the next step is to set prices for our goods that will yield the maximum profit. This amounts to finding, for each good, the value for $price_{d,g}$ that satisfies the relation

$$\frac{A_{d,g}}{D^{eff}_{d,g}} = P(order|price_{d,g}) \tag{4.12}$$

which is a simple rearrangement of Eq. 4.9.

This could be solved analytically or numerically, assuming we have reasonable functions for $D^{eff}_{d,g}$ and $P(order|price_{d,g})$. In general, however, one or both of these functions are likely to be empirically-derived. Under the previous assumption of most sales occurring close to a market clearing price, we can approximate $D^{eff}_{d,g}$ using $price^{est}_{d,g}$, reducing the computation to finding the value of $price_{d,g}$ that satisfies

$$\frac{A_{d,g}}{D^{eff}_{d,g}(price^{est}_{d,g})} = P(order|price_{d,g}) \tag{4.13}$$

When prices are set in this way, the resulting customer orders provide an additional signal from the market that can be used to refine our estimate of $price^{est}_{d,g}$. If $O_{d,g}$ is the number of orders placed for good g on day d (as a result of offers made on day $d-1$), then a refined estimate of the actual market prices on that day $price^{act}_{d-1,g}$ can be found by finding an adjusted probability distribution $P^{adj}(order|price_{d-1,g})$ such that

$$\frac{O_{d,g}}{D^{eff}_{d-1,g}(price_{d-1,g})} = P^{adj}(order|price_{d-1,g}) \tag{4.14}$$

and computing an estimated actual price $price^{act}_{d-1,g}$ such that

$$\frac{A_{d-1,g}}{D^{eff}_{d-1,g}(price_{d-1,g})} = P^{adj}(order|price^{act}_{d-1,g}) \tag{4.15}$$

See Fig. 4.2 for a graphical visualization of this relationship. For simplicity, we illustrate an approximate adjustment made by shifting the location of the probability curve along the price axis without changing its shape.

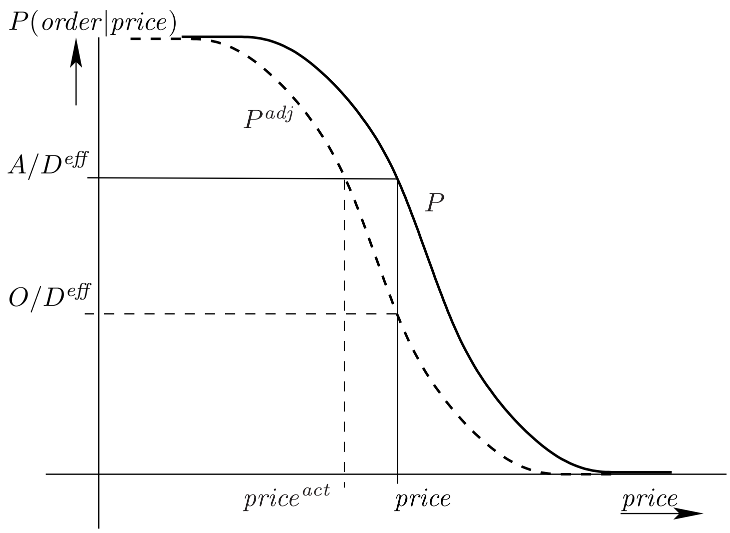


Figure 4.2: Estimating actual market price, given order volume O and an estimate of the order probability function P.

The resulting estimate $price^{act}_{d-1,g}$ is subject to the randomness of the market, and therefore we use an exponentially smoothed offset to produce a refined value of $price^{est}_{d,g}$ each day, as

$$price^{est}_{d,g} = price^{pred}_{d,g} + \delta_{d,g} \tag{4.16}$$

where $price^{pred}_{d,g}$ is the *predicted* market price for product g (see Sect. 5.5), and $\delta_{d,g}$ is updated daily as

$$\delta_{d,g} = \alpha\delta_{d-1,g} + (1-\alpha)(price^{act}_{d-1,g} - price_{d-1,g}) \tag{4.17}$$

for some appropriate value of α.

Chapter 5

Economic Regimes

Market conditions change over time, and this should affect the strategy used by an agent in procurement, production planning, resource allocation, and pricing of goods. For example, in order to make the strategic allocation decision described in Section 4.1, the agent needs predictions of the future values of $P(order|price)$.

Economic theory [Kearl, 1989] suggests that economic environments exhibit three dominant market patterns: scarcity, balanced, and oversupply. We define a scarcity condition if there is more customer demand than product supply in the market, a balanced condition if demand is approximately equal to supply, and an oversupply condition if there is less customer demand than product supply in the market. When there is scarcity, prices are higher, so the agent should price more aggressively. In balanced situations, prices are lower and have more spread, so the agent has a range of options for maximizing expected profit. In oversupply situations prices are lower. The agent should primarily control costs, and therefore either do pricing based on costs, or wait for better market conditions.

5.1 Interpretation of order probability

Since the prediction of the probability of receiving a customer order for a given price, $P(order|price)$, is an integral part of any sales agent we need to consider which parameters are available during the game, and decide how to use them. From a sales perspective it would be ideal to know all the offer and order prices and their associate quantities, but as described in Chapter 3, this is not the case in TAC SCM. We have only limited price visibility, and do not know the associated quantities at all. Everyday the agent receives a price report with the minimum and maximum product order prices from the previous day. Since the agent has only access to the accepted offer prices we need to produce a formulation of the probability of order that only includes those, and not the rejected offers.

We define RFQ_i^j as the number of RFQs per computer type i over a particular time period j. What is not observable during the TAC SCM game is the level of supply $Offers_i^j$, i.e., the number of all the offers made per computer type i during period j. Typically $Offers_i^j > RFQ_i^j$. In each time period j the following condition between RFQ_i^j and $Offers_i^j$ holds

$$Offers_i^j = E\left(\beta_i^j\right) \times RFQ_i^j \tag{5.1}$$

$E\left(\beta_i^j\right)$ represents the expected mean number of bids per computer type i over the time period j. If we assume that all six agents always bids on every RFQ then $\beta = 6$; on the other hand, if all the agents are equally likely to bid or not to bid then $\beta = 3$.

However, if we assume that bidding is stable, i.e., the expectation on the number of bids can be computed in an unbiased manner, then for every price the number of accepted bids affects the probabilities at higher prices in a similar manner. This implies that probabilities are scaled up in a similar fashion. Consider the relationship between the actual order probability $P(S_i^j|price)$ which includes rejected bids and the cumulative demand order probability $P(D_i^j|price)$ which does not consider rejected bids. We can calculate $P(D_i^j|price)$ as we will show in Equation 5.33. However the question remains on what is the relationship between $P(D_i^j|price)$ and $P(S_i^j|price)$ and how to use $P(D_i^j|price)$ for making optimal pricing decisions.

Let's consider the relationship between the two measures first. Let $A_i^j(price)$ be the number of accepted offers for computer type i in time period j at or below *price*, and X_i^j the total number of accepted offers for computer type i in the same time period j. Then we get

$$P(D_i^j|price) = \frac{A_i^j(price)}{X_i^j} \tag{5.2}$$

By definition the total number of bids for computer type i in time period j equals

$\beta_i^j \times X_i^j$. Therefore

$$P(S_i^j|price) = \frac{A_i^j(price)}{\beta_i^j \times X_i^j} \quad (5.3)$$

$$= \frac{P(D_i^j|price)}{\beta} \quad (5.4)$$

Equation 5.1 implies that for every sale there were on average β offers made. Therefore the actual order probability is simply scaled by β as compared to our estimated cumulative demand order probability $P(D_i^j|price)$, as Equation 5.3 indicates. From Equation 5.3, we can derive

$$\frac{P(S_i^j|price_1)}{P(S_i^j|price_2)} = \frac{P(D_i^j|price_1)}{P(D_i^j|price_2)} \quad (5.5)$$

i.e., the relative order probability at two prices is the same. Therefore, if we want to calculate the relative likelihood of a sale at two different prices then both measures can be used to arrive at the same optimal price choice through revenue maximization and a mathematical programming model. Therefore we can base our order probability curves only on accepted offers. This of course assumes that the TAC SCM environment is dynamically stable. Based on observed game data we have evidence that this assumption holds.

Figure 5.1 shows sample curves for $P(order|price)$, the probability of receiving an order for a given offer price. The shape of the curve and its position changes over time. According to economic theory, consistent high prices correspond to a situation of scarcity, where price elasticity is small, while a less steep slope corresponds to a balanced market where the range of prices is larger.

We believe that even though the market is constantly changing, there are some underlying dominant patterns that characterize the aforementioned market conditions. We define a specific mode a market can be in as a *regime*. A way of solving the decision problem an agent is faced with is to characterize those regimes and to apply specific decision making methods to each regime. This requires the agent to have methods for figuring out what is the current regime and for predicting which future regimes to expect in its planning horizon.

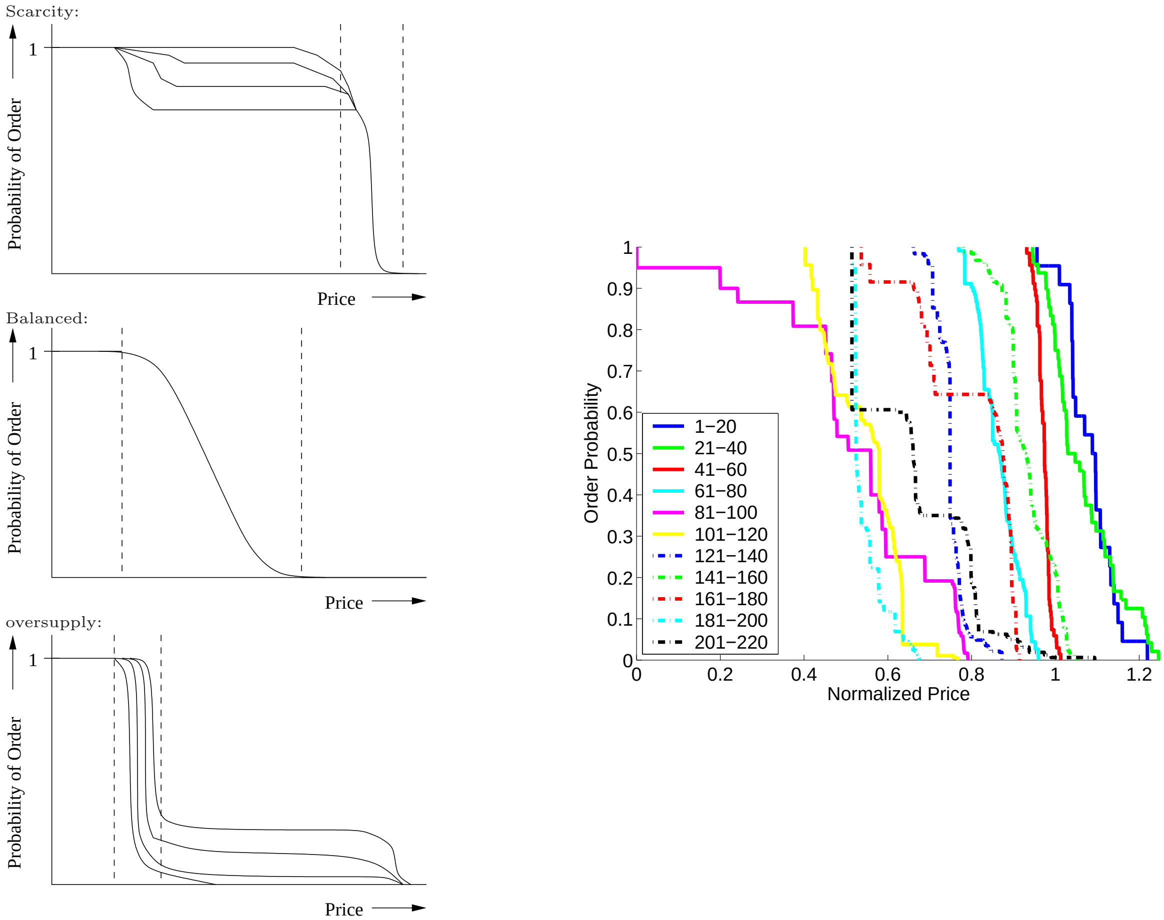


Figure 5.1: The reverse cumulative density function represents the probability of order. Typical order probability curves during scarcity (left top), balanced (left middle) and oversupply (left bottom) regimes and experimental order probability curves (right).

5.2 Learning from historical sales data

The first phase in our approach is to identify and characterize market regimes by analyzing data from past sales. The assumption we make is that enough historical data are available for the analysis and that historical data are sufficiently representative of possible market conditions. Information observable in real-time in the market is then used to identify the current regime and to forecast regime transitions.

Since prices are likely to have different ranges for different goods, we normalize them.

We call np_g the normalized price for good g and define it as follows:

$$\mathrm{np_g} \quad = \quad \frac{price_g}{AssemblyCost_g + \sum_{j=1}^{numParts} NominalPartCost_{g,j}} \tag{5.6}$$

where $NominalPartCost_{g,j}$ is the nominal cost of the j-th part for good g, $numParts$ is the number of parts needed to make the good g, and $AssemblyCost_g$ is the cost of manufacturing the good g. An advantage of using normalized prices is that we can easily compare price patterns across different goods. In the following we just use np, since we normalized goods across one market.

Historical data are used to estimate the price density, $p(\mathrm{np})$, and to characterize regimes. We estimate the price density function by fitting a Gaussian mixture model (GMM)([Titterington *et al.*, 1985]) to historical normalized price, np, data.

We present results using a GMM with fixed means, μ_i, and fixed variances, σ_i, since we want one set of Gaussians to work for all games off-line and online. We use the Expectation-Maximization (EM) Algorithm ([Dempster *et al.*, 1977]) to determine the prior probability, $P(\zeta_i)$, of the Gaussians components of the GMM. The means, μ_i, are uniformly distributed and the variances, σ_i^2, tile the space. Specifically variances were chosen so that adjacent Gaussians are two standard deviations apart.

The density of the normalized price can be written as:

$$p(\mathrm{np}) = \sum_{i=1}^{N} p(\mathrm{np}|\zeta_i)\, P(\zeta_i) \tag{5.7}$$

where $p(\mathrm{np}|\zeta_i)$ is the i-th Gaussian from the GMM, i.e.,

$$p(\mathrm{np}|\zeta_i) = p(\mathrm{np}|\mu_i, \sigma_i) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{\left[\frac{-(\mathrm{np}-\mu_i)^2}{2\times\sigma_i^2}\right]} \tag{5.8}$$

where μ_i is the mean and σ_i is the standard deviation of the i-th Gaussian from the GMM. An example of a GMM is shown in Figure 5.2.2.

5.2.1 Reasons for selecting a Gaussian Mixture Model

We use a GMM since it is able to approximate arbitrary density functions. Another advantage is that the GMM is a semi-parametric approach which can be computed fast and uses less memory than other approaches. The GMM is one way of modeling a probability density function $p(x)$, given a finite number of data points X^n, n = 1,...,N drawn from that density.

We decided to advocate this model over other possible models for various reasons. First, there are three alternatives to approach the density estimation problem: parametric, non-parametric and semi-parametric. In parametric methods one assumes a specific functional form for the density model. Its parameters are then optimized by fitting the model to the data set, e.g. functional form: normal or Gaussian distribution and parameters: the mean μ and standard deviation σ of that distribution. The drawback is that the functional form might not be consistent with the data and may result in unsatisfactory estimation.

In non-parametric estimation methods the form of the density is determined entirely by the data, i.e. no particular functional form is assumed, e.g., histograms, kernel-based methods, K-nearest-neighbors and Parzen window [Bishop, 1995, Duda *et al.*, 2000, Nabney, 2001]. The drawback is that huge data sets are needed for good models and that parameter tuning is critical for performance of the method.

In semi-parametric estimation methods a general class of functional forms is allowed and the number of adaptive parameters can be adapted in a systematic way allowing even more flexible models, e.g. more hidden units, multi-layer perceptrons, radial basis functions and Gaussian mixture models. The advantage of this method is that it combines the best characteristics of parametric and non-parametric methods. This is especially important in electronic marketplaces where the demand might be low for a long period, but then rapidly change to high. The complexity of the model only increases with the total number of parameters in the model, and not simply with the size of the data set.

The demand characteristics in electronic marketplaces have been found to be fractal, that is the short-term demand pattern has much larger variation and mean than the long-term time-averaged demand pattern [Gupta *et al.*, 1997]. This means that while there are periods of no or little demand there will be periods when demand will be extremely high. The pricing strategy of an agent needs to take this into account. Traditionally parameterized econometric models perform extremely poor in these situations. On the contrary, non-parametric approaches do an excellent job in estimation, but these methods are usually computationally too expensive. In the

TAC SCM domain we need to make decisions fast and do not have time to make time consuming calculations. Therefore, we decided to adopt a semi-parametric approach, and in particular the GMM.

5.2.2 Determination of optimal number of Gaussians for a GMM

We developed an algorithm (see Appendix C) to find the optimal number of Gaussians in a GMM and applied it to the training data. The algorithm iterates over 1 to N Gaussian components and for each set of Gaussians it fits a GMM to all collected historical normalized prices, np, of the training set. New normalized price samples are generated from each fitted GMM model via Monte-Carlo sampling, with the number of new samples matching the original data size. Price histograms are generated using the same bins for original and sampled data, and are compared with the help of the KL-divergence ([Kullback and Leibler, 1951, Kullback, 1959])[1]. For each set of Gaussians we iterate the resampling and the KL-divergence steps. Finally we calculate the mean KL-divergence of all sets of Gaussians. The set with the minimum mean KL-divergence is the set that most closely reproduces the original distribution.

The results of the optimization algorithm are shown in Figure 5.3, which shows the mean KL-divergence of 10 fits for 4 to 25 Gaussians and their corresponding standard deviations (left).

Figure 5.2.2 shows that the price density function, $p(\text{np})$, (right) estimated by the GMM with 16 components fits well the historical normalized price data (left y-axis represents good quantity) for a sample market. The optimal number of Gaussians for this sample market is 24.

While the optimization algorithm suggests a choice of 24, the number of Gaussians', should reflect a balance between accuracy and computational overhead. We consider mean prediction accuracy instead of accuracy of fit. Creating a model with a very good fit to the observed data does not usually translate well into predictions. If the model has too many degrees of freedom there is high likelihood of overfitting([Mitchell, 1997], [Russell and Norvig, 2002]). Therefore, we decided to take $N = 16$ Gaussians to avoid overfitting the training data and better able to adapt to unseen instances.

[1] With the KL-divergence we are able to measure the closeness of two distributions. If the two distributions are completely the same, then the KL-divergence is zero. A deeper discussion about the KL-divergence can be found in Section 6.

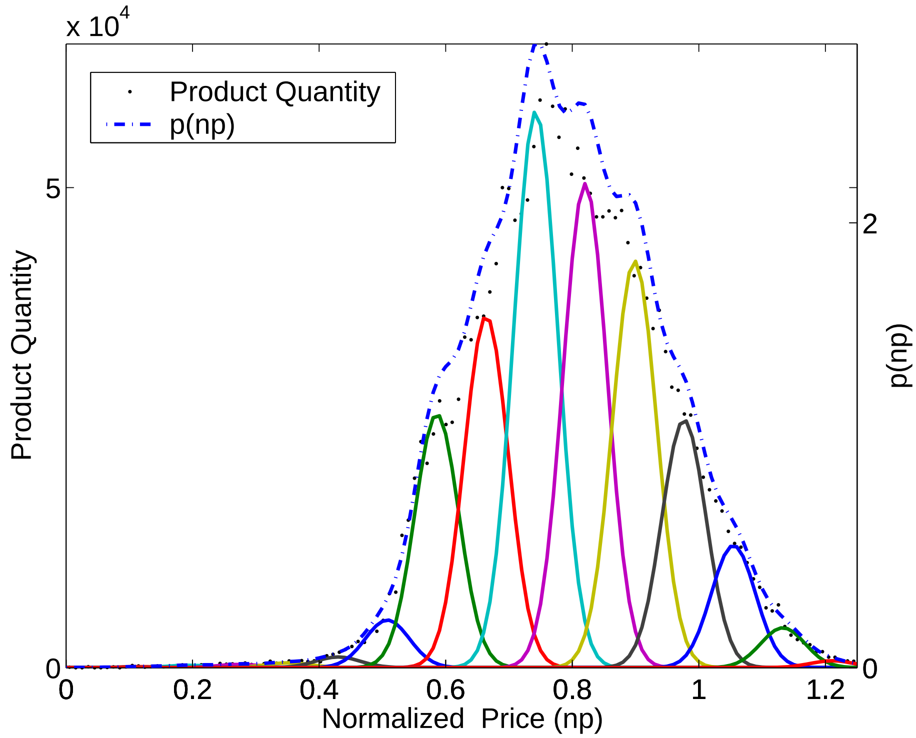


Figure 5.2: The price density density function, $p(np)$, (right y-axis) estimated by the GMM with 16 components fits well the historical normalized price data (left y-axis represents good quantity) for the medium market segment. Data are from 18 games from semi-finals and finals of TAC SCM 2005.

For $N = 16$ Gaussians the KL-value is around 0.01, which is close enough to have a very good fit to the actual data. A similar approach was used by [Zhang and Cheung, 2005] and [Beygelzimer and Rish, 2003] to select an appropriate model that reduces the computational complexity of graphical models in the medical domain.

5.2.3 Regime definition

Next, coming back to our regime method, we apply Bayes' rule to determine the posterior probability:

$$P(\zeta_i|\mathrm{np}) = \frac{p(\mathrm{np}|\zeta_i)\,P(\zeta_i)}{\sum_{i=1}^{N} p(\mathrm{np}|\zeta_i)\,P(\zeta_i)} \qquad \forall i = 1, \cdots, N \tag{5.9}$$

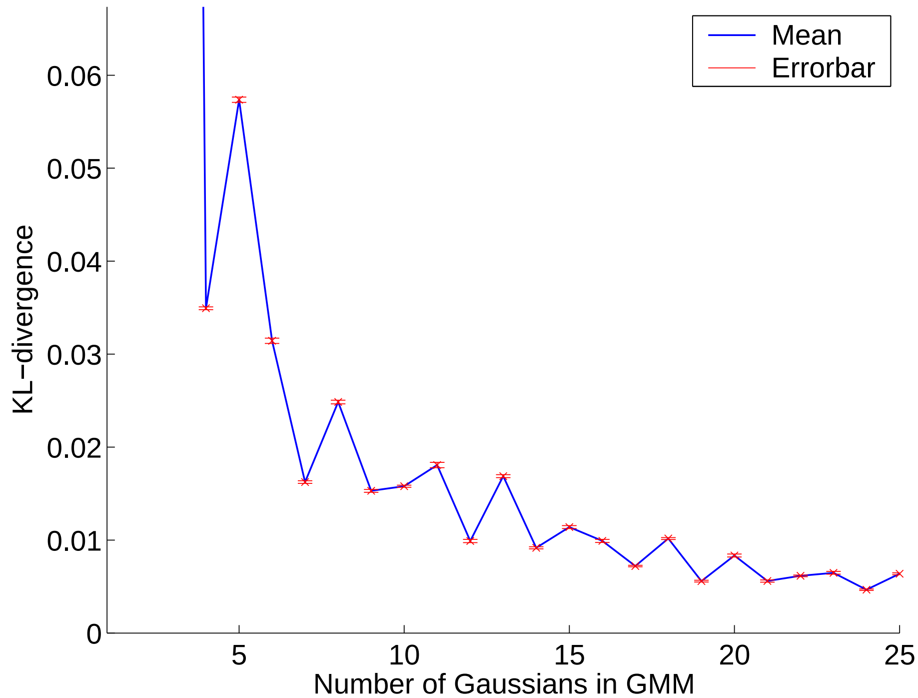


Figure 5.3: Mean KL-divergence of 10 fits for 4 to 25 Gaussians and their corresponding standard deviations. The KL-divergence values for one to three Gaussians are not displayed since the values are much larger and would make it impossible to see the KL-divergence values of the other Gaussians. The mean KL-divergence value for one Gaussian equals 2.64, for two Gaussians equals 0.58, and for three Gaussians equals 0.44. Data are from 18 games from semi-finals and finals of TAC SCM 2005.

We then define the posterior probabilities of all Gaussians' given a normalized price, np, as the following N-dimensional vector:

$$\vec{\eta}(\text{np}) = [P(\zeta_1|\text{np}), P(\zeta_2|\text{np}), \ldots, P(\zeta_N|\text{np})]. \tag{5.10}$$

For each normalized price np_j we compute the vector of the posterior normalized price probabilities, $\vec{\eta}(\text{np}_j)$, which is $\vec{\eta}$ evaluated at each observed normalized price np_j.

The intuitive idea of a regime as a recurrent economic condition is captured by discovering price distributions that recur across days. We define *regimes* by clustering price distributions across days. This is done with the k-means algorithm, using a similarity measure on the probability vectors $\vec{\eta}(\text{np}_j)$ and normalized prices np. The clusters found by this method correspond to frequently occurring price distributions with support on contiguous range of np. We have found that sometimes data points corresponding to specific regimes are close in probability space, but not in price space. Specifically it can happen that one regime dominates the extreme low and the extreme high price range, with different regimes in between. This regime is more difficult to interpret in terms of market concepts like oversupply or scarcity. To circumvent this problem we perform clustering in an augmented space formed by appending a rescaled version of np to the probability vector. Specifically, the mean of np is subtracted and np is scaled so that its standard deviation matches the largest standard deviation of the probability vectors.

The center of each cluster (ignoring the last component which contains the rescaled price information) is a probability vector that corresponds to regime $r = R_k$ for $k = 1, \cdots, M$, where M is the number of regimes. We selected a priori the number of regimes, after examining the data and looking at economic analyses of market situations. In our experiments we found out that the number of regimes chosen does not significantly affect the results on price trend predictions. We tried computing the GMM and k-means clustering with different initial conditions, but consistently converged to the same results.

Collecting these vectors into a matrix yields the conditional probability matrix $\mathbf{P}(c|r)$. The matrix has N rows, one for each component of the GMM, and M columns, one for each regime.

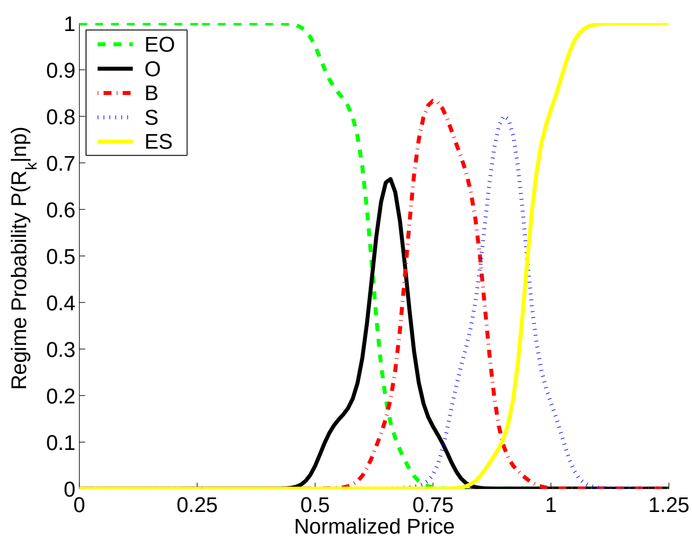


Figure 5.4: Learned regime probabilities, $P(R_k|\text{np})$, over normalized price np, for a sample market after training.

In Figure 5.4 we distinguish five regimes, which we can call extreme oversupply (R_1 or EO), oversupply (R_2 or O), balanced (R_3 or B), scarcity (R_4 or S), and extreme scarcity (R_5 or ES). We decided to use five regimes instead of the three basic regimes which are suggested by economic theory because in this way we are able to isolate outlier regimes, such as extreme oversupply and extreme scarcity. Regimes R_1 and R_2 represent a situation where there is a glut in the market, i.e. an oversupply situation, which depresses prices. Regimes R_3 represents a balanced market situation, where most of the demand is satisfied. In regime R_3 the agent has a range of options of price vs sales volume. Regimes R_4 and R_5 represent a situation where there is scarcity of goods in the market, which increases prices. In this case the agent should price as close as possible to the estimated maximum price a customer is willing to pay.

We marginalize the density of the normalized price, np, given the i-th Gaussian of the GMM, $p(\mathrm{np}|\zeta_i)$, and the conditional probability clustering matrix, $P(\zeta_i|R_k)$, over all Gaussians ζ_i. We obtain the density of the normalized price np dependent on the regime R_k:

$$p(\mathrm{np}|R_k) = \sum_{i=1}^{N} p(\mathrm{np}|\zeta_i)\, P(\zeta_i|R_k). \tag{5.11}$$

The probability of regime R_k dependent on the normalized price np can be computed using Bayes rule as:

$$P(R_k|\mathrm{np}) = \frac{p(\mathrm{np}|R_k)\, P(R_k)}{\sum_{k=1}^{M} p(\mathrm{np}|R_k)\, P(R_k)} \quad \forall k = 1, \cdots, M. \tag{5.12}$$

where M is the number of regimes. The prior probabilities, $P(R_k)$, of the different regimes are determined by a counting process over past data. Figure 5.4 depicts the regime probabilities for a sample market in TAC SCM. Each regime is clearly dominant over a range of normalized prices.

The intuition behind regimes is that prices communicate information about future expectations of the market. However, absolute prices do not mean much because the same price point can be achieved in a static mode (i.e., when prices don't change), when prices are increasing, or when prices are decreasing. In the construction of a regime the variation in prices (the nature, variance, and the neighborhood) are considered thereby providing a better assessment of market conditions.

5.3 Real time regime identification

Every day the agent receives a report which includes the minimum and maximum prices of the computers sold the day before, but not the quantities sold. The mid-range price, $\overline{\text{np}}$, i.e. the price midway between the minimum and maximum, can be used to approximate the mean price. However, this is not always an accurate estimate of the mean price, because of local fluctuations in minimum and maximum prices. and because the distribution of prices is not known, only the minimum and maximum price.

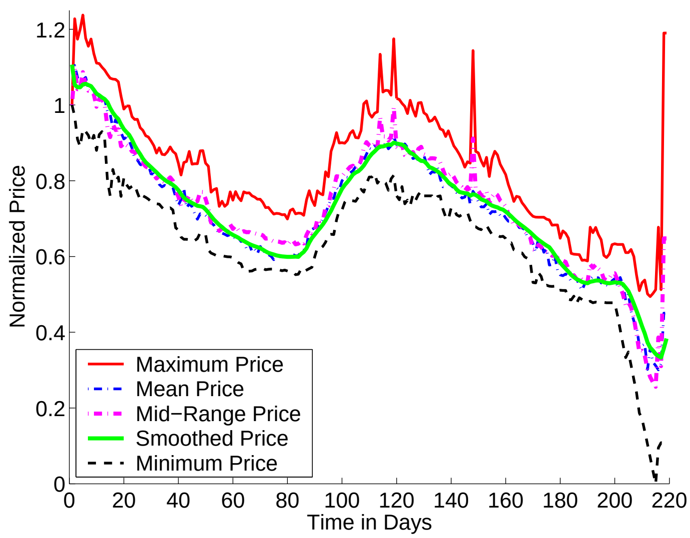


Figure 5.5: Minimum, maximum, mean, mid-range, and smoothed mid-range daily normalized prices of computers sold, as reported during the game every day for the medium market segment in the 3721@tac3, one of the final games. The mean price is computed after the game using the game data, which include complete information on all the transactions.

An example which shows how the mid-range value differs from the mean value is in Figure 5.5. The mean value was computed after the game, when the entire game data are available. In this example, especially on day 110, 120, 140 and at the end, we observe a high spike in the maximum price. This was caused by an opportunistic agent who discovered a small amount of unsatisfied demand, but most of that day's orders were sold at a much lower price. To lower the impact of sudden price changes we implemented a Brown linear (i.e. double) exponential smoother ([Brown *et al.*, 1961]) with $\alpha = 0.5$. The general form of this smoother is:

$$\widetilde{\mathrm{np}}'_{d-1} = \alpha \cdot \overline{\mathrm{np}}_{d-1} + (1-\alpha) \cdot \widetilde{\mathrm{np}}'_{d-2} \tag{5.13}$$

$$\widetilde{\mathrm{np}}''_{d-1} = \alpha \cdot \widetilde{\mathrm{np}}'_{d-1} + (1-\alpha) \cdot \widetilde{\mathrm{np}}''_{d-2} \tag{5.14}$$

where $\widetilde{\mathrm{np}}'$ denotes the singly-smoothed mid-range normalized price series obtained by applying simple exponential smoothing to mid-range normalized price $\overline{\mathrm{np}}$, and $\widetilde{\mathrm{np}}''$ denotes the doubly-smoothed normalized price series obtained by applying simple exponential smoothing (using the same α) to $\widetilde{\mathrm{np}}'$. We model yesterday's smoothed price estimate $\widetilde{\mathrm{np}}_{d-1}$ is given by:

$$\widetilde{\mathrm{np}}_{d-1} = 2 \cdot \widetilde{\mathrm{np}}'_{d-1} - \widetilde{\mathrm{np}}''_{d-1} \tag{5.15}$$

Finally, since we only have the minimum and maximum prices from the previous day and not the mean, we model $\widetilde{\mathrm{np}}_{d-1}$ as follows:

$$\widetilde{\mathrm{np}}_{d-1} = \frac{\widetilde{\mathrm{np}}^{min}_{d-1} + \widetilde{\mathrm{np}}^{max}_{d-1}}{2} \tag{5.16}$$

where $\widetilde{\mathrm{np}}^{min}_{d-1}$ is the exponentially smoothed minimum normalized price and $\widetilde{\mathrm{np}}^{max}_{d-1}$ is the exponentially smoothed maximum normalized price from the previous day. This results in a better approximation of the real mean price than smoothing only the mid-range price from the previous day. Figure 5.5 shows that the smoothed mid-range price, $\widetilde{\mathrm{np}}$, is closer to the mean price.

During the game, on day d the agent estimates the current regime by calculating the smoothed mid-range normalized price $\widetilde{\mathrm{np}}_{d-1}$ for the previous day (recall that the agent every day receives the prices for the previous day) and by selecting the regime which has the highest probability, i.e.

$$\hat{R}_{max_1} \; s.t. \; max_1 = \operatorname*{argmax}_{1 \le k \le M} \vec{P}(\hat{R}_k | \widetilde{\mathrm{np}}_{d-1}). \tag{5.17}$$

Figure 5.6 shows how to use the smoothed mid-range price to identify the corresponding regime probabilities online over the course of a game. The graph shows that different regimes are dominant at different points in the game, and that there are brief intervals during which two regimes are almost equally likely. An agent could use this information to decide which strategy, or mixture of strategies, to follow.

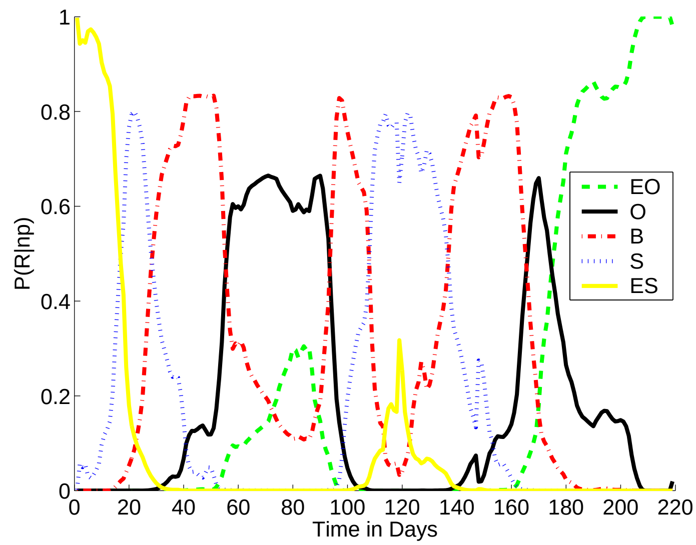


Figure 5.6: Regime probabilities computed online every day for the medium market segment in game 3721@tac3.

The smoothed mid-range price can be used to identify the corresponding regime in real time, as shown in Figure 5.7 (right). The data are from game 3721@tac3, which was not in the training set of games used to develop the regime definitions. The top left, middle left, and bottom left parts of Figure 5.7 show respectively the probability of receiving an order in an extreme scarcity, balanced and in an extreme over-supply situation for different prices. Scarcity typically occurs early in the game and at other times when supply is low. These probabilities are computed from past game data for each regime.

A measure of the confidence in the regime identification is the entropy of the set S of probabilities of the regimes given the normalized mid-range price from the daily price reports $\tilde{\mathrm{np}}_{day}$, where

$$S = \{P(R_1|\widetilde{\mathrm{np}}_d), \cdots, P(R_M|\widetilde{\mathrm{np}}_d)\} \tag{5.18}$$

and

$$\mathrm{Entropy}(S) \equiv \sum_{k=1}^{M} -P(R_k|\widetilde{\mathrm{np}}_d) \log_2 P(R_k|\widetilde{\mathrm{np}}_d) \tag{5.19}$$

An entropy value close to zero corresponds to a high confidence in the current regime

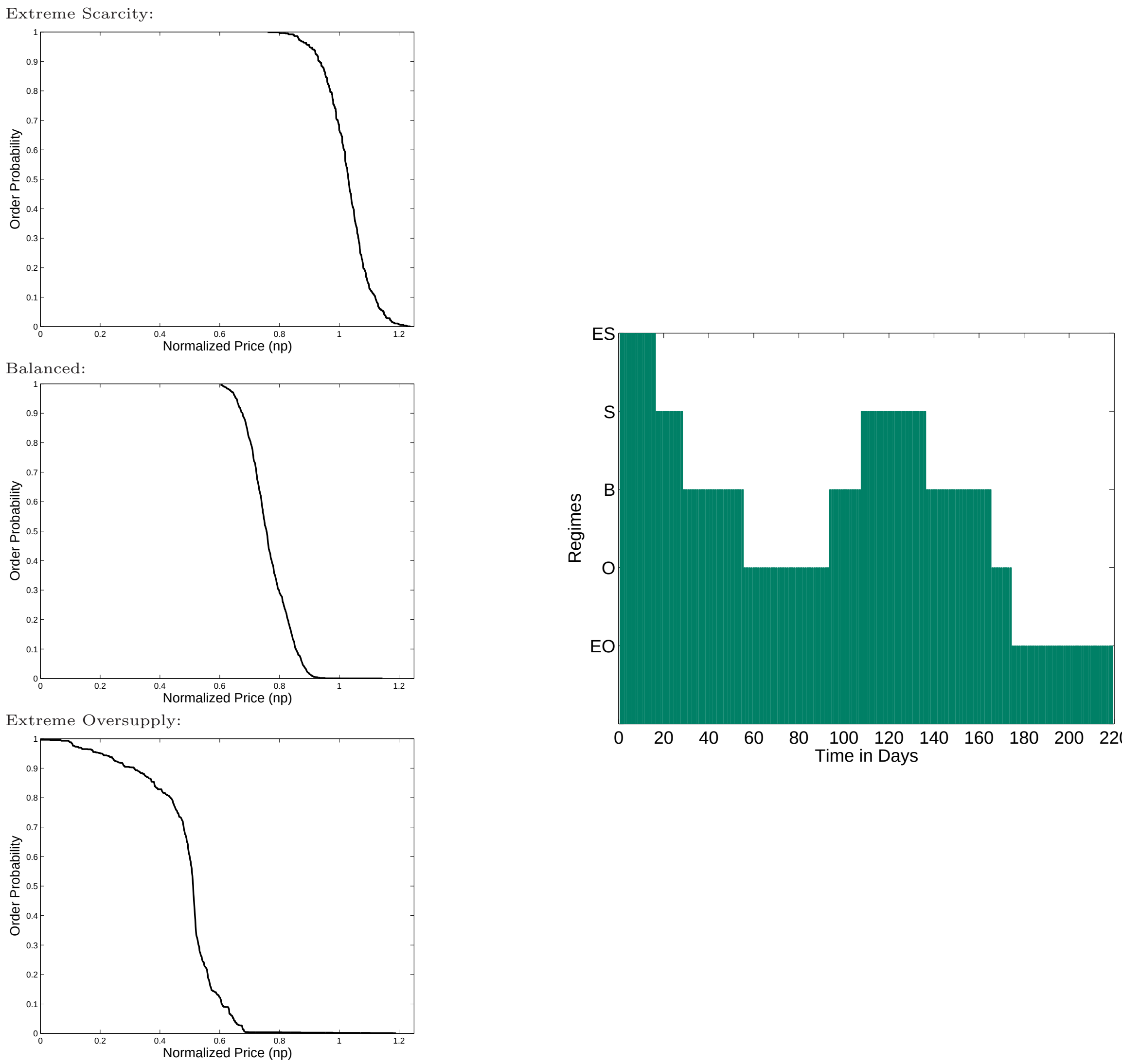


Figure 5.7: Game 3721@tac3 (Final TAC SCM 2005) – Regimes over time for the medium market computed online every day (right), probability of receiving an order by normalized price for an extreme scarcity situation (R_5 indicated by ES) (top left), for a balanced situation (R_3 indicated by B) (middle left) and for an extreme oversupply situation (R_1 indicated by EO) (bottom left).

and an entropy value close to its maximum, i.e. for M regimes $\log_2 M$, indicates that the current market situation is a mixture of M almost equally likely regimes. An example for the medium market segment in game 3721@tac3 is shown in Figure 5.8. In Section 6.1.2 we develop an alternative measure for regime confidence, and a more precise measure for regime uncertainty.

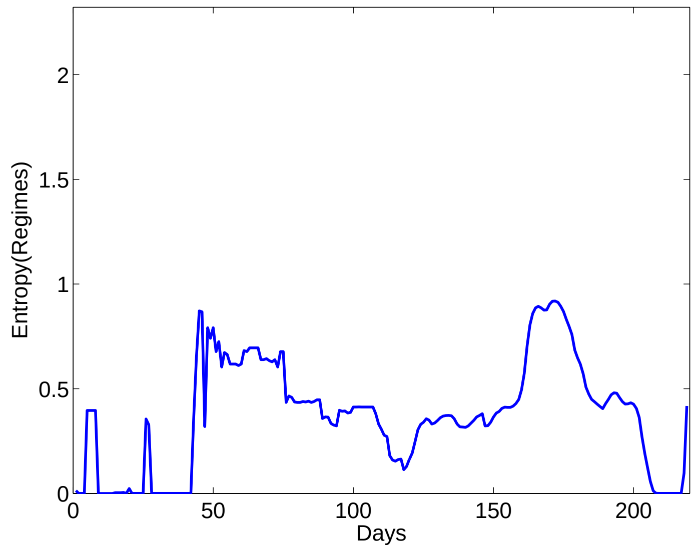


Figure 5.8: Daily entropy values of the five regimes for the medium market segment in game 3721@tac3. Notice how the entropy values match the regime probabilities shown in Figure 5.6.

Figure 5.9 (left) shows the percentage of the factory utilization (FU), the ratio of offer to demand, which represents the portion of the market demand that is satisfied, and the normalized price (np) over time. On the right side we display the quantity of the unsold finished goods inventory (FG) instead of factory utilization[2]. The regimes identified by our approach are superimposed, where ES (or R_5) represents extreme scarcity, S (or R_4) scarcity, B (or R_3) balanced, O (or R_2) oversupply, and EO (or R_1) extreme oversupply. These factors clearly correlate with market regimes, but they are not directly visible to the agent during the game. For example, the figure shows that when the offer to demand ratio is high (i.e. oversupply) prices are low and vice versa. We can observe that the ratio of offer to demand changes significantly during the game. For instance, on day 111 the ratio of offer to demand is 1.95 and prices are high. On day 208 the ratio of offer to demand is much higher, 5.38, and prices are lower. We can also observe that prices tend to lag changes in ratio of offer to demand.

We have reported results on correlation between regimes and market parameters in [Ketter *et al.*, 2005].

[2] The quantity of the finished goods inventory is affected by other factors, such as storage cost, which have changed in the TAC SCM 2005 games. In 2005 and 2006 games agents tend to build to order and keep most of their inventory in the form of parts, not finished products.

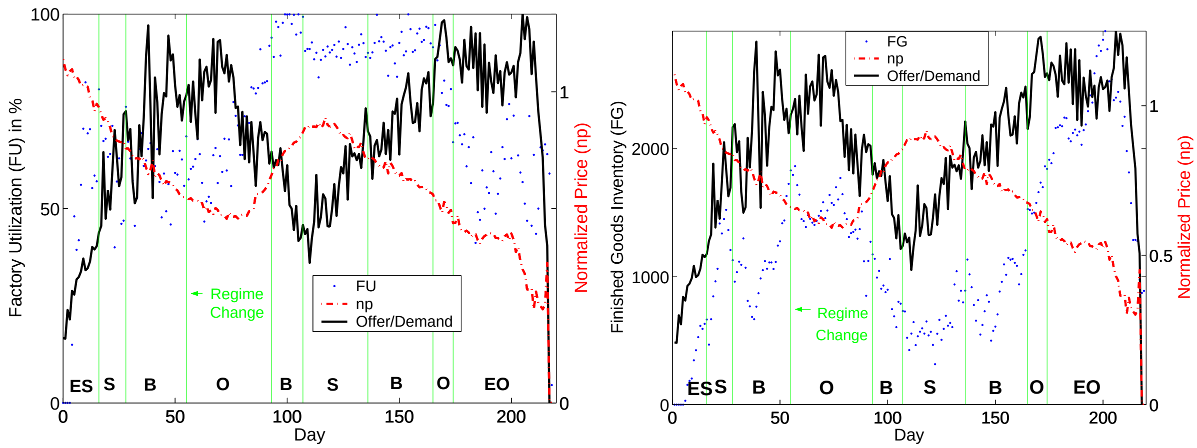


Figure 5.9: Game 3721@tac3 (Final TAC SCM 2005) – Relationships between regimes and normalized prices in the medium market. On the left axis, we show in the left figure the daily factory utilization and in the right figure the available finished goods inventory for all agents. In both figures we also display on the left axis the ratio of offer to demand (which ranges from 0 to 5.38), which is scaled to fit between the minimum and maximum values of the left axis. On the right axis we show the normalized prices. The dominant regimes are labeled along the bottom.

5.4 Regime prediction

Since the behavior of the agent should depend not just on the current regime but also on expected future regimes, the agent needs to predict future regimes. In this Section we describe three different methods to generate regime predictions. The first a Markov prediction process, the second is a Markov correction-prediction process, and the last is based on exponential smoothing. Each of these methods has other characteristics and should be used for different purposes. The exponential smoother regime prediction process is an ideal candidate to estimate the current regime distribution, since it is more reactive to the current market condition than any other method. The Markov prediction process is a good choice for short-term and mid-term predictions, and the Markov correction-prediction process is suited more for long-term predictions.

5.4.1 Markov prediction

We model the prediction of short-term future regimes for tactical decision making as a Markov prediction process [Isaacson and Madsen, 1976]. The prediction is based on the last price measurement.

We construct a Markov transition matrix, $\mathbf{T}(r_{d+n}|r_d)$, off-line by a counting process over past games. This matrix represents the posterior probability of transitioning on day $d+n$ to regime r_{d+n} given the current regime on day d, r_d. We distinguish between two types of Markov prediction: (1) interval and (2) repeated 1-day prediction. The interval prediction is based on training a separate Markov transition matrix for each day in the planning horizon h, i.e. $\mathbf{T_n}(r_{d+n}|r_d)$, where $\forall n = 1, \cdots, h$. The same type of prediction can be done by repeating a single day prediction matrix, $\mathbf{T_1^h}(r_{d+1}|r_d)$, rather than training an interval prediction matrix.

The prediction of the posterior distribution of regimes n days into the future, $\vec{P}(\hat{r}_{d+n}|\{\widetilde{\mathrm{np}}_{d-1}\})$, is done recursively as follows:

1. **Interval prediction:**

$$\vec{P}(\hat{r}_{d+h}|\{\widetilde{\mathrm{np}}_{d-1}\}) = \sum_{r_{d+h}} \cdots \sum_{r_{d-1}} \left\{ \vec{P}(\hat{r}_{d-1}|\{\widetilde{\mathrm{np}}_{d-1}\}) \cdot \mathbf{T}_n(r_{d+n}|r_{d+n-1}) \right\} \tag{5.20}$$

2. **Repeated 1-day prediction:**

$$\vec{P}(\hat{r}_{d+h}|\{\widetilde{\mathrm{np}}_{d-1}) = \sum_{r_{d+h}} \cdots \sum_{r_{d-1}} \left\{ \vec{P}(\hat{r}_{d-1}|\{\widetilde{\mathrm{np}}_{d-1}) \cdot \mathbf{T_1^{h+1}}(r_d|r_{d-1}) \right\}, \tag{5.21}$$

where

$$\mathbf{T_1^{h+1}}(r_d|r_{d-1}) = \prod_{n=0}^{h} \mathbf{T_1}(r_d|r_{d-1}) \tag{5.22}$$

5.4.2 Markov correction-prediction

We model the prediction of long-term future regimes for strategic decision making as a Markov correction-prediction process, where the prediction part is similar to the Markov prediction described above but taking the entire price history into account.

The method is based on two distinct operations:

1. a **correction** (recursive Bayesian update) of the posterior probabilities for the regimes based on the history of measurements of the smoothed mid-range normalized price $\widetilde{\mathrm{np}}$ obtained since the time of the first measurement until the previous day, $d-1$. We use $\vec{P}(\hat{r}_{d-1}|\{\widetilde{\mathrm{np}}_1, \ldots, \widetilde{\mathrm{np}}_{d-1}\})$, to indicate a vector of the posterior probabilities of all the regimes on day $d-1$.

2. a **prediction** of regime posterior probabilities for the current day, d. The prediction of the posterior distribution of regimes n days into the future, $\vec{P}(\hat{r}_{d+n}|\{\widetilde{\text{np}}_1, \ldots, \widetilde{\text{np}}_{d-1}\})$, is done recursively as follows:

   (a) **Interval prediction:**

$$\begin{aligned}&\vec{P}(r_{d+h}|\{\widetilde{\text{np}}_1, \ldots, \widetilde{\text{np}}_{d-1}\})\\ &\quad = \sum_{r_{d+h}} \cdots \sum_{r_{d-1}} \left\{ \vec{P}(r_{d-1}|\{\widetilde{\text{np}}_1, \ldots, \widetilde{\text{np}}_{d-1}\}) \cdot \mathbf{T_n}(r_{d+n}|r_{d+n-1}) \right\}\end{aligned} \tag{5.23}$$

   (b) Repeated 1-day prediction:

$$\begin{aligned}&\vec{P}(r_{d+h}|\{\widetilde{\text{np}}_1, \ldots, \widetilde{\text{np}}_{d-1}\})\\ &\quad = \sum_{r_{d+n}} \cdots \sum_{r_{d-1}} \left\{ \vec{P}(r_{d-1}|\{\widetilde{\text{np}}_1, \ldots, \widetilde{\text{np}}_{d-1}\}) \cdot \mathbf{T_1^{h+1}}(r_d|r_{d-1}) \right\},\end{aligned} \tag{5.24}$$

where

$$\mathbf{T_1^{h+1}}(r_d|r_{d-1}) = \prod_{n=0}^{h} \mathbf{T_1}(r_d|r_{d-1}) \tag{5.25}$$

On the first day in TAC SCM we set the prior regime probability to 100% extreme scarcity, since all the agents start out with zero inventory on the first day. This has a strong effect on pricing. Whoever has something to sell early on is able to sell it at a high price.

5.4.3 Exponential smoother prediction

As an alternative to the Markov prediction process, we designed a method for regime predictions based on exponentially smoothed price predictions. In Equations 5.13 to 5.16 we describe how to obtain an estimate of the smoothed mid-range price from the previous day, $\widetilde{\text{np}}_{d-1}$. As the first step in predicting prices we calculate the price trend, tr_{d-1}, using the results from Equation 5.13 and Equation 5.14, as:

$$tr_{d-1} = \frac{\alpha}{1-\alpha} \cdot (\widetilde{\text{np}}'_{d-1} - \widetilde{\text{np}}''_{d-1}) \tag{5.26}$$

Since we estimate the smoothed mid-range price using the daily minimum and maximum price, see Equation 5.16, we have to do the same while calculating the trend estimate:

$$\widetilde{tr}_{d-1} = \frac{\widetilde{tr}^{min}_{d-1} + \widetilde{tr}^{max}_{d-1}}{2} \tag{5.27}$$

where $\widetilde{tr}^{min}_{d-1}$ is the exponentially smoothed minimum normalized trend and $\widetilde{tr}^{max}_{d-1}$ is the exponentially smoothed maximum normalized trend from the previous day. Using the trend and yesterday's price estimate, see Equation 5.16, we estimate today's and future daily smoothed prices as:

$$\widetilde{\mathrm{np}}_{d+n} = \widetilde{\mathrm{np}}_{d-1} + (1+n) \cdot \widetilde{tr}_{d-1}, \qquad \forall n = 1, \cdots, h \tag{5.28}$$

We obtain the density of the normalized price, $\widetilde{\mathrm{np}}_{d+n}$, dependent on the regime R_k:

$$p(\widetilde{\mathrm{np}}_{d+n}|\hat{R}_k) = \sum_{i=1}^{N} p(\widetilde{\mathrm{np}}_{d+n}|\zeta_i)\, P(\zeta_i|R_k). \tag{5.29}$$

The predicted probability of regime R_k dependent on the predicted exponentially smoothed normalized price n days into the future $\widetilde{\mathrm{np}}_{d+n}$ can be computed using Bayes rule as:

$$P(\hat{R}_k|\widetilde{\mathrm{np}}_{d+n}) = \frac{p(\widetilde{\mathrm{np}}_{d+n}|\hat{R}_k)\, P(R_k)}{\sum_{k=1}^{M} p(\widetilde{\mathrm{np}}_{d+n}|\hat{R}_k)\, P(R_k)} \;\; \forall k = 1, \cdots, M. \tag{5.30}$$

where M is the number of regimes. The prior probabilities, $P(R_k)$, of the different regimes are determined by a counting process over past data after the k-means clustering. We also developed a semi-Markov prediction process. Its description and some results can be found in Appendix D.

Examples of regime predictions for game 3721@tac3 for the medium market segment are shown in Figure 5.10 and Figure 5.11. The figures show the real regimes measured after the game from the game data and the predictions made by our method during the game. As it can be seen in the figures, the match between predictions and real data is very good.

Figure 5.10 shows a predicted change from an oversupply situation to a balanced situation. This means that the agent should sell less today and build up more inventory for future days when prices will be higher. On the other hand we see in Figure 5.11 a predicted change from scarcity to the balanced regime. In this case the agent should try to sell more aggressively the current day, since prices will be decreasing in the

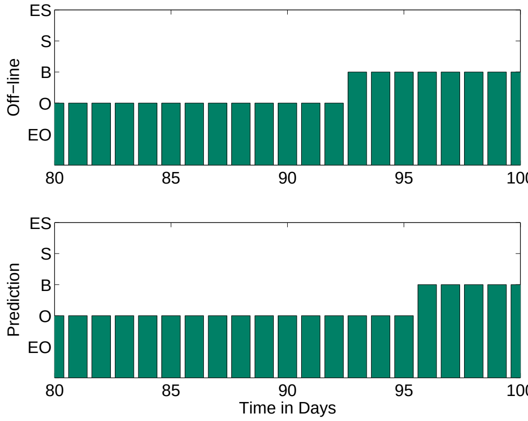


Figure 5.10: Regime predictions for game 3721@tac3 starting on day 80 for 20 days into the future for the medium market segment. Data are shown as computed after the game using the complete set of data, and as predicted by our method during the game.

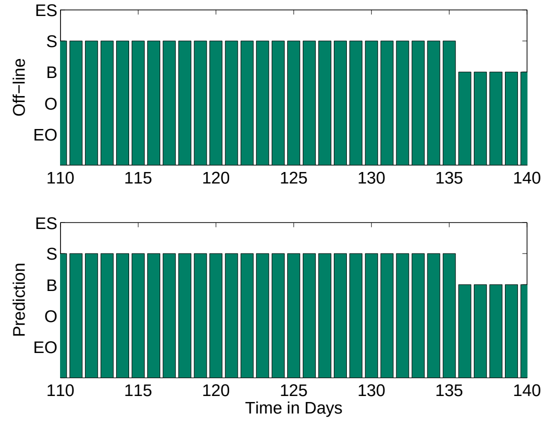


Figure 5.11: Regime predictions for game 3721@tac3 starting on day 110 for 30 days into the future for the medium market segment.

next days.

5.5 Prediction of price distribution and price trend

In this Section we describe a method for generating price trend prediction based on regime prediction. An agent successful at price trend prediction can use this information for guiding its procurement, production, and pricing decisions. Equation 5.31 describes a computation that calculates a price prediction distribution based on a given predicted regime distribution. M represents the number of different regimes and N the number of Gaussians used in the GMM (see Equation 5.7). A point $p(\widehat{\mathrm{np}}_{d+n}|\{\widetilde{\mathrm{np}}_d, \ldots, \widetilde{\mathrm{np}}_{d-1}\})$ on the distribution, given a value for normalized price np, is given by

$$
\begin{aligned}
p(\widehat{\mathrm{np}}_{d+n}|\{\widetilde{\mathrm{np}}_1, \ldots, \widetilde{\mathrm{np}}_{d-1}\}) &= \sum_{i=1}^{M} p(\mathrm{np}|R_i)\, P(\hat{R}_{i,d+n}|\{\widetilde{\mathrm{np}}_1, \ldots, \widetilde{\mathrm{np}}_{d-1}\}) \\
&= \sum_{j=1}^{N} \underbrace{\sum_{i=1}^{M} P(\zeta_j|R_i)\, P(\hat{R}_{i,d+n}|\{\widetilde{\mathrm{np}}_1, \ldots, \widetilde{\mathrm{np}}_{d-1}\})}_{P(\zeta_{j,d+n})}\, p(\mathrm{np}|\zeta_j)
\end{aligned}
$$

$$= \sum_{j=1}^{N} P(\zeta_{j,d+n})\, p(\mathrm{np}|\zeta_j), \qquad \forall n = 1, \cdots, h \tag{5.31}$$

where $P(\hat{R}_{i,d+n}|\{\widetilde{\mathrm{np}}_1, \ldots, \widetilde{\mathrm{np}}_{d-1}\})$ is one element of the predicted regime probability vector given by Equations 5.21, 5.20, 5.24, and 5.23. To obtain a predicted price distribution we sample Equation 5.31 for every day over the planning horizon h with values for np between 0 and 1.25. Examples of price distributions are given in Figure 5.12 (left) and in Figure 5.13 (left).

After sampling the mixture distribution over the set of np values, the distribution is renormalized to sum to one. This discretizes the continuous distribution, which simplifies all subsequent probability calculations. For instance the mean of the distribution can be computed as:

$$E[\widehat{\mathrm{np}}_{d+n}] = \sum_{j=1}^{Ns} p_{norm}\left(\widehat{\mathrm{np}}_{d+n}(j)|\{\widetilde{\mathrm{np}}_1, \ldots, \widetilde{\mathrm{np}}_{d-1}\}\right) \cdot \mathrm{np}(j), \qquad \forall n = 1, \cdots, h \tag{5.32}$$

To predict price trends we use also the 10%, 50%, and 90% percentile of the predicted price distribution, which are interpolated from the discretized cumulative distribution.

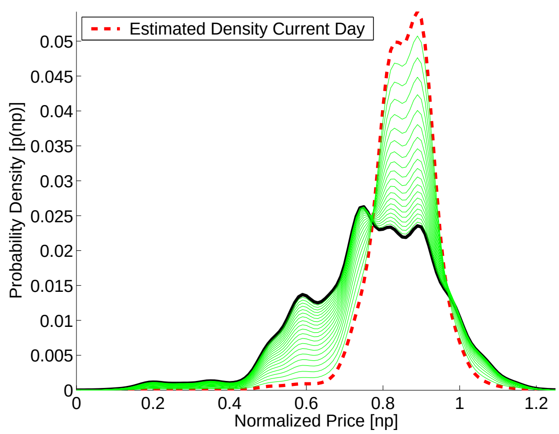
Figure 5.12 (left) shows the forecast price density, based on a 1-day trained Markov matrix, for game 3717@tac3, for 20 days starting at day 115. The dashed curve represents the price density for the first forecast day, the thick solid line shows the price density for the last forecast day, and the thin solid curves show the forecast for the intermediate days. As expected the predicted price density broadens as we forecast further into the future, reflecting a decreasing certainty in the prediction.

We can also compare the actual price trends with our predictions. Figure 5.12 (right) shows the real mean price trend along with forecast price trends based on the different predictors, the expected mean Markov prediction, the 10%, 50% and the 90% Markov density percentiles, and the exponential smoother. All the curves in the figure represent a relative price trend – to better compare the different predictors which each other graphically, we subtracted from each forecast value the first predicted value, so that they all start at zero.

The exponential smoother prediction in this example is not good[3], since the smoother

[3]Usually the exponential smoother predicts much better (see Section 6), but we use this example to explain one of the advantages of our method.

is too myopic, i.e. it puts a weight that is too high on recently observed prices. Figure 5.5 shows that before day 115 the prices were increasing. When performing a prediction the exponential smoother takes the recent slope and extrapolates it into the future. On the contrary, our method learns during training how long, dependent on the preconditions, a particular regime is active.



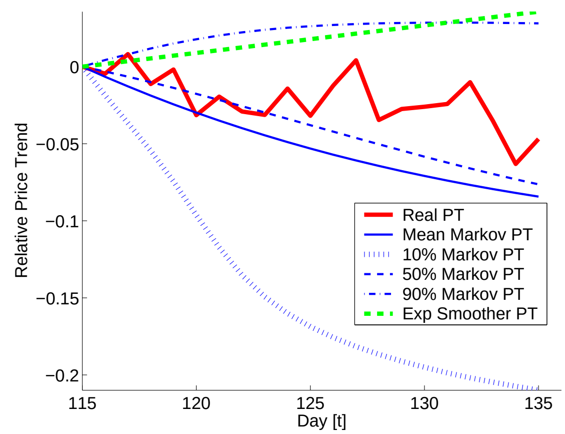


Figure 5.12: Predicted price density (left) and predicted price trend (right) using the repeated 1-day trained Markov matrix for game 3717@tac3 from day 115 until day 135 in the low market segment. In the price density figure (left), the thick dashed curve is the price density estimate for the current day, the thick black solid curve is the price density estimate for the last day in the planning horizon, and the thin solid curves are the estimates for the intermediate days.

Figure 5.13 (left) shows the forecast price density, based on a n-day trained Markov matrix, for game 3717@tac3, for 20 days starting at day 115. We observe that the predicted price density, which is generated using an n-day Markov matrix, broadens much less than the one using a 1-day Markov matrix. This reflects an increasing certainty in the prediction when switching from a 1-day to a n-day Markov matrix. Figure 5.13 (right) shows the appropriate relative price trend. The increased certainty in prediction is also reflected in the limits of the predicted density. The 10% and 90% percentiles forming a good prediction envelope for the tracked price.

5.6 Prediction of order probability

Mathematically speaking the curves in Figure 5.1 represent different order probability distribution functions $P(order|\text{np})$. Because np represents normalized order prices, the cumulative density function $CDF(\text{np})$ describes the proportion of orders that will

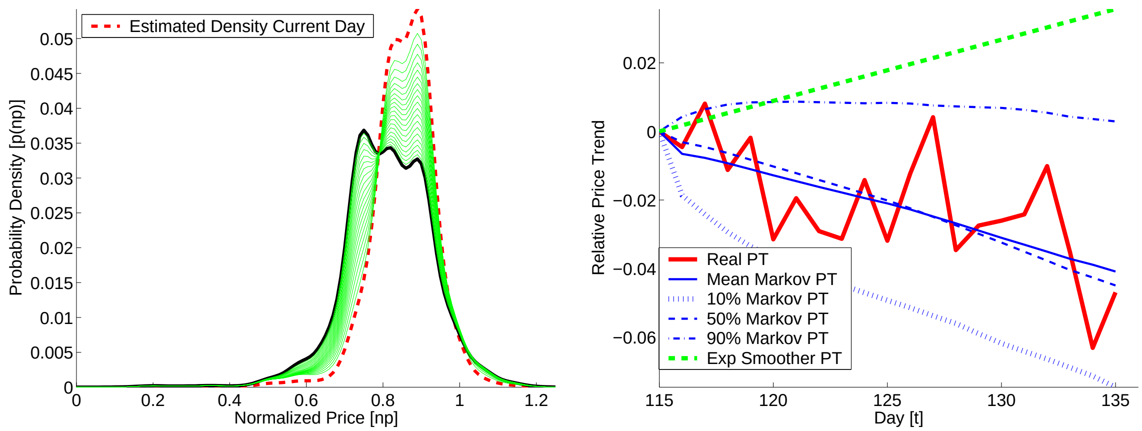


Figure 5.13: Predicted price density (left) and predicted price trend (right) using n-day trained Markov matrix for game 3717@tac3 from day 115 until day 135 in the low market segment.

be placed at or below a value of normalized price np.

$$P(order|\mathrm{np}) = 1 - CDF(\mathrm{np}) \tag{5.33}$$

Where the CDF is related to a probability density function $p(\mathrm{np})$ (see Equation 5.31 and left side of Figures 5.12, 5.13) by

$$CDF(\mathrm{np}) = \int_0^{\mathrm{np}} p(\mathrm{np}')\,d\mathrm{np}' \tag{5.34}$$

in the TAC SCM case $\mathrm{np}_{max} = 1.25$, so that $CDF(\mathrm{np}_{max}) = 1$.

As an example, in Figure 5.14 we see the curve for $P(order|\mathrm{np}) = 1 - CDF(\mathrm{np})$ corresponding to the estimated current day density shown as the dashed curve in Figures 5.12 and 5.13 (left side).

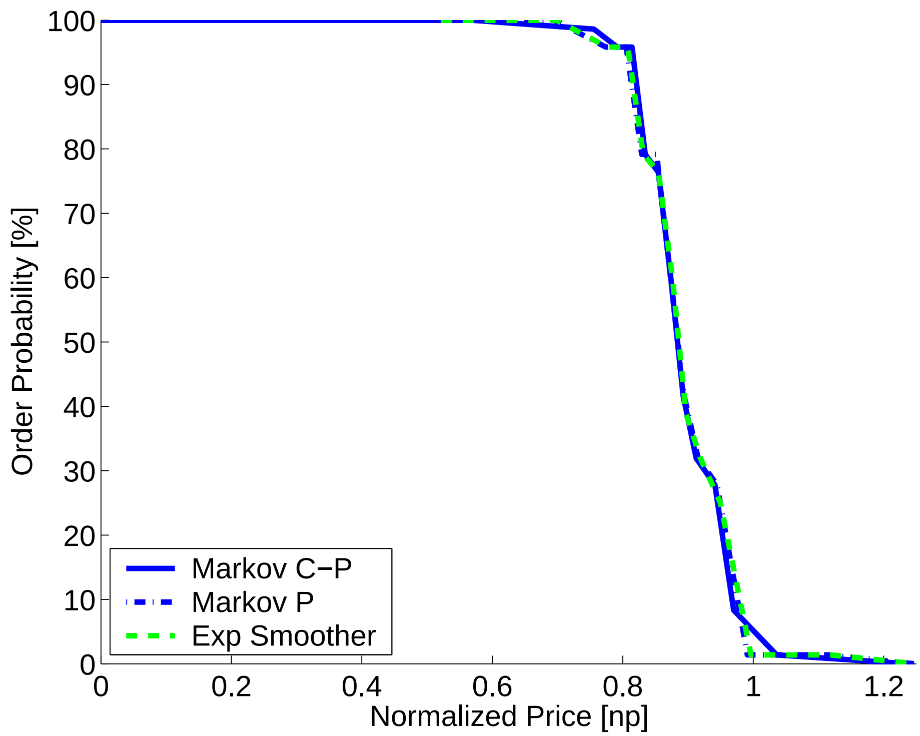


Figure 5.14: Real-time order probability curve, $P(order|\mathrm{np}) = 1 - CDF(\mathrm{np})$, for day 115 for the low market segment in game 3717@tac3.

Chapter 6

Performance Evaluation

A critical element of a trading agent is the determination of the current market prices, probability of order, prediction of prices, price trends, resource allocation, and setting of sales prices. We have presented methods in Chapters 4 and 5 to facilitate tactical and strategic decision via economic regimes. In this Chapter we focus on the evaluation of those methods.

We begin by presenting compelling off-line results in the TAC SCM domain. Since all our explanations so far were based on market segments we continue this in our off-line experiments. We selected market segments over products because the movements of markets, e.g. the movement of the low end computers vs high end computers, are more intuitive than individual products, and a company can make decisions on an entire market instead of individual products. In the second part of this Chapter, we will cover the real-time experiments. We have implemented our methods in a real agent, MinneTAC. Since the supply for individual products in one market segment may vary, we implemented our methods for this set of experiments at the product level. This makes the agent more reactive to the supplier market. Finally, we present an application of economic regimes to financial markets and we are also able to show compelling results in this domain.

6.1 TAC SCM - Off-line

Our method is useful to the extent that it characterizes and predicts real qualities of the market. There are many hidden variables in a competitive market, such as the inventory positions and procurement arrangements of the competitors. Our method uses observable historical and current data to guide tactical and strategic decision processes. In this Section we evaluate the practical value of regime identification, prediction and the estimation of order probability in TAC SCM.

6.1.1 Experimental setup

For our experiments, we used data from a set of 28 games (18 for training[1] and 10 for testing[2]) played during the semi-finals and finals of TAC SCM 2005. The mix of players changed from game to game, the total number of players was 12 in the semi-finals and 6 in the finals. Since supply and demand in TAC SCM change in each of the market segments (low, medium, and high) independently of the other segments, our method is applied to each individual market segment. Each type of computer has a nominal cost, which is the sum of the nominal cost of each of the parts needed to build it. In TAC SCM the cost of the facility is sunk, and there is no per-unit assembly cost. We normalize the prices across the different computer types in each market segment.

6.1.2 Identification of regime uncertainty

An automated agent can use the entire daily estimated regime distribution as input to other algorithms to facilitate decision making. On the contrary a human decision maker might just want to have an estimate of the daily dominant regime to get a quick and intuitive estimate of the current market condition.

For each training game we calculate everyday the 1-norm between the estimated daily online regime probability and the actual off-line regime probability. After all games are processed we compute the RMS error between online and the off-line regime probabilities over 2% probability bins from 0% to 100%. The off-line regime probability is calculated based on the actual price on a given day, where the estimated online regime probabilities are based on price estimates and Markov matrices. During training we used a Markov prediction, a Markov correction-prediction, and an exponential

[1] 3694@tac3, 3700@tac3, 4229@tac4, 4234@tac4, 7815@tac5, 7821@tac5, 5638@tac6, 5639@tac6, 3719@tac3, 3720@tac3, 3721@tac3, 3722@tac3, 3723@tac3, 4255@tac4,4256@tac4, 4257@tac4, 4258@tac4, 4259@tac4 – To obtain the complete path name append .sics.se to each game number.

[2] 3697tac3, 4235tac4, 7820tac5, 5641tac6, 3717tac3, 3718tac3, 3724tac3, 4253tac4, 4254tac4, 4260tac4

smoother process to calibrate different online regime probability estimates.

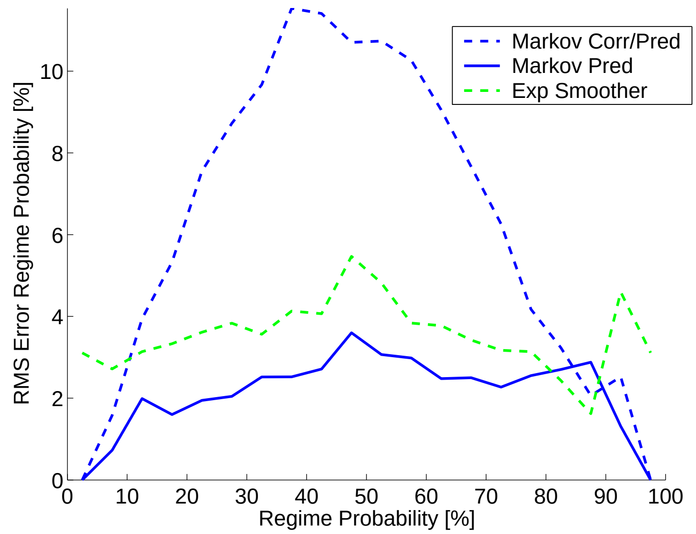


Figure 6.1: RMS Error for identification of daily regime probabilities based on the training data.

Figure 6.1 presents the RMS error in the daily regime probability identification for the three predictors. From the Figure we can determine that the Markov prediction process has the lowest error in identifying the current regime probabilities followed by the exponential smoother process and last the Markov correction-prediction process. This results make sense, since the Markov prediction process is myopic and therefore relies strongly on the current market situation, tactical decision, whereas the Markov correction-prediction process is designed for long-term, strategic decision.

Our online approach to signal regime uncertainty is based on using the learned look-up table for each identified regime probability to determine the appropriate RMS regime probability error. We subtract the error, $Err(P(\hat{R}_{max_1}))$, from the probability of $\hat{R}_{max_1}$, which is the regime with the highest probability, and add the error, $Err(P(\hat{R}_{max_2}))$, to the probability of $\hat{R}_{max_2}$, the regime with the 2nd highest probability, and take the difference between them. If the difference is positive, which means that the error regions don't cross, than there is no uncertainty in the regime identification, otherwise there is as expressed. Please see the following Equation 6.1

for details:

$$RU = \begin{cases} 0 & if \quad \{P(\hat{R}_{max_1}) - Err(P(\hat{R}_{max_1}))\} \\ & \quad -\{P(\hat{R}_{max_2})) + Err(P(\hat{R}_{max_2}))\} \leq 0 \\ \frac{P(\hat{R}_{max_2})}{P(\hat{R}_{max_1})} & otherwise \end{cases} \tag{6.1}$$

In addition we have defined regime confidence as:

$$RC = \frac{P(\hat{R}_{max_1})}{P(\hat{R}_{max_2})} \tag{6.2}$$

The top row of Figure 6.2 shows regime confidence, RC, results and the bottom row shows regime uncertainty, RU, results for game 3717tac3 for the low market segment. The left two quadrants of Figure 6.2 were generated using a Markov prediction process and the two quadrants to the right using an exponential smoother process. We observe that in testing, as we did in training, on a daily basis the Markov prediction process has less uncertainty than the exponential smoother process in identifying the current dominant regime.

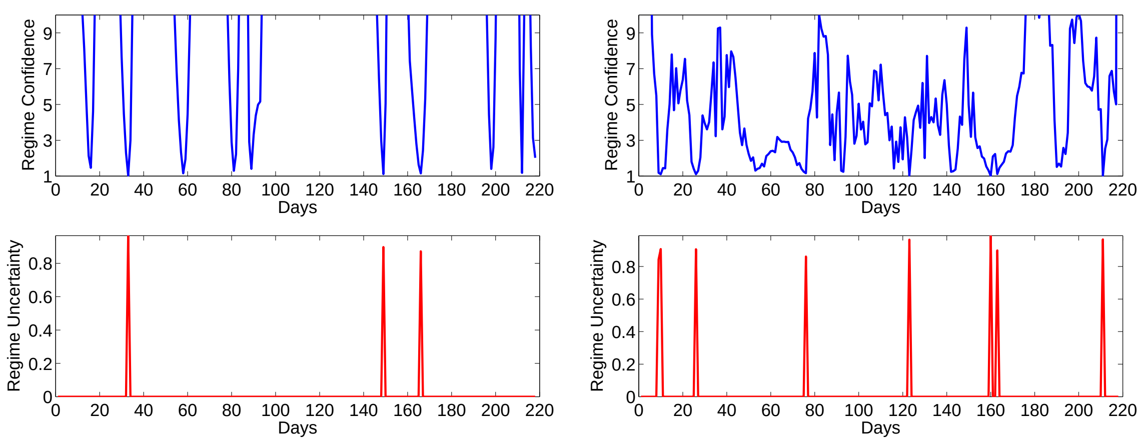


Figure 6.2: Game 3717@tac3 - low market segment: Uncertainty in the identification of the daily dominant regime based on Markov prediction process (left) vs exponential smoother regime prediction (right).

6.1.3 Relationship between identified regime and market variables

We expect identified regimes to qualitatively represent the status of important hidden market factors. A correlation analysis of market parameters of the training set is shown in Figure 6.3. The p-values for the correlation analysis are all less than 0.01.

Regime EO (extreme oversupply) correlates positively with quantity of finished goods inventory, negatively with percent of factory utilization, positively with the ratio of offer to demand, and negatively with normalized price. On the other hand, in Regime ES (extreme scarcity) we observe a negative correlation with the amount of unsold finished goods inventory, with the percent of factory utilization, and the ratio of offer to demand, and positively with normalized price.

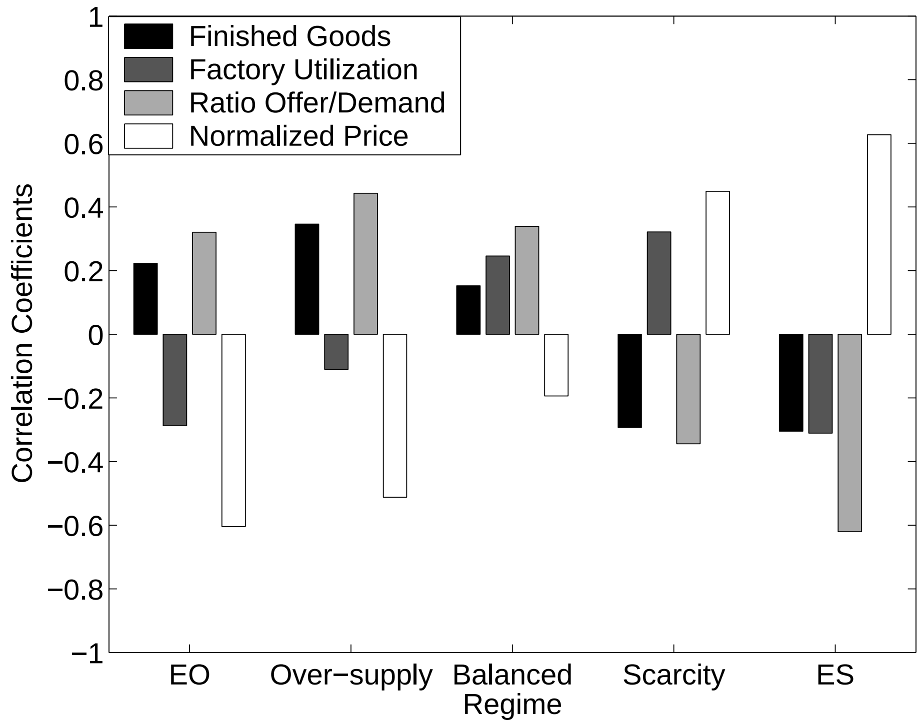


Figure 6.3: Training set (18 games) – Correlation coefficients between regimes and quantity of finished goods inventory, factory utilization, the ratio of offer to demand, and normalized price (np) in the medium market segment. All values are significant at the $p = 0.01$ level.

An advantage of using five regimes instead of three regimes is that we gain two degrees of freedom. This enables better decision making, by isolating the outliers in the market. For example, regime EO (extreme oversupply) is different from Regime O (oversupply) since it presents a potential price war situation. Another difference between regime EO and regime O is that regime EO is universally unprofitable and that regime O is marginally profitable for most agents. Regimes B and S are universally profitable and in regime ES some agents have left the market. The major difference between the scarcity regime, S, and the extreme scarcity regime, ES, is that in regime S the factory runs at full capacity, because of excess demand, while in regime ES we observe a scarcity of parts, which results in underutilization of the production capacity.

Another way to evaluate the quality of regime identification is given by an interpretation of the k-means clustering algorithm. Essentially, it finds points along the path that connects the regime centers in the regime probability space. In Figure 6.4 we represent the results of the k-means clustering algorithm, or the learned regime probabilities. For ease of visualization we use only three regimes to explain the learned behavior; the five regime case produces similar results, but they are harder to visualize. We can see that the learned regime probabilities in the posterior probability space connect the regimes in the "expected" way. In other words, we do not see points directly between scarcity and oversupply; instead, the path leads from scarcity through balance to oversupply.

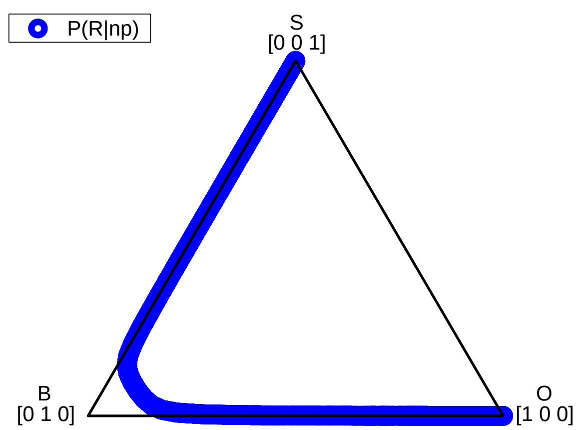


Figure 6.4: An example of learned regime probabilities, $P(R_k|\text{np})$, for the medium market segment in TAC SCM after training.

We expect a dynamic regime prediction algorithm to move along this path of learned regime probabilities.

6.1.4 Prediction of dominant regime

We measure the accuracy of regime prediction by counting how many times the regime predicted is the correct one. As ground truth we measure the number of regime switches and when they happen off-line using data from the game. Starting with day 1 until day 179, we forecast every day the regimes for the next 40 days and we forecast when a regime transition would occur. Experimental results for discrete regime change prediction are shown in Figure 6.5 using a 1-day (left) vs a n-day (right) Markov matrix. We observe that when using a n-day as opposed a 1-day Markov matrix, the prediction accuracy increases on average about 10% in all market segments starting from 15 days into the future.

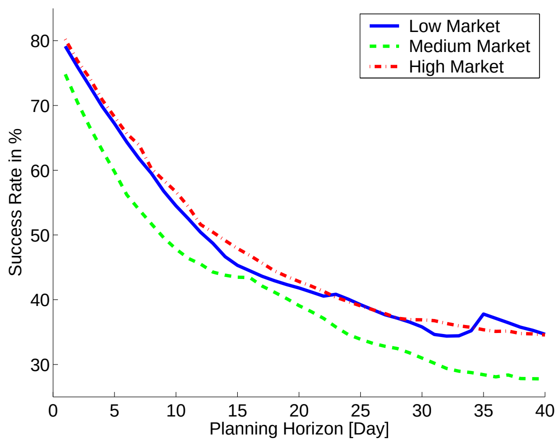


Figure 6.5: Success rate of correct regime shift prediction using a 1-day (left) vs a n-day (right) Markov matrix.

6.1.5 Prediction of regime distribution

The above results are based on discrete regimes, i.e., using only the dominant regime of each predicted day to the actual value of any given day. One measure which can be used in determining the closeness of all individual predicted regime probabilities to the actual ones is called the *Kullback-Leibler (KL) divergence* [Kullback and Leibler, 1951, Kullback, 1959]. This is a quantity which measures the difference between two probability distributions in bits, meaning the smaller the measure the closer the predictions are to optimal. We can calculate the Kullback-Leibler divergence, $KL(\vec{P}(\hat{R})\|\vec{P}(R))$ as:

$$KL(\vec{P}(\hat{R})\|\vec{P}(R)) = \sum_{r\in\mathcal{R}} \vec{P}(\hat{R}) \log\left(\frac{\vec{P}(\hat{R})}{\vec{P}(R)}\right) \tag{6.3}$$

The KL difference can be interpreted in terms of how much additional data is needed to achieve optimal prediction performance. The precision of this data is given by the number of bits in the KL-divergence measure. For example a 1 bit difference would require an additional binary piece of information [Shannon, 1948], like: "Were yesterday's bids all satisfied?" If the difference between the two distributions is 0 than the predictions are optimal in sense that the predicted and actual distributions match.

If the time-dependent distribution of a Markov process, in our case $\vec{P}(\hat{R})$, converges to a limit

$$\vec{\Pi} = \lim_{m \to \infty} \{\vec{P}(\hat{R})\}^m \tag{6.4}$$

then Π is called the stationary distribution. When the stationary distribution exists it is characterized by the fix-point equation

$$\vec{\Pi} = \vec{\Pi} \cdot \mathbf{T_n} \tag{6.5}$$

There are several ways to compute the stationary distribution, Π, but all involve solving the eigenvalue problem specified in the above equation. The reason for introducing the n-day Markov matrix is that we hypothesized that it takes longer to reach the stationary distribution of its Markov process than opposed a 1-day Markov matrix, and therefore deliver a better prediction performance. We prove this hypothesis empirically by calculating the stationary distribution for the 1-day and each n-day Markov matrix and compare it with Markov predicted regime distribution. For this we utilize again the KL-divergence measure:

$$KL(\vec{P}(\hat{R}) \| \vec{\Pi}) = \sum_{r \in \mathcal{R}} \vec{P}(\hat{R}) \log \left(\frac{\vec{P}(\hat{R})}{\vec{\Pi}} \right) \tag{6.6}$$

In Figure 6.6 we show prediction results in terms of KL-divergence for a GMM with 16 components using a 1-day Markov matrix (left) vs using a n-day Markov matrix (right). The KL-divergences are computed using 5 regimes for the low market segment over the testing set. Points represent the KL-divergences between the Markov predicted regime distribution and the actual distribution, $KL(\vec{P}(\hat{R})_{Markov} \| \vec{P}(R))$. Diamonds represent the KL-divergences between the double exponentially smoothed predicted distribution and the actual distribution $KL(\vec{P}(\hat{R})_{ExpS} \| \vec{P}(R))$, and pluses represent the KL-divergences between the Markov predicted regime distribution, and the stationary distribution $KL(\vec{P}(\hat{R})_{Markov} \| \vec{\Pi})$ over the planning horizon.

Our predictions differ between 0.28 bits (current day), 0.80 bits (20 days), and 0.95 bits (40 days) of information when using the repeated 1-day Markov matrix, and

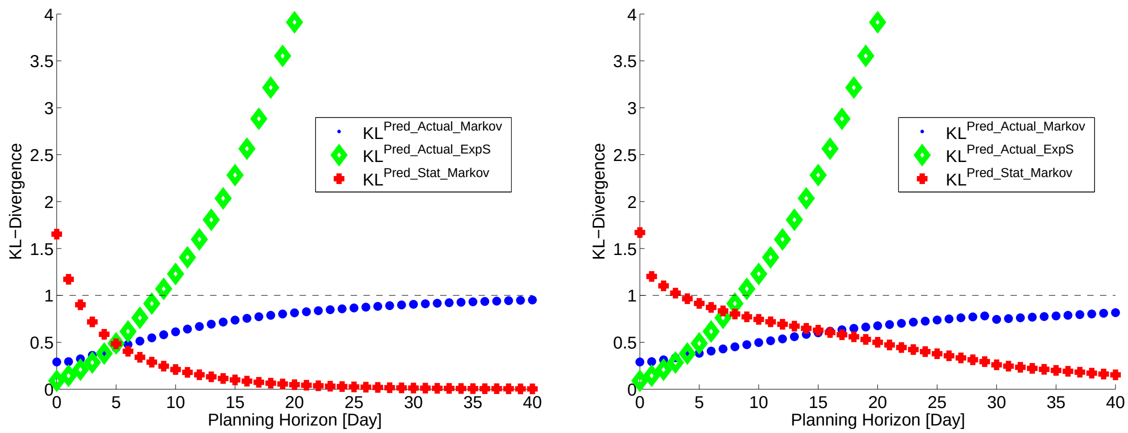


Figure 6.6: KL-divergence between predicted, actual, and stationary regime distribution using a repeated 1-day (right) vs n-day (left) Markov matrix. KL-divergence between the Markov predicted regime distribution and the actual distribution (point), the double exponentially smoothed predicted distribution and the actual distribution (diamond), and the Markov predicted regime distribution,and the stationary distribution (plus) over the planning horizon. KL-divergences computed using five regimes for the low market segment over the testing set. The left figure is generated using a repeated 1-day trained Markov matrix and the right figure with n-day trained Markov matrices.

between 0.28 bits (current day), 0.66 bits (20 days), and 0.81 bits (40 days) of information when using the n-day interval Markov matrix, as opposed to the Exponential Smoother predictions which vary between 0.09 bits (current day), 3.55 bits (20 days), and 12.62 bits (40 days). It is typically acceptable having a KL-divergence less than or close to one [Zhang and Cheung, 2005]. There will not be significant gains by obtaining more information in the estimation procedure. We only show values of KL-divergences up to 4, since we want to highlight the initial exponential smoother and the Markov predictions. The current day exponential smoother predictions are approximately 1.14 times better as the repeated 1-day and n-day interval Markov predictions. On the other hand the 20 and 40 days exponential smoother predictions are approximately 6.73 and 3259 times worse than the repeated 1-day Markov predictions and 7.42 and 3591 times worse than the n-day interval Markov predictions.

The KL-divergences calculated using the n-day interval Markov matrix are always smaller than the one using a repeated 1-day Markov matrix, especially in the long-term. This indicates a better fit between the predicted and the actual regime probabilities for the n-day interval Markov matrix and as a consequence should be used for strategic decision making as opposed to the predictions generated by the repeated 1-day Markov matrix. The best estimate for the short-term (current day up to 4 days into the future) is given by the exponential smoother and as a result should be used as an input to generate the price densities for the short-term (explained in Section 5.5) and sales offer prices for the current day, i.e. for tactical decision making.

6.1.6 Price distribution

Every day we forecast the price density for the next n days into the future, where $p(\widehat{\mathrm{np}}_d)$[3] is the predicted price density for the current day, and $p(\widehat{\mathrm{np}}_{d+n})$ is the predicted price density on the n-day into the future of the planning horizon h. In our experiments we choose $h = 40$. We calculated the expected mean price (see Equation 5.32), and track different areas (10%, 50%, and 90%) of the price density curve. We furthermore calculated the expected mean price using exponential smoother regime and density prediction and the pure exponential smoother price prediction. We calculate the root mean square error, $RMSE(\widehat{\mathrm{np}}_n, \mathrm{np}_n)$, between the predicted normalized price, $\widehat{\mathrm{np}}_n$, and the actual normalized price, np_n, over a prediction interval, n, between the current day and the end of the planning horizon, h, averaged across days and games, to determine the accuracy of the price prediction as:

$$RMSE(\widehat{\mathrm{np}}_n, \mathrm{np}_n) = \sqrt{\frac{\sum_{\gamma=1}^{N_G} \sum_{d=1}^{N_D-n} (\widehat{\mathrm{np}}_d^{n,\gamma} - \mathrm{np}_d^{n,\gamma})^2}{N_G \cdot (N_D - n)}}, \qquad \forall n = 1, \cdots, h \tag{6.7}$$

where N_D is the length of a TAC SCM game in days and N_G is the number of test games.

Figure 6.7 shows the RMS error of the Markov predictors using a repeated 1-day matrix (left) vs an interval matrix (right) and compares it to the RMS error of the price generated by exponential smoother regime lookup and to the pure implementation of the exponential smoother. An RMS error of 0.05 corresponds to an average prediction error of 4% and an RMS error of 0.25 corresponds to an average prediction error of 20%. We observe that when switching from a repeated 1-day to a n-day interval Markov matrix the overall price prediction using the density improves, especially at the limits of the distribution. Looking at the n-day interval Markov matrix

[3]For simplicity we leave out the dependence on historical normalized prices.

results shows that on the current day the pure exponential smoother is doing best when compared to the real mean price and on the second day the pure exponential smoother and the price prediction using regime lookup via the exponential smoother is the best. Since the differences in terms of price prediction between the pure exponential smoother and the regime look-up via exponential smoother are so small we think it is best to go with the predicted price density generated via exponential smoothed regime look-up, since you have the whole price density available to make an informed decision instead only having the mean price. From day 3 until the end of the planning horizon the expected mean and median Markov price predictions are the best and about the same in quality, followed by the 10% and 90% Markov price predictions, followed by the predicted price using regime lookup via the exponential smoother and last on all the days is the price prediction by the pure implementation of the exponential smoother.

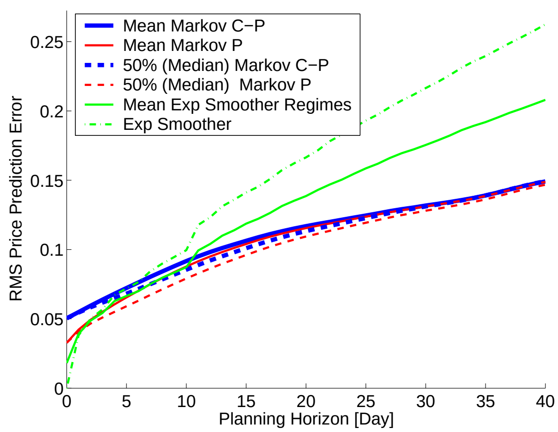


Figure 6.7: RMS price prediction error based on a repeated 1-day (left) vs n-day interval (right) trained Markov matrix.

We tested whether the Markov correction-prediction predicted normalized prices, $\widehat{\mathrm{np}}_n$, were different from the actual observed prices, np_n, for any day, n, in the planning horizon, h, using a pairwise T-test and failed to reject the equality of price hypothesis at $p = 0.05$ significance level. This indicates that our predicted prices, $\widehat{\mathrm{np}}_n$, are statistically following the real prices np_n.

6.1.7 Price trend

To obtain the price trend, for every day, n, over the planning horizon, h, we take the forecast from the different predictors and apply it to compute:

$$\widehat{Tr}_n = sgn(\widehat{\text{np}}_{d+n} - \widehat{\text{np}}_d), \qquad \forall n = 1, \cdots, h \tag{6.8}$$

where $\widehat{Tr}_n$ represents the price trend for n days into the future, $\widehat{\text{np}}_d$ is the predicted price for the current day, and $\widehat{\text{np}}_{d+n}$ is the predicted price (a point on the predicted price distribution, e.g. in our examples 10%, 50%, and 90%) n number of days into the future. Each day the agent has access only to the minimum and maximum prices of the previous day, so it needs a one day forecast to estimate the price for the current day. If $\widehat{Tr}_n$ is positive, then the predicted prices are increasing, and if $\widehat{Tr}_n$ is negative, then predicted prices are decreasing. Otherwise we predict prices will remain stable at the current level.

Figure 6.8 displays the success rate of price trend sign prediction using a repeated 1-day Markov matrix (left) and a n-day interval Markov matrix (right) over the planning horizon. Since the price trend is used for tactical decision making, we calculate the success rate only after five days in the future. We observe that the n-day interval Markov correction-prediction process forms a nice line along the 70% success rate and outperforms the Markov prediction process in terms of predicting correctly the sign of the price trend.

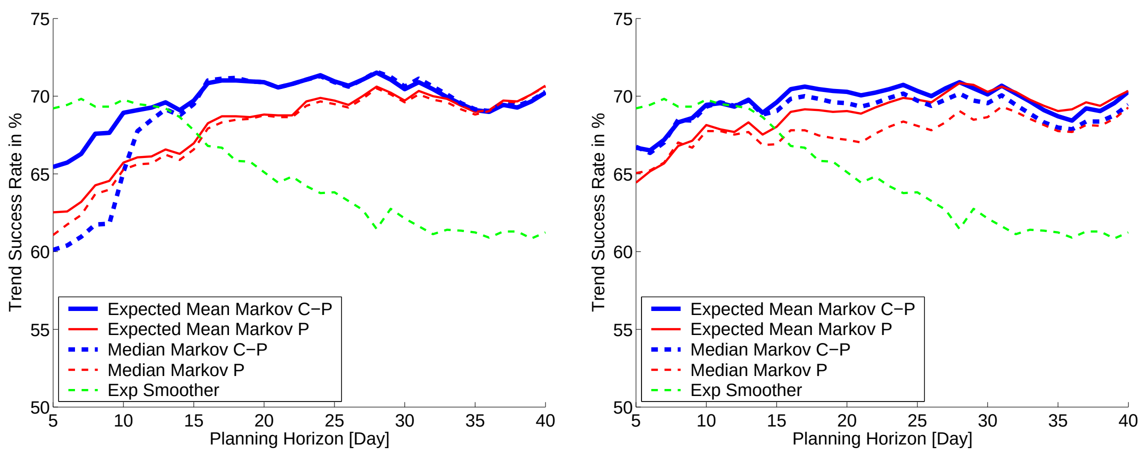


Figure 6.8: Success-rate of price trend sign prediction based on repeated 1-day (left) vs n-day interval (right) trained Markov matrices.

Often (especially in the short-term) applying a n-day interval trained Markov matrix results in better performance in terms of predicting falling or rising prices compared to the repeated 1-day trained Markov matrix. The reason for the behavior might be that any prediction further out than one day needs to be estimated by the 1-day trained Markov matrix which is is multiplied n-times, whereas the n-day interval trained Markov matrix is based on actual observed regime changes. Each time we apply the 1-day trained Markov matrix to itself the uncertainty in regime prediction increases.

Next, we analyze the sensitivity of the price trend prediction when using a different number of regimes and compare it to results obtained using an additional prediction method, which we call "Q Bayes Net."

To evaluate the quality of the predictions, we compare the performance of regime-based price trend prediction with two "baseline" methods, one which is based on predicted demand, the other based on an exponential smoother. The demand-based predictor is based on the economic theory that prices are influenced by demand. For TAC SCM, customer demand is generated by a random walk algorithm, as specified in [Collins *et al.*, 2004]. We can use a Bayesian inversion of this algorithm to predict demand, as shown in [Kiekintveld *et al.*, 2004]. In Table 6.1, this prediction is labeled "Q Bayes Net." The exponential predictor is based on a Brown's linear (i.e., double) exponential smoother with $\alpha = 0.5$. In Table 6.1, this prediction is labeled "Exp Smoother." The other predictors are the 5%, 10% and 50% percentiles of the Markov model.

For the experiments. every day we forecast the price trend for the next 20 days, using Equation 6.8. The experiments used the training and testing data sets specified earlier. In addition to the baseline comparison, we explored the parameter space in two dimensions: the number of regimes used, and the percentile on the predicted price distribution used to determine the price trend. The "Success Rate" for our prediction method is computed using Equation 6.8. The price trend prediction is successful when the predictor's forecast price trend has the same sign as the real price trend.

A surprising outcome that can be seen in Table 6.1 is that customer demand is not a strong predictor (around 50% to 57%), contrary to our initial assumption. We can see that price trend predictions based on regimes outperform predictions based on demand projection by a significant margin. Across all market segments our prediction

Predictors	*Market Segment*	*Success Rate (%)*	
		Number of Regimes	
		3	5
Q Bayes Net	Low	54.93	54.93
Exp Smoother	Low	66.66	66.66
5 % Markov	Low	71.23	71.04
10 % Markov	Low	70.95	71.32
50 % Markov	Low	67.32	66.48
Q Bayes Net	Medium	57.45	57.45
Exp Smoother	Medium	66.66	66.66
5 % Markov	Medium	70.67	68.34
10 % Markov	Medium	71.23	68.99
50 % Markov	Medium	72.25	67.78
Q Bayes Net	High	56.70	56.70
Exp Smoother	High	67.22	67.22
5 % Markov	High	69.46	70.39
10 % Markov	High	70.11	73.18
50 % Markov	High	71.14	70.48
Mean Markov	All	70.48	70.00

Table 6.1: Prediction results on the testing set of games from day 1 to day 179 (for a total of 1074 trials) with a 20 day forecast horizon, i.e. the last day of a forecast is day 199.

methods outperforms the stable and Q Bayes net prediction results. We can also see that the prediction quality is not especially sensitive to the number of regimes used or to the chosen percentile on the predicted distribution. Figure 6.9 depicts the success rate for the different predictors over a varying planning horizon up to 50 days into the future for the high market segment.

We tested whether the Markov predictions were different from actual observed price trends using pairwise Binomial hypothesis test and failed to reject the equality of trends hypothesis at $p = 0.01$ significance level. This indicates that our predictions are statistically following the same trend as the real price trends.

Finally, we tested the influence of different training and clustering methods on price trend predictions. As an alternative training method we used all order prices on a given day to determine the dominant regime instead of only the mean price, and as a second clustering method we used a hard version of the kmeans clustering algorithm

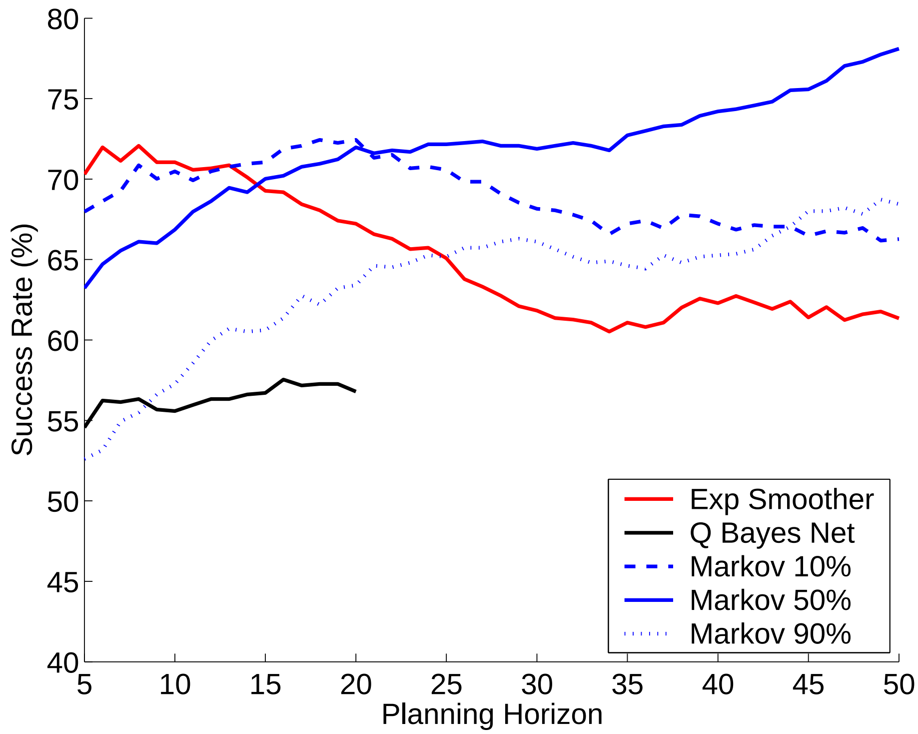


Figure 6.9: Success rate over the planning horizon over the test set of games for the high market segments. These are computed off-line from games out of the training set.

called kmns [McCallum and Nigam, 1988]. The results of these experiments are listed in Table 6.2.

We expected the results of the experiments which were trained with all order prices to have a higher success rate than the one with just the mean prices, but the results are almost equal and the training method does not appear to have a strong influence. Furthermore, the different clustering methods did not influence the result strongly. Across all market segments our prediction method outperforms Bayes net prediction results. We tested the significance between prediction success rates using a pairwise Binomial hypothesis test and recored a significance at the $p = 0.01$ level.

6.1.8 Prediction error

We also measured the prediction error for the price distributions generated by our model. Because the mixture of Gaussians can only approximate the true price distributions, we measured the difference between observed price frequencies and model

Predictors	*Market Segment*	*Success Rate (%)*			
		Clustering Method			
		kmeans		kmns	
		Training Method			
		mean price	all prices	mean price	all prices
Q Bayes Net	Low	52.89	52.89	52.89	52.89
5 % Markov	Low	68.99	69.09	69.09	68.99
10 % Markov	Low	68.72	68.90	69.09	68.72
Q Bayes Net	Medium	50.37	50.37	50.37	50.37
5 % Markov	Medium	66.20	66.48	66.57	65.92
10 % Markov	Medium	66.20	66.01	66.20	65.92
Q Bayes Net	High	56.61	56.61	56.61	56.61
5 % Markov	High	74.12	73.84	75.42	75.88
10 % Markov	High	74.21	74.30	74.67	75.79

Table 6.2: Prediction results based on the testing set of games from day 1 to day 179 (altogether 1074 trials) with a 20 day forecast horizon, i.e. the last day of a forecast is day 199. On average prediction results that differ by more than 5% are significant at the $p = 0.01$ level given the sample size.

predictions using a Monte Carlo method. In particular, price frequencies were computed for 64 bins from game data to form an empirical histogram. Simulated price data was sampled from the mixture model and binned as per real data. Prediction error was defined as the 1-norm (sum of absolute differences) distance between simulated and measured histograms, averaged across 1000 simulated data samples. In Figure 6.10 we present the algorithm used to analyze price predictions when sampling from the learned GMM with 16 and 25 components. Table 6.3 displays the results for the fitted GMMs.

	low market	medium market	high market	# Gaussians
Prediction Error in %	6.69	6.89	8.99	16
Prediction Error in %	5.75	5.48	5.95	25

Table 6.3: Overall prediction error for a 16 and a 25 GMM in the three market segments. Results were obtained after averaging over 1000 iterations.

The results show that the total error introduced by the mixture model approximation varied between $5\% - 8\%$, with more components resulting in slightly lower errors.

Inputs:
  pnp_{avg}: original normalized price density
  $numBins$: the number of histogram bins
  $numNP$: number of np in the training set
  GMM: learned Gaussian Mixture Model
  $maxIter$: number of iterations
Output:
  PredErr: the overall mean prediction error
Process:
  for $j = 1$ **until** $maxIter$
    $pnp_{samp} = Monte\_Carlo\_Sampling(GMM, numNP)$
    $Error(j) = \frac{|pnp_{avg} - pnp_{samp}|}{numBins}$
  end
  $\text{meanErr} = \overline{\text{Error}}$
  $\text{PredErr} = numBins \cdot \sum \text{meanErr}$
  return PredErr

Figure 6.10: Prediction error algorithm.

6.1.9 Order probability

We verify the goodness of the current day order probability estimation by determining the normalized prices, np, which represent 10%, 25%, 50%, 75%, and 90% of the area under the cumulative distribution function (see Equation 5.34). If the estimated cumulative distribution function is correct, i.e. mirroring the current market condition, then this should directly translate into 90%, 75%, 50%, 25%, and 10% daily order probability (see Equation 5.33). Next, we determine off-line how many auctions we would have won on each day if we had bid those estimated prices.[4] If we take the percentage of the winning auctions to the overall auctions we get the actual order probability for each of those prices. For our experiment we estimated 2200 (10 games times 220 days) order probability curves for a sample market. Figure 6.11 shows the results of the experiments for the Markov correction-prediction, the Markov prediction, and the exponential smoother process. The y-axis shows the estimated order probability, and the bar graphs show the actual mean order probability and their standard deviations. We observe that all three predictors estimate the daily order probability well. Furthermore we observe, that the Markov correction-prediction process estimates the higher order probabilities, 75% and 90%, better than the Markov

[4] In TAC SCM customers always buy from the cheapest manufacturer agent.

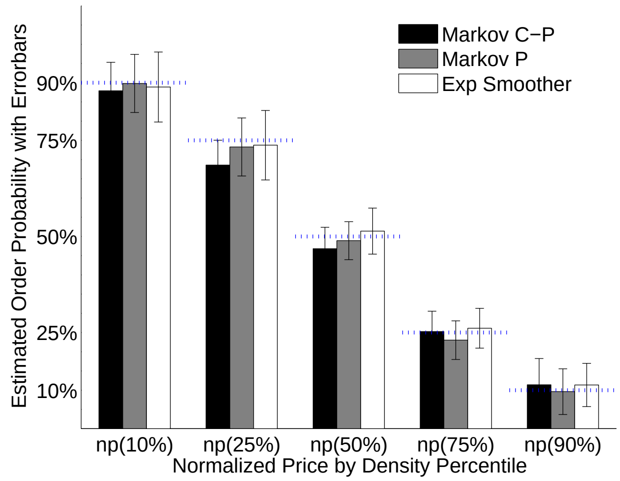


Figure 6.11: Daily order probability estimation (mean/std) for the 10th, 25th, 50th, 75th, and 90th percentile using different predictors.

prediction and the exponential smoother process, but performs a little less in the lower percentiles of the order probability estimation.

6.2 TAC SCM - Real-time

In the previous Section we evaluated our approach on data which are available from the game servers after the games have been played. In this Section we evaluate our methods in the context of playing against five other agents during a game in real-time. We focus on regime identification and prediction at the individual product level instead of the market segment, as described in the previous Section. This allows an agent to be more reactive in the supplier market, e.g. the market segment of a particular product might be in an over-supply situation, but one product might miss a part and therefore it is in a scarcity situation, which has a totally different impact on sales prices.

6.2.1 Experimental setup

We implemented all three regime identification and prediction methods, i.e. Markov prediction (MP), Markov correction-prediction (MCP), and exponential smoother lookup (ExpS) process, with the help of evaluators, as mentioned in Section 3.2. More details on evaluators are in [Collins *et al.*, 2007]. We also designed a training data evaluator which is shared (superclass) by the individual regime evaluators (subclasses). The training data evaluator reads in the configuration of the regime classes, e.g. 3 vs 5 regimes, which type of training data to use, median vs mean prediction, etc. and reads in the appropriate off-line learned training data files before the beginning of the game. The different regime evaluators inherit the training data and update them separately during a game. Figure 6.12 shows the chain of the evaluators used to make offer prices.

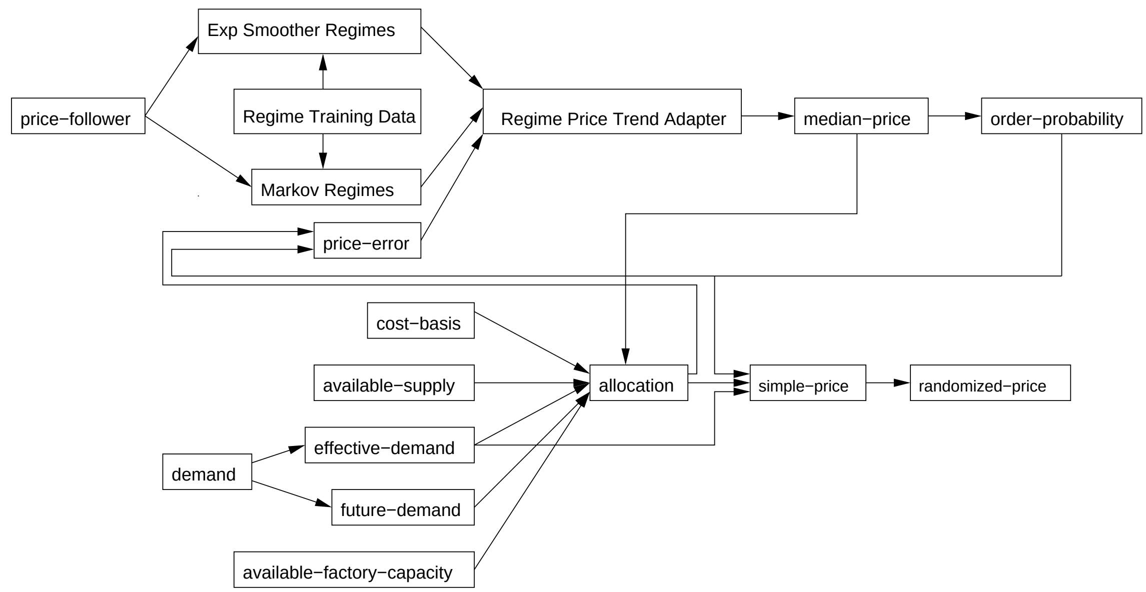


Figure 6.12: Evaluator chain for sales quota and pricing using regimes to determine prices, price trends and order probability.

The agents we use for our experiments have been obtained from the TAC SCM agent repository[5]. We selected five of the finalists from the 2006 competition and an agent from the 2005 competition. The agents are:

1. DeepMaize – University of Michigan
2. Maxon – Xonar Inc
3. MinneTAC – University of Minnesota
4. PhantAgent – Politechnica University of Bucharest
5. RationalSCM – Australian National University; competed in 2005.

[5] `http://www.sics.se/tac/showagents.php`

6. TacTex – University of Texas; winner TAC SCM 2006

Agent performance in TAC SCM is affected not only by the set of agents playing together but also by random variations in supply, demand, and other market parameters. To compare different variations of our own agent without having to run a very large number of games, we use the controlled server [Borghetti *et al.*, 2006, Sodomka *et al.*, 2007] which has been developed as a tool to evaluate agents. The controlled server allows for repeatable pseudo-random sequences of any individual market factor or combination of factors.

We run N_G games, each with a different pseudo-random sequence, using MinneTAC, and then run N_G games with the *same market factors* (the same set of N_G pseudo-random sequences) this time using a different version of our agent, MinneTAC'. In other words, all the pseudo-random sequences in the first set of N_G games are repeated in the second set of N_G games. For our tests, we chose $N_G = 23$. This method removes the profit variability due to the agents seeing different market conditions, and, at the same time, allows for testing multiple variations of our agent, one for each set of games. In this way our agent plays always against five different agents, and not against its onw clone(s).

Each version of our MinneTAC agent uses a different technique for tactical decisions (order probability calculation) and for strategic decisions (price and price trend prediction).

At the strategic level we distinguish three different price prediction methods. The first one uses a price-following method, the second one uses regimes (as described is Section 5.5), and the third one (called Combo in Table 6.4) also uses regimes, but reverts to a price-following method if the agent identifies a "price war" situation, i.e. an extreme over-supply regime. The reason for this combination predictor is based on our hypothesis that in a extreme-oversupply regime we are faced with a mismatch between the training data and the actual game data. This means in particular that we might observe prices in extreme-oversupply regimes during actual games which are by far lower than the prices we observed during the economic regime training phase. As a consequence during actual games the regime price prediction might, after some days, point toward higher prices again, whereas the exponential smoother predictor will simply follow the actual prices, which in extreme-oversupply are just a low flat price line.

At the tactical level we differentiate two methods to calculate order probability, one based on a linear interpolation between the estimated minimum and maximum good prices and the other based on regimes (as described is Section 5.6).

6.2.2 Real-time results

Our tests includes eight sets of twenty-three games each, one set for each different configuration of our MinneTAC agent, using the same twenty-three pseudo-random sequences for each set. For each method we compare the difference in profit, and compute the standard error associated with each mean difference. As the primary measure of agent performance we list in Table 6.4 and Table 6.5 the mean total profit per agent over a game. From Table 6.4 and Table 6.5 we see that MinneTAC when playing with this set of agents always comes in fifth, but for the purpose of testing the different configurations of MinneTAC, we are only interested in its relative performance.

The results of the different experiments are as follows:

1. In the first experiment MinneTAC uses a linear interpolation to determine the probability of order and an exponential smoother to predict price trends. MinneTAC reaches a final mean profit of 1.347 million.

2. The configuration of the second experiment uses economic regimes (based on a repeated 1-day Markov prediction) to predict price trends instead of the exponential smoother predictor. The final result of this combination amounts to 1.813 million. We already reported before that price predictions generated by economic regime have a higher accuracy and a smaller error in terms of the root mean square price error, but now we have also empirically shown that regime price predictions outperform exponential smoother predictions in terms of final mean profit.

3. In the third experiment we test if our hypothesis that a combination predictor consisting of economic regimes and exponential smoother beats an agent which uses pure regime price predictions. Since the final profit of MinneTAC for this experiment amounts to 1.780 million we have to reject our hypothesis, and

conclude that it is better to use regime price predictions in all different market conditions. The resulting small profit difference between experiment two and three makes intuitively sense, since in an extreme over-supply situation there is hardly any profit to be made, and it is more important to control losses than to make profit.

4. Our fourth experiment uses regimes for tactical decisions (determination of order probability based on exponentially smoothed predicted regimes) and for strategic decisions (price and price trend prediction based on a repeated 1-day Markov matrix). The final mean profit for this experiment is 2.017 million and beats in that respect all other combinations.

5. Experiment five uses a Markov n-day interval prediction to determine price trends. Its final mean profit comes to 1.567 million. This result shows a difference when predicting aggregate vs individual product data. We expected that in real-time the Markov n-day interval prediction would outperform the repeated 1-day Markov prediction as reported in Section 6.1.6, but the outcome of our experiments shows that a repeated 1-day Markov prediction performs better in terms of final profit than a Markov n-day interval prediction. The reason could be that off-line we use for every day in the planning horizon a separately trained Markov matrix, but since we only have limited time during a game in real-time we use only a 1, 10, and 20 day Markov prediction matrix. Then we perform regime and price density predictions for these three matrices and interpolate the missing prices between them. Here we make the assumption that the prices in between are related linearly to each other, which is most likely not the case, since we actually expect prices to flatten out further into the future.

6. In the sixth experiment we use a Markov correction-prediction with a repeated 1-day Markov matrix to determine the price trends. The final mean profit amounts to 1.889 million. This outcome fits our observation on the market segment level, where the Markov prediction method outperforms the Markov correction-prediction method.

7. The seventh experiment uses a Markov correction-prediction process with an n-day interval trained Markov matrix to determine price trends. Its final mean profit comes to 1.670 million. This outperforms the results of experiment five, which uses a Markov prediction with an n-day interval trained Markov matrix

to determine price trends. Here we observe the same phenomenon as in experiment five, where the n-day interval matrix performs worse than the 1-day repeated Markov matrix. The most likely reason again is that we interpolated the predicted prices in between linearly and they might not have a linear form. We are currently working on an implementation where we use matrices for all days in the planning horizon.

8. Experiment number eight uses an exponential smoother regime lookup predictor to determine the order probability and median prices as well as to determine the future price trends. It has a final mean profit of 1.545 million. The exponentially smooth look-up regime process works well for determining the current day order price distribution, but has a price divergence that is too large at the end of the planning horizon.

	Mean Profit / Standard Deviation (in million)				
Experiment #	1	2	3	4	5
Strategic:	Price-Follower	Regimes (MP 1-Day)	Combo	Regimes (MP 1-day)	Regimes (MP n-day)
Tactical:	Linear	Linear	Linear	Regimes (ExpS)	Regimes (ExpS)
Agent:					
TacTex06	8.752/5.682	8.873/5.600	8.399/5.173	9.205/5.385	9.061/5.331
DeepMaize06F	8.839/4.629	8.713/4.846	8.403/4.710	8.318/4.181	8.652/4.865
PhantAgent06	8.049/5.422	7.991/5.384	7.895/5.326	8.173/5.437	7.953/5.247
Maxon06F	4.243/4.516	3.767/4.288	3.808/4.254	4.019/4.181	3.945/4.396
MinneTAC	1.347/3.703	1.813/4.017	1.780/4.536	2.117/3.764	1.567/3.796
Rational05	0.739/4.912	0.669/4.692	0.710/4.692	1.305/4.527	1.115/4.682

Table 6.4: Experimental setup with controlled market conditions and different variations of MinneTAC for order probability, price and price trend predictions. Mean profit and standard deviation results are based on 23 games per set of experiments. Regime MP 1-day stands for regime prediction using a 1-day Markov transition matrix, Regime MP n-day uses the n-day interval Markov transition matrix, and Regime ExpS does regime prediction via an exponential smoother lookup process.

	Mean Profit / Standard Deviation (in million)		
Experiment #	6	7	8
Strategic:	Regimes (MCP 1-day)	Regimes (MCP n-day)	Regimes (ExpS)
Tactical:	Regimes (ExpS)	Regimes (ExpS)	Regimes (ExpS)
Agent:			
TacTex06	9.039/5.075	9.311/5.203	9.302/5.343
DeepMaize06F	8.648/4.521	8.515/4.488	8.921/4.733
PhantAgent06	8.082/5.126	7.966/4.940	8.029/5.425
Maxon06F	3.988/3.976	4.138/4.381	4.214/4.628
MinneTAC	1.889/3.740	1.670/3.867	1.545/3.898
Rational05	1.211/4.346	0.668/4.440	1.032/4.898

Table 6.5: Experimental setup with controlled market conditions and different variations of MinneTAC for order probability, price and price trend predictions. Regime MCP stands for regime prediction via a Markov correction-prediction process, and 1-day or n-day refer respecitivey to using the 1-day or n-day Markov transition matrix.

We used the Wilcoxon signed rank test [Gibbons, 1986, Hollander and Wolfe, 2000] to assess statistical significance between the first four experiments. This is a *non-parametric* test of the difference between the medians of two samples that does not require the samples to come from normal (or even the same) distribution. We focused on the first four experiments because they constitute improvements steps over the original method, while the other experiments test the effects of variations in the price-trend prediction method based on regimes. The Wilcoxon signed rank test is used to test whether the median of a symmetric population is 0. First, the data are ranked without regard to sign. Second, the signs of the original observations are attached to their corresponding ranks. Finally, the one sample z statistic (mean / standard error of the mean) is calculated from the signed ranks.

Test #	1: Exp 4 - Exp 1			2: Exp 4 - Exp 2			3: Exp 4 - Exp 3			4: Exp 2 - Exp 1		
$\alpha = 0.05$	p	h	srank	p	h	srank	p	h	srank	p	h	srank
All	0.0138	1	57	0.1137	0	86	0.3155	0	105	0.0727	0	79
Positive	0.0054	1	13	0.2769	0	40	0.7615	0	54	0.0256	1	21
Negative	0.4258	0	15	0.4258	0	15	0.4961	0	16	0.9102	0	21

Table 6.6: Wilcoxon signed rank test of equality of medians. The test were performed at a significance level of $\alpha = 0.05$ based on 23 data points. p represents the p-value, h is the result of the hypothesis test, srank contains the value of the signed rank statistic.

In Table 6.6 we show the results of the Wilcoxon test. Rejecting the null hypothesis means the medians from the two different samples are different. p is the probability of observing a result equally or more extreme than the one using the data (from both samples) if the null hypothesis is true. If p is near zero, this casts doubt on this hypothesis. The field "srank" contains the value of the signed rank statistic.

We performed the tests on the set of all games, on only positive profit games, and on only negative profit games. As a result of these tests we observe that the configuration of experiments four and two, and four and three are not significantly different, and we cannot reject the null hypothesis. On the other hand we find significant differences between the outcome of experiments four and one. We are able to reject the null hypothesis of equal median for the set of all games and the set of all positive games, but not for the set of negative games. The most likely reason why we are not able to reject the null hypothesis of equal median for the set of negative games is that in negative games an agent is more concerned with controlling cost than making profit and so the differences between the configuration is not much apparent. We are also able to show significance between the experiments two and one for the set of all positive games. Although we could not reject the null hypothesis for the set of all games at $\alpha = 0.05$ significance level with a p-value of 0.0726, we are able to reject the null hypothesis for the set of all games at $\alpha = 0.1$ significance level. We believe that the test would likely show significance with a larger sample size. With this we have shown statistical significance between the original configuration and the regime/regime configuration. Our results also suggests that the regime/regime configuration performs better than the linear/regime configuration, although we need more data to show that conclusively.

6.3 Financial markets

In the following we present an application of the regime method to the stock market domain. An investor could use this to decide whether to keep his stocks, to buy more, or to sell in time to make a profit. We define that in financial markets regime R_1 represents a more *bearish* signal, as opposed regime R_5 which represents a more *bullish* signal. We have done these experiments on different stocks, but present here the General Electric, GE, stock as an example of many. We are not claiming that this works better than other time series prediction methods, this is just a proof of concept on data outside of the TAC SCM domain.

6.3.1 Experimental setup

Stock market prices are characterized by a time series, and when we perform the regime training we need to pick a continuous price stream, as opposed to TAC SCM where we randomly pick training games of a pool of games. We obtained the stock market data from the Yahoo finance[6] service.

The left side of Figure 6.13 displays the time series of our training price data from October 1st 2005 until December 31st 2005 and the right side shows the appropriate GMM. For these experiments we use a GMM with 20 Gaussians.

Figure 6.14 (left) shows the learned regime probabilities over price. We experimented with different number of regimes on different stocks and found that 5 regimes results in the highest success-rate of price trend predictions. Figure 6.14 (right) displays the

[6]Yahoo finance: http://finance.yahoo.com/

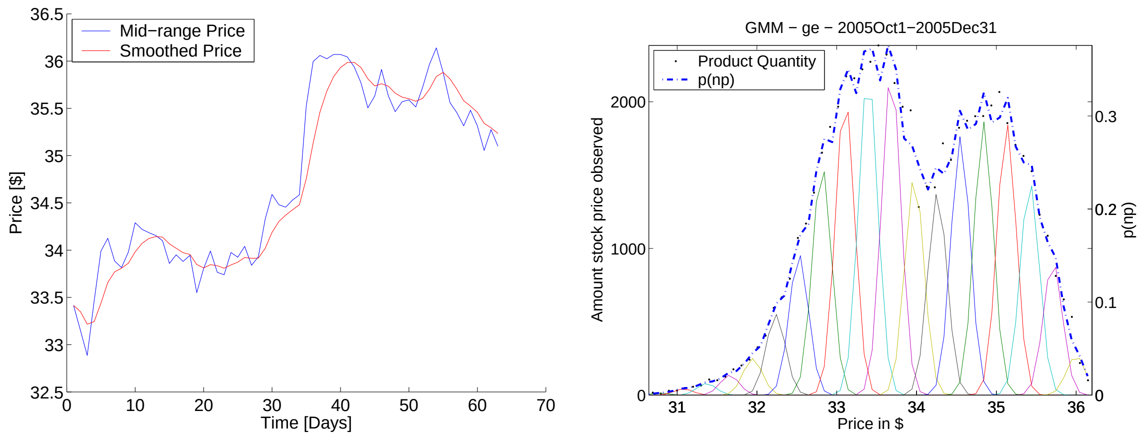


Figure 6.13: General Electric: Historical prices from Oct 1st 2005 until Dec 31st 2005 (left) and the appropriate GMM (right).

time series of our testing set. We recorded historical prices from January 1st 2006 until September 26th 2006. We calculate the price limits of the GMM as follows:

$$price\_limit_{max}^{GMM} = max((price_d + \Delta price), highest\ price\ in\ training\ data) \quad (6.9)$$

$$price\_limit_{min}^{GMM} = min((price_d - \Delta price), lowest\ price\ in\ training\ data) \quad (6.10)$$

where $price_d$ is the average stock price for the current day and $\Delta price$ is the biggest price difference over continuous intervals of the length of the planning horizon over the training set.

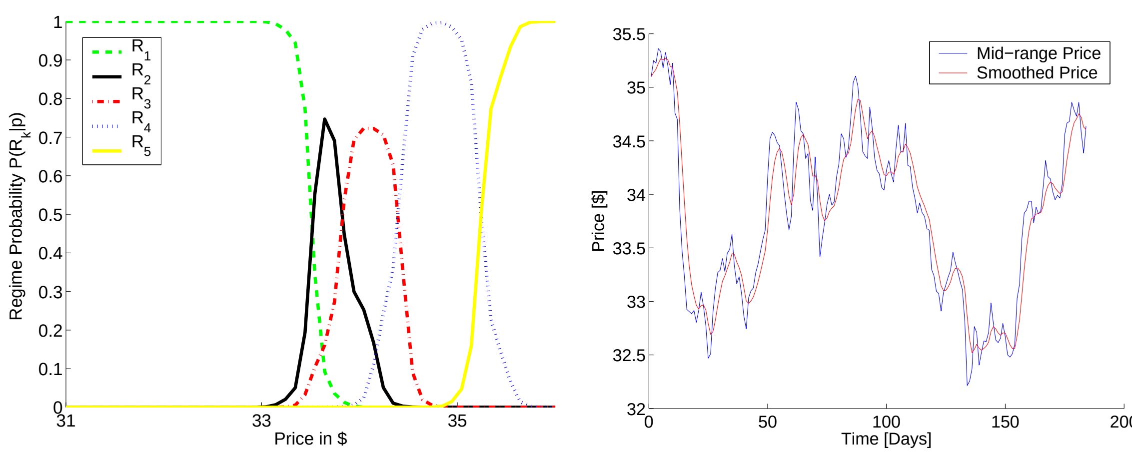


Figure 6.14: General Electric: Learned regime distributions (left) and test data from Jan 1st 2006 until Sep 26th 2006 (right).

6.3.2 Off-line results

Figure 6.15 (left) shows the real testing price data, the exponential smoother predictions, and the Markov predictions using a repeated 1-day Markov transition matrix. The right side of Figure 6.15 displays the Markov predictions using a N-day interval Markov transition matrix.

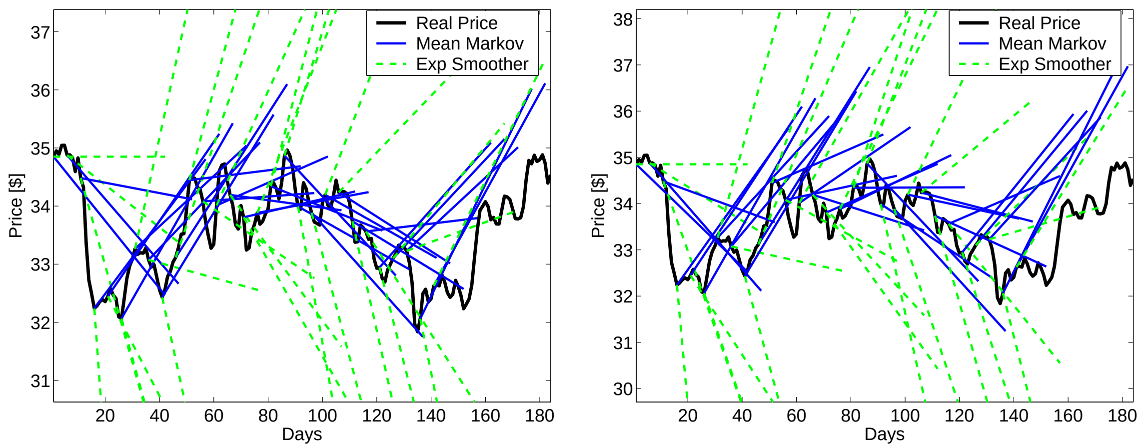


Figure 6.15: Price prediction based on 1-day (left) vs n-day (right) trained Markov matrix.

Finally Figure 6.16 visualizes the success-rate of price trend predictions using a 1-day (left) and a n-day (right) Markov transition matrix. We observe that here, the same as in TAC SCM, using a n-day Markov matrix leads to a higher success rate in terms of price trend predictions for the overall distribution.

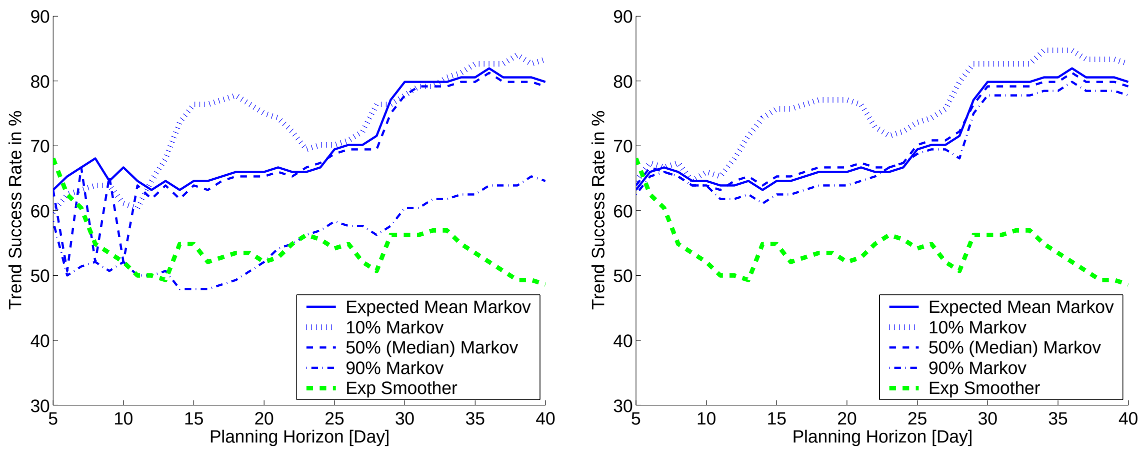


Figure 6.16: Success-rate of price trend prediction based on 1-day (left) vs n-day (right) trained Markov matrix.

Chapter 7

Conclusions

We have presented an approach for identifying and predicting market conditions in markets for durable goods. We have demonstrated the effectiveness of our approach using games played in the semi-finals and finals from TAC SCM 2005 and on a controlled server environment, and showed initial research on stock market data. An advantage of the proposed method is that it works in any market for durable goods, since the computational process is completely data driven and that no classification of the market structure (monopoly vs competitive, etc) is needed.

7.1 Contributions

Our approach recognizes that different market situations have qualitative differences that can be used to guide the strategic and tactical behavior of an agent. Unlike regression-based methods that try to predict prices directly from demand and other observable factors, our approach recognizes that prices are also influenced by non-observable factors, such as the inventory positions of other agents. Our approach learns the dynamics and durations of different regimes, and when to expect a shift in the dominant regime. This is important information that is difficult to represent with regression-based methods. For example, regression in an expanding market (where prices increase) will extrapolate increasing prices using the slope of recent price data. On the other hand, the regime approach can learn that expansion (or scarcity) regimes are typically limited in duration and predictably followed by other regimes. When prices are increasing, it is more important to know if prices will fall by the end of the planning horizon, which can be invaluable information for a decision maker. Our method can enable an agent to anticipate and prepare for regime changes, for example by building up inventory in anticipation of better prices in the future or by selling in anticipation of an upcoming oversupply situation.

7.2 Future directions

Our approach maintains the uncertainty in price prediction by maintaining a price distribution. This allows an agent to avoid over-committing to risky decisions. We intend to apply our method in other domains where predicting price distributions appears fruitful, including domains such as Amazon.com, eBay.com, and to further deepen our research in financial applications like stock tracking and forecasting.

We have implemented the regime identification and prediction method in a TAC SCM agent and integrated it into the overall decision making process. Currently we are using regime predictions for tactical and strategic decision making in the sales component of our agent. Ultimately, we plan to combine probability information supplied by our method with information about possible consequences of actions to optimize decision making.

With TAC SCM and the stock market we have presented two applications of our research. The stock market generates a continuous flow of pricing data, whereas in TAC SCM we have an artificial start and end. Figure 7.1 shows the minimum and maximum prices, available in the daily price report, for a typical game, 4260@tac4, for the low market segment.

We observe that the spread between minimum and maximum prices tends to be relatively constant for much of the game, but in the last 10 days the minimum and maximum prices diverge strongly. This is a typical behavior in TAC SCM games, since unsold inventories have no residual value at the end of the game, and some agents get very aggressive in attempting to sell it off. We have analyzed the mean, median, and standard deviation of all prices of all the final games in 2005 TAC SCM tournament (see Figure 7.2), and found that the price divergence in the last 10 days is very apparent in the aggregate data. Since the last 10 days of the game do not appear to exhibit any consistent structure, we are thinking it might be best to apply a simple linear approximation of the order probability curve to react faster to unseen

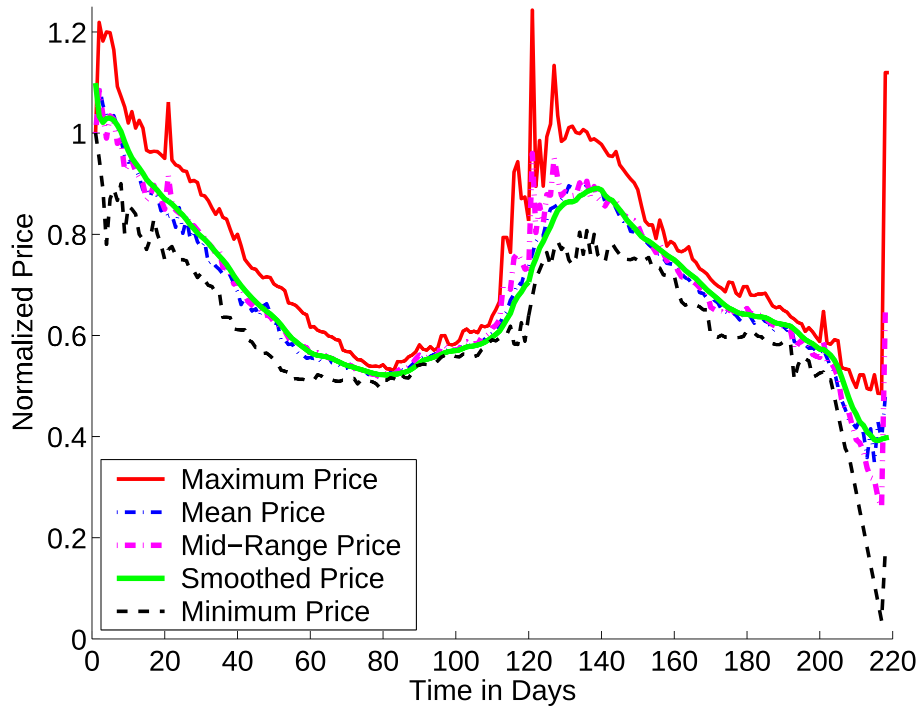


Figure 7.1: 4260tac4 (Final TAC SCM 05) – Minimum and maximum daily prices of computers sold, as reported during the game every day, and mean price for the low market segment. The mean price is computed after the game using the game data, which include complete information on all the transactions.

market conditions.

Based on many games we also found out that the beginning of the game, usually the first 30 days, has an apparent downward trend. The reason is that all manufacturer agents start out with nothing in their inventories. Once they have something to offer they can sell for a high price, since the market is in an extreme scarcity regime. Since the start effect is quite predictable and the end game effect is hardly predictable we are considering applying three different models to better determine the order probability. We have started training a separate GMM for each product for the start phase – day 1 to day 30 (see Figure 7.2, and one for the mid game – day 20 to day 210 (see Figure 7.2), and a linear model for the last 10 days of the game. As we can see in Figure 7.2, high initial prices are reflected in the early GMM through a large density mass above the point where np = 1.0. Towards the end of the game we hardly see any prices above 1.0. The reason is that the competition is much stronger after day 30, i.e. after each agent has enough inventory to respond to the majority of the customer

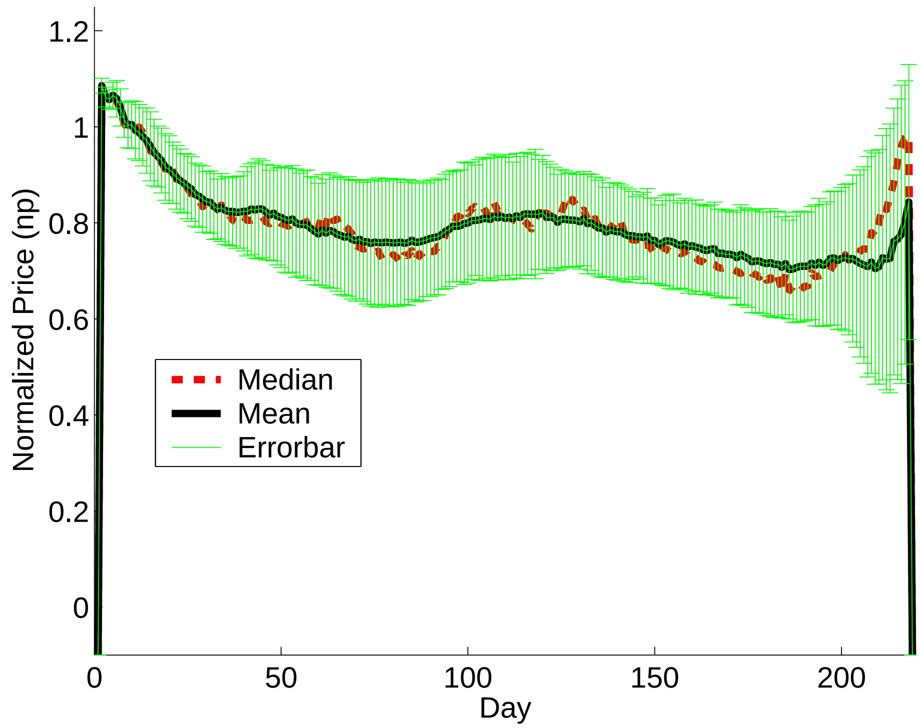


Figure 7.2: Mean, median prices and error bars (+/ − 1 standard deviation) for all final games in TAC 2005 for the low market segment.

RFQs.

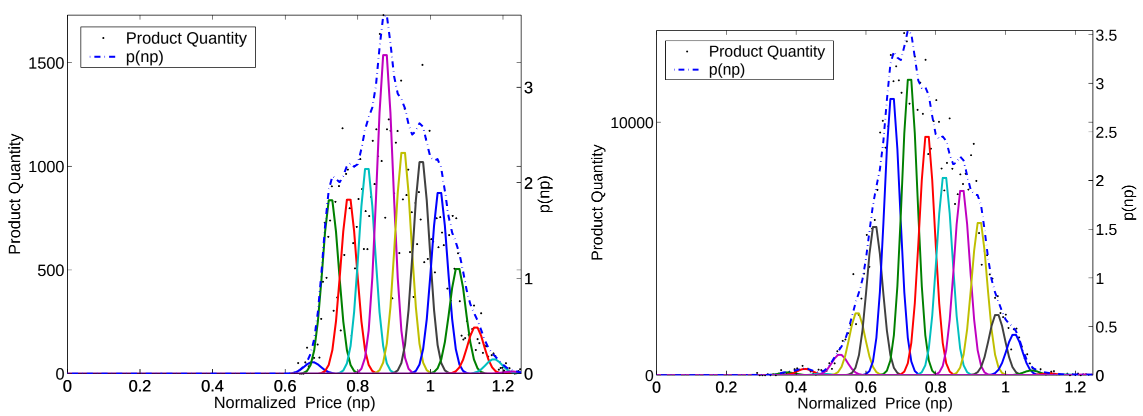


Figure 7.3: The price density density function for product 16, $p(np)$, (right y-axis) estimated by the GMM with 25 components from day 1 to day 30 (left) and from day 21 to day 210 (right). We clearly observe a shift in the distribution.

We have purposely trained the early and middle GMM model with a time overlap, since the initial early model might predict after day 20 a too strong price trend downwards. We have developed a model combiner that takes a transition length and the start and end times of the different models as input. Both models together generate weighted predictions, i.e. medians, price trends and order probabilities. Their weights assignments are linearly reciprocal, i.e. on the first day the early model has 100% weight and the middle one 0%, after 10% of the transition time, the first model has 90% and the middle one has 10%, and so on until the end where the first model has 0% and the middle model has 100%. We plan to thoroughly test and analyze the model in the near future.

In addition, we plan to apply reinforcement learning [Sutton and Barto, 1998] to map economic regimes to internal operational regimes and operational regimes to actions, such as procurement and production scheduling. Under operational regimes we understand a state which includes which actions to take next while knowing the current regime and receiving the regime forecast.

There are several improvements to the prediction process we are also pursuing. We plan to implement a boosting algorithm [Schapire *et al.*, 2002, Stone *et al.*, 2003] to perform dynamic model selection between the two Markov, the exponential smoother and potentially other regime predictors. Since the methods might work well for different time-steps in the planning horizon, we plan to perform boosting for all days in the planning horizon, which will provide a way to focus on the most successful predictors during the online use of the algorithm.

We also would like to research the impact of different training models and their resulting real-time performance. One way to modify the regime training model is to train one model for all the products in one market segment, and then in real-time the agent will update all the products in one market segment individually. The reason is that all the products in one market segment see the same demand level, and we normalize the prices across products.

Appendix A

Summary of notation

Symbol	**Definition**
$\mathcal{C}$	Set of all available component types
$\mathcal{G}$	Set of all goods (product types)
d	Current day
$D_{d,g}$	Customer demand for a good g on day d
$D_{d,g}^{eff}$	Effective customer demand for a good g on day d
Φ	Total profit
$A_{d,g}$	Allocated quota for a good g on day d
F	Factory capacity
h	Planning horizon
np	Normalized price
$\overline{\mathrm{np}}$	Mid-range normalized price
$\widetilde{\mathrm{np}}^{min}$	Smoothed minimum normalized price
$\widetilde{\mathrm{np}}^{max}$	Smoothed maximum normalized price
$\widetilde{\mathrm{np}}$	Smoothed mid-range normalized price
α	Smoothing coefficient
$p(\mathrm{np})$	Density of the normalized price
GMM	Gaussian Mixture Model
N	Number of Gaussians of the GMM
$p(\mathrm{np}\vert\zeta_i)$	Density of the normalized price, np, given i-th Gaussian of the GMM
μ_i	Mean of i-th Gaussian of the GMM
σ_i	Standard deviation of i-th Gaussian of the GMM

$P(\zeta_i)$	Prior probability of i-th Gaussian of the GMM
$P(\zeta_i\|\text{np})$	Posterior probability of the i-th Gaussian of the GMM given a normalized price np
$\vec{\eta}(\text{np})$	N-dimensional vector with posterior probabilities, $P(\zeta_i\|\text{np})$, of the GMM
M	Number of regimes
R_k	k-th Regime, $k = 1, \cdots, M$
$\mathbf{P}(c\|r)$	Conditional probability matrix (N rows and M columns) resulting from k-means clustering
$p(\text{np}\|R_k)$	Density of the normalized price np given a regime R_k
$P(R_k\|\text{np})$	Probability of regime R_k given a normalized price np
$P(order\|\text{np})$	Probability of order given a normalized price np
$\mathbf{T}$	Markov transition matrix
Π	Stationary distribution of a Markov process
$KL(\vec{P}(\hat{R})\\|\vec{P}(R))$	KL-divergence between the predicted regime probability distribution and the actual regime probability distribution

Appendix B

Parameter determination of a Gaussian Mixture Model

B.1 Maximum Likelihood

Maximum likelihood estimation [Mitchell, 1997] begins with the mathematical expression known as a likelihood function of the sample data. Loosely speaking, the likelihood of a set of data is the probability of obtaining that particular set of data given the chosen probability model. This expression contains the unknown parameters. Those values of the parameter that maximize the sample likelihood are known as the maximum likelihood estimates.

B.2 The EM-Algorithm

To calculate the posterior probability $P(c|np)$ Equation 5.9 assumes that we know the distribution that each training instance came from, and the parameters of the mixture model. The problem is that we do not know either of these things. Here we describe how to determine the parameters, μ, σ and prior probability of a GMM from a data set with the help of an unsupervised clustering method called the *EM algorithm* [Dempster *et al.*, 1977].

The EM algorithm does the clustering from a probabilistic perspective. It attaches to each observation, even training examples, a certain probability of belonging to each cluster. The EM algorithm presumes that the data are generated from a mixture distribution, P. Such a distribution has N *components*, each of which is a distribution in its own right. A data point is generated by first choosing a component and then generating a sample from that component. Let the random variable C denote the component, with values $1, \ldots, N$; then the mixture distribution is given by:

$$P(x) = \sum_{i=1}^{N} \{P(x|c_i) \times P(c_i)\} \tag{B.1}$$

where x refers to the values of the attribute for a data point. In our case this corresponds to Equation 5.7 where a mixture of Gaussians is the choice of the components distributions. In a GMM $P(c_i)$ corresponds to the weight of each component, w_i, and $P(x|c_i)$ corresponds to the normal distribution $N(x; \mu_i, \sum_i)$ for each component. In our specific case we are looking only at the density for a particular parameter, normalized unit order price np, which collapses the covariance matrix to one element, σ_i. For general understanding we keep this description open to the multi-parameter case.

The underlying idea of the EM algorithm is to pretend that we know the parameters of the mixture model and then infer the probability that each data point belongs to each component. In a second step each component is re-estimated to fit the entire data set with each point weighted by the probability that it belongs to that component. These two steps are iteratively repeated until the procedure converges to a stationary values for μ_i. The GMM parameters are initialized arbitrarily and the following two steps are iterated:

1. E-step: Compute the probabilities $p_{ij} = P(c_i|x_j)$, the probability that datum x_j was generated by component i. With the help of Bayes' rule we determine $p_{ij} = \alpha \times P(x_j|c_i) \times P(c_i)$. The term $P(x_j|c_i)$ is the probability at x_j of the ith Gaussian, and the term $P(c_i)$ is the weight parameter or prior probability for the ith Gaussian. Define $p_i = \sum_j p_{ij}$

2. M-step: Compute the new mean, μ_i, covariance $\sum_i$, and component weights, w_i, as follows:

$$\mu_i \leftarrow \sum_j \frac{p_{ij} \times x_j}{p_i}$$

$$\sum_i \leftarrow \sum_j \frac{p_{ij} \times (x_j - \mu_i) \times (x_j - \mu_i)^T}{p_i}$$

$$w_i \leftarrow p_i$$

In the first phase, E-step or *expectation* step, the algorithm calculates the expected values p_{ij} of the hidden indicator variables Z_{ij}, where Z_{ij} is 1 if datum x_j was generated by the ith component and 0 otherwise. In the second phase, M-step or *maximization* step, the algorithm finds the new values of the parameters that maximize the log likelihood of the data, given the expected values of the hidden indicator variables.

Appendix C

Optimal number of Gaussians

Inputs:
- pnp_{avg}: original normalized price density
- $maxNumGauss$: the maximum number of Gaussians
- $maxFits$: iterations of GMM fitting
- NP: set of all normalized prices used for training
- $numNP$: length of NP

Output:
- optNumGauss: the optimal number of Gaussians

Process variables:
- GMM: Gaussian mixture model
- pnp_{samp}: sampled estimated normalized price density
- KL: KL divergence
- KL_{avg}: average KL divergence

Process:

for $comp = 1$ **until** $maxNumGauss$
    for $fits = 1$ **until** $maxFits$
    $GMM = Expectation\_Maximization(NP, comp)$
    $pnp_{samp} = Monte\_Carlo\_Sampling(GMM, numNP)$
    $KL(comp, fits) = KL\_divergence(pnp_{avg}, pnp_{samp})$
    end
$KL_{avg}(comp) = mean(KL(comp))$
end
$\text{Index\_KL}_{\text{min}} = min(KL_{avg})$
$\text{optNumGauss} = KL_{avg}(\text{Index\_KL}_{\text{min}})$
return optNumGauss

Figure C.1: Algorithm to find the optimal number of Gaussians in a GMM.

Appendix D

Semi-Markov Process

We hypothesized that regime switches are not exponential (Markov), i.e. the future depends not only on the present state, but also on the length of time the process has spent in that state. This requires modeling the regime transition as a semi-Markov process [Levinson, 1986].

To model this we modify the Markov transition matrix, $\mathbf{T}_{predict}$, to be a weighted sum of two matrices, the steady state matrix $\mathbf{T}_{steady}$ and the change matrix $\mathbf{T}_{change}$. $\mathbf{T}_{steady}$ is the $M \times M$ identity matrix, where M is the number of regimes. $\mathbf{T}_{change}$ is the Markov transition matrix, which is computed off-line as described earlier.

$$\mathbf{T}_{predict}(r_{t+1}|r_t) = (1 - \omega(.))\mathbf{T}_{steady} + \omega(.)\mathbf{T}_{change}(r_{t+1}|r_t) \tag{D.1}$$

where $\omega(.)$ represents the probability of a regime change, and r_t represents the current regime. To compute the value of $\omega(.)$, we need to introduce a few variables. We define Δt as the time since the last regime transition at t_0: $\Delta t = t - t_0$. We model the time τ_i spent in regime R_i before the transition to regime R_j occurs as a random variable with distribution F_{ij}. τ_i is estimated from historical data. We hypothesized that the probability density of τ_i is dependent on the current regime, R_i, i.e. $p(\tau_i|R_i)$. We computed the frequency of all values of τ_i in ascending order and fitted different distributions. The Gamma distribution, $g(t; \alpha, \lambda)$ is a reasonable fit to the data. (see Figure D.1).

The gamma density function, $g(t; \alpha, \lambda)$, depends on two parameters, α and λ:

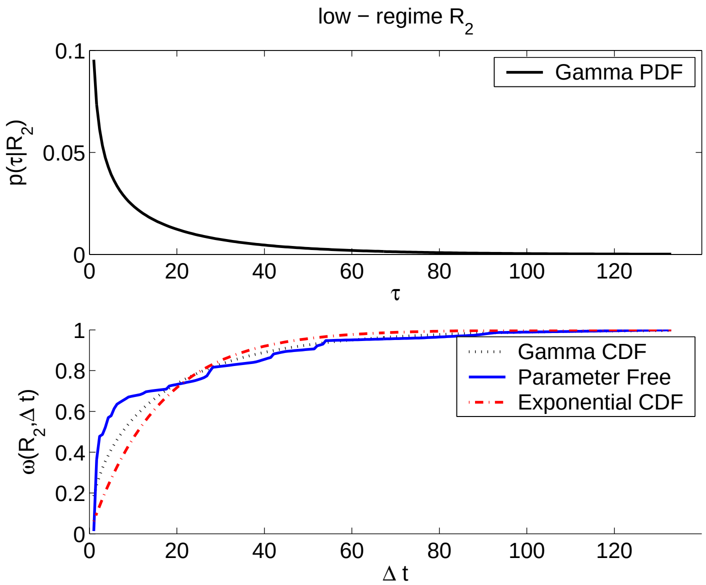


Figure D.1: Fitted Gamma pdf for $p(\tau|R_2)$ (top); Cumulative distributions: $\omega(r = R_2, \Delta t)$ is the probability of transitioning out of regime R_2, Δt is the elapsed time since the last regime change (bottom). Data are for the low market segment.

$$g(t;\alpha,\lambda) = \begin{cases} \frac{\lambda^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\lambda t} & \text{for } t \geq 0 \\ 0 & \text{for } t < 0 \end{cases}, \tag{D.2}$$

where $\Gamma(x)$ is the gamma function, which is defined as $\Gamma(x) = \int_0^\infty u^{x-1} e^{-u}\,\mathrm{d}u \quad x \geq 0, \alpha > 0, \lambda > 0$. The parameters were fitted separately for each regime using a maximum likelihood procedure. After applying the fitting procedure we obtained $\alpha_2 = 0.5193$ and $\lambda_2 = 0.0264$ for regime R_2 in the medium market segment. The probability of a regime transition $\omega(r, \Delta t)$ from the current regime, r, with respect to the time Δt that has elapsed since the last regime transition, t_0, is given by:

$$\omega(r = R_i, \Delta t) = \int_0^{\Delta t} p(\Delta t|r = R_i)\,\mathrm{d}\Delta t \tag{D.3}$$

where $p(\Delta t|r = R_i) = g(\Delta t; \alpha_i, \lambda_i)$. Equation D.4 describes a recursive computation for predicting the posterior distribution of regimes at time $t + n$ days into the future, where $k = n + 1$, for the semi-Markov process.

$$\vec{P}(r_{t+k}|\mathrm{np}_{t-1}) = \sum_{r_{t+k-1}} \cdots \sum_{r_{t-1}} \vec{P}(r_{t-1}|\mathrm{np}_{t-1}) \cdot \prod_{j=1}^{k} \mathbf{T}_{\mathbf{predict}}(r_{t+j}|r_{t+j-1}, \Delta t + j - 1) \tag{D.4}$$

When we model the process as a semi-Markov process we obtain in the high market 2989 out of 3184 (i.e. 93.88% success rate), in the mid market 2395 out of 3184 trials (i.e. 75.22% success rate), and in the low market 2451 out of 3184 trials (i.e. 76.98% success rate).

Bibliography

[Babanov *et al.*, 2003a] Alexander Babanov, Wolfgang Ketter, and Maria Gini. An evolutionary approach for studying heterogenous strategies in electronic markets. In G. Di Marzo Serugendo, A. Karageorgos, O. F. Rana, and F. Zambonelli, editors, *Engineering Self-Organising Applications (ESOA 2003)*, pages 157 – 168. LNCS 2977, Springer Verlag, 2003.

[Babanov *et al.*, 2003b] Alexander Babanov, Wolfgang Ketter, and Maria Gini. An evolutionary framework for large-scale experimentation in multi-agent systems. In Thomas Wagner, editor, *Toward an Application Science: MAS Problem Spaces and Their Implications to Achieving Globally Coherent Behavior*. Kluwer, 2003.

[Benisch *et al.*, 2004] Michael Benisch, Amy Greenwald, Ioanna Grypari, Roger Lederman, Victor Naroditskiy, and Michael Tschantz. Botticelli: A supply chain management agent designed to optimize under uncertainty. *ACM Trans. on Comp. Logic*, 4(3):29–37, 2004.

[Benisch *et al.*, 2005] Michael Benisch, James Andrews, David Bangerter, Timothy Kirchner, Benjamin Tsai, and Norman Sadeh. Cmieux analysis and instrumentation toolkit for tac scm. Technical Report CMU-ISRI-05-127, School of Computer Science, Carnegie Mellon University, September 2005.

[Berry *et al.*, 2006] P. Berry, K. Conley, M. Gervasio, B. Peintner, T. Uribe, and N. Yorke-Smith. Deploying a personalized time management agent. In *Proc. of the Fifth Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, Hakodate, Japan, May 2006.

[Beygelzimer and Rish, 2003] Alina Beygelzimer and Irina Rish. Approximability of probability distributions. In *Proc. of the 17th Conf. on Neural Information Processing Systems*, Vancouver, BC, Canada, December 2003.

[Bishop, 1995] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, 1995.

[Borghetti *et al.*, 2006] Brett Borghetti, Eric Sodomka, Maria Gini, and John Collins. A market-pressure-based performance evaluator for tac-scm. In *AAMAS06: Workshop on Trading Agent Design and Analysis (TADA/AMEC)*, 2006.

[Bradshaw *et al.*, 1997] Jeffrey M. Bradshaw, Stewart Dutfield, Pete Benoit, and John D. Wolley. KAoS: Toward an industrial-strength open agent architecture. In Jeffrey M. Bradshaw, editor, *Software Agents*, pages 375–418. AAAI Press, 1997.

[Bradshaw, 1997] Jeffrey M. Bradshaw, editor. *Software Agents*. AAAI Press, 1997.

[Brazier *et al.*, 2002] Frances M. T. Brazier, Catholign M. Jonker, and Jan Treur. Principles of component-based design of intelligent agents. *Data and Knowledge Engineering*, 41(1):1–28, April 2002.

[Brooks *et al.*, 2002] Christopher H. Brooks, Robert Gazzale, Rajarshi Das, Jeffrey O. Kephart, Jeffrey K. MacKie-Mason, and Edmund H. Durfee. Model selection in an information economy: Choosing what to learn. *Computational Intelligence*, April 2002.

[Brown *et al.*, 1961] Robert G. Brown, Richard F. Meyer, and D. A. D'Esopo. The fundamental theorem of exponential smoothing. *Operations Research*, 9(5):673–687, 1961.

[Buschmann *et al.*, 1996] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-Oriented Software Architecture: a System of Patterns*. Wiley, 1996.

[Carmel and Markovitch, 1993] D. Carmel and S. Markovitch. Learning models of opponent's strategy in game playing. Technical report, Technion-Israel Institute of Technology, 1993.

[Chajewska *et al.*, 2001] Urszula Chajewska, Daphne Koller, and Dirk Ormoneit. Learning an agent's utility function by observing behavior. In *Proc. of the 18th Int'l Conf. on Machine Learning*, pages 35–42, Lafayette, June 2001.

[Chavez and Maes, 1996] Anthony Chavez and Pattie Maes. Kasbah: An agent marketplace for buying and selling goods. In *Proc. of the First Int'l Conf. on the*

Practical Application of Intelligent Agents and Multi-Agent Technology, pages 75–90, London, UK, April 1996. Practical Application Company.

[Cherkassky and Mulier, 1998] Vladimir Cherkassky and Filip Mulier. *Learning from data - Concepts, Theory, and Methods*. John Wiley & Sons, INC., New York, 1998.

[Choi and Liu, 2001] Samuel P. M. Choi and Jiming Liu. A dynamic mechanism for time-constrained trading. In *Proc. of the Fifth Int'l Conf. on Autonomous Agents*, pages 568–575, 2001.

[Collins *et al.*, 2001] John Collins, Corey Bilot, Maria Gini, and Bamshad Mobasher. Decision processes in agent-based automated contracting. *IEEE Internet Computing*, 5(2):61–72, March 2001.

[Collins *et al.*, 2002a] John Collins, Wolfgang Ketter, and Maria Gini. A multi-agent negotiation testbed for contracting tasks with temporal and precedence constraints. *Int'l Journal of Electronic Commerce*, 7(1):35–57, 2002.

[Collins *et al.*, 2002b] Michael Collins, Robert E. Schapire, and Yoram Singer. Logistic Regression, AdaBoost and Bregman Distances. *Journal of Machine Learning*, 48, 2002.

[Collins *et al.*, 2004] John Collins, Raghu Arunachalam, Norman Sadeh, Joakim Ericsson, Niclas Finne, and Sverker Janson. The Supply Chain Management Game for the 2005 Trading Agent Competition. Technical Report CMU-ISRI-04-139, Carnegie Mellon University, Pittsburgh, PA, December 2004.

[Collins *et al.*, 2005] John Collins, Raghu Arunachalam, Norman Sadeh, Joakim Ericsson, Niclas Finne, and Sverker Janson. The Supply Chain Management Game for the 2006 Trading Agent Competition. Technical Report CMU-ISRI-05-132, Carnegie Mellon University, Pittsburgh, PA, November 2005.

[Collins *et al.*, 2007] John Collins, Wolfgang Ketter, Maria Gini, and Amrudin Agovic. Software architecture of the MinneTAC supply-chain trading agent. Technical Report 07-006, University of Minnesota, Dept of Computer Science and Engineering, Minneapolis, MN, February 2007.

[CombineNet, 2006] CombineNet. Sourcing solutions. http://www.combinenet.com/sourcing_solutions/, 2006.

[Dahlgren and Wurman, 2004] Erik Dahlgren and Peter Wurman. PackaTAC: A conservative trading agent. *SIGecom Exchanges*, 4(3):33–40, 2004.

[Dempster *et al.*, 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Stat. Soc.*, Series B, 39(1):1–38, 1977.

[Duda *et al.*, 2000] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[Foundation, 2006] Apache Software Foundation. Apache excalibur. http://excalibur.apache.org/, 2006.

[Ghani, 2005] Rayid Ghani. Price prediction and insurance for online auctions. In *Int'l Conf. on Knowledge Discovery in Data Mining*, pages 411–418, Chicago, Illinois, August 2005.

[Ghose *et al.*, 2006] Anindya Ghose, Michael D. Smith, and Rahul Telang. Internet exchanges for used books: An empirical analysis of product cannibalization and welfare impact. *Information Systems Research*, 17(1):3–19, 2006.

[Gibbons, 1986] Jean Dickinson Gibbons. Nonparametric statistical inference. *Technometrics*, 28(3):275, 1986.

[Gupta *et al.*, 1996] Alok Gupta, Dale O. Stahl, and Andrew Whinston. Economic Issues in Electronic Commerce. In Ravi Kalakota and Andrew B. Whinston, editors, *Readings in Electronic Commerce*, pages 197–227, Reading, MA, 1996. Addison Wesley.

[Gupta *et al.*, 1997] Alok Gupta, Dale O. Stahl, and Andrew Whinston. The Internet: A Future Tragedy of the Commons? In H. Amman, B. Rustem, and A. B. Whinston, editors, *Computational Approaches to Economic Problems*, pages 347–361, Dordrecht, The Netherlands, 1997. Kluwer Academic Publishers.

[Gupta *et al.*, 2000] Alok Gupta, Boris Jukic, Dale O. Stahl, and Andrew Whinston. Extracting Consumers' Private Information for Implementing Incentive Compatible Internet Traffic Pricing. *Journal of Management Information Systems*, 17(1):9–29, 2000.

[He *et al.*, 2006] Minghua He, Alex Rogers, Xudong Luo, and Nicholas R. Jennings. Designing a successful trading agent for supply chain management. In *Proc. of the Fifth Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, pages 1159–1166, May 2006.

[Hillier and Lieberman, 1990] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research*. McGraw-Hill, 1990.

[Hollander and Wolfe, 2000] Myles Hollander and Douglas A. Wolfe. Nonparametric statistical methods. *Journal of the American Statistical Association*, 95(449):333, 2000.

[I2, 2006] I2. Next-generation planning. http://i2.com/solution_library/ng_planning.cfm, 2006.

[Isaacson and Madsen, 1976] Dean L. Isaacson and Richard W. Madsen. *Markov Chains – Theory and Applications*. John Wiley & Sons, 1976.

[Jaiswal *et al.*, 2003] Ashutosh Jaiswal, Yongdae Kim, and Maria Gini. Security model for a multi-agent marketplace. In *Proc. of 5th Int'l Conf. on Electronic Commerce*, pages 119–124, Pittsburgh, PA, October 2003.

[Jordan *et al.*, 2006] Patrick R. Jordan, Christopher Kiekintveld, Jason Miller, and Michael P. Wellman. Market efficiency, sales competition, and the bullwhip effect in the tac scm tournaments. In *AAMAS06: Workshop on Trading Agent Design and Analysis (TADA/AMEC)*, 2006.

[Karacapilidis and Moraïtis, 2001] Nikos Karacapilidis and Pavlos Moraïtis. Intelligent agents for an artificial market system. In *Proc. of the Fifth Int'l Conf. on Autonomous Agents*, pages 592–599, May 2001.

[Kearl, 1989] James R. Kearl. *Contemporary Economics*. Scott, Foresman and Company, 1989.

[Kearns and Ortiz, 2003] Michael Kearns and Luis Ortiz. The Penn-Lehman Automated Trading Project. *IEEE Intelligent Systems*, pages 22–31, 2003.

[Keller *et al.*, 2004] Philipp W. Keller, Felix-Olivier Duguay, and Doina Precup. Redagent - winner of the TAC SCM 2003. *SIGecom Exchanges*, 4(3):1–8, 2004.

[Kephart *et al.*, 2000] Jeffrey O. Kephart, James E. Hanson, and Amy R. Greenwald. Dynamic pricing by software agents. *Computer Networks*, 32(6):731–752, 2000.

[Ketter *et al.*, 2003] Wolfgang Ketter, Alexander Babanov, and Maria Gini. An evolutionary framework for studying behaviors of economic agents. In *Proc. of the Second Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, pages 1030–1031, Melbourne, Australia, July 2003.

[Ketter *et al.*, 2004a] Wolfgang Ketter, Elena Kryzhnyaya, Steven Damer, Colin McMillen, Amrudin Agovic, John Collins, and Maria Gini. Analysis and design of supply-driven strategies in TAC-SCM. In *Workshop: Trading Agent Design and Analysis at the Third Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, pages 44–51, New York, July 2004.

[Ketter *et al.*, 2004b] Wolfgang Ketter, Elena Kryzhnyaya, Steven Damer, Colin McMillen, Amrudin Agovic, John Collins, and Maria Gini. Design and analysis of the MinneTAC-03 Supply-Chain Trading Agent. Technical Report 04-016, University of Minnesota, Dept of Computer Science and Engineering, Minneapolis, MN 55455, 2004.

[Ketter *et al.*, 2004c] Wolfgang Ketter, Elena Kryzhnyaya, Steven Damer, Colin McMillen, Amrudin Agovic, John Collins, and Maria Gini. MinneTAC sales strategies for supply chain TAC. In *Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, pages 1372–1373, New York, July 2004.

[Ketter *et al.*, 2005] Wolfgang Ketter, John Collins, Maria Gini, Alok Gupta, and Paul Schrater. A Computational Approach to Predicting Economic Regimes in Automated Exchanges. In *Proc. of the Fifteenth Annual Workshop on Information Technologies and Systems*, pages 147–152, Las Vegas, Nevada, USA, December 2005.

[Kiekintveld *et al.*, 2004] Christopher Kiekintveld, Michael P. Wellman, Satinder Singh, Joshua Estelle, Yevgeniy Vorobeychik, Vishal Soni, and Matthew Rudary. Distributed feedback control for decision making on supply chains. In *Int'l Conf. on Automated Planning and Scheduling*, pages 384–392, Whistler, BC, Canada, June 2004.

[Kiekintveld *et al.*, 2005] Christopher Kiekintveld, Yevgeniy Vorobeychik, and Michael P. Wellman. An Analysis of the 2004 Supply Chain Management Trading Agent Competition. In *IJCAI 2005 Workshop on Trading Agent Design and Analysis*, pages 61–70, Edinburgh, Scotland, August 2005.

[Kullback and Leibler, 1951] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[Kullback, 1959] Solomon Kullback. *Information Theory and Statistics*. Dover Publications, New York, 1959.

[Lawrence, 2003] Richard Lawrence. A machine learning approach to optimal bid pricing. In *8th INFORMS Computing Society Conf. on Optimization and Computation in the Network Era*, pages 1–22, Arizona, January 2003.

[Levinson, 1986] S E Levinson. Continuously variable duration hidden markov models for automatic speech recognition. *Comput. Speech Lang.*, 1(1):29–45, 1986.

[Littman, 1994] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning (ML-94)*, pages 157–163, New Brunswick, NJ, 1994. Morgan Kaufmann.

[Mark and Perrault, 2006] Bill Mark and Raymond C. Perrault. Calo: Cognitive assistant that learns and organizes. http://www.ai.sri.com/project/CALO, 2006.

[Massey and Wu, 2005] Cade Massey and George Wu. Detecting regime shifts: The causes of under- and overestimation. *Management Science*, 51(6):932–947, 2005.

[McCallum and Nigam, 1988] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *Proceedings of the 7th National Conference on Artificial Intelligence in Workshop on Learning for Text Categorization*, St. Paul, MN, August 1988.

[McMillen, 2003] Colin McMillen. Toward the development of an intelligent agent for the supply chain management game of the 2003 trading agent competition. Undergraduate honors thesis, University of Minnesota, Minneapolis, MN, 2003.

[Mitchell, 1997] Tom M. Mitchell. *Mchine Learning*. McGraw-Hill, 1997. ISBN: 0071154671.

[Moraitis *et al.*, 2003] P. Moraitis, E. Petraki, and N. Spanoudakis. Engineering JADE agents with the Gaia methodology. In R. Kowalszyk, J. Miller, H. Tianfield, and R. Unland, editors, *Agent Technologies, Infrastructures, Tools, and Applications for e-Services*, volume 2592 of *Lecture Notes in Computer Science*, pages 77–91. Springer-Verlag, 2003.

[Nabney, 2001] Ian T. Nabney. *NETLAB Algorithms for Pattern Recognition.* Springer, 2001.

[Ng and Russell, 2000] Andrew Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Proc. of the 17th Int'l Conf. on Machine Learning*, pages 663–670, Palo Alto, June 2000.

[Norman *et al.*, 1997] Timothy J. Norman, Nicholas R. Jennings, Peyman Faratin, and E. H. Mamdani. Designing and implementing a multi-agent architecture for business process management. In M. J. Wooldridge, J. P. Müller, and N. R. Jennings, editors, *Intelligent Agents III*, volume 1193 of *Lecture Notes in Artificial Intelligence*, pages 261–275. Springer-Verlag, Berlin, 1997.

[Osborn and Sensier, 2002] Denise R. Osborn and Marianne Sensier. The prediction of business cycle phases: financial variables and international linkages. *National Institute Econ. Rev.*, 182(1):96–105, 2002.

[Papaioannou and Cassaigne, 2000] V. Papaioannou and N. Cassaigne. A critical analysis of bid pricing models and support tool. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Piscataway, NJ, 2000.

[Pardoe and Stone, 2004] David Pardoe and Peter Stone. Bidding for Customer Orders in TAC SCM: A Learning Approach. In *Workshop on Trading Agent Design and Analysis at AAMAS*, pages 52–58, New York, July 2004.

[Pardoe and Stone, 2006] David Pardoe and Peter Stone. Tactex-05: A champion supply chain management agent. In *Proc. of the Twenty-First National Conference on Artificial Intelligence*, pages 1389–1394, Boston, Mass., July 2006. AAAI.

[Pauwels and Hanssens, 2002] Koen Pauwels and Dominique Hanssens. Windows of Change in Mature Markets. In *European Marketing Academy Conf.*, Braga, Portugal, May 2002.

[Pauwels and Hanssens, 2004] Koen Pauwels and Dominique Hanssens. Performance Regimes and Marketing Policy Shifts. In *Marketing Science*, page submitted, April 2004.

[Pindyck and Rubinfeld, 1998] Robert S. Pindyck and Daniel L. Rubinfeld. *Econometric Models and Econometric Forecasts; 4th Edition.* Irwin/McGraw-Hill, 1998.

[Russell and Norvig, 2002] Stuart Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach; 2nd Edition.* Prentice Hall, 2002. ISBN: 0137903952.

[Sandholm, 2006] Tuomas Sandholm. Expressive commerce and its application to sourcing. In *Proc. of the Twenty-First National Conference on Artificial Intelligence*, pages 1736–1743, Boston, MA, July 2006. AAAI.

[Schapire *et al.*, 2002] Robert E. Schapire, Peter Stone, David McAllester, Michael L. Littman, and János A. Csirik. Modeling auction price uncertainty using boosting-based conditional density estimation. In *Proceedings of the Nineteenth International Conference on Machine Learning*, Sydney, Australia, July 2002.

[Shannon, 1948] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423 and 623–656, July and October 1948.

[Shoham, 1997] Yoav Shoham. An overview of agent-oriented programming. In Jeffrey M. Bradshaw, editor, *Software Agents*, pages 271–290. AAAI Press, 1997.

[Sodomka *et al.*, 2007] Eric Sodomka, John Collins, and Maria Gini. Efficient statistical methods for evaluating trading agent performance. Technical Report 07-005, University of Minnesota, Dept of Computer Science and Engineering, Minneapolis, MN, February 2007.

[Stone and Greenwald, 2005] Peter Stone and Amy Greenwald. The first international trading agent competition: Autonomous bidding agents. *Electronic Commerce Research*, 5(1):229–64, January 2005. To appear.

[Stone *et al.*, 2003] Peter Stone, Robert E. Schapire, Michael L. Littman, János A. Csirik, and David McAllester. Decision-theoretic bidding based on learned density models in simultaneous, interacting auctions. *Journal of Artificial Intelligence Research*, 19:209–242, 2003.

[Stone, 2003] Peter Stone. Multiagent competitions and research: Lessons from RoboCup and TAC. In Gal A. Kaminka, Pedro U. Lima, and Raul Rojas, editors, *RoboCup-2002: Robot Soccer World Cup VI*, pages 224–237. Springer Verlag, Berlin, 2003.

[Sun *et al.*, 2004] Shuang Sun, Viswanath Avasarala, Tracy Mullen, and John Yen. PSUTAC: A Trading Agent Designed from Heuristics to Knowledge. In *Workshop on Trading Agent Design and Analysis at AAMAS*, pages 15–20, New York, July 2004.

[Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning - An Introduction*. The MIT Press, Cambridge, 1998.

[Sycara and Pannu, 1998] Katia Sycara and Anandeep S. Pannu. The RETSINA multiagent system: towards integrating planning, execution, and information gathering. In *Proc. of the Second Int'l Conf. on Autonomous Agents*, pages 350–351, 1998.

[Szyperski, 1998] Clemens Szyperski. *Component Software: Beyond Object-Oriented Programming*. ACM Press, 1998.

[Titterington *et al.*, 1985] D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, New York, 1985.

[Vetsikas and Selman, 2003] Joannis A. Vetsikas and Bart Selman. A principled study of the design tradeoffs for autonomous trading agents. In *Proc. of the Second Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, 2003.

[von Neumann and Morgenstern, 1947] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 2nd edition, Princeton, N.J., 1947.

[Weiss, 1999] Gerhard Weiss. *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, Massachusetts, 1999.

[Wellman and Wurman, 1998] Michael P. Wellman and Peter R. Wurman. Market-aware agents for a multiagent world. *Robotics and Autonomous Systems*, 24:115–125, 1998.

[Wellman *et al.*, 2001] Michael P. Wellman, Peter R. Wurman, K. O'Malley, R. Bangera, S. Lin, D. Reeves, and William E. Walsh. Designing the market game for a trading agent competition. *IEEE Internet Computing*, 5(2):43–51, 2001.

[Wellman *et al.*, 2003] Michael P. Wellman, Daniel M. Reeves, Kevin M. Lochner, and Yevgeniy Vorobeychik. Price prediction in a trading agent competition. *Journal of Artificial Intelligence Research*, 2003.

[Wellman *et al.*, 2005] Michael P. Wellman, Daniel M. Reeves, Kevin M. Lochner, Shih-Fen Cheng, and Rahul Suri. Approximate strategic reasoning through hierarchical reduction of large symmetric games. In *Proc. of the Twentieth National Conference on Artificial Intelligence*, Pittsburgh, 2005.

[Wellman *et al.*, 2006] Michael P Wellman, Patrick R. Jordan, Christopher Kiekintveld, Jason Miller, and Daniel M. Reeves. Empirical game-theoretic analysis of the tac market games. In *AAMAS06: Workshop on Trading Agent Design and Analysis (TADA/AMEC)*, 2006.

[Witten and Frank, 2000] Ian H. Witten and Eibe Frank. *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers, San Francisco, CA, 2000.

[Zhang and Cheung, 2005] Xiaofeng Zhang and William K. Cheung. Learning global models based on distributed data abstractions. In *Nineteenth International Joint Conference on Artificial Intelligence*, pages 1645–1646, Edinburgh, Scottland, August 2005.

[Zhang *et al.*, 2004] Dongmo Zhang, Kanghua Zhao, Chia-Ming Liang, Gonelur Begum Huq, and Tze-Haw Huang. Strategic trading agents via market modeling. *SIGecom Exchanges*, 4(3):46–55, 2004.