Applications of Classifying Bidding Strategies for the CAT Tournament

Mark L. Gruman and Manjunath Narayana

Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{mgruman, narayana}@cs.umass.edu

Abstract

In the CAT Tournament, specialists facilitate transactions between buyers and sellers with the intention of maximizing profit from commission and other fees. Each specialist must find a well-balanced strategy that allows it to entice buyers and sellers to trade in its market while also retaining the buyers and sellers that are currently subscribed to it. Classification techniques can be used to determine the distribution of bidding strategies used by all traders subscribed to a particular specialist. Our experiments showed that Hidden Markov Model classification yielded the best results. The distribution of strategies, along with other competition-related factors, can be used to determine the optimal action in any given game state. Experimental data shows that the GD and ZIP bidding strategies are more volatile than the RE and ZIC strategies, although no traders ever readily switch specialists. An MDP framework for determining optimal actions given an accurate distribution of bidding strategies is proposed as a motivator for future work.

1 Introduction

The field of Catallactics, or the science of exchanges, has received significant attention in the Artificial Intelligence community over the past few years, in large part, due to increasing use of e-commerce environments such online auctions and ticket vendors. In particular, significant attention has been given to designing efficient markets in which traders of numerous roles and preferences interact and exchange goods while utilizing various bidding strategies and trading tactics. The CAT Tournament [1], an offshoot of the original Trading Agents Competition introduced in 2007, is a contest in which markets (hereon referred to as "specialists") attempt to lure buyers and sellers (hereon collectively referred to as "traders") to their respective trading platforms in hopes of maximizing profit. Unlike the original TAC Classic and TAC SMC competitions in which specialists fulfilled stationary requests, CAT specialists must respond to a variety of bidding techniques employed by the traders who also wish to maximize their own profits; this dynamic environment serves as the main motivation for developing adaptive markets that actively respond to the traders' ever-changing preferences.

Successful specialist design requires a balanced decision-making strategy that entices new traders to subscribe to the specialist while also retaining existing traders. One method of developing such a strategy involves creating a model for every trader and determining how each action affects each model. This approach is highly infeasible, however, because the specialist does not receive any information regarding the traders other than which ones are currently subscribed to it; all incoming bids are masked before they reach the specialist, so the specialist is unable to definitively link each bid with a particular trader. Creating a model for each trader is also highly ineffective given that there may be hundreds of traders interacting with one another. Processing hundreds of models can take a significant amount of time on even the most powerful systems and may require more system memory .than is available.

One important feature of the CAT Tournament is that all traders use strictly one of four previously-defined bidding strategies. In this paper we describe how classification techniques can be used to exploit the fact that traders must use one of four bidding strategies, reducing the number of models required to accurately represent all traders to just four. We provide experimental results indicating how certain actions affect the trader pool, especially groups of traders utilizing the same bidding strategy. We also discuss how these group models can be used to train the specialist, allowing it make decisions quickly during the competition.

The paper first provides a brief description of each bidding strategy in Section 2. The benefits of classifying traders according to bidding strategies are described in detail in Section 3, along with a comprehensive analysis of various classification techniques that have been applied to this problem and their final results. Section 4 briefly describes how classification (in conjunction with other game factors) can be used to determine the optimal action the specialist should take at any particular point in the competition. Experimental results are presented in Section 5, followed by a brief conclusion and a discussion of possible future work in Section 6.

2 Bidding Strategies

All traders in the CAT Tournament are required to utilize one of four bidding strategies. A brief description of each strategy is provided here, but the reader is urged to consult the original publications (see references [2]-[5]) that describe the strategies in detail.

A trader using the Double Auction strategy [3] (henceforth called GD) keeps track of the number of bids accepted and rejected by the market at a particular price. Subsequent values for the bids are chosen depending on the probability of acceptance of a bid, given the past history. Utilizing the Extensive Form Game strategy [5] (hereon referred to as RE), a trader alters its future bid values based on the profits that were observed for the previous bids. The Zero Information-Constrained (ZI-C) strategy [4] involves generating random bids constrained between a maximum and minimum value. A buyer using the ZI-C strategy will never bid more than what it believes a good is worth. Likewise, a seller will never sell a good for less than the amount it cost the seller to obtain the good. Finally, if a trader uses the Zero Information, Plus (ZIP) strategy [2], it utilizes the same trading techniques as a trader that employs ZI-C, but it also updates the constraints based on feedback from the market. Thus, each process (except when using the ZI-C strategy) receives feedback from the market in various forms and updates itself to generate new bids.

3 Classifying Traders by Bidding Strategies

In the CAT Tournament, all traders must employ one of four previously-defined bidding strategies when placing bids with their respective specialist. With this stipulation, it is reasonable to assume that all traders utilizing the same bidding strategy will behave similarly (at least more so than the other traders). Likewise, any action taken by the specialist, such as raising or lower a particular fee, will most likely have a similar affect on all traders utilizing the same bidding strategy. These assumptions allow the specialist to reason about how its actions may affect an entire group of traders rather than individuals, turning the trader-modeling problem into a classification problem. We discuss the validity of our assumption in the Experimental Results section (5).

For simplicity, we consider only the bids that the specialist has received from its traders. Unfortunately, the data cannot be used for classification in its rawest form because bids are masked before they reach the specialist, making it virtually impossible to determine from which trader each bid originated. As a result, a set of collected data was manually unmasked in order to train and then test various classifiers. Furthermore, based on the assumption that traders act as groups, it is sufficient for the classifier to predict the overall representation of each bidding strategy rather than link each trader to a specific strategy. This subtle difference becomes crucial during the actual competition when all of the bids are again masked and the specialist is unable to determine the origin of each bid it receives.

Bid sequences were collected for 400 traders (100 traders for each strategy), with their identities unmasked. We decided that focusing on the selling trader sequences alone would suffice to evaluate the efficacy of the classification strategy. In all, 2076 bid sequences were generated by the system. Two-thirds (1384) of these samples were randomly chosen to be the training set and one-third of the data (692) was used for testing.

We describe our data collection methods in Section 3.1 and then examine two classification techniques in detail, focusing our attention on SVM Classification in Section 3.2 and HMM Classification in Section 3.3. In Section 3.4 we briefly discuss other classification techniques that were considered but not explored in detail.

3.1 Data Collection

In CAT, each trader makes a bid to the market and continues to update it until another trader in the same market accepts the bid price and a transaction takes place. We call each string of updated bids from the same trader a "bid sequence". Sample bid sequences can be seen in Figure 1. Given the competitive nature of the market with several traders attempting to make the transaction, a buyer's bid could be accepted by a seller at any point during the bid process. This means that the number of bids a trader has to make before successfully concluding a transaction is not constant. As a result, the number of bids in each bid sequence can vary significantly. There is no upper bound on the length of the sequence. The number of bids in the sequence can range from 1 to any large number depending on the state of the market and the strategies of other traders in that market. We witnessed a number of occasions in which bid sequences contained more than 200 bids. The problem is further complicated when we consider multiple traders using different strategies.

```
Seller_GD_1- 84.0, 90.4, 102.9
Seller_RE_1- 93.6
Seller_GD_3- 153.9, 140.5, 75.6, 90.3
Seller_ZIP_1- 100.2, 98.7, 89.6, 77.6, 109.4
Seller_ZIP_4- 59.0, 85.2, 82.7, 81.3, 73.4
Seller_ZIC_3- 34.5, 56.7, 78.9
Seller_GD_3- 152.8, 132,6
...
```

Figure 1: Illustration of bid sequences from a sample market with multiple traders employing different bidding strategies.

The traders' bidding data was collected through numerous simulations of a typical CAT competition[1].

[1] Simulations were run using the TAC Market Design Competition Platform[6], which was obtained at https://sourceforge.net/projects/jcat

Although there were no problems collecting a sufficient amount of data (one could always run more simulations), the "raw" data collected could not immediately be used for classification purposes for a number of reasons. The most significant obstacle of data classification was dealing with anonymized data. By "anonymized" we mean that the true origin of each bid was masked. This occurred per the specification of the CAT Competition Protocol, in which bids shouted by traders first reached the server, which replaced the identity of the bid source with a unique bid identifier used only for that particular sequence of bids. Once a transaction completed, the bid identifier was discarded and a new bid identifier was assigned to the next sequence of bids from the same trader. As a result, all bids were masked by the time they reached the specialist. Thus, the specialist could determine neither the true identity of each bid, nor which bid sequences originated from the same trader.

The random order in which the bids arrived also further complicated the process of determining the true origin of each bid. Fortunately, the CAT source code (freely available to all CAT Competition participants and researchers) gives users access to all functional modules that make up the competition. With these additional resources, and a number of code modifications, the authors were able to obtain the required data in its "unmasked" form, allowing them to identify the true source of each bid that the specialist received. This data was collected under the assumption that it would be used for training only, since the identity of each bid would not be available to the specialist during an actual competition.

3.2 Classification Using a Support Vector Machine

Support Vector Machines are a set of popular classification algorithms that strive to simultaneously minimize classification error and maximize the margin of separation of data [7].

In order to perform classification using the SVM, collected data was first converted to the appropriate data format. Each sequence of bids was represented by a unique vector, and each bid in the sequence became a feature in the corresponding vector. Feature numbers were assigned incrementally, so a bid sequence made up of n bids was represented by a vector of features 1 through n. Each vector was assigned a classification as follows: a bid sequence coming from a trader using the GD strategy was classified as class 1, RE as class 2, ZIP as class 3, and ZIC as class 4. An example of the data can be seen in Figure 2.

SVM training and classification was performed using LibSVM® v2.85 [8]. Prediction results using SVM classification varied greatly, depending largely on the type of kernel that was used for training. Training on the sigmoid kernel (with default gamma and coefficient values) yielded the worst results, predicting only 28.2% of the testing set correctly (only 3% better than completely random prediction). Training under the linear and polynomial kernels also yielded rather poor results, predicting only 32.8% and 38.4% of the testing data correctly, respectively. The sigmoid kernel, however, produced much better results, predicting 53.8% of the testing data correctly under default parameters and 59.7% of the data correctly when gamma was set to 0.8. An observation was also made that the GD and ZI-C strategies were predicted with a high degree of accuracy, while data from the ZIP and RE strategies was more difficult to classify.

```
1   1 : 83.02029493660217
1   1 : 84.91796358721331
1   1 : 113.72054114472121   2 : 116.14283320959657   3 : 116
1   1 : 84.9176358721331
. . .
2   1 : 112.84017410967499
2   1 : 192.3076814096421    2 : 111.3076814096421
. . .
3   1 : 89.74849000047514
3   1 : 109.12522211910922
3   1 : 80.72191004970509
3   1 : 139.5462076581298    2 : 107.85463726359954
. . .
4   1 : 95.08454458555207
4   1 : 91.39030987862498
4   1 : 94.43402012111619
```

Figure 2: Bid sequence data converted to SVM format. The number of bids in each sequence varied and was heavily dependent on the bidding strategy utilized by each bidding trader.

3.3 Classification Using a Hidden Markov Model

Hidden Markov Models are graphical models that can be used to model the underlying process that generates a given set of data [9]. They are most widely used for classifying time-series data, (e.g. speech processing [10]).

HMM-based classification showed a slight improvement over the SVM-based method that was used earlier, supporting the authors' expectation that a Hidden Markov Model would most effectively model the variables involved in the bidding process. Several HMM runs were executed with different values for the parameters (number of hidden states and mixture components). A summary of the results can be seen in Figure 3. The HMM models took between 10 and 68 minutes to train depending on the number of states and mixture components. In contrast, SVM training ranged between only a few seconds and 2-3 minutes. Nevertheless, the accuracy for the HMM method was always greater than 52%, and the authors were able to achieve about 62% accuracy by tuning the parameters (number of hidden states = 10, number of mixture components = 10).

A small experiment was also carried out to explore the efficacy of feature reduction of the observed dataset in the HMM classification framework. We could not use standard reduction methods like PCA because the dataset included instances of varying feature lengths (treating each bid as a single feature). Most samples had only one feature, while some had as many as 200. Experiments showed that when only two features were used, the HMM accuracy fell to 49%, while using only the first 10 features improved upon earlier best results slightly. Thus, we concluded that a moderate reduction in the number of features could result in improved performance while also maintaining the integrity of the observed data.

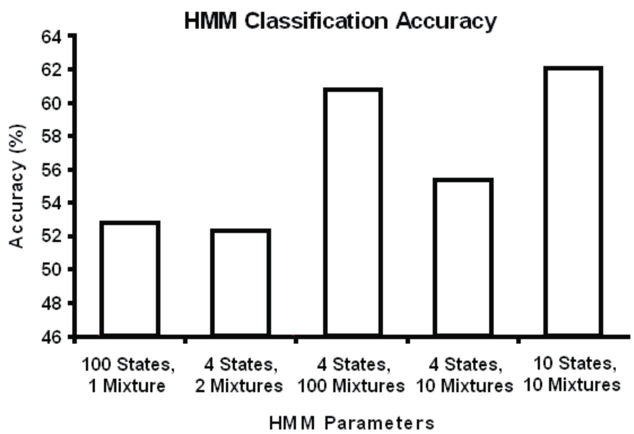

Figure 3: HMM Classification results for varying parameters

3.4 Alternative Classification Techniques

A number of other interesting methods were explored for improving the results of the HMM. We tried to fit a Conditional Random Fields (CRF) model to the data, using the *CRF Toolbox for MATLAB®* source code provided by Professor Kevin Murphy (U. of B.C.). The model failed to converge in many cases, however, and resulted in accuracy that was not much better than random performance.

Since SVM is the most popular and successful classification method in most applications, we decided to also try the time-dependent Fourier kernel [11]. We thought that the power of SVM when combined with some time information provided by the Fourier kernel might show significant improvement in the classification accuracies. A Fourier kernel was calculated for all the instances in the training and testing data set, as required by the LibSVM framework for implementing user-defined kernels in LibSVM. Accuracy of between 28% and 42% was observed for different cases of the Fourier kernel, suggesting that it was not an improvement over the other kernel functions of the SVM framework.

We briefly contemplated the use of the pyramid kernel [12] because it is designed to work with datasets that have a varying number of features. However, due to a lack of time we decided to pursue this as future work.

4 Utilizing Strategy Classification to Determine Optimal Action Policies

An accurate model of CAT is required for the specialist to make sound decisions throughout the competition. The specialist is privy to a large amount of data, most notably every bid it receives from its traders, but it is not immediately clear how this data should be represented in the game model. Clearly, modeling every individual bid is not only impractical but also very likely infeasible and probably unnecessary. The classification techniques we discussed in the previous section allow us to model all bids using a simple yet highly-descriptive distribution. We can then use this distribution, along with a number of other observable factors, to uniquely and correctly identify the *state* of the competition and take the *action* that is deemed optimal for that state.

Preliminary data suggested that an accurate description of the state of the CAT competition should include at least the following characteristics:

State:

1) **Distribution ::= bidding strategy combinations**
2) **Position ::= numerical score position**
3) **Trader Count ::= 0 – total_trader_count**
4) **Current Fees ::= 0 – ∞ flat fees, 0 – 1 profit fee**

Utilizing the classification results from Section 3, each bidding strategy combination can be described in a sequence of bidding strategies (most popular, 2nd most popular, 3rd most popular, and least popular), thus giving us 24 (4!) total combinations.

In addition to the distribution of bidding strategies, the specialist should consider a number of other observable factors such as Position, Trader Count, and Current Fees. All of these factors may be subject to a wide range of values or may not be bounded at all, so some sort of mapping is required to reduce the number of total possible states to a finite and practical quantity. For example, it may be sufficient to categorize Trader Count based on a particular range of total traders (e.g. 0-10% of all traders, 10-20%, etc…). Likewise, the Current Fees factor can be classified in relation to some numerical constants determined to be "threshold boundaries" for certain groups of traders.

The actions each specialist can take can be limited to raising, lowering, or maintaining each of the five fees the specialist charges each trader. However, because four of the five fees do not have an upper bound, it does not make sense to enumerate the actions based on raw values (of which there are infinitely many). Similar to the Trader Count factor used to define the state of the CAT Tournament, raw actions must also be mapped to produce a finite and enumerable set of distinct actions. One such mapping involves simply determining whether each fee has been raised, lowered, or unmodified, resulting in 3^5, or 243 unique actions. We describe our implementation of this framework in section 5.4.

5 Experimental Results

A set of experimental test runs was executed to determine if certain actions had a more profound effects on specific groups of traders; they yielded a number of interesting properties for various bidding strategies. We also describe our attempt to determine optimal bidding policies using a Markov Decision Process (MDP) framework outlined earlier.

We first describe the environment in which our experiments were executed, as well as the algorithm that was implemented to adjust fees throughout the experiments.

5.1 Testing Environment

All non-clustering experiments were run on a Compaq C712NR laptop with the following specifications:

Intel® Pentium® Dual-core CPU T2310 @ 1.46GHz
1 GB DDRAM, 789 MHz
Windows XP with Service Pack 2
Java™ SE Runtime Environment (build 1.6.0_05-b13)

All CAT agents (server, specialists, and traders) were run on the same machine. The server and traders were run using the *tournament.params* parameter file provided with the CAT source code. Important features included:

400 total traders (100 for each bidding strategy)
200 buyers, 200 sellers (8 groups of 50 traders total)
Game length of 4000 days (usually terminated earlier)
Day length of 20 rounds
Round length of 1000 milliseconds

All experiments were run with a total of 5 specialists. To simulate a realistic tournament environment, publicly available binaries from the 2007 CAT Tournament[2] were used as specialist adversaries, specifically *CrocodileAgent*, *jackaroo*, *PersianCat*, and *TaxTec*.

5.2 Fee Adjustments

Following the discussion in the previous section, fees were randomly adjusted in one of three ways:

i) increase a fee by a factor of 2
ii) decrease a fee by a factor of 2
iii) retain an existing fee

The likelihood of all outcomes was set to the same frequency (1/3), although it relied heavily on Java's implementation of the generating random values of type Double. Our fee-adjustment algorithm ensured that the profit fee would not surpass 100% (per the specification of the CAT protocol) and set an arbitrary amount for a fee when it was being increased from 0, since increasing 0 by a factor of two again results in 0.

Finally, a special "stabilization" algorithm was implemented for a subset of the experiments. Specifically, the algorithm maximized the specialist's chances of regaining traders if the number of subscribed traders reached 0. In this case, all fees were immediately reduced to 0 and maintained at that level until at least 10% of the trader pool was again subscribed to the specialist.

Algorithm for Non-Stabilizing Fee Increases:

```
funct IncreaseFee(fee):
  if (increase_fee)
    if (fee > 0)
      fee = fee * 2.0;
      if (fee > 1.0 && isProfitFee(fee))
        fee = 1.0;
    else
      if (fee.type == profitFee)
        fee = 0.1;
      else
        fee = 1.0;
```

5.3 Experiments

A number of experiments were run and a large amount of data was collected. We separate the results we deemed most interesting into the following categories: largest increases in traders, largest decreases in traders, and largest discrepancy in trader strategies.

I. Largest Increases in Trader Count

Unsurprisingly, some of the largest increases in trader count came during the first day of a stabilization sequence when all fees were reset to 0. The increase in traders ranged from 9 to 24 traders. The traders that immediately subscribed to the stabilizing specialist represented all of the bidding strategies fairly equally, although the total number of recently-joined sellers was often higher than the total number of recently-joined buyers.

More surprisingly, a continued state of stabilization did not yield a constant increase in traders. In one case it took 356 trading days before the stabilizing specialist had regained 10 percent of the total trader pool (although it had regained 9% of the trader pool in 191 days). This result, presented in Figure 4, conveyed that once traders had settled upon a particular specialist a significant decrease in fees of another specialist was not sufficient in tearing the traders away from their host, and only action taken by the host specialist resulted in traders looking for another specialist.

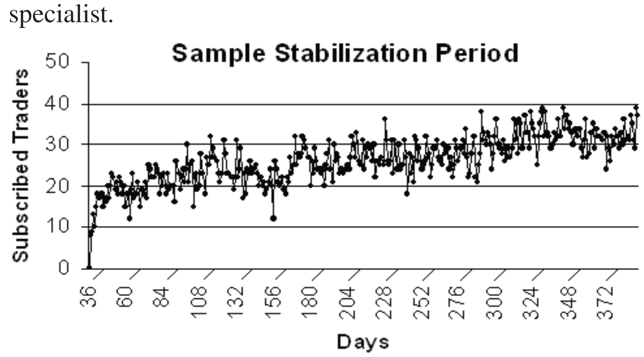

Figure 4: Stabilization sequence which lasted 356 days

II. Largest Decreases in Trader Count

Analogous to the largest increases, the most significant decreases in trader count occurred when multiple fees

[2] Binaries of 2007 CAT Tournament specialists were obtained from http://www.sics.se/tac/showagents.php

increased simultaneously, sometimes bringing the specialist's trader count to zero and initiating a stabilization sequence.

Interestingly, an increase in multiple fees almost always resulted in a significant decrease in trader count regardless of whether other fees had decreased or remained the same. Additionally, the data suggested that trader count decreased when multiple fees increase regardless of the actual amount by which the fees rose. For example, three fees doubling from 1.0 to 2.0 during one day resulted in a decrease in trader count with the same magnitude as that of three fees doubling from 16 to 32. A small link appeared to exist between a decrease in the registration and information fees and a decrease in the buyers using GD bidding and sellers using RE bidding.

III. Largest Discrepancy in Bidding Strategies

After modifying the provided CAT source code, we were able to identify the bidding strategies of all traders placing bids through our specialist. This allowed us to analyze the distribution of strategies at any point during the CAT competition and yielded some interesting results. Here we note that this information is not available during the actual competition when bids are masked, so the results we have gathered should be used only to identify general properties of the bidding strategies.

Our first observation was that traders employing the GD and ZIP bidding strategies were generally more volatile than traders who were utilizing the RE and ZIC bidding strategies. For example, during a sequence of days in which the total count of traders decreased we observed that the percentage of total traders utilizing the ZIP strategy nearly doubled while the percentage of total traders utilizing the GD strategy decreased by nearly a factor of 2 (see Figure 5).

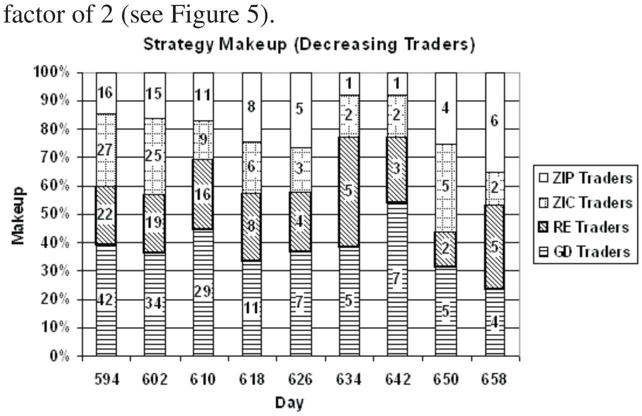

Figure 5: Bidding strategy makeup during a sequence of decreasing traders

In the same experiment, we also observed the representation of each bidding strategy during a sequence of days in which the total number of traders gradually increased. Under this scenario, the percentage of total traders utilizing the ZIP strategy decreased by a factor of 2 while the proportion of traders utilizing the GD strategy nearly doubled (see Figure 6). Although the proportion of traders employing the other two strategies also changed during the same sequences, the changes were not reciprocal of one another when comparing the sequences.

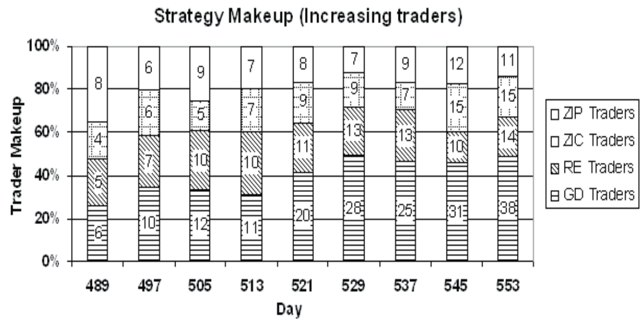

Figure 6: Bidding strategy makeup during a sequence of increasing traders.

5.4 Determining Optimal Actions Using an MDP

Very early in our testing cycle we observed that CAT states possess the Markov Property, suggesting that a Markov Decision Process (MDP) framework could be used to determine the optimal action for each state. Thus, we attempted to address the decision-making problem presented in section 4 with the following MDP definition:

<u>CAT MDP Definition</u>:

Let $< S, A, R, P >$ represent the MDP decision-making problem in the CAT tournament, where

- S is a CAT state, further decomposed into 4 criterion:
  - i) Distribution ::= S_1, S_2, S_3, S_4, represents the strategies utilized by the subscribed traders in decreasing order of popularity.
  - ii) Position ::= <FIRST, MIDDLE, LAST>, refers to the specialist's overall score when compared with other specialists' scores.
  - ii) Traders ::= <0-10%, 10-20%, 20-30%, 30%+>, represents the percentage of all traders subscribed to our specialist.
  - iv) Fees ::= <Flat Fees | Profit Fee>, where
    Flat::=<0, 0–1, 1–20, 20–100, 100–1000, 1000+>
    Profit ::= <0, 0-0.25, 0.25-0.5, 0.5-0.75, 0.75-1, 1>
- A is a CAT action, represented by the following tuple: $<f_1, f_2, f_3, f_4, f_5>$ where f_x ::= <raise, lower, keep> represents an action for each of the five fees.
- R is the reward function for each CAT state. Since the ultimate goal in CAT is to maximize profit, we defined the reward of each state to be the profit earned during the most recent day of trading.
- P is the transition probability matrix for every pair of states. Matrix values were experimentally obtained.

After a number of experiments[3] we realized that our state definition was not sufficient in accurately representing the CAT game state. We based this conclusion on the observation that the reward of a given state often varied greatly (often by thousands of points), suggesting that our states were not sufficiently unique. It was not immediately obvious what additional criteria could be used to better identify CAT game states and remains a top priority for future work.

6 Conclusion and Future Work

We present a number of techniques for classifying traders according to their bidding strategies and show that a Hidden Markov Model yields the best results. Our experimental data presents a number of conclusions regarding fees and how they affect bidding strategies. Most notably, we demonstrate that traders utilizing the GD and ZIP strategies are more volatile than those employing the RE a ZIC strategies. We also note that multiple fee increases generally lead to a loss of traders. Finally, we show evidence supporting the claim that traders are hesitant about switching, regardless of their strategies.

We also proposed a model for discovering optimal action policies while also exploiting the strategy-specific properties presented here. Although our framework appears to support the Markov Property (suggesting that MDP-related algorithms would do well in determining optimal action policies), the criteria we chose were not sufficient to uniquely identify CAT game states. Establishing which criteria should be used to identify CAT states remains an interesting and unsolved problem. Additional experiments should also be performed to determine if bidding strategies can be further exploited. Especially interesting is the interrelationship of the various bidding properties and whether or not certain actions affect the strategies in a similar fashion.

Acknowledgements

First and foremost, we would like to thank Professor Shlomo Zilberstein for his support of this project over the past two years, including regular meetings regarding research progress and advice regarding methodology. We would also like to thank Professor Sridhar Mahadevan for providing us with an opportunity to discuss our work and how it relates to Machine Learning, and for providing us with a MATLAB® implementation of the Expectation-Maximization Algorithm using a Hidden Markov Model. Finally, we would also like to thank Professor Kevin Murphy for his CRF Toolbox source code, which allowed us to evaluate the effectiveness of using a Conditional Random Fields model to classify traders.

References

[1] Enrico Gerding, Peter McBurney, Jinzhong Niu, Simon Parsons and Steve Phelps [2007]: Overview of CAT: A Market Design Competition. Technical Report ULCS-07-006. Department of Computer Science, University of Liverpool. Liverpool, UK. Available from: http://www.csc.liv.ac.uk/research/techreports/tr2007/tr07006abs.html

[2] D. Cliff. Minimal-intelligence agents for bargaining behaviors in market-based environments. Technical Report HP--97--91, Hewlett Packard Laboratories, Bristol, England, 1997.

[3] Gjerstad, Steven, and J. Dickhaut [1998]. Price Formation in Double Auctions. *Games and Economic Behavior*, 22:1-29, 1998.

[4] D. K. Gode and S. Sunder. Allocative efficiency of markets with zero-intelligence traders: Markets as a partial substitute for individual rationality. *The Journal of Political Economy*, 101(1):119-137, February 1993.

[5] A. E. Roth and I. Erev. Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term. *Games and Economic Behavior*, 8:164-212, 1995.

[6] J. Niu, K. Cai, S. Parsons, E. Gerding, P. McBurney, T. Moyaux, S. Phelps and D. Shield [2008]: JCAT: A platform for the TAC Market Design Competition. *Proceedings of the Seventh International Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS 2008). Estoril, Portugal, May 2008.

[7] Christopher J. C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition". Data Mining and Knowledge Discovery 2:121 - 167, 1998

[8] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[9] Jeff A. Bilmes**.** What HMMs can do. *IEICE Transactions on Information and Systems*, E89-D(3):869-891, 2006.

[10] Lawrence R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE International Conference*, 77 (2), 257–286, February 1989

[3] Our implementation of the MDP framework utilized source code written for the AIMA[13] textbook and can be obtained from http://code.google.com/p/aima-java

[11] Stephen Ruping, (2001). *SVM kernels for time series analysis* (Technical Report). Department of Computer Science, University of Dortmund, Dortmund, Germany.

[12] K. Grauman and T. Darrell. The Pyramid Match kernel: discriminative classification with sets of image features. In *Proceedings of the IEEE International Conference on Computer Vision* (ICCV), Beijing, China, October 2005.

[13] Russell S, Norvig P (1995) *Artificial Intelligence: A Modern Approach*, Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, New Jersey