

Bidding in Non-Stationary Energy Markets

(Extended Abstract)

Pablo Hernandez-Leal
Instituto Nacional de
Astrofísica, Óptica y
Electrónica
Sta. María Tonantzintla,
Puebla, México
pablohl@ccc.inaoep.mx

Matthew E. Taylor
Washington State University
Pullman, Washington, USA
taylorm@eecs.wsu.edu

Enrique Munoz de Cote
and L. Enrique Sucar
Instituto Nacional de
Astrofísica, Óptica y
Electrónica
Sta. María Tonantzintla,
Puebla, México
{jemc,esucar}@inaoep.mx

ABSTRACT

The PowerTAC competition has gained attention for being a realistic and powerful simulation platform used for research on retail energy markets, in part because of the growing number of energy markets worldwide. Agents in this complex environment typically use multiple strategies, changing from one to another, posing a problem for current learning algorithms. This paper introduces DriftER, an algorithm that learns an opponent model and tracks its error rate. We compare our algorithm in the PowerTAC simulator against the champion of the 2013 competition and a state of the art algorithm tailored for interacting against switching (non-stationary) opponents. The results show that DriftER outperforms the competition in terms of profit and accuracy.

Categories and Subject Descriptors

I.2.11 [Distributed Artificial Intelligence]: Multiagent Systems

General Terms

Algorithms, Experimentation

Keywords

PowerTAC; opponent modeling; non-stationary strategies; Markov decision processes, energy markets

1. INTRODUCTION

One of the consequences of shifting towards smarter energy (consumption, generation and distribution) is the deregulation of the energy supply and demand. These deregulated grids have enabled producers to sell energy to consumers by using a *broker* as an intermediary. However, these broker agents need to interact in a highly dynamic environment, where other agents are competing against each other. Autonomous brokers can succeed because of their computation power and fast reaction times, but are still challenged by the

scenario's complexity (rich state spaces, high dimensionality, partial observability and non-stationarity [5]) and straightforward game-theoretic, machine learning, and artificial intelligence techniques fall short. Moreover, in this complex environment, it is reasonable to expect that agents will use different strategies throughout their interaction and change from one to another.

Recent approaches based on multiagent systems have been proposed for energy markets. PowerTAC simulates a retail electrical energy market, where competing brokers (trying to maximize their profits) offer energy services to customers through tariff contracts, and must then serve those customers by trading in a wholesale market [4]. However, neither of the agents winning 2013 (TacTex [5]) nor 2014 (AgentUDE [1]) can efficiently compete against non-stationary opponents (that switch between stationary strategies), even though agents tend to change their strategy in competitions over time (e.g., to keep their opponent guessing) [2]. Some works have addressed this problem, like the MDP-CL framework [3]. One major drawback is that it has many parameters that need to be tuned by an expert. This paper's main contribution is to introduce *DriftER*, Drift (based on) Error Rate, an algorithm that uses concept drift ideas for adapting quickly to non-stationary opponents and has few parameters to tune. The results show the effectiveness of our approach, obtaining better results in total profit and accuracy, relative to existing approaches.

2. DRIFTER

When facing non-stationary opponents two aspects are important: exploring the opponent actions to detect switches and tracking the opponent model. DriftER treats the opponent as a stationary (Markovian) environment and uses concept drift ideas to track the quality of the learned model as an indicator of a possible change in the opponent strategy. When a switch in the opponent strategy is detected, DriftER resets its learned model and restarts the learning. In this work, DriftER uses the same representation as TacTex [5] for modeling the wholesale market as a Markov Decision Process. Because the agent has no initial information, it must collect data to develop a transition function. It starts with exploratory actions during the first k timeslots (learning phase), after which the MDP can be solved. We assume that during learning phase the opponent remains stationary.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*
Copyright © 2015, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

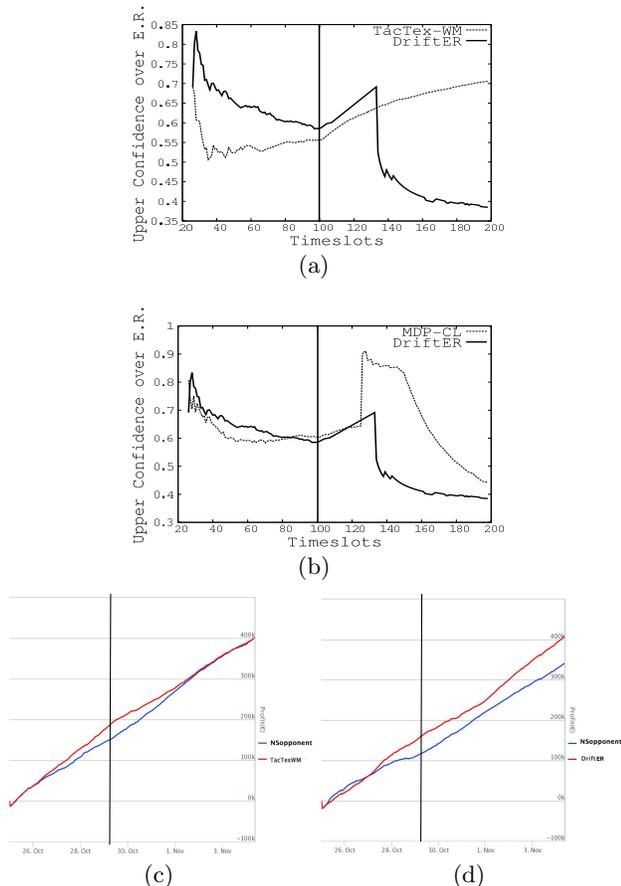


Figure 1: Upper confidence over the error rate of (a) TacTex-WM and (b) MDP-CL while comparing with DriftER. Cumulative profits of (c) TacTex-WM and (d) DriftER (red line) against the non-stationary opponent (blue line) in a competition of 250 timesteps. The time when opponent switches between strategies is displayed with a vertical line.

2.1 Switch detection

DriftER learns online: at each timestep the algorithm decides to continue with the current model or change to a new one. Once DriftER has learned an opponent model, it can predict the next state of the opponent \hat{s}_i on each timestep and can be compared with the true state, s_i . This comparison can be seen as a Bernoulli trial. We assume a sequence of i.i.d. events will produce a Bernoulli process. Then, for each i in the sequence, the error rate $error(s_i)$ is the probability of observing *incorrect*. Statistical theory guarantees that while the class distribution of the examples is stationary, the error rate $error(s_i)$ will decrease when i increases. At this point the error rate can be improved by taking into consideration a confidence interval over the error rate $conf(s_i)$. DriftER keeps track of this $conf(s_i)$ value at each timestep, where a decrease in this value indicates that the current model is correct and useful and not useful otherwise. However, $conf(s_i)$ may increase for two reasons: (i) noise in the opponent, in this case, we do not want to learn a new model but instead should stay with the current one, or (ii) the opponent has switched to a different strategy and the learned model is no longer useful for predictions. In this case, we want to stop using the current model and learn a new one. In order to

fulfill these requirements we keep track of the first derivative $conf'(s_i)$ of the last n timesteps. If $conf'(s_i) > 0 == true$ in at least last m out of n steps, the algorithm decides the opponent has switched strategies and restarts the learning phase.

3. EXPERIMENTS

Experiments were performed on the PowerTAC simulator. We model a non-stationary opponent that uses two stationary strategies: it starts with a fixed limit price P_l and then in the middle of the interaction changes to a different (higher) fixed limit price P_h .

We compare three learning algorithms against the switching opponent. Figure 1 (a) shows the upper confidence of the error rate of TacTex-WM and DriftER. We can observe that starting from round 100 (when the opponent changes its strategy) the error rate of TacTex-WM increases as it is not able to adapt to the opponent. In contrast, DriftER shows an increase in the error rate after the opponent switch (timeslots 100 to 110), but then re-enters the learning phase (time steps 110-135). At this point, its confidence over the error rate is high and it shows a peak. At this point, DriftER has learned a new MDP and a new policy which reduces the error rate consistently. Figure 1 (b) shows the error rates of MDP-CL and DriftER. We can observe that both algorithms detect the opponent's switch. However, MDP-CL performs comparisons to detect switches every w steps ($w = 25$ in this case), unlike DriftER. Figure 1 shows the cumulative profit of (c) TacTex-WM and (d) DriftER against the non-stationary opponent. TacTex-WM profits decrease after the opponent's switch, while DriftER's profits increase even after the switch. Both algorithms reach similar cumulative profits, but DriftER obtained an average of 80k € more than the non-stationary opponent.

4. CONCLUSIONS AND FUTURE WORK

This paper introduces DriftER, an algorithm that learns a model of the opponent in the form of a MDP and keeps tracks of its error rate. DriftER's success is shown empirically by comparing with other approaches in PowerTAC, a complex energy market simulator. Future work will address using transfer learning ideas to promote a fast learning.

REFERENCES

- [1] J. Babic and V. Podobnik. An analysis of power trading agent competition 2014. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*. Springer, 2014.
- [2] D. Fudenberg and J. Tirole. *Game Theory*. The MIT Press, Aug. 1991.
- [3] P. Hernandez-Leal, E. Munoz de Cote, and L. E. Sucar. A framework for learning and planning against switching strategies in repeated games. *Connection Science*, 26(2):103–122, Mar. 2014.
- [4] W. Ketter, J. Collins, and P. Reddy. Power TAC: A competitive economic simulation of the smart grid. *Energy Economics*, 39:262–270, Sept. 2013.
- [5] D. Urieli and P. Stone. TacTex'13: A Champion Adaptive Power Trading Agent. In *Association for the Advancement of Artificial Intelligence 2014*, Quebec, Canada, May 2014.