Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

# Model AI Assignments 2013

**Todd W. Neller**
Gettysburg College
*tneller@gettysburg.edu*

**Marc Lanctot**
Maastricht University
*marc.lanctot@maastrichtuniversity.nl*

**Devika Subramanian**
Rice University
*devika@cs.rice.edu*

**Stephanie E. August**
Loyola Marymount University
*saugust@lmu.edu*

**Abstract**

The Model AI Assignments session seeks to gather and disseminate the best assignment designs of the Artificial Intelligence (AI) Education community. Recognizing that assignments form the core of student learning experience, we here present abstracts of three AI assignments from the 2013 session that are easily adoptable, playfully engaging, and flexible for a variety of instructor needs. Assignment specifications and supporting resources may be found at http://modelai.gettysburg.edu.

## Recreating TD-Gammon

### Devika Subramanian

In this programming assignment, students in a junior/senior level undergraduate AI class reconstruct the classic TD-Gammon program from the original specifications in (Tesauro 1995). We provide them with infrastructure code in Java for the backgammon player and the neural net evaluation function. The assignment builds on a prior assignment in which students build a hill-climbing learner that tunes a parametric evaluation function. The assignment specifies the TD algorithm in detail and identifies key design choices that have to be made. Students empirically determine a good set of design parameters for the algorithm. The assignment also has an analytical component in which students compare and explain the performance of the TD learner relative to the hill climbing learner. They are also invited to discuss the utility of this learning approach for other stochastic games.

## Tsunami Warning System: A Case Study of Intelligent Agents in Action

### Stephanie E. August

This case study analyzes a tsunami warning system (TWS) from the perspective of an intelligent agent, the Tsunami Activity Reporter. Students receive a description of the reporter and an overview of a warning system from the National Oceanic and Atmospheric Administration and design an intelligent agent using the PEAS (performance measure, environment, actuators, and sensors) framework. Students independently complete a draft of the assignment then collaborate in small groups on a robust solution. TWS reinforces concepts of agent architecture and critical thinking skills while presenting a real-world and familiar context for problem solving. It presents an opportunity to invite in experts from other disciplines and an opportunity to discuss the ethics involved in warning systems and the dangers of false negatives and false positives. Experience indicates that as students debate the capabilities of various architectures and apply their knowledge of environments, they gain a deeper understanding of issues involved in working with intelligent agents. TWS is designed for Introduction to AI students but is easily adapted. K-12/CS1/CS2 students can learn the potential and challenges of intelligent systems. Students studying knowledge-based systems can implement prototype warning systems. Systems engineering students can consider interfaces between various sensors and software systems.

## An Introduction to Counterfactual Regret Minimization

### Todd W. Neller and Marc Lanctot

In 2000, Hart and Mas-Colell introduced the important game-theoretic algorithm of regret matching (Hart and Mas-Colell 2000). Players reach equilibrium play by tracking regrets for past plays, making future plays that are proportional to positive cumulative regrets (i.e. how much they wished they had made the moves on average). The technique is not only simple and intuitive; it inspired the counterfactual regret minimization algorithm that has experienced much success in the annual computer poker competitions.

Since these algorithms are relatively recent, there are few curricular materials available to introduce students, researchers, and practitioners to regret-based algorithms. The tutorial PDF, suggested exercises, and sample code offered below represent a modest first step towards making such recent innovations more accessible.

Using Java code examples in Donald Knuth's literate programming style, we will present example applications of the regret matching algorithm for normal form "one-shot" games, counterfactual regret minimization (CFR) (Zinkevich et al. 2008) for extensive form games, and fixed-strategy iteration counterfactual regret minimization (FSICFR) (Neller and Hnath 2012). We also briefly touch on strategy cleaning and how one might take maximal advantage of opponent error. Throughout, the reader first sees an example application and is then invited to deepen understanding through application to additional challenge problems.

## References


Hart, S., and Mas-Colell, A. 2000. A simple adaptive procedure leading to correlated equilibrium. *Econometrica* 68(5):1127–1150.

Neller, T. W., and Hnath, S. 2012. Approximating optimal dudo play with fixed-strategy iteration counterfactual regret minimization. In van den Herik, H. J., and Plaat, A., eds., *LNCS 7168: Advances in Computer Games, 13th International Conference, ACG 2011, Tilburg, The Netherlands, November 20-22, 2011, Revised Selected Papers*, 169–182. Springer.

Tesauro, G. 1995. Temporal difference learning and td-gammon. *Commun. ACM* 38(3):58–68.

Zinkevich, M.; Johanson, M.; Bowling, M.; and Piccione, C. 2008. Regret minimization in games with incomplete information. In Platt, J.; Koller, D.; Singer, Y.; and Roweis, S., eds., *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press. 1729–1736.