Learning and Modeling Player Behavior in Games

Ben G. Weber
bgweber@gmail.com
Big Data meets Game Design

- How can we use **big data** to enable **new types of playable experiences**?
Big Data

- Telemetry
- Replays
- Social Media
Research Questions

- What can an AI system learn from players?

- What can big data tell us about players?
Projects

- Madden NFL Mining
- EISBot
Madden NFL Mining

- Pilot study at Electronic Arts

 Goals

- Analyze data from *Madden NFL 11*
- Determine why players quit playing
- Identify potential changes for *Madden NFL 12*
Madden Mining Question

- How do modifications in the design impact player retention?
Current Approaches

- **User studies**
  - Survey human participants
  - Collect physiological data

- **A/B testing**
  - Deploy several game variations with single-variable test samples
Robust Unique Effect Analysis

- An algorithm that performs **regression** and analyzes **unique effects** to rank features

**Algorithm overview**

1. Build regression models for predicting retention
2. Perturb the inputs to the models
3. Compute the impact of individual features
Most Influential Features

The following features were identified as the **most influential** in predicting **player retention**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Correlation</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offense Play Diversity</td>
<td>( - )</td>
<td>55.4</td>
</tr>
<tr>
<td>Defense Play Diversity</td>
<td>( - )</td>
<td>34.2</td>
</tr>
<tr>
<td>Interceptions Caught</td>
<td>( + )</td>
<td>24.6</td>
</tr>
<tr>
<td>Online Franchise Wins</td>
<td>( + )</td>
<td>15.7</td>
</tr>
<tr>
<td>Running Play Ratio</td>
<td>( + )</td>
<td>10.1</td>
</tr>
<tr>
<td>Multiplayer Wins</td>
<td>( + )</td>
<td>9.3</td>
</tr>
</tbody>
</table>
Win Rate Influence on Player Retention

![Graph showing the relationship between win rate and player retention for different game modes: PlayNow, Ranked, Superstar, and Franchise. The graph plots the predicted number of games against the win rate on a linear scale for each mode.]
Design Recommendations

- Simplify playbooks
- Clearly present the controls
- Provide the correct challenge
Project Impact

OFFENSIVE GAMEPLAN

SITUATION - 3RD AND SHORT
3rd down, with less than 3 yards to go

Playbook:
TB - R.Morris

Pass Plays
- Quick Pass 18
- Standard Pass 96
- Shotgun Pass 67
- Play Action Pass 52

Run Plays
- Inside Handoff 33
- Outside Handoff 13
- Pitch 9
- Counter 9
Telemetry-Supported Game Design

- **Question:** Identify questions about the design
- **Record:** Determine which data needs to be recorded and deploy the game
- **Analyze:** Check if the collected data matches expectations
- **Refine:** Given the findings, analyze the design and formulate additional questions
EISBot

- An agent for the real-time strategy game **StarCraft: Brood War**

- **Goals**
  - Explore capabilities necessary for expert StarCraft gameplay
  - Investigate techniques for learning from replays

[Flash, Pro-gamer]
Why StarCraft?

- Many real-world properties
- Evolving meta-game
- Multi-scale
What competencies are necessary for StarCraft gameplay?

Which competencies can be learned from demonstrations?
StarCraft Gameplay

- Expand Tech Tree
- Attack Opponent
- Manage Economy
- Produce Units
Agent Overview

- Implemented in the **ABL** planning language

**Architecture**

- Extension of McCoy & Mateas’ (2008) integrated agent framework
- Partitions gameplay into distinct competencies
- Uses a blackboard for coordination
Multi-Scale Idioms

- Design patterns for enabling authoring of agents that perform multi-scale reasoning

**Idioms**
- Daemon behaviors
- Message passing
- Managers
- Unit subtasks
EISBot Managers

- Strategy Manager
  - Income Manager
    - Gather Resources
  - Production Manager
    - Construct Buildings
  - Tactics Manager
    - Attack Opponent
  - Recon Manager
    - Scout Opponent
Learning in EISBot

- Build-order prediction
- State estimation
- Strategy learning

Diagram:
- Replay
  - Training Process
  - Gameplay Model
Build-Order Prediction

- **Goals**
  - Identify opponent build orders
  - Predict when buildings will be constructed

**Spawning Pool Timing**

**Factory Timing**
Approach

- **Feature encoding**
  - Each player’s actions are encoded in a single vector
  - Vectors are labeled using a build-order rule set

- Features describe the game cycle when a unit or building type is first produced by a player

\[
f(x) = \begin{cases} 
  t, & \text{time when } x \text{ is first produced by } P \\
  0, & x \text{ was not (yet) produced by } P 
\end{cases}
\]
Build-Order Prediction Results

- **NNge**
- **Boosting**
- **Rule Set**
- **State Lattice**

Game Time (minutes) vs. Recall Precision
State Estimation

- **Goal**
  - Estimate enemy positions given prior observations

- **Particle Model**
  - Apply movement model
  - Remove visible particles
  - Reweight particles
Parameter Selection

- **Free parameters**
  - Trajectory weights
  - Decay rates

- State estimation is represented as an optimization problem
  - **Input:** parameter weights
  - **Output:** particle model error

- Replays are used to implement a particle model error function
State Estimation Results

![Graph showing State Estimation Results with various models: Null Model, Perfect Tracker, Default Model, Optimized Model. The x-axis represents Game Time (Minutes), and the y-axis represents Threat Prediction Error. The graph illustrates the performance of each model over the course of a game.](image-url)
Strategy Learning

- **Goals**
  - Learn build-orders from demonstration
  - Formulate goals based on examples

- **Trace Algorithm**
  - Converts replays to a trace representation
  - Formulates goals based on most similar situation
  - Utilizes retrieved goals for strategy selection and opponent modeling
Strategy Learning Results

Opponent modeling with a window size of 20

Prediction Error (RMSE) vs. Actions performed by player

- Null
- IB1
- Trace
- MultiTrace
Evaluation

- Ablation Study
- User Study
Project Contributions

- **Idioms** for authoring **multi-scale** agents
- Methods for **learning from replays**
Next Directions

- What is the future of learning from demonstration?

- What is the future of game analytics?
Learning from Demonstration

- **Mixed-initiative AI development**
  - How can we integrate learning from demonstration with other AI methods?

- **Stylized gameplay**
  - Can we develop game AI which emulates the style of a specific player or archetype?
Game Analytics

- **Data integration**
  - How can we integrate data harnessed from diverse data sources?

- **Continuous feedback**
  - How can analytics be utilized throughout the development cycle?
What is the Future of Games?

- Using **Big Data** to enable new types of **AI development** and gameplay

- Integrating **player feedback** in the game design process
Learning and Modeling Player Behavior

- Ben G. Weber
  - bgweber@gmail.com
  - @bgweber
  - http://bgweber.com