Probabilistic Approaches to Cheating Detection in Online Games

Laetitia Chapel, Dmitri Botvich, **David Malone**

19 August 2010

Motivation: Cheating in Games

- Cheating believed to be common in online games.
- Can decrease player satisfaction.
- Various existing schemes. E.g.
  - Check for running debugger.
  - Look for known exploited mechanism in use.
  - Look for bot-like traits of player.
- Can we base learning only on game results?
- Number of players can be small ($N = 2$) or very large ($N = 10^6$).
- Should be a chance to learn who cheaters are.

Cheating

What is Cheating?

> *activities that modify the game experience to give one player an advantage over another player(s)*

- Can we spot this?
- Idea: look for better than expected performance.
- Problem: How do we know what to expect?
- Have a look at two techniques: LLN, BT.

Game Model

- We have N players, who play in pairs.
- Each player i has a strength/rank/skill: $\pi_i > 0$.
- i playes j some number of times $s_{ij} \geq 0$.
- i beats j some number of times X_{ij}.
- This might be repeated.

Game is of skill and chance, so:

$$\mathbb{P}[i \text{ beats } j] = g(\pi_i, \pi_j).$$

For simplicity, we use:

$$g(\pi_i, \pi_j) = \frac{\pi_i}{\pi_i + \pi_j}$$

All games independent.

LLN Test

- Knowing $\pi_i \Rightarrow$ distribution of results.
- Cheater should win more often than expected in long run.
- Can look at number of wins X_i for player i.
- If games independent, can use the Central Limit Theorem.

$$\frac{X_i - E(X_i)}{\sigma_i} \sim \mathcal{N}(0,1)$$

Use this to construct a hypothesis test.

LLN: Some Results

3 cheaters in 20 players, 500 reps.

		Identified As	
		Cheaters	Non-cheaters
Actual Cheaters	5 games	99.9%	0.1%
	10 games	100%	0%
	20 games	100%	0%
	80 games	100%	0%
Actual Non-cheaters	5 games	0.1%	99.9%
	10 games	0%	100%
	20 games	0%	100%
	80 games	0%	100%

LLN: Some More Results

3 cheaters in 5 players, 500 reps.

		Identified As	
		Cheaters	Non-cheaters
Actual Cheaters	10 games	60.2%	39.8%
	20 games	70.1%	29.9%
	40 games	76.1%	23.9%
	80 games	83.7%	16.3%
Actual Non-cheaters	10 games	0%	100%
	20 games	0%	100%
	40 games	0%	100%
	80 games	0%	100%

Problem

- How do we know the π_i?
- Classic machine learning problem: Bradley Terry (1952).
- Max likelihood estimator from given set of results.
- Known to converge, given some constraints on games.

Require: any initial p_j^0, $j = 1, ..., N$, such that $\sum_{j=1}^{N} p_j^0 = 1$

repeat

Let $s = (k \bmod N) + 1$.

$$\mathbf{p}^{k+1} \equiv \left[p_1^k, ..., p_{s-1}^k, \frac{\sum_{i,i\neq s} w_{si}}{\sum_{i,i\neq s} \frac{w_{si}+w_{is}}{p_s^k+p_i^k}}, p_{s+1}^k, ..., p_N^k \right]^T$$

Normalize $\mathbf{p}^{k+1}$: $\sum_{i=1}^{N} p_i^{k+1} = 0$

until $\frac{\partial \ell(\mathbf{p}^{k+1})}{\partial p_j} = 0, \forall j$

BT Test

- Could use this to estimate π_i.
- What if players cheat in particular circumstances?
- E.g. cheat when playing stronger players.
- Similar problem to BT with home advantage (Agresti 1990).

Each player has strength of advantage θ_i and plays with strength $\theta_i \pi_i$ when playing stronger players.

For honest players $\theta_i = 1$.

BT Test

- We can build ML estimator.
- However, we need an estimate of when $\pi_i > \pi_j$.
- Estimator uses BT based on previous tournament.
- π_i jittered between tournament.
- More requirements for convergence.
- Quite a lot slower than LLN.

BT: Some Results

3 cheaters in 20 players, 500 reps.

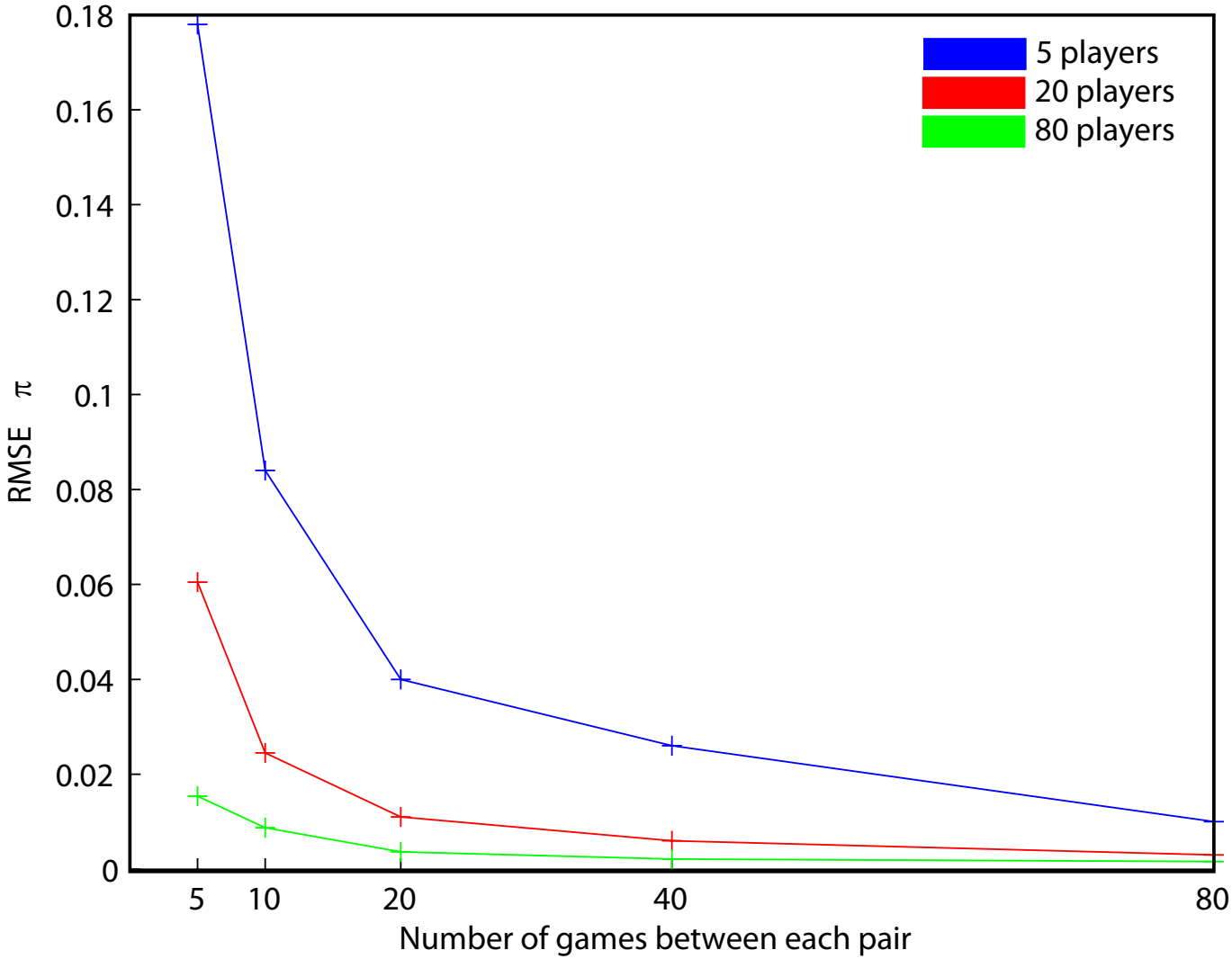
		Identified As	
		Cheaters	Non-cheaters
Actual Cheaters	5 games	60.3%	39.7%
	10 games	81.0%	19.0%
	20 games	92.0%	8.0%
	80 games	99.7%	0.3%
Actual Non-cheaters	5 games	0.6%	99.4%
	10 games	0.6%	99.4%
	20 games	0.6%	99.4%
	80 games	0.8%	99.2%

BT: Some More Results

3 cheaters in 5 players, 500 reps.

		Identified As	
		Cheaters	Non-cheaters
Actual Cheaters	10 games	63.7%	36.3%
	20 games	88.3%	11.7%
	40 games	96.7%	3.3%
	80 games	99.7%	0.3%
Actual Non-cheaters	10 games	0%	100%
	20 games	0%	100%
	40 games	0%	100%
	80 games	0%	100%

How Good Are π_i Estimates?



Conclusions, Ongoing & Future Work

- Looked at two ways to detect cheating.
- Simple tests show it might just work. . .
- . . . but you can also see how it could go wrong.
- In practice, would probably use as one factor.
- Have checked LLN based on BT.
- Some analysis of estimator convergence.
- Looking at Sumo data, actually easier!