Deterministic Calibration with Simpler Checking Rules

Dean Foster
University of Pennsylvania

and

Sham Kakade
University of Pennsylvania

April 12, 2005

The problem: Learning Nash equilibria

Current methods are slow and involve exhaustive search.

Can a fast method be found?

How about for special form games?

Measuring complexity

Two definitions of speed of convergence:

- total CPU used
- number of rounds of play

History

	Forecast probability	**Forecast utility**
Blackwell	**CE** Calibration (F. and Vohra, '97)	**CE** No regret (F. and Vohra '97) (Hart and Mas-Colell '00)
Exhaustive search	**NE** Hypothesis testing (F. and Young '03)	**NE** Regret testing (F. and Young '05) (Germano & Lugosi '05)
Public methods	**NE** Weak calibration yesterday's talk (Kakade and F. '04)	**NE** Weak utility estimation today's talk (Kakade and F. '05)

Speed (rounds of play)

	Forecast probability	Forecast utility
Blackwell (→ CE)	$(1/\epsilon)^{a^n}$	$(a/\epsilon)^2$
Exhaustive search (→ Nash)	$\gg (1/\epsilon)^{a^n}$	$\gg (1/\epsilon)^{an}$
Public methods (→ Nash)	$(1/\epsilon)^{a^n}$ $2^{\lvert\mathcal{I}\rvert}$	$(1/\epsilon)^{an}$ $\lvert\mathcal{I}\rvert^{\log\log\lvert\mathcal{I}\rvert}$ (with constant a)

$$\begin{aligned} n &= \text{number of players} \\ a &= \text{number of actions per player} \\ \epsilon &= \text{desired accuracy} \\ |I| = a^n &= \text{input size } (a \text{ is fixed}) \end{aligned}$$

(CE: Blackwell gives fast approx algo. NE: slow, few computational results known.)

Background: Testing functions in calibration

- X_t sequence to be forecast by p_t

- Weak calibration, means

$$\frac{1}{T}\sum_{t=1}^{T}(X_t - p_t)\, w(p_t) \to 0$$

  - $w()$ is any smooth function.
  - What Sham talked about yesterday.

- Today's twist: Use other testing functions. Eg

$$\frac{1}{T}\sum_{t=1}^{T}(X_t - p_t)\, w(p_t, X_{t-1}) \to 0$$

  Would test for Markov patterns.

Relationship between testing functions and conditional expectation

- "Advanced" version of conditional expectation

$$E\left[(X - E(X|Y))\ w(Y)\right] = 0.$$

  - X, and Y are random variables
  - $w()$ is measurable. (Can restrict $w()$ to be smooth.)
  - We should assume $E(X|Y) = h(Y)$ for some measureable function $h()$

- Contrast with our definition:

$$\frac{1}{T}\sum_{t=1}^{T}(X_t - p_t)\, w(p_t, X_{t-1}) \to 0$$

  - can think of $p_t = \widehat{E}(X_t|X_{t-1}, p_t)$
  - If we could enforce measurability we might get uniqueness and then this notation would be useful.

Individual vs Public calibration

- Game setting for calibration
  - $X_{i,t}$ is the observable that player i cares about at time t
  - $p_{i,t}$ is a forecast of $X_{i,t}$

- Individual calibration:

$$(\forall i) \qquad \frac{1}{T}\sum_{t=1}^{T}(X_{i,t} - p_{i,t})\, w(p_{i,t}) \longrightarrow 0$$

- Public calibration:

$$(\forall i) \qquad \frac{1}{T}\sum_{t=1}^{T}(X_{i,t} - p_{i,t})\, w(\vec{p}_t) \longrightarrow 0$$

The game model

- Player i uses $p_{i,t}$ to predict the round t
- Player i then use smooth decision rule $s_i(p_{i,t})$ to pick the probability of their play in round t.
- Player i then randomly action S_i from this distribution

Observables

- Game setup:
  - Take $X_i = S_{-i}$ (i.e. all actions but player i)
  - $p_{i,t}$ is forecast of $X_{i,t}$

- Individual calibration:

$$(\forall i) \qquad \frac{1}{T}\sum_{t=1}^{T}(X_{i,t} - p_{i,t})\, w(p_{i,t}) \longrightarrow 0$$

- Public calibration:

$$(\forall i) \qquad \frac{1}{T}\sum_{t=1}^{T}(X_{i,t} - p_{i,t})\, w(\vec{p}_t) \longrightarrow 0$$

Convergence

- Suppose players play a smooth best reply to forecast $p_{i,t}$.
  - Traditional calibration $\rightarrow$ correlated equilibria
  - Public calibration $\rightarrow$ Nash equilibria

- Speed of convergence is related to dimension of the “Hilbert space” of the testing functions
  - For individual: dimension $(1/\epsilon)^{a^n}$
  - For public: dimension is $(1/\epsilon)^{na^n}$
  - Hence convergence is slow in both cases.

- Need lower dimensional space, but what can be changed?

Proof: Public calibration converges to NE

- Truth $\approx$ prediction
  - via calibration

- Truth is independent
  - Given $\vec{p}$ each player is in fact playing independently

- ϵ-rationality
  - ϵ-BR to prediction
  - p_i includes information about what all other players will do

- Independence $+$ ϵ-rationality $=$ ϵ-NE.

What can be changed?

Utility estimation

- Take $X_{i,t}$ to be the vector of potential payoffs
  - $\vec{S}_{-i}$ is the vector of everyone else's play
  - $u_{i,t}(k) = u_i(k, \vec{S}_{-i,t})$
  - $X_{i,t} = (u_{i,t}(1), \ldots, u_{i,t}(a))$
- Utility model
  - $p_{i,t}$ is an estimate of $X_{i,t}$ made at time $t - 1$
  - For CE we need

$$(\forall i) \qquad \frac{1}{T}\sum_{t=1}^{T}(X_{i,t} - p_{i,t})\, w(p_{i,t}) \to 0$$

  - For NE we need

$$(\forall i) \qquad \frac{1}{T}\sum_{t=1}^{T}(X_{i,t} - p_{i,t})\, w(\vec{p}_t) \to 0$$

Speed of convergence of utility estimation

- For CE: number of rounds is $O((n/\epsilon)^a)$
- For NE: number of rounds is $O((n/\epsilon)^{an})$
- Looks almost polynomial in length of input
  - $|I| = a^n =$ input size (a is fixed)
  - number of rounds is $O(|\mathcal{I}|^{\log\log|\mathcal{I}|})$
  - "pseudo Poly".
- Although exp in a, little known computationally.

Graphical Models for Game Theory

- Undirected graph capturing local (strategic) interactions (Kearns, Littman, & Singh)
  - Each "player" represented by a vertex
  - Payoff to i, is only a function of neighbors actions
  - Compact (yet general) representation of game
  - Assume max degree is d, then representation is $O(na^d)$ instead of $O(a^n)$.

- Can graphical games be learned faster than general games?

Need smaller observable set

- $X_{i,t}$ need only capture plays of neighbors
  - $N(i)$ is the set of neighbors of i (assume $|N(i)| \leq d$)
  - $S_{N(i)-i}$ is actions of all neighbors excluding self
  - $u_{i,t} = u_i(S_{i,t}, S_{N(i)-i})$
  - $p_{i,t}$ is forecast of $X_{i,t}$

- Same proof as before shows that for a NE we need

$$(\forall i) \qquad \frac{1}{T}\sum_{t=1}^{T}(X_{i,t} - p_{i,t})\, w(\vec{p}_t) \to 0$$

- But we desire to to better for structured games.

  (This is $(1/\epsilon)^{na^d}$, while the representation of a graphical game is na^d.)

Don't need to check as much

- We don't need to check $w(\vec{p}_t)$
- Instead we can check only

$$(\forall i) \qquad \frac{1}{T}\sum_{t=1}^{T}(X_{i,t}-p_{i,t})\,w(\vec{p}_{N(i),t}) \to 0$$

  where $\vec{p}_{N(i),t}$ is a vector of all the p's of all the neighbors of i.
- Since this is all that matters in $u_i()$, rationality against this set is rationality against the entire $\vec{p}$.
- Complexity: $n(1/\epsilon)^{a^{2d}}$
- The complexity is $|\mathcal{I}|$.
- NOTE TO SELF: No matter how excited you are about a complexity, never, write it as $|\mathcal{I}|$!

A even smaller observable set

- $X_i =$ personal utility
- $p_i =$ forecast of personal utility
- $w()$ is local:

$$(\forall i) \qquad \frac{1}{T}\sum_{t=1}^{T}(X_{i,t} - p_{i,t})\, w(\vec{p}_{N(i),t}) \to 0$$

- Converges to NE.
- Complexity: $n(1/\epsilon)^{a^d}$

A system based on trust

- X_i = action taken
- p_i = forecast of own action
- decisions are made based on other peoples forecast of themselves
- $w()$ is local:

$$(\forall i) \qquad \frac{1}{T}\sum_{t=1}^{T}(X_{i,t} - p_{i,t})\, w(\vec{p}_{N(i),t}) \rightarrow 0$$

- Converges to NE.
- Complexity: $n(1/\epsilon)^{a^d}$
- Violations can cause the system to crumble

Summary: Complexity of Learning in Graphical Games

Speed of convergence:

- Complexity: $n(1/\epsilon)^{da^d}$
- Recall, game representation is na^d
- Hence, the max degree is the bottleneck!
- Can get better results with utility forecasts: $n(1/\epsilon)^{da}$

CPU time:

- For tree games, fast per round computation
- Total CPU time comparable to NashProp
- For general graphs, could be hard to make forecast each round

See reverse side of handout for related readings