Learning for Multi-Agent Decision Problems

Geoff Gordon

ggordon@cs.cmu.edu

A learning problem

Observation → Decision → Reward →

→ Observation → Decision → Reward → ...

Maximize (say) discounted sum of rewards

Standard RL problem, but devil is in the details

Details

What do we get to observe?

What kinds of decisions can we make?

What does the environment remember about our past decisions?

Is there anybody out there?

Is there anybody out there?

Q: Why model other agents explicitly? Why are they any different from the rest of the environment?

Is there anybody out there?

Q: Why model other agents explicitly? Why are they any different from the rest of the environment?

A: Because it helps us predict the future.

Is there anybody out there?

Q: Why model other agents explicitly? Why are they any different from the rest of the environment?

A: Because it helps us predict the future.

A': Because it helps us act.

Is there anybody out there?

Q: Why model other agents explicitly? Why are they any different from the rest of the environment?

A: Because it helps us predict the future.

A': Because it helps us act.

Agent = part of the environment which we model as choosing actions in pursuit of goals

Problem

Many popular agent models don't help much in predicting or acting

. . . unless restrictive assumptions hold

Agent models

Part of environment:

- Independent, identically distributed actions
- Finite state machine
- Mixture of FSMs

The “who needs more than Bayes’ rule” view

Correct, but unhelpful if many FSMs or states

Lots of FSMs, states in realistic priors

Agent models

As decision maker:

- helpful teammate
- implacable enemy
- general-sum utility maximizer

First 2 are OK if true, last is not enough to predict actions

Rest of talk

Simplify the world drastically, step by step, preserving agent-modeling aspect of problem

(Start to) add complications back in

First simplification

Observation $\longrightarrow$ Decision $\longrightarrow$ Reward $\longrightarrow$ $\ldots$

Small discrete set of actions

Known payoff matrix

Observe past actions of all agents

$\Rightarrow$ Ignore all state *except* other agents; only learning problem is how to influence them

Repeated matrix game

Battle of the Sexes

	O	F
O	$4, 3$	$0, 0$
F	$0, 0$	$3, 4$

Outcomes of learning

Q: What are possible/desireable outcomes of learning in repeated matrix games?

Outcomes of learning

Q: What are possible/desireable outcomes of learning in repeated matrix games?

A: Equilibria.

Outcomes of learning

Q: What are possible/desireable outcomes of learning in repeated matrix games?

A: Equilibria.

But which equilibria?

Some kinds of equilibria

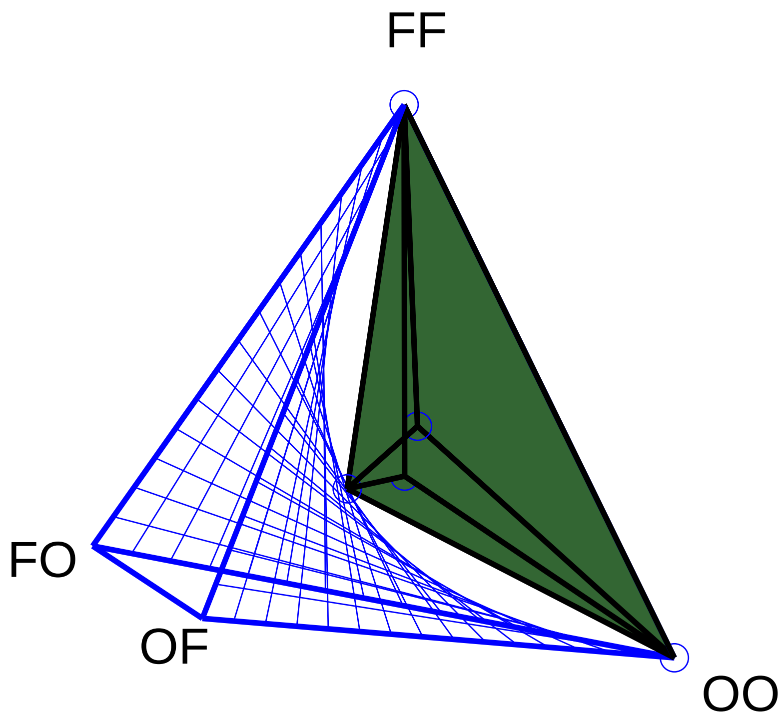
Equilibrium: distribution P over joint actions s.t. no player wants to deviate unilaterally from P

Nash equilibrium: P factors into independent row and column choices

Correlated equilibrium: general P

- executing P requires “moderator” or “correlation device”
- “unilaterally deviate” means, on recommendation of action a, play b

Equilibria in BoS



Nash: OO, FF, $[\frac{4}{7}$O, $\frac{3}{7}$O$]$ (last equalizes alternatives)

Correlated: e.g., coin flip

Equilibria in BoS

$$4\frac{a}{a+b} + 0\frac{b}{a+b} \geq 0\frac{a}{a+b} + 3\frac{b}{a+b} \qquad \text{if } a+b>0$$

$$\begin{aligned}
4a + 0b &\geq 0a + 3b \\
0c + 3d &\geq 4c + 0d \\
3a + 0c &\geq 0a + 4c \\
0b + 4d &\geq 3b + 0d \\
a, b, c, d &\geq 0 \\
a + b + c + d &= 1
\end{aligned}$$

Equilibria as outcomes

Are any of the above reasonable outcomes of learning?

- Coin flip: yes
- OO, FF: maybe
- $[\frac{4}{7}$O, $\frac{3}{7}$O$]$: no!

Equilibria as outcomes

Are there reasonable outcomes not included?

Equilibria as outcomes

Are there reasonable outcomes not included?

Yes: minimax is reasonable if our model is wrong or if negotiation fails

Minimax: forget their payoffs, they're out to get me!

Minimax payoffs may not be result of any equilibrium

Equilibria of repeated game

Can't learn from a single game of BoS

We're playing *repeated* BoS

Equilibria of repeated game include minimax point and all above equilibria (and much, much more. . .)

(Note: imprecision)

Folk theorem

Luckily, equilibria of repeated game are *easier* to characterize

Folk theorem: any feasible and strictly individually rational reward vector is the payoff of a subgame-perfect Nash equilibrium of the repeated game

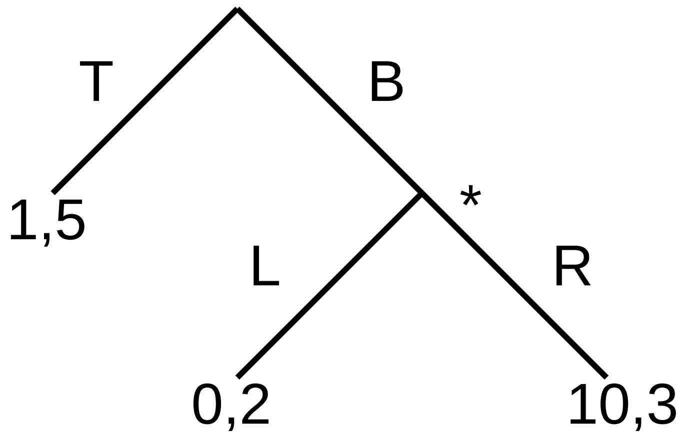
Subgame-perfect Nash

Nash equilibrium gives recommended play for each history

Some legal histories may not be reachable

Recommended plays for these histories don't have to be rational

Incredible threats

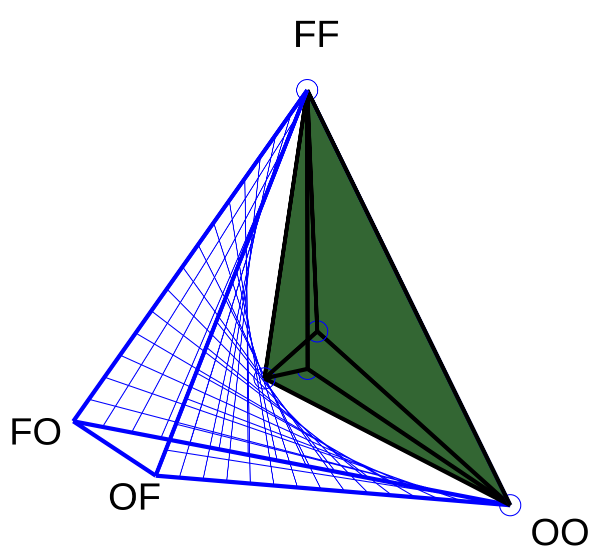


Two Nash equilibria:

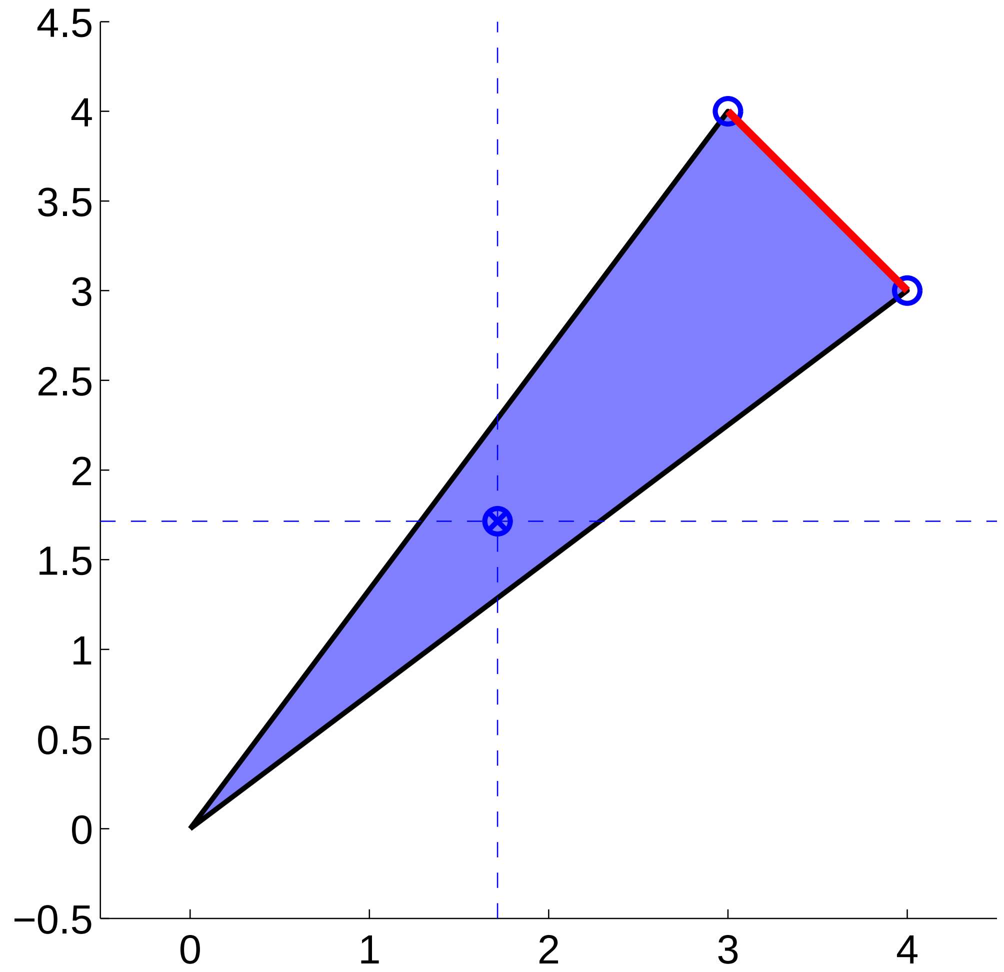
- T,L w/ payoffs $1, 5$
- B,R w/ payoffs $10, 3$

Only 2nd is subgame perfect: no one wants to deviate at *any* history (even unreachable ones)

Folk theorem, illustrated



$\Rightarrow$

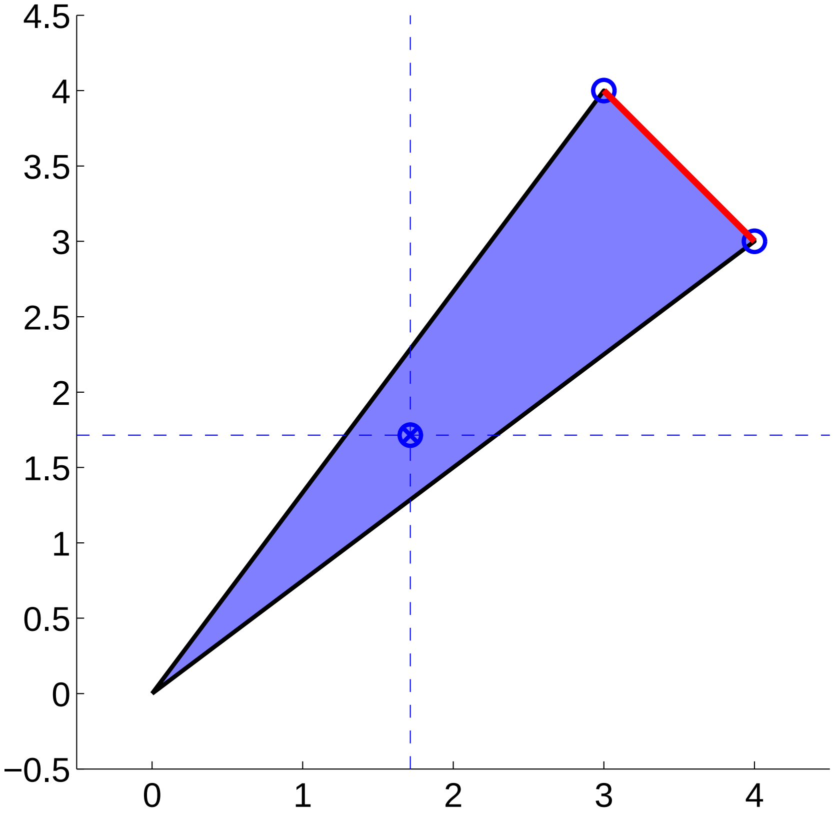


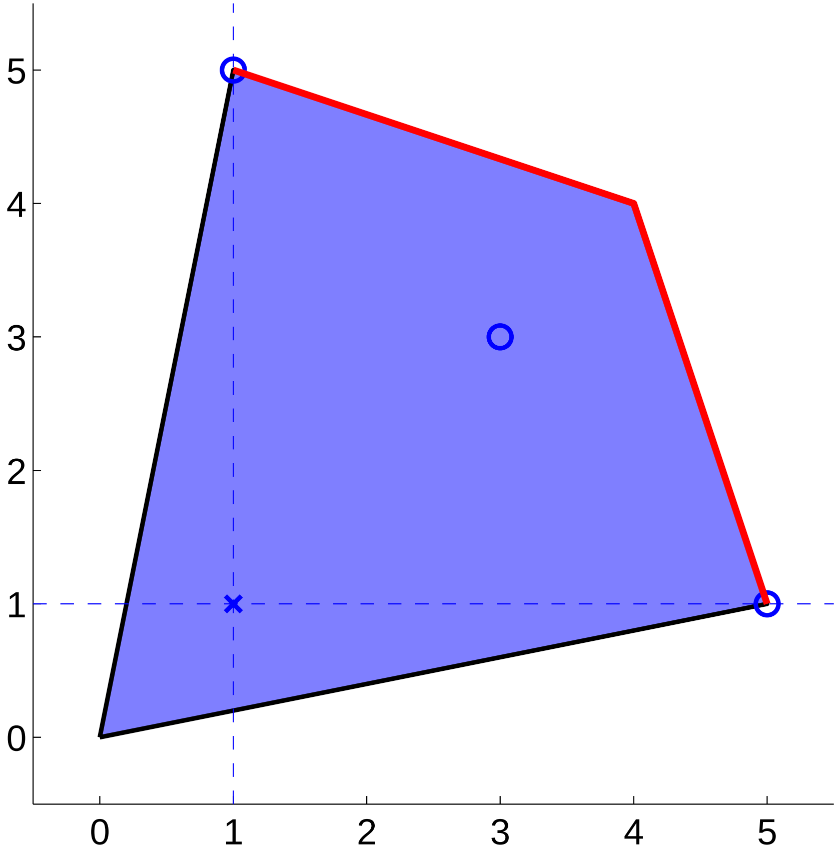
Are we done?

Not quite: minimax point is only a reasonable outcome if negotiation fails

If other players are “reasonable,” want better

Pareto optimality





Conjecture

For some reasonable definition of “reasonable,” a reasonable learner will converge to:

- its part of a Pareto-optimal subgame-perfect Nash equilibrium of the repeated game, if other players are also reasonable
- a best response, if other players are stationary
- payoffs $\geq$ its minimax value, o/w

Cf: ELE [Brafman & Tennenholtz, AIJ 2005]

Note: sufficient patience

Am I being reasonable?

OK, I've conjectured requirements for "reasonable" algorithms

Are these requirements reasonable?

Maybe. . .

A learning strategy

Based on two ideas:

- No-regret algorithms
- Proof of Folk Theorem

Run a no-regret algorithm which leaves some action choices free

Fix those free choices to a folk-theorem-like strategy

No-regret algorithms

Regret

No regret

An algorithm

Regret

Regret vs. strategy $\pi = \rho_\pi$ = how much do I wish I had played π?

E.g., other played OOOOOOOFOOOFOOFOOOOO, I played at random

Lots of regret for not playing "O all the time"

Lots of negative regret v. "F all the time"

Overall regret

Overall regret ρ v. "comparison class" $\mathcal{H}$ = worst regret v. any strategy in $\mathcal{H}$

We will take $\mathcal{H}$ = all constant-action strategies (e.g. "O all the time")

No-regret algorithms

Guarantee ρ_t grows slower than $O(t)$, often $O(\sqrt{t})$

Average regret $\frac{\rho_t}{t} \to 0$ as $t \to \infty$ at rate $1/\sqrt{t}$

Guarantee is for *all* sequences of opp plays

$\Rightarrow$ approach equilibrium if opponent tries to hurt us, something like CLT if fixed opponent strategy

Algorithm for BoS

Keep track of *regret vector*, S_t

- S_t will tell us our regret ρ_t

Compute $[S_t]_+$

Renormalize to get $q = \alpha[S_t]_+$

Randomize according to q

Or play arbitrarily if $S_t \leq 0$

“External regret matching” [Hannan 1957]

Regret vector

$$x_t = \begin{pmatrix} \text{1 if I played O} \\ \text{1 if I played F} \end{pmatrix}$$

y_t = same for opponent

$My_t =$ my payoffs for each action at time t, where M is my payoff matrix

Regret vector, cont'd

$r_t = x_t \cdot M y_t =$ my payoff

$s_t = M y_t - r_t \mathbf{1} =$ my regret vector

$S_t = \sum_t s_t$

$\rho_t = \max S_t$

Why does it work?

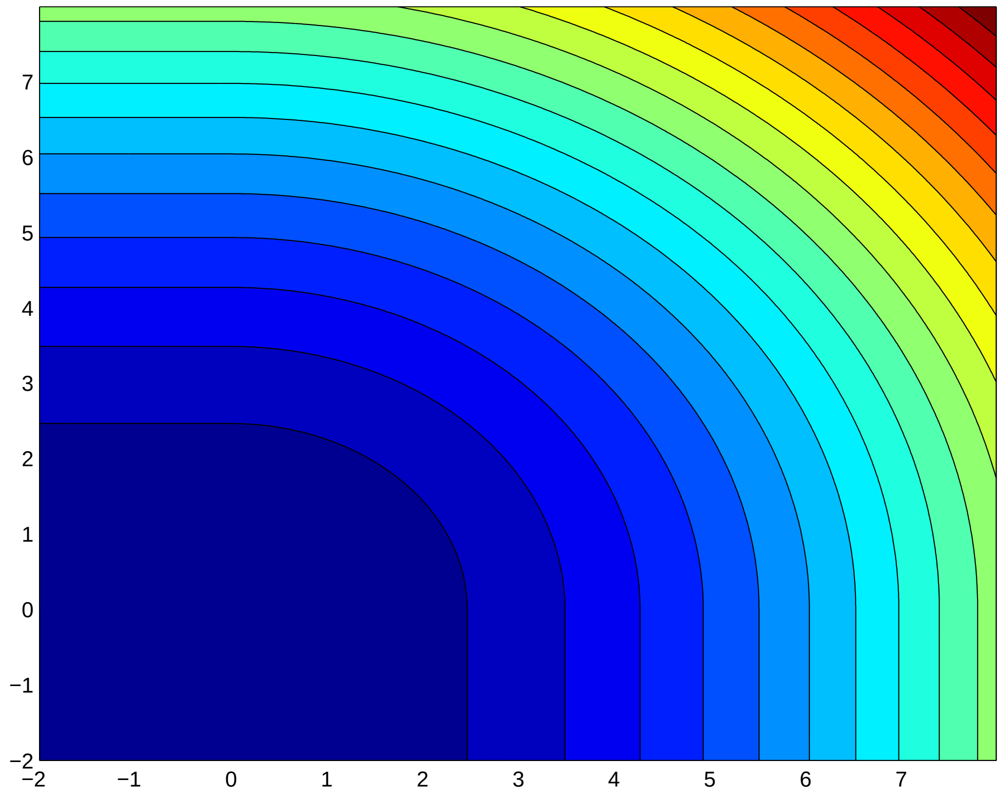
Potential function, $F(S)$

- low $F(S) \Rightarrow$ low regret

Gradient $F'(S_t)$ used to select plays

Prescribed play limits motion along gradient

Potential



$$F(s) = \| [s]_+ \|_2^2$$

Building on no regret

By itself, external regret matching:

- never gets less than minimax (“rational”)
- converges to best response v. stationary (“teachable”)

Can we get last property as well (Pareto SP Nash)?

Building on no regret

ERM allows arbitrary play if $S_t \leq 0$

Can generalize to use $S_t - \lambda \mathbf{1}$ for fixed λ

$\Rightarrow$ we can start off with any strategy, then switch to no-regret if it isn't working

So, do something with this flexibility…

Proof sketch of Folk Theorem

Constructive proof: exhibit SPNE strategy which has desired payoffs

$\exists$ a sequence of pure action profiles which has (arbitrarily close to) desired average payoff

Start off playing this sequence repeatedly

Punish deviations

Punishments

Simplest punishment: grim trigger

After a single deviation, play to minimize deviator's payoff forever

Nash, but not subgame perfect

More complicated punishments allow deviator to "pay restitution" and maintain subgame perfection

Combining NR & FT

Pick large λ so many initial free plays

Pick some Pareto-optimal payoffs

Use free plays to play grim trigger w/ those payoffs

$\Rightarrow$ everything but subgame perfection

Discussion

How to choose a Pareto point?

Can we incorporate more sophisticated bargaining r.t. “take it or leave it”?

Why is subgame perfection hard?

Bargaining

Important quantity: excess over minimax

Nash: maximize product of excesses

K-S: share sum of excesses proportional to each player's largest possible excess

If utilities are transferable, everything reduces to: share sum of excesses equally

Backing off simplifications

Environment = Nature, other agents; Nature resets every stage

Everything is observable

Actions are in $\{1 \ldots k\}$ for small k

Can we add complications back in?

Relaxing observability

Possible observables:

- my payoff (“bandits problem”)
- my payoff vector for all acts (“experts problem”)
- entire payoff matrix (“perfect monitoring”)
- my action v. all actions

$\exists$ no-regret algorithms for all cases

Relaxing observability, cont'd

Difficulty is Folk Theorem strategies

Brafman & Tennenholtz proved $\neg\exists$ ELE in some cases of imperfect monitoring

Open question: are there interesting subcases of imperfect monitoring where we can find "reasonable" algorithms?

Relaxing finiteness of actions

Suppose $\mathcal{A}_i$ is an arbitrary compact convex set

Payoffs are multilinear in $a_1, a_2, \ldots$

Called “online convex programming”

$\exists$ no-regret algorithms for OCP

- Some allow “free” action choices
- E.g., [Gordon 2005]

OCP examples

Disjoint paths in a graph

Rebalancing trees

. . .

Paths as OCP

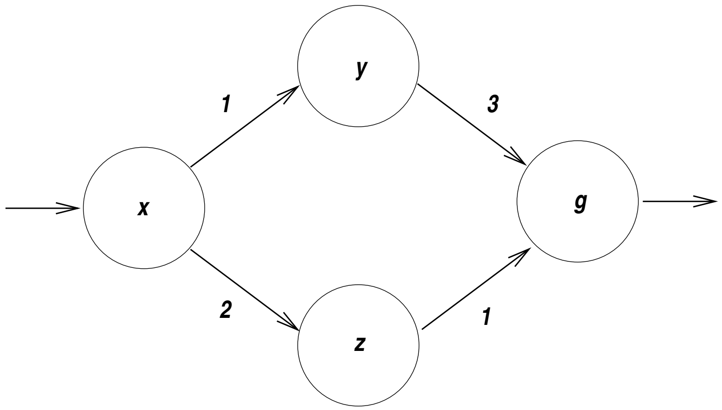
$\mathcal{A}_0 =$ paths in graph

- One indicator variable for each edge $ij \in E$
- $a_{ij} = 1$ iff edge ij in path

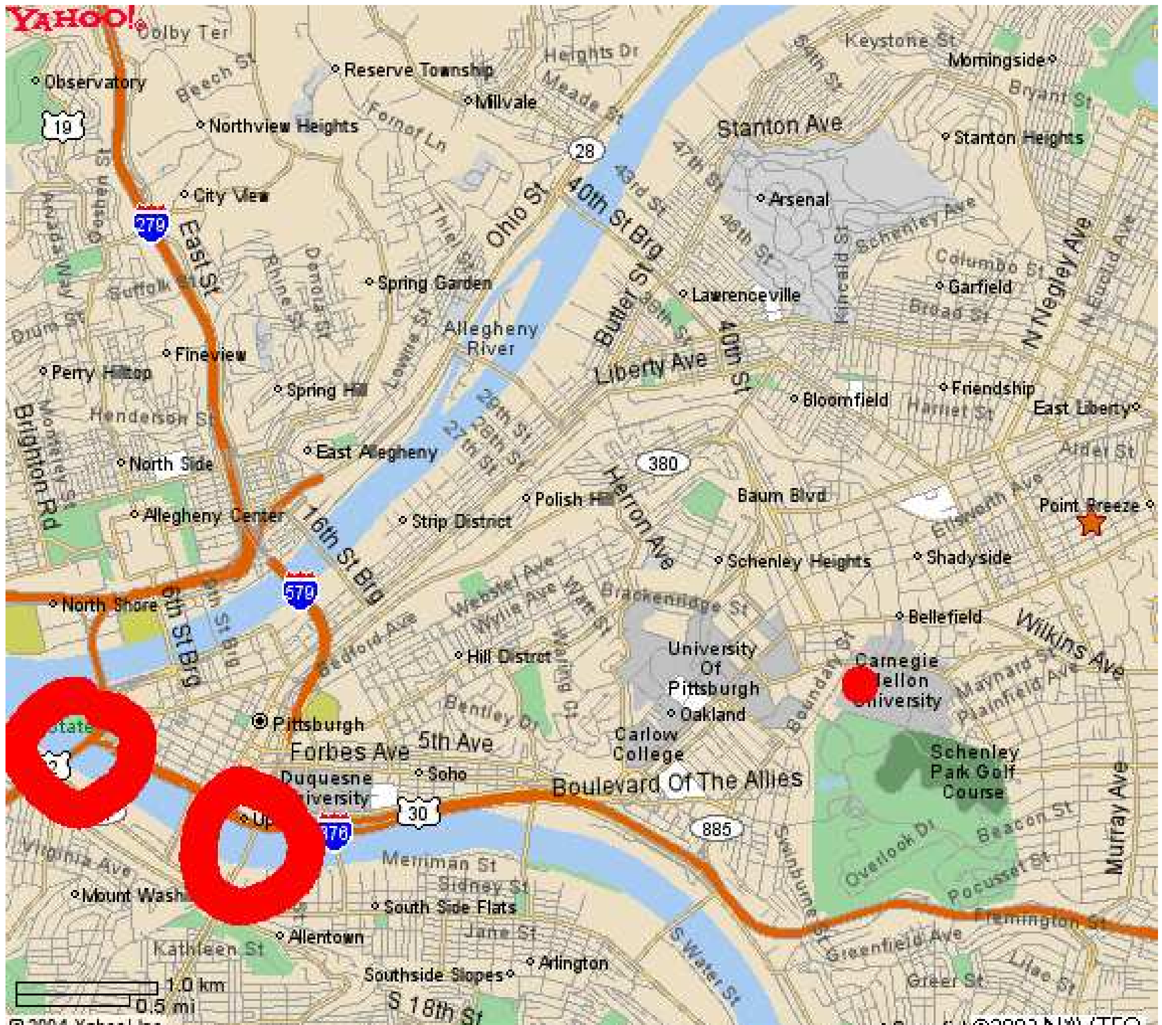
$\mathcal{A} =$ hull$(\mathcal{A}_0) =$ rand. paths

Cost to i: $c_i \cdot a_i + a_1 \cdot a_2$

- $c_i =$ edge costs, player i
- $a_1 \cdot a_2 =$ collision count



Example: avoiding detours



Generalizing the algorithm

Can do same trick

Start w/ no-regret for OCP

Replace flexible action choices w/ a folk-theorem-like strategy

Relaxing independence

What happens if Nature doesn't reset every step?

Assume Nature always resets eventually

Between resets: extensive form game (or stochastic game, or POSG, or ...)

Relaxing independence

Strategies in EF games form convex set

Sequence weights

Example: one-card poker

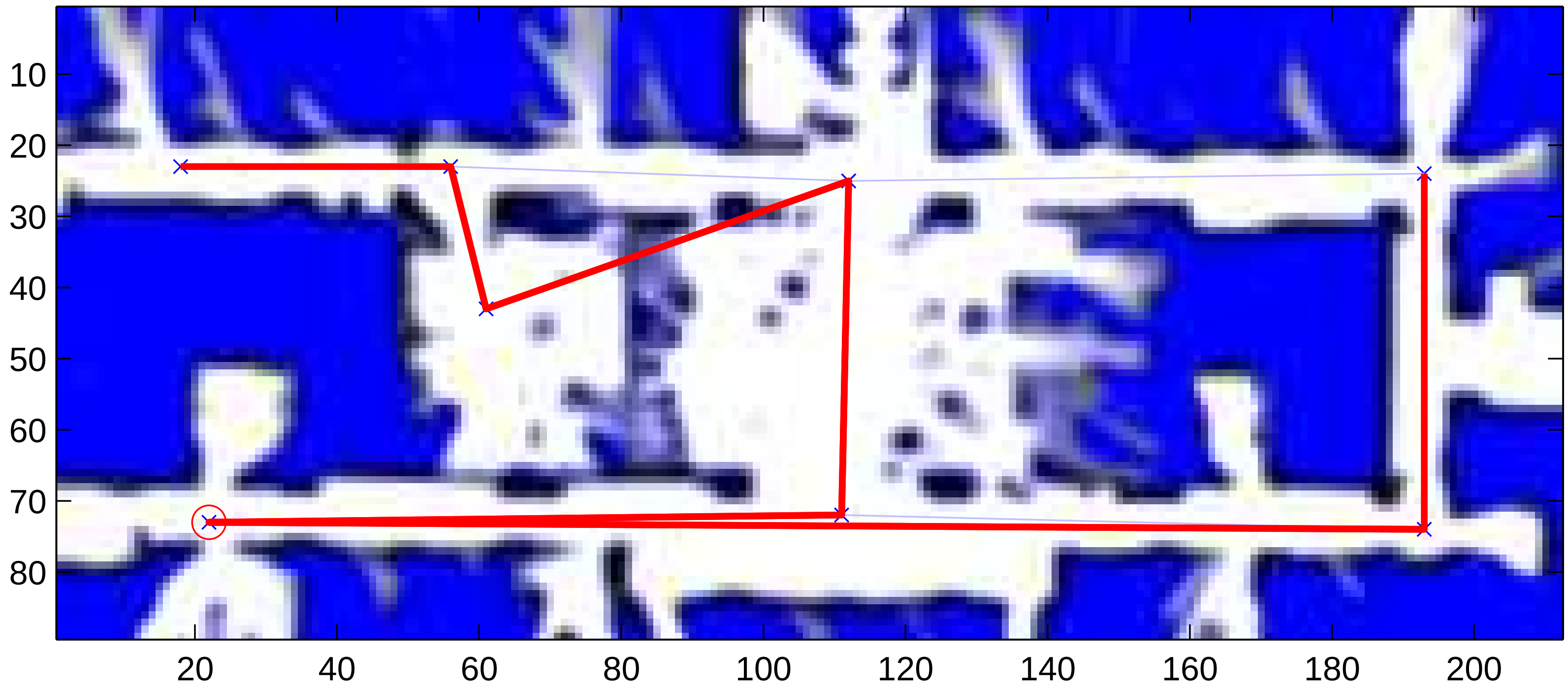
Relaxing independence, cont'd

Sequence weights are sometimes a big set! Can we get smaller?

Yes, in special cases ([randomized] path planning w/ detours, key-finding, multiagent linear regression)

Don't know in general

Example: keys



Searching as OCP

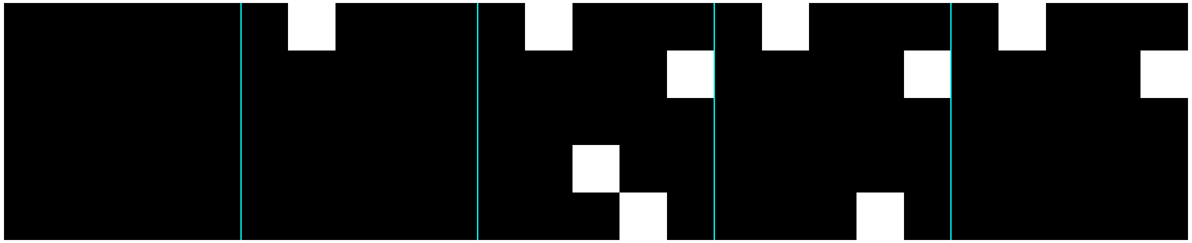
Strategies = (randomized) paths which visit every node

payoff = total cost of edges visited before finding keys

Note: convexity

Searching as OCP

h_{ijk} = did we traverse ij before visiting k

E.g., $[12543] =$ 

$$\ell_t(h) = c_{t,ijk} \cdot h$$

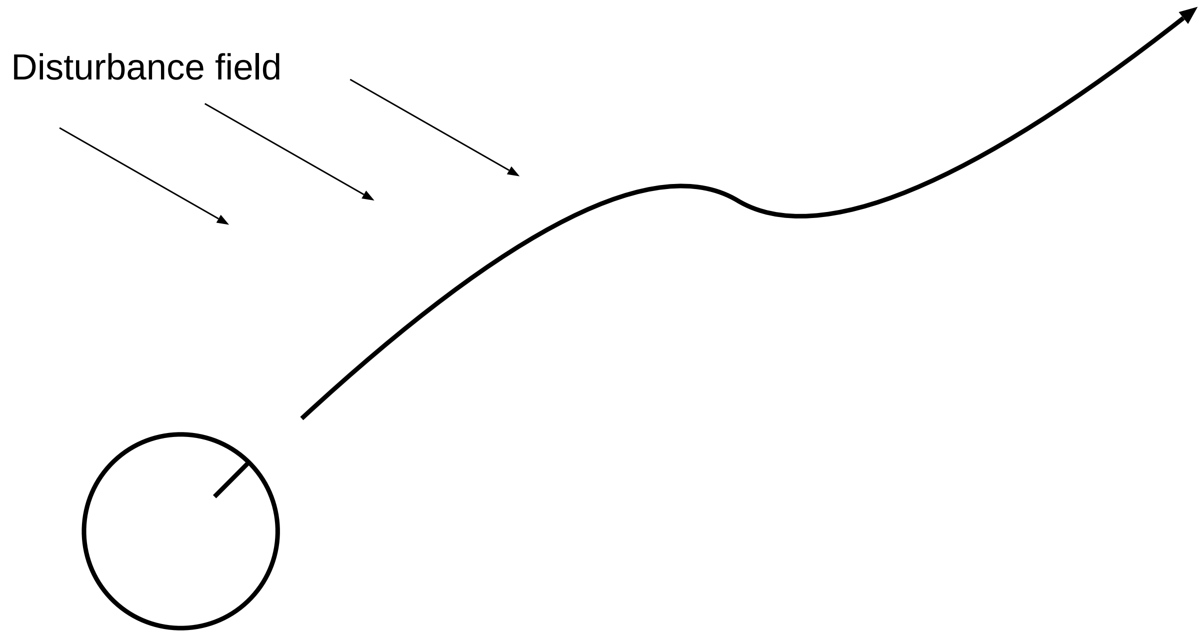
$$c_{t,ijk} = \begin{cases} c_{ij} & \text{keys at } k \text{ on trial } t \\ 0 & \text{otherwise} \end{cases}$$

Example: regression

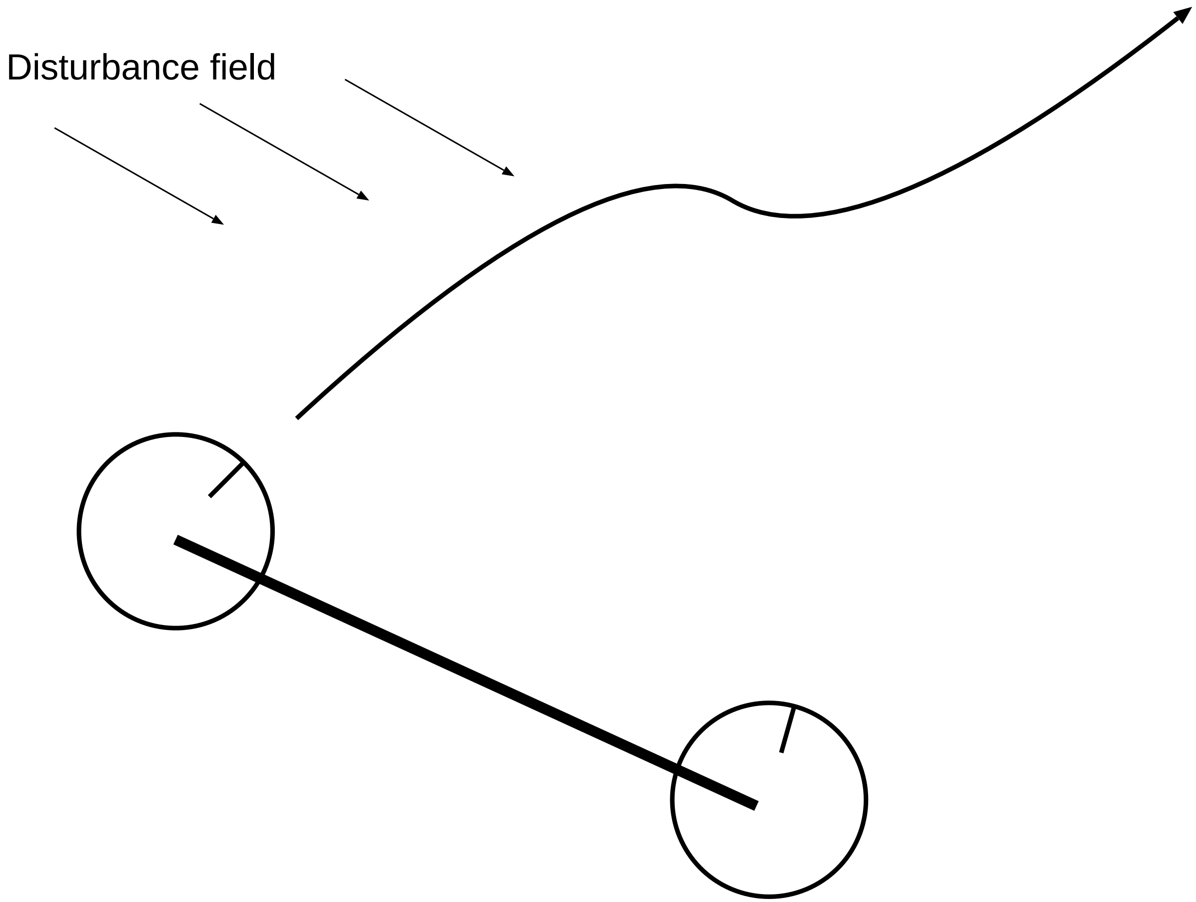
Linear regression w/ 2 agents

Motivation: compensation for drift in a controller, or actor-critic

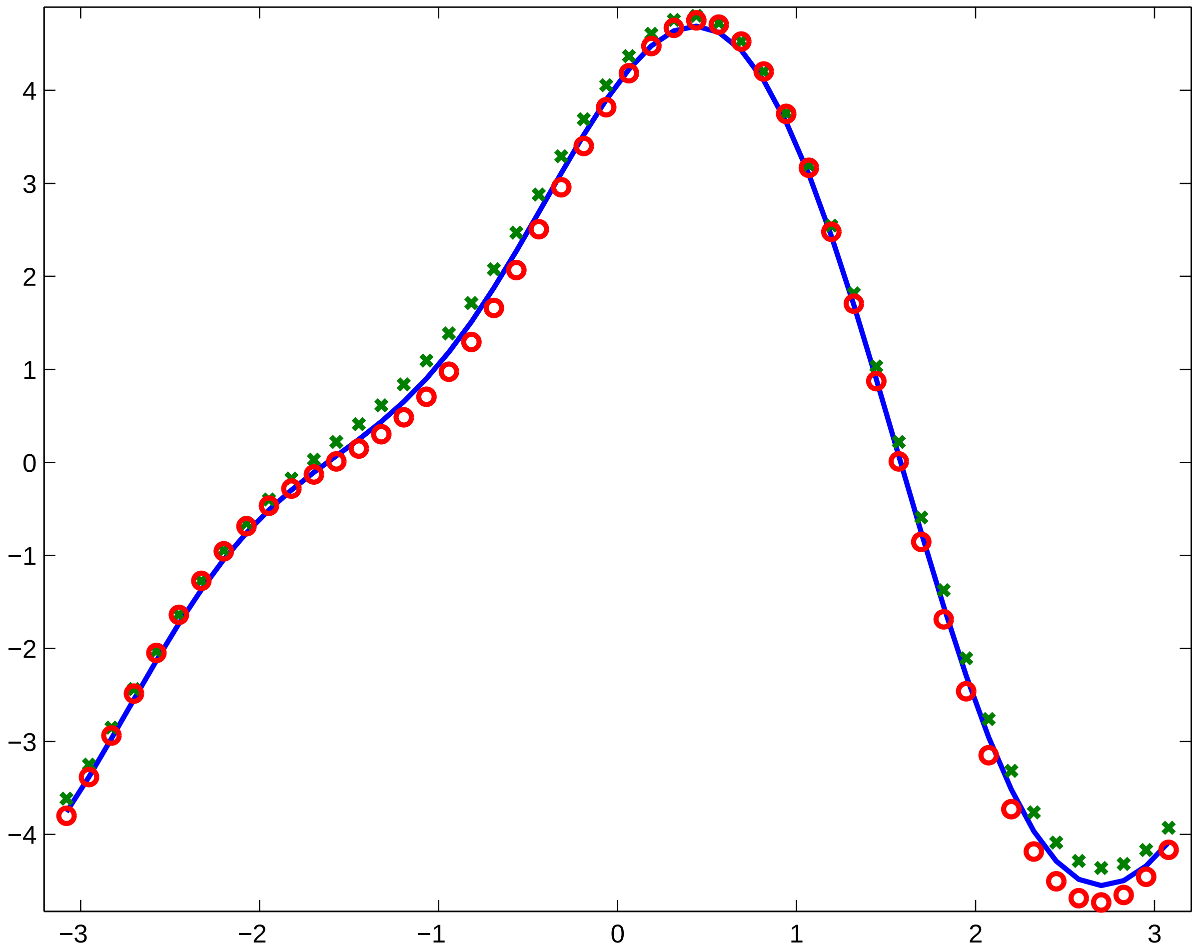
Drift compensation



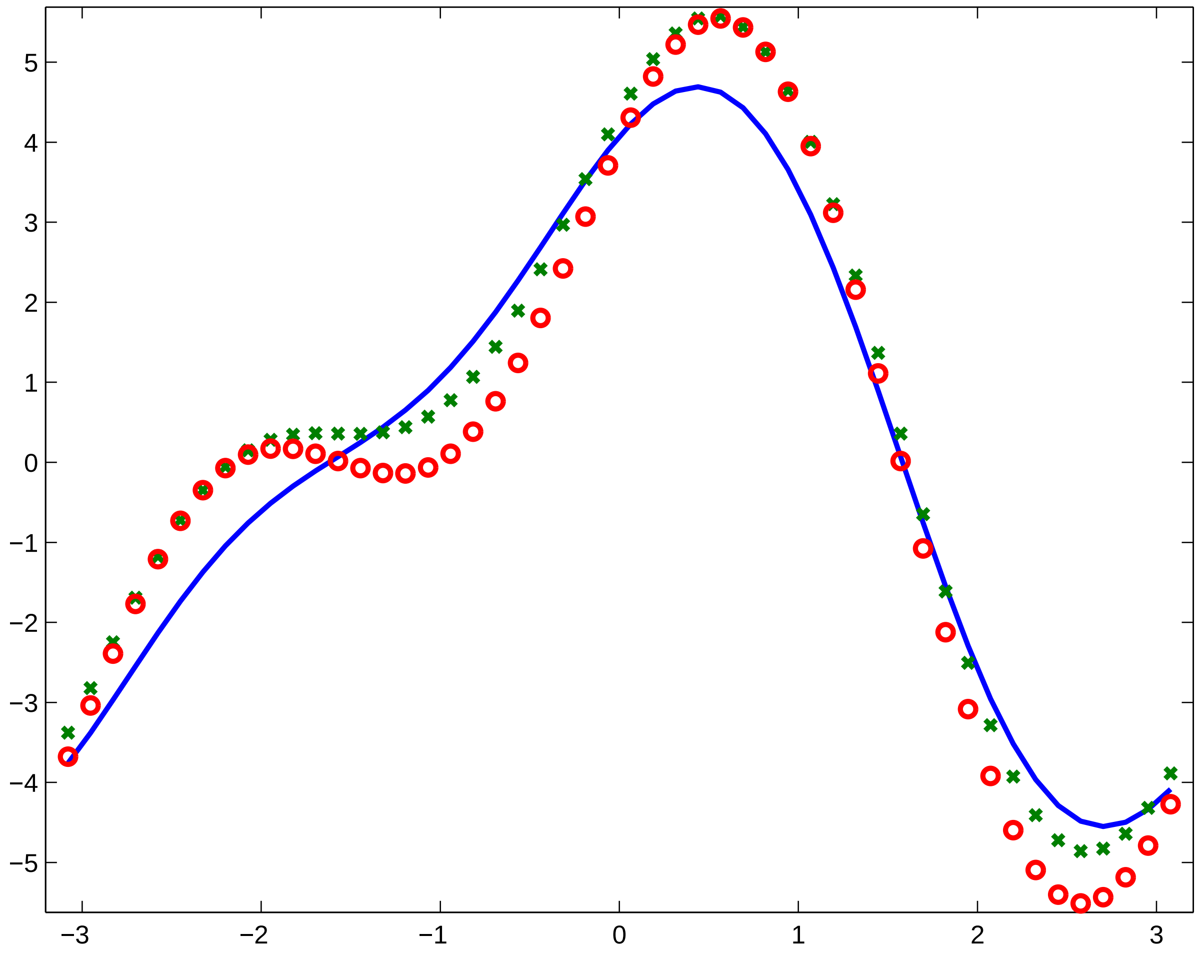
Drift compensation



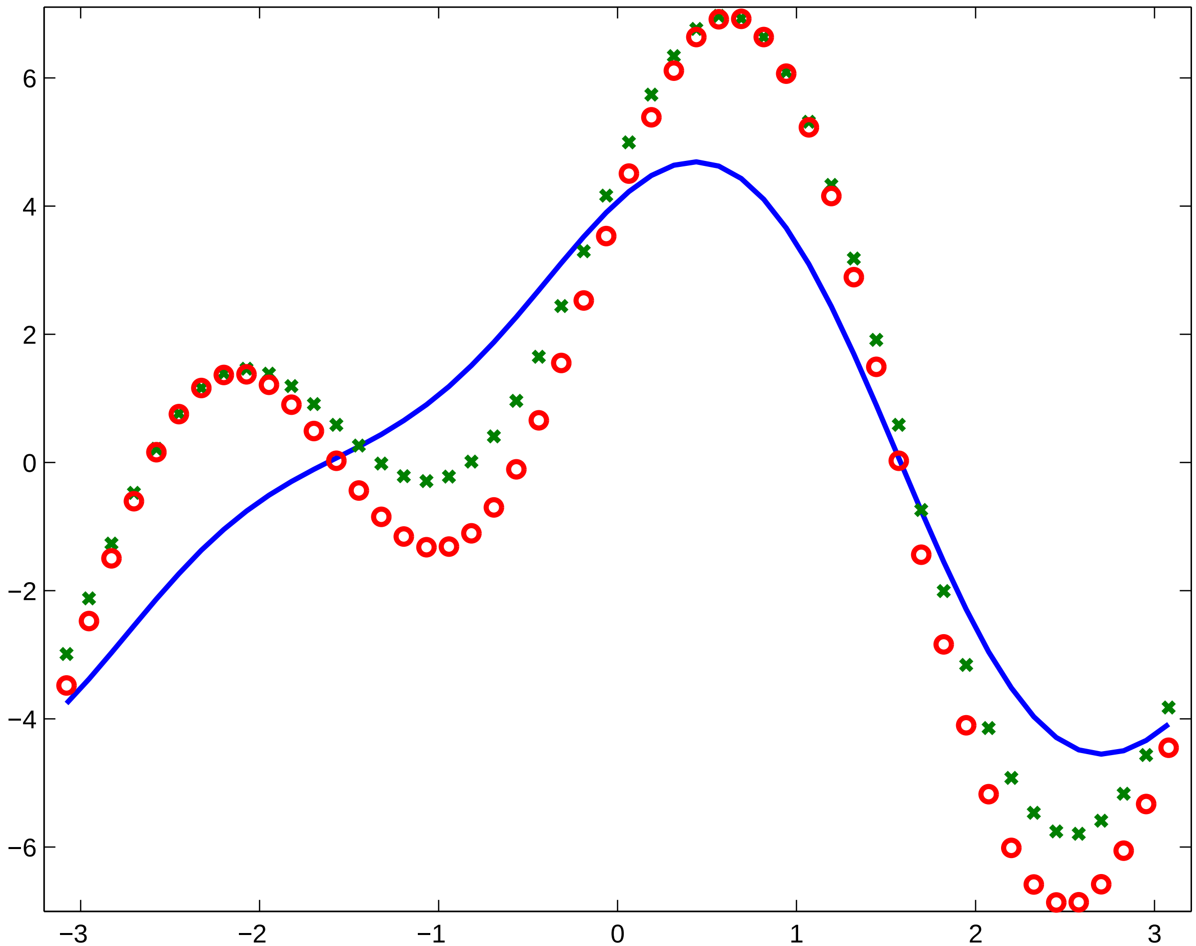
One-d view



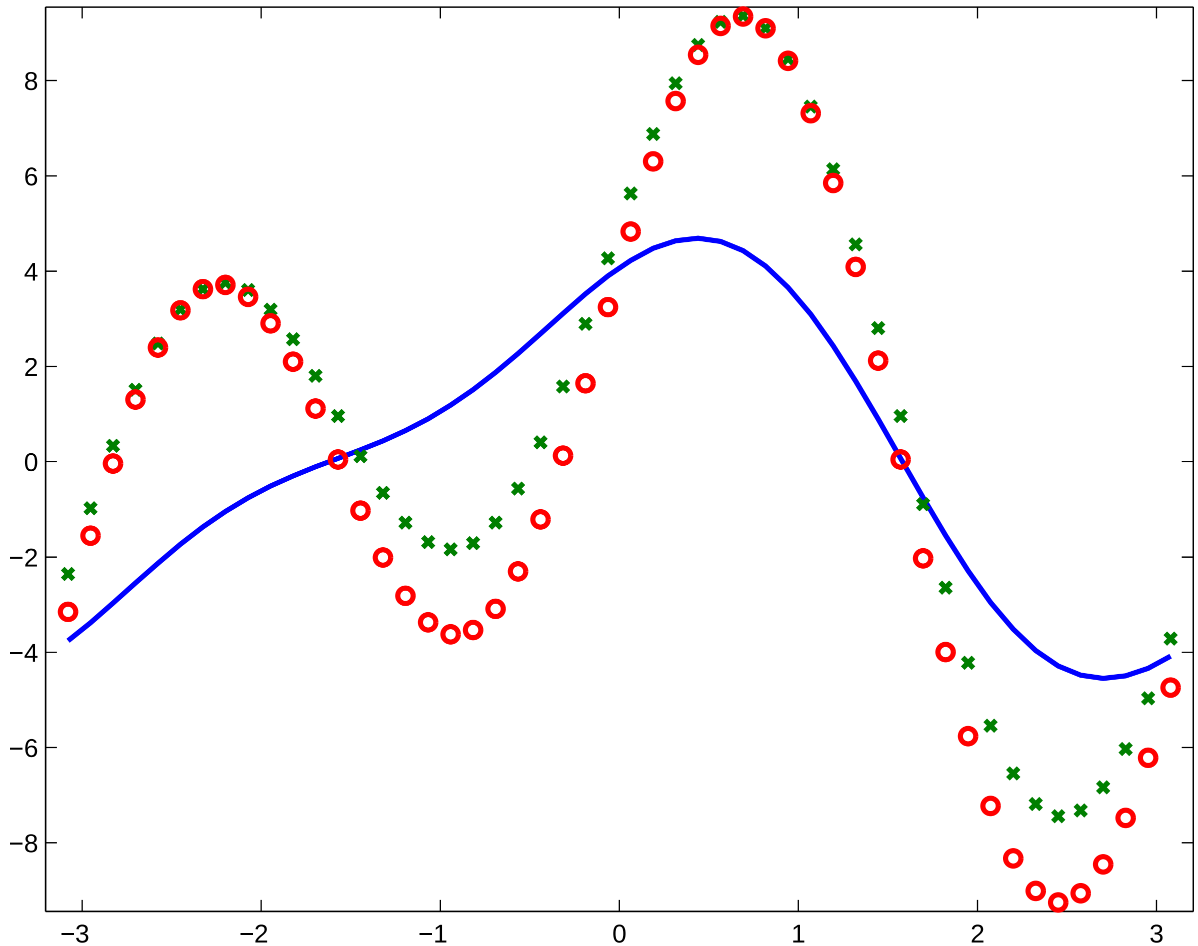
After 5 steps



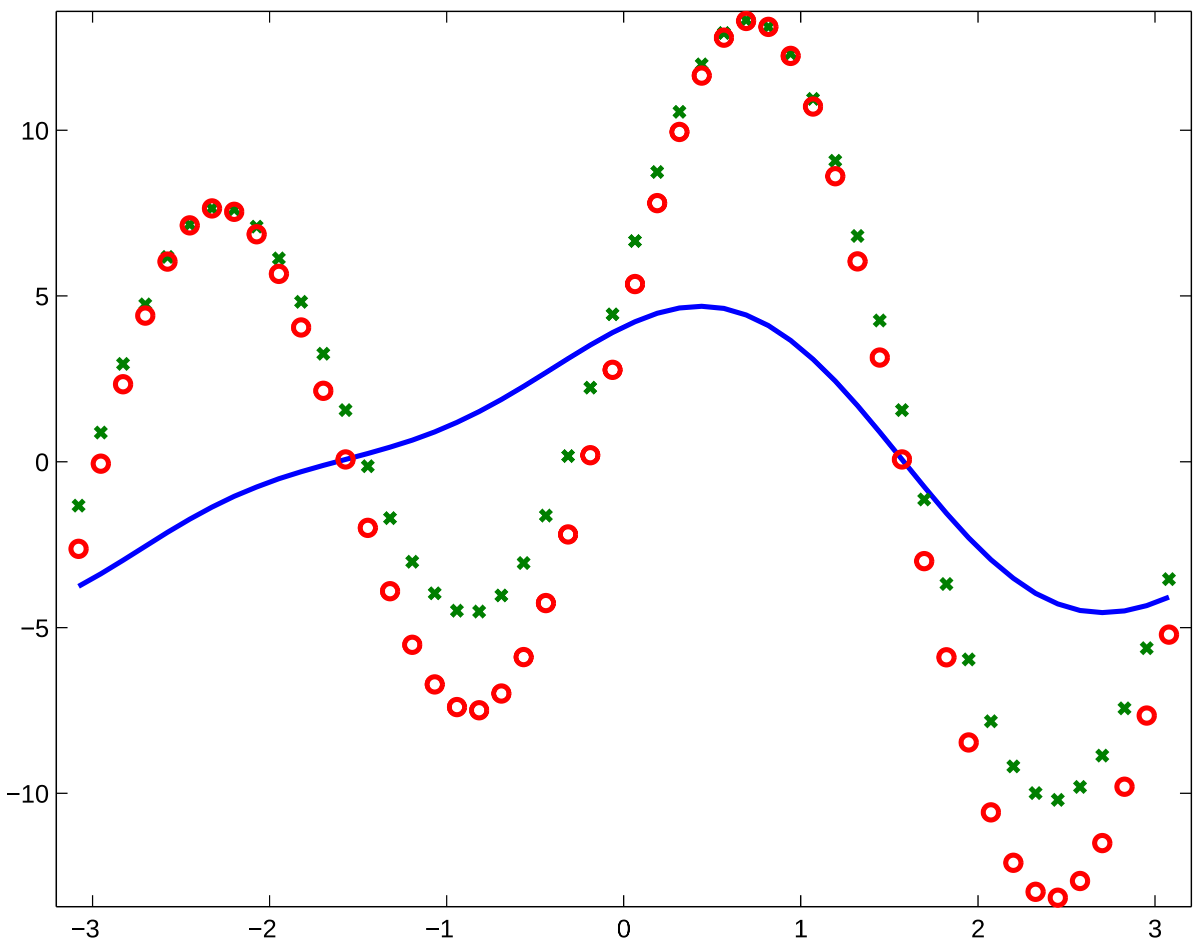
After 10 steps



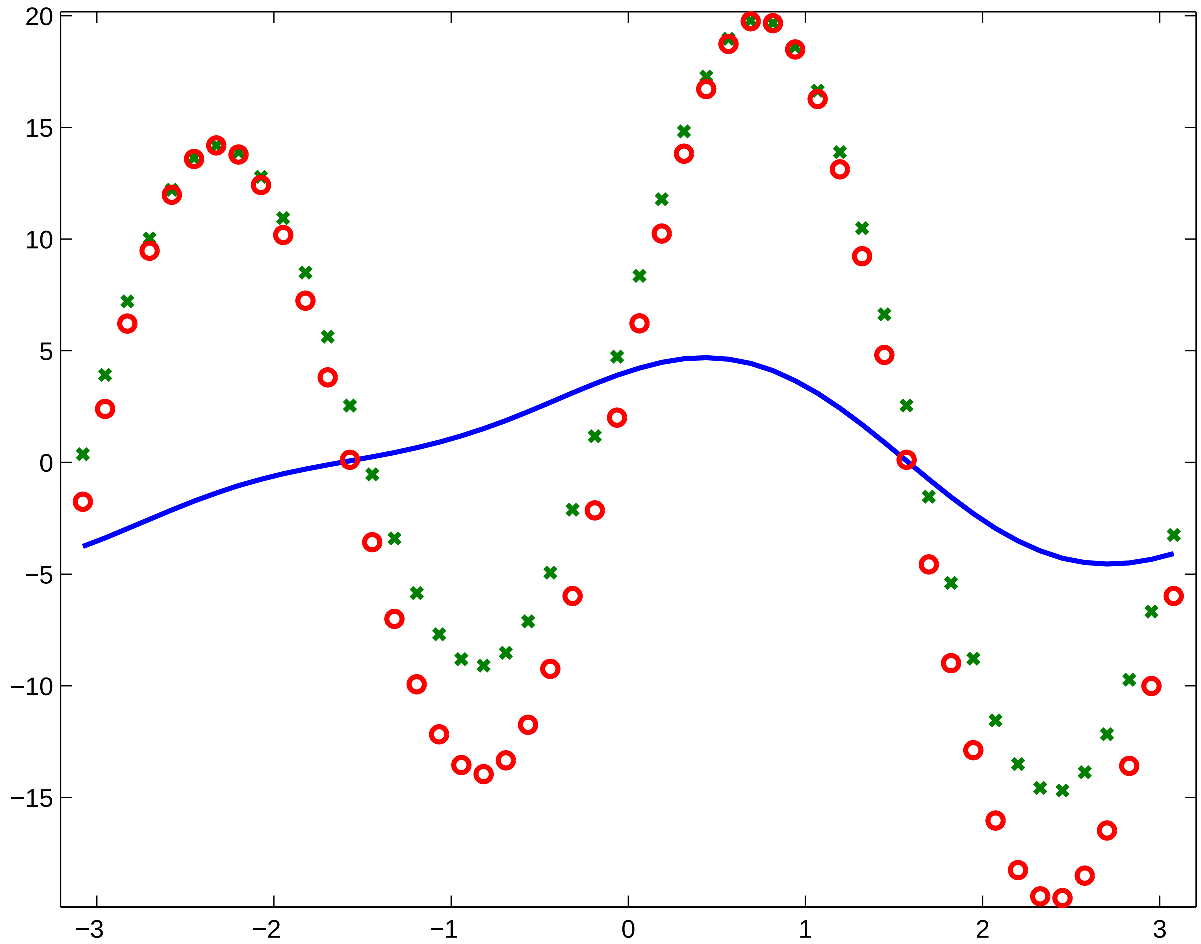
After 15 steps



After 20 steps



After 25 steps



Conclusions

Argued that “reasonable” learners in repeated matrix games should seek feasible, IR, and Pareto-optimal payoffs

If other players reasonable, should converge to equilibrium

If others stationary, best response

If others unreasonable, minimax

Conclusions, cont'd

If Nature has state, move to repeated OCP

Open questions:

- reducing requirements for observability
- achieving subgame perfection
- reducing size of representations

Thanks: Ron Parr, Yoav Shoham's group, Sebastian Thrun