



Machine Learning: Dynamical, Statistical and Economic Perspectives

Michael Jordan

University of California, Berkeley

Machine Learning (aka AI) Successes

- First Generation ('90-'00): the **backend**
 - e.g., fraud detection, search, supply-chain management
- Second Generation ('00-'10): the **human side**
 - e.g., recommendation systems, commerce, social media
- Third Generation ('10-now): **pattern recognition**
 - e.g., speech recognition, computer vision, translation

What Intelligent Systems Currently Exist?

What Intelligent Systems Currently Exist?

- Brains and Minds

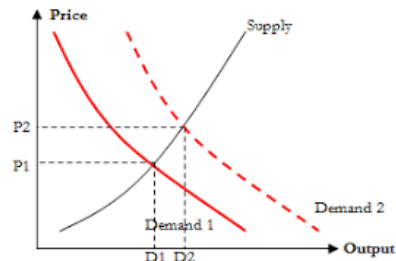
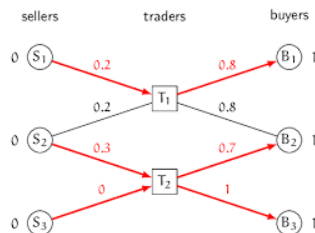


What Intelligent Systems Currently Exist?

- Brains and Minds



- Markets



AI (aka Machine Learning) Successes

- First Generation ('90-'00): the **backend**
 - e.g., fraud detection, search, supply-chain management
- Second Generation ('00-'10): the **human side**
 - e.g., recommendation systems, commerce, social media
- Third Generation ('10-now): **pattern recognition**
 - e.g., speech recognition, computer vision, translation
- Fourth Generation (emerging): **markets**
 - not just one agent making a decision or sequence of decisions
 - but a huge interconnected web of data, agents, decisions
 - many new challenges!

Decisions

- It's not just a matter of a threshold

Decisions

- It's not just a matter of a threshold
- Real-world decisions with consequences
 - counterfactuals, provenance, relevance, dialog

Decisions

- It's not just a matter of a threshold
- Real-world decisions with consequences
 - counterfactuals, provenance, relevance, dialog
- Sets of decisions across a network
 - false-discovery rate (instead of precision/recall/accuracy)
- Sets of decisions across a network over time
 - streaming, asynchronous decisions (cf. Zrnic, Ramdas & Jordan, *Asynchronous online testing of multiple hypotheses*, arXiv, 2019)

Decisions

- It's not just a matter of a threshold
- Real-world decisions with consequences
 - counterfactuals, provenance, relevance, dialog
- Sets of decisions across a network
 - false-discovery rate (instead of precision/recall/accuracy)
- Sets of decisions across a network over time
 - streaming, asynchronous decisions (cf. Zrnic, Ramdas & Jordan, *Asynchronous online testing of multiple hypotheses*, arXiv, 2019)
- Decisions when there is scarcity and competition
 - need for an economic perspective

Decisions

- It's not just a matter of a threshold
- Real-world decisions with consequences
 - counterfactuals, provenance, relevance, dialog
- Sets of decisions across a network
 - false-discovery rate (instead of precision/recall/accuracy)
- Sets of decisions across a network over time
 - streaming, asynchronous decisions (cf. Zrnic, Ramdas & Jordan, *Asynchronous online testing of multiple hypotheses*, arXiv, 2019)
- Decisions when there is scarcity and competition
 - need for an economic perspective
- The current (human-imitative) focus on pattern recognition and reinforcement learning is limiting

Consider Classical Recommendation Systems

- A record is kept of each customer's purchases
- Customers are “similar” if they buy similar sets of items
- Items are “similar” are they are bought together by multiple customers

Consider Classical Recommendation Systems

- A record is kept of each customer's purchases
- Customers are “similar” if they buy similar sets of items
- Items are “similar” if they are bought together by multiple customers
- Recommendations are made on the basis of these similarities
- These systems have become a commodity

Multiple Decisions with Competition

- Suppose that recommending a certain movie is a good business decision (e.g., because it's very popular)
- Is it OK to recommend the same movie to everyone?
- Is it OK to recommend the same book to everyone?
- Is it OK to recommend the same restaurant to everyone?
- Is it OK to recommend the same street to every driver?
- Is it OK to recommend the same stock purchase to everyone?

The Alternative: Create a Market

- A two-way market between consumers and producers
 - based on recommendation systems on both sides
- E.g., diners are one side of the market, and restaurants on the other side
- E.g., drivers are one side of the market, and street segments on the other side
- This isn't just classical microeconomics; the use of recommendation systems is key

AI = Data + Algorithms + Markets

- Computers are currently gathering huge amounts of data, for and about humans, to be fed into learning algorithms
 - often the goal is to learn to **imitate** humans
 - a related goal is to provide **personalized services** to humans
 - but there's a lot of guessing going on about what people want

AI = Data + Algorithms + Markets

- Computers are currently gathering huge amounts of data, for and about humans, to be fed into learning algorithms
 - often the goal is to learn to **imitate** humans
 - a related goal is to provide **personalized services** to humans
 - but there's a lot of guessing going on about what people want
- Services are best provided in the context of a **market**; market design can eliminate much of the guesswork
 - when **data flows** in a market, the underlying system can learn from that data, so that the market provides better services
 - **fairness** arises not from providing the same service to everyone, but by allowing individual utilities to be expressed

Social Consequences

- By creating a market based on the data flows, new jobs are created!
- So here's a way that AI can be a job creator, and not (mostly) a job killer
- This can be done in a wide range of other domains, not just music
 - entertainment
 - information services
 - personal services
- The markets-meets-learning approach deals with other problems that a pure learning approach does not
 - e.g., recommendations when there is scarcity

Example: Music in the Data Age

- More people are making music than ever before, placing it on sites such as SoundCloud
- More people are listening to music than ever before
- But there is no economic value being exchanged between producers and consumers
- And, not surprisingly, most people who make music cannot do it as their full-time job
 - i.e., human happiness is being left on the table

Example: Music in the Data Age

- More people are making music than ever before, placing it on sites such as SoundCloud
- More people are listening to music than ever before
- But there is no economic value being exchanged between producers and consumers
- And, not surprisingly, most people who make music cannot do it as their full-time job
 - i.e., human happiness is being left on the table
- There do exist companies who make money off of this; they stream data from SoundCloud to listeners, and they make their money ... from advertising! ☹️

The Alternative: Create a Market

- Use data to provide a **dashboard** to musicians, letting them learn where their audience is
- The musician can give shows where they have an audience
- And they can make **offers** to their fans

The Alternative: Create a Market

- Use data to provide a **dashboard** to musicians, letting them learn where their audience is
- The musician can give shows where they have an audience
- And they can make **offers** to their fans
- I.e., consumers and producers become linked, and value flows: a market is created
 - the company that creates this market profits simply by taking a cut from the transactions

The Alternative: Create a Market

- Use data to provide a **dashboard** to musicians, letting them learn where their audience is
- The musician can give shows where they have an audience
- And they can make **offers** to their fans
- I.e., consumers and producers become linked, and value flows: a market is created
 - the company that creates this market profits simply by taking a cut from the transactions
- In the US, the company *United Masters* is doing precisely this; see www.unitedmasters.com

Perspectives on AI

- The classical “human-imitative” perspective
 - cf. AI in the movies, interactive home robotics
- The “intelligence augmentation” (IA) perspective
 - cf. search engines, recommendation systems, natural language translation
 - the system need not be intelligent itself, but it reveals patterns that humans can make use of
- The “intelligent infrastructure” (II) perspective
 - cf. transportation, intelligent dwellings, urban planning
 - large-scale, distributed collections of data flows and loosely-coupled decisions

M. Jordan (2018), “Artificial Intelligence: The Revolution Hasn’t Happened Yet”, *Medium*.

Near-Term Challenges in II

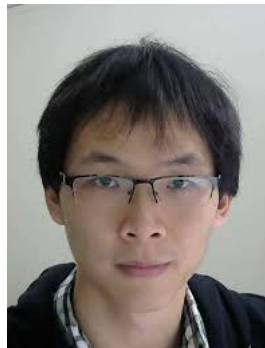
- Error control for **multiple** decisions
- Systems that create **markets**
- Designing systems that can provide meaningful, calibrated notions of their **uncertainty**
- Managing **cloud-edge** interactions
- Designing systems that can find **abstractions** quickly
- **Provenance** in systems that learn and predict
- Designing systems that can **explain** their decisions
- Finding causes and performing **causal** reasoning
- Systems that pursue **long-term goals**, and actively collect data in service of those goals
- Achieving **real-time** performance goals
- Achieving **fairness** and **diversity**
- Robustness in the face of **unexpected situations**
- Robustness in the face of **adversaries**
- **Sharing data** among individuals and organizations
- Protecting **privacy** and data ownership

Algorithmic and Theoretical Progress

- Nonconvex optimization
 - avoidance of saddle points
 - rates that have dimension dependence
 - acceleration, dynamical systems and lower bounds
 - statistical guarantees from optimization guarantees
- Computationally-efficient sampling
 - nonconvex functions
 - nonreversible MCMC
 - links to optimization
- Market design
 - approach to saddle points
 - recommendations and two-way markets

Part I: How to Escape Saddle Points Efficiently

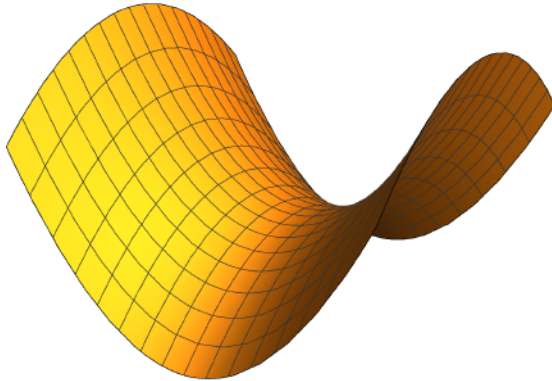
with Chi Jin, Praneeth Netrapalli, Rong Ge,
and Sham Kakade



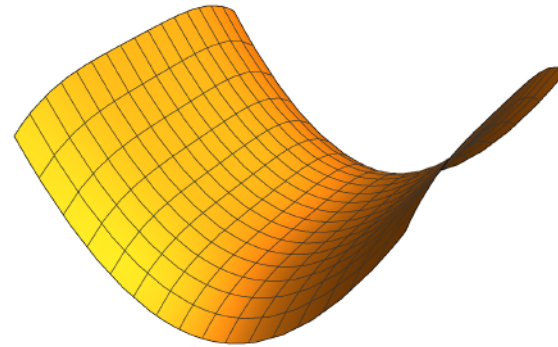
Nonconvex Optimization in Machine Learning

- Bad local minima used to be thought of as the main problem on the optimization side of machine learning
- But many machine learning architectures either have no local minima (see list later), or stochastic gradient seems to have no trouble (eventually) finding global optima
- But **saddle points** abound in these architectures, and they cause the learning curve to flatten out, perhaps (nearly) indefinitely

The Importance of Saddle Points



Strict saddle point



Non-strict saddle point

- How to escape?
 - need to have a negative eigenvalue that's strictly negative
- How to escape **efficiently**?
 - in high dimensions how do we find the direction of escape?
 - should we expect exponential complexity in dimension?

A Few Facts

- Gradient descent will **asymptotically** avoid saddle points (Lee, Simchowitz, Jordan & Recht, 2017)
- Gradient descent can take **exponential time** to escape saddle points (Du, Jin, Lee, Jordan, & Singh, 2017)
- Stochastic gradient descent can escape saddle points in **polynomial** time (Ge, Huang, Jin & Yuan, 2015)
 - but that's still not an explanation for its practical success
- Can we prove a stronger theorem?

Optimization

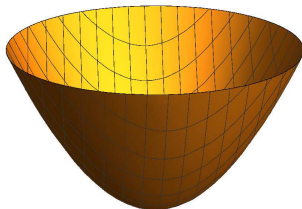
Consider problem:

$$\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$$

Gradient Descent (GD):

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t).$$

Convex: converges to global minimum; **dimension-free** iterations.



Convergence to FOSP

Function $f(\cdot)$ is l -smooth (or gradient Lipschitz)

$$\forall \mathbf{x}_1, \mathbf{x}_2, \|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \leq l\|\mathbf{x}_1 - \mathbf{x}_2\|.$$

Point \mathbf{x} is an ϵ -first-order stationary point (ϵ -FOSP) if

$$\|\nabla f(\mathbf{x})\| \leq \epsilon$$

Theorem [GD Converges to FOSP (Nesterov, 1998)]

For l -smooth function, GD with $\eta = 1/l$ finds ϵ -FOSP in iterations:

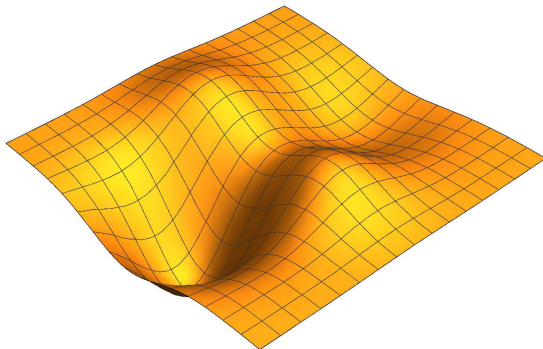
$$\frac{2l(f(\mathbf{x}_0) - f^*)}{\epsilon^2}$$

*Number of iterations is dimension free.

Nonconvex Optimization

Non-convex: converges to Stationary Point (SP) $\nabla f(\mathbf{x}) = 0$.

SP : local min / local max / saddle points



Many applications: no spurious local min (see full list later).

Definitions and Algorithm

Function $f(\cdot)$ is ρ -**Hessian Lipschitz** if

$$\forall \mathbf{x}_1, \mathbf{x}_2, \quad \|\nabla^2 f(\mathbf{x}_1) - \nabla^2 f(\mathbf{x}_2)\| \leq \rho \|\mathbf{x}_1 - \mathbf{x}_2\|.$$

Point \mathbf{x} is an ϵ -**second-order stationary point** (ϵ -SOSP) if

$$\|\nabla f(\mathbf{x})\| \leq \epsilon, \quad \text{and} \quad \lambda_{\min}(\nabla^2 f(\mathbf{x})) \geq -\sqrt{\rho\epsilon}$$

Definitions and Algorithm

Function $f(\cdot)$ is ρ -**Hessian Lipschitz** if

$$\forall \mathbf{x}_1, \mathbf{x}_2, \quad \|\nabla^2 f(\mathbf{x}_1) - \nabla^2 f(\mathbf{x}_2)\| \leq \rho \|\mathbf{x}_1 - \mathbf{x}_2\|.$$

Point \mathbf{x} is an ϵ -**second-order stationary point** (ϵ -SOSP) if

$$\|\nabla f(\mathbf{x})\| \leq \epsilon, \quad \text{and} \quad \lambda_{\min}(\nabla^2 f(\mathbf{x})) \geq -\sqrt{\rho\epsilon}$$

Algorithm Perturbed Gradient Descent (PGD)

1. **for** $t = 0, 1, \dots$ **do**
2. **if** perturbation condition holds **then**
3. $\mathbf{x}_t \leftarrow \mathbf{x}_t + \xi_t, \quad \xi_t$ uniformly $\sim \mathbb{B}_0(r)$
4. $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)$

Adds perturbation when $\|\nabla f(\mathbf{x}_t)\| \leq \epsilon$; no more than once per T steps.

Main Result

Theorem [PGD Converges to SOSP]

For ℓ -smooth and ρ -Hessian Lipschitz function f , PGD with $\eta = O(1/\ell)$ and proper choice of r, T w.h.p. finds ϵ -SOSP in iterations:

$$\tilde{O}\left(\frac{\ell(f(\mathbf{x}_0) - f^*)}{\epsilon^2}\right)$$

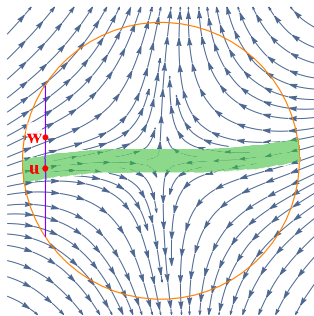
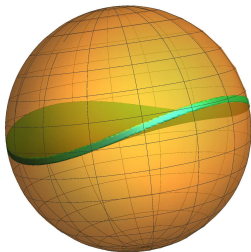
*Dimension dependence in iteration is $\log^4(d)$ (almost dimension free).

	GD (Nesterov 1998)	PGD (This Work)
Assumptions	ℓ -grad-Lip	ℓ -grad-Lip + ρ -Hessian-Lip
Guarantees	ϵ -FOSP	ϵ -SOSP
Iterations	$2\ell(f(\mathbf{x}_0) - f^*)/\epsilon^2$	$\tilde{O}(\ell(f(\mathbf{x}_0) - f^*)/\epsilon^2)$

Geometry and Dynamics around Saddle Points

Challenge: non-constant Hessian + large step size $\eta = O(1/\ell)$.

Around saddle point, **stuck region** forms a non-flat “pancake” shape.



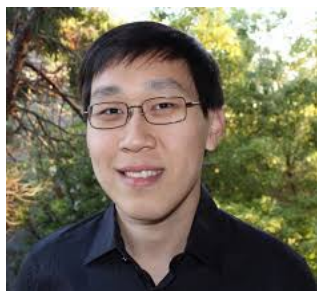
Key Observation: although we don't know its shape, we know it's thin!
(Based on an analysis of two nearly coupled sequences)

How Fast Can We Go?

- Important role of **lower bounds** (Nemirovski & Yudin)
 - strip away inessential aspects of the problem to reveal fundamentals
- The **acceleration** phenomenon (Nesterov)
 - achieve the lower bounds
 - second-order dynamics
 - a conceptual **mystery**
- Our perspective: it's essential to go to **continuous time**
 - the notion of "acceleration" requires a continuum topology to support it

Part II: Variational, Hamiltonian and Symplectic Perspectives on Acceleration

with Andre Wibisono, Ashia Wilson and Michael Betancourt



Accelerated gradient descent

Setting: Unconstrained convex optimization

$$\min_{x \in \mathbb{R}^d} f(x)$$

- ▶ Classical gradient descent:

$$x_{k+1} = x_k - \beta \nabla f(x_k)$$

obtains a convergence rate of $O(1/k)$

Accelerated gradient descent

Setting: Unconstrained convex optimization

$$\min_{x \in \mathbb{R}^d} f(x)$$

- ▶ Classical gradient descent:

$$x_{k+1} = x_k - \beta \nabla f(x_k)$$

obtains a convergence rate of $O(1/k)$

- ▶ Accelerated gradient descent:

$$\begin{aligned}y_{k+1} &= x_k - \beta \nabla f(x_k) \\x_{k+1} &= (1 - \lambda_k)y_{k+1} + \lambda_k y_k\end{aligned}$$

obtains the (optimal) convergence rate of $O(1/k^2)$

Accelerated methods: Continuous time perspective

- ▶ Gradient descent is discretization of gradient flow

$$\dot{X}_t = -\nabla f(X_t)$$

(and mirror descent is discretization of natural gradient flow)

Accelerated methods: Continuous time perspective

- ▶ Gradient descent is discretization of gradient flow

$$\dot{X}_t = -\nabla f(X_t)$$

(and mirror descent is discretization of natural gradient flow)

- ▶ Su, Boyd, Candes '14: Continuous time limit of accelerated gradient descent is a second-order ODE

$$\ddot{X}_t + \frac{3}{t}\dot{X}_t + \nabla f(X_t) = 0$$

Accelerated methods: Continuous time perspective

- ▶ Gradient descent is discretization of gradient flow

$$\dot{X}_t = -\nabla f(X_t)$$

(and mirror descent is discretization of natural gradient flow)

- ▶ Su, Boyd, Candes '14: Continuous time limit of accelerated gradient descent is a second-order ODE

$$\ddot{X}_t + \frac{3}{t}\dot{X}_t + \nabla f(X_t) = 0$$

- ▶ These ODEs are obtained by taking continuous time limits. Is there a deeper generative mechanism?

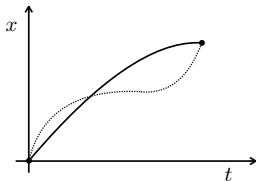
Our work: A general variational approach to acceleration
A systematic discretization methodology

Bregman Lagrangian

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma t + \alpha t} \left(D_h(x + e^{-\alpha t} \dot{x}, x) - e^{\beta t} f(x) \right)$$

Variational problem over curves:

$$\min_X \int \mathcal{L}(X_t, \dot{X}_t, t) dt$$



Optimal curve is characterized by **Euler-Lagrange** equation:

$$\frac{d}{dt} \left\{ \frac{\partial \mathcal{L}}{\partial \dot{x}}(X_t, \dot{X}_t, t) \right\} = \frac{\partial \mathcal{L}}{\partial x}(X_t, \dot{X}_t, t)$$

E-L equation for Bregman Lagrangian under ideal scaling:

$$\ddot{X}_t + (e^{\alpha t} - \dot{\alpha}_t) \dot{X}_t + e^{2\alpha t + \beta t} \left[\nabla^2 h(X_t + e^{-\alpha t} \dot{X}_t) \right]^{-1} \nabla f(X_t) = 0$$

Mysteries

- **Why** can't we discretize the dynamics when we are using exponentially fast clocks?
- **What** happens when we arrive at a clock speed that we can discretize?
- **How** do we discretize once it's possible?

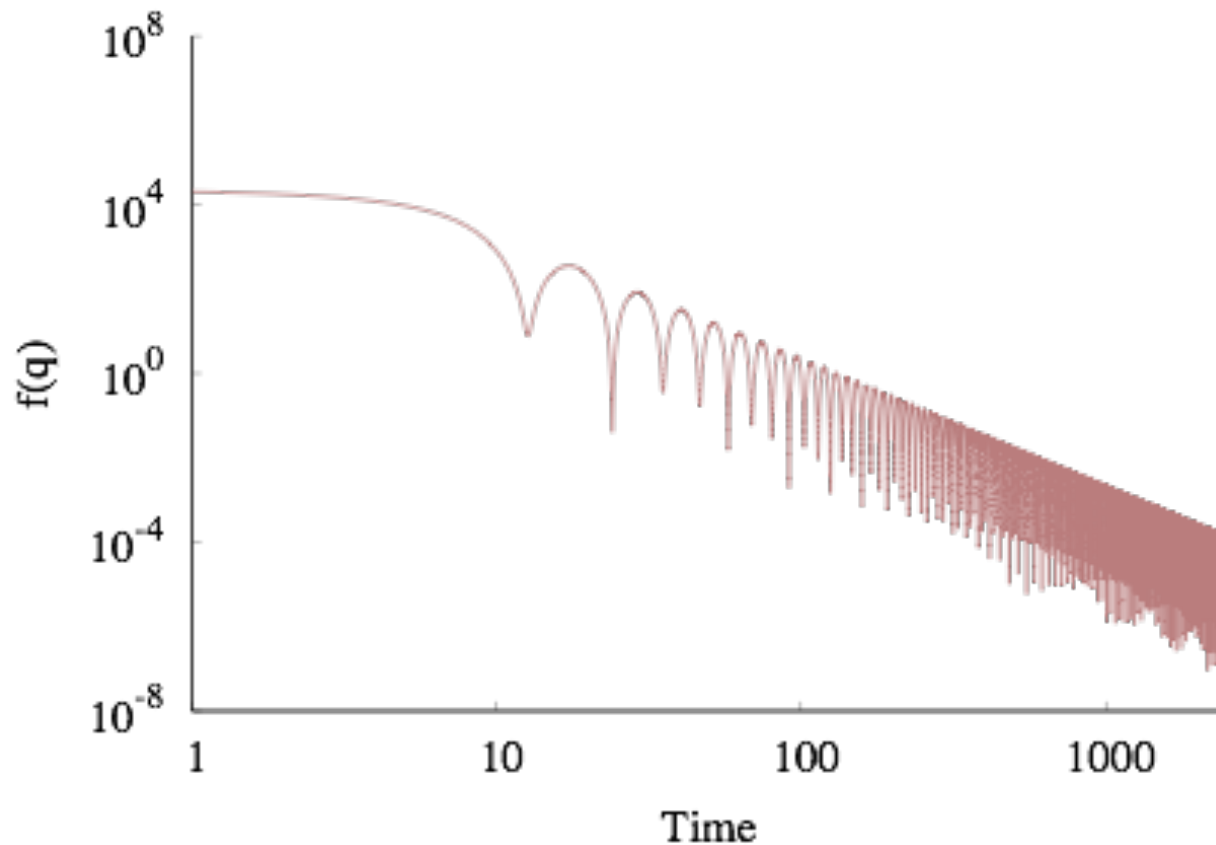
Towards A Symplectic Perspective

- We've discussed discretization of Lagrangian-based dynamics
- Discretization of Lagrangian dynamics is often fragile and requires small step sizes
- We can build more robust solutions by taking a Legendre transform and considering a *Hamiltonian* formalism:

$$L(q, v, t) \rightarrow H(q, p, t, \mathcal{E})$$

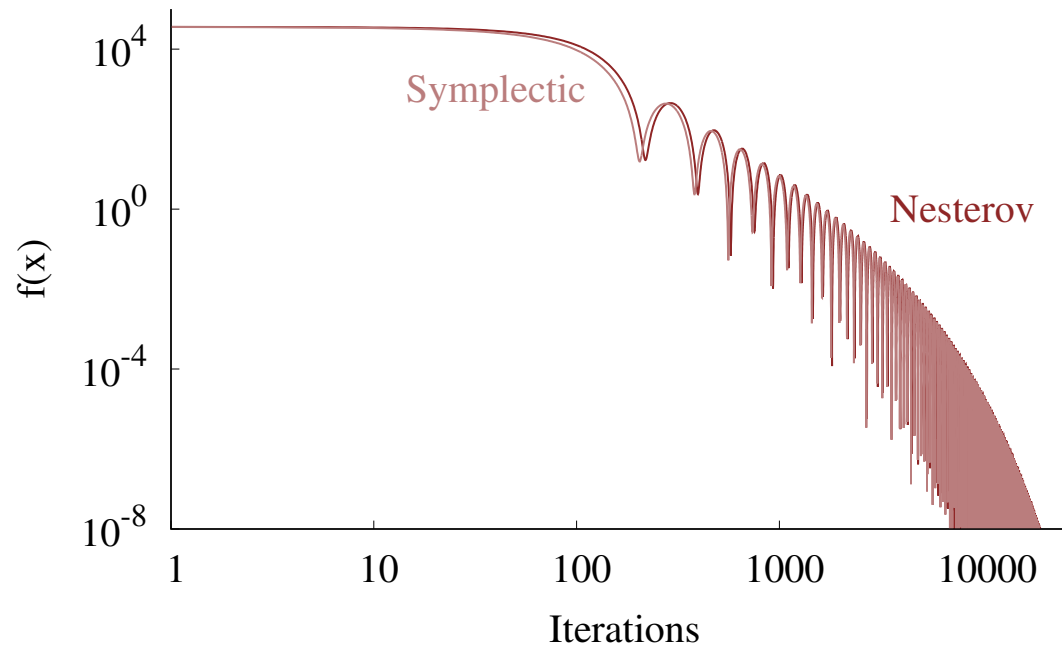
$$\left(\frac{dq}{dt}, \frac{dv}{dt} \right) \rightarrow \left(\frac{dq}{d\tau}, \frac{dp}{d\tau}, \frac{dt}{d\tau}, \frac{d\mathcal{E}}{d\tau} \right)$$

Symplectic Integration of Bregman Hamiltonian



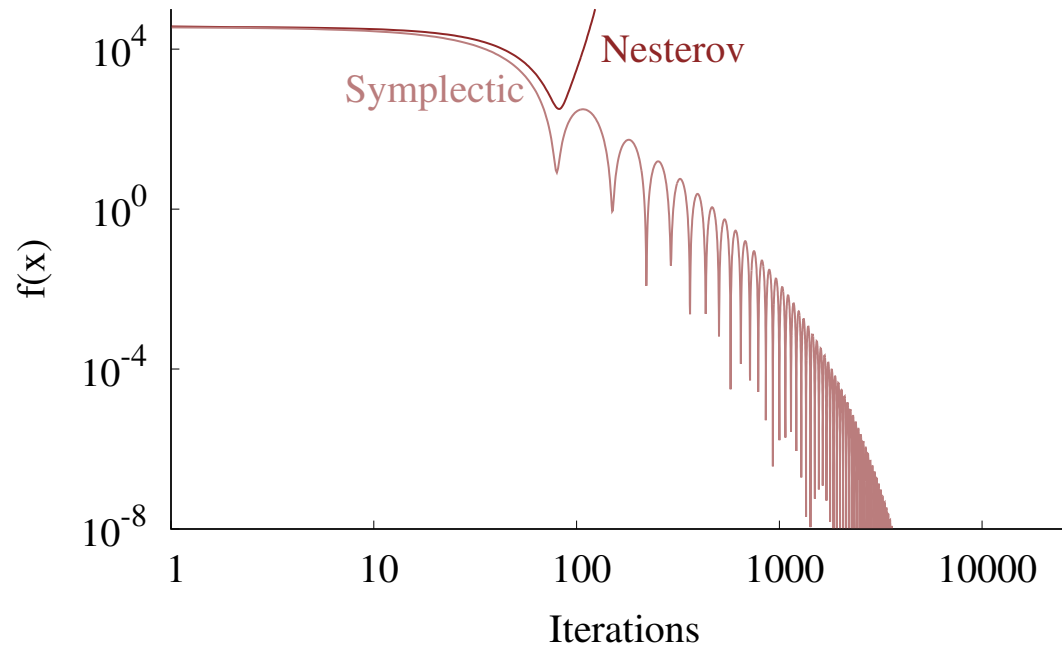
Symplectic vs Nesterov

$p = 2, N = 2, C = 0.0625, \epsilon = 0.1$



Symplectic vs Nesterov

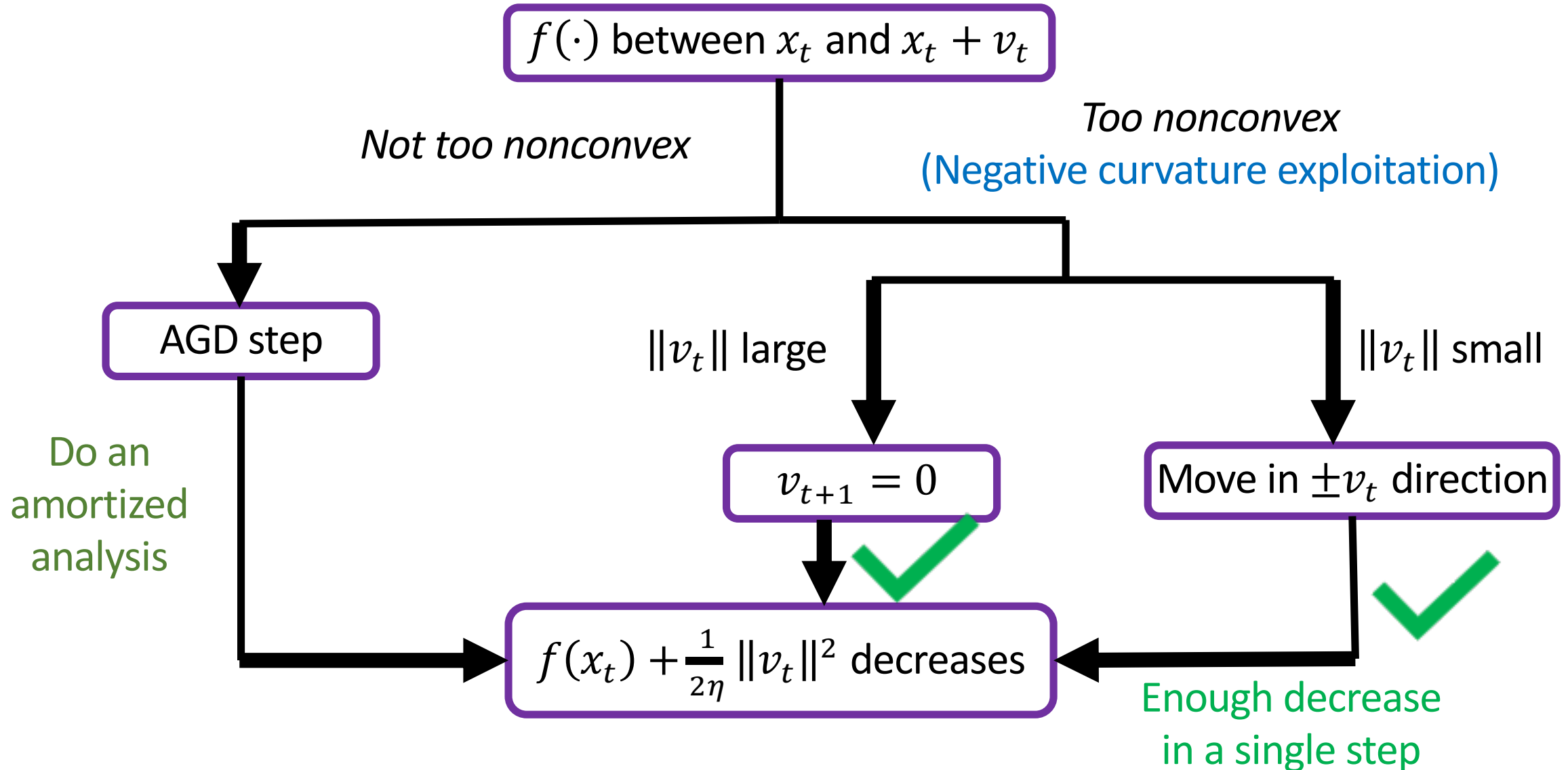
$p = 2, N = 2, C = 0.0625, \varepsilon = 0.25$



Part III: Acceleration and Saddle Points

with Chi Jin and Praneeth Netrapalli

Hamiltonian Analysis



Convergence Result

PAGD Converges to SOSP Faster (Jin et al. 2017)

For l -gradient Lipschitz and ρ -Hessian Lipschitz function f , PAGD with proper choice of $\eta, \theta, r, T, \gamma, s$ w.h.p. finds ϵ -SOSP in iterations:

$$\tilde{O}\left(\frac{l^{1/2}\rho^{1/4}(f(\mathbf{x}_0) - f^*)}{\epsilon^{7/4}}\right)$$

	Strongly Convex	Nonconvex (SOSP)
Assumptions	l -grad-Lip & α -str-convex	l -grad-Lip & ρ -Hessian-Lip
(Perturbed) GD	$\tilde{O}(l/\alpha)$	$\tilde{O}(\Delta_f \cdot l/\epsilon^2)$
(Perturbed) AGD	$\tilde{O}(\sqrt{l/\alpha})$	$\tilde{O}(\Delta_f \cdot l^{1/2}\rho^{1/4}/\epsilon^{7/4})$
Condition κ	l/α	$l/\sqrt{\rho\epsilon}$
Improvement	$\sqrt{\kappa}$	$\sqrt{\kappa}$

Part IV: Acceleration and Stochastics

with Xiang Cheng, Niladri Chatterji and Peter
Bartlett

Acceleration and Stochastics

- Can we accelerate diffusions?
- There have been negative results...
- ...but they've focused on classical **overdamped** diffusions

Acceleration and Stochastics

- Can we accelerate diffusions?
- There have been negative results...
- ...but they've focused on classical **overdamped** diffusions
- Inspired by our work on acceleration, can we accelerate **underdamped** diffusions?

Overdamped Langevin MCMC

Described by the Stochastic Differential Equation (SDE):

$$dx_t = -\nabla U(x_t)dt + \sqrt{2}dB_t$$

where $U(x): R^d \rightarrow R$ and B_t is standard Brownian motion.

The stationary distribution is $p^*(x) \propto \exp(U(x))$

Corresponding Markov Chain Monte Carlo Algorithm (MCMC):

$$\tilde{x}_{(k+1)\delta} = \tilde{x}_{k\delta} - \nabla U(\tilde{x}_{k\delta}) + \sqrt{2\delta}\xi_k$$

where δ is the *step-size* and $\xi_k \sim N(0, I_{d \times d})$

Guarantees under Convexity

Assuming $U(x)$ is L -smooth and m -strongly convex:

Dalalyan'14: Guarantees in Total Variation

$$\text{If } n \geq O\left(\frac{d}{\epsilon^2}\right) \text{ then, } TV(p^{(n)}, p^*) \leq \epsilon$$

Durmus & Moulines'16: Guarantees in 2-Wasserstein

$$\text{If } n \geq O\left(\frac{d}{\epsilon^2}\right) \text{ then, } W_2(p^{(n)}, p^*) \leq \epsilon$$

Cheng and Bartlett'17: Guarantees in KL divergence

$$\text{If } n \geq O\left(\frac{d}{\epsilon^2}\right) \text{ then, } \text{KL}(p^{(n)}, p^*) \leq \epsilon$$

Underdamped Langevin Diffusion

Described by the *second-order* equation:

$$dx_t = v_t dt$$

$$dv_t = -\gamma v_t dt + \lambda \nabla U(x_t) dt + \sqrt{2\gamma\lambda} dB_t$$

The stationary distribution is $p^*(x, v) \propto \exp\left(-U(x) - \frac{|v|^2}{2\lambda}\right)$

Intuitively, x_t is the position and v_t is the velocity

$\nabla U(x_t)$ is the force and γ is the drag coefficient

Quadratic Improvement

Let $p^{(n)}$ denote the distribution of $(\tilde{x}_{n\delta}, \tilde{v}_{n\delta})$. Assume $U(x)$ is strongly convex

Cheng, Chatterji, Bartlett, Jordan '17:

If $n \geq O\left(\frac{\sqrt{d}}{\epsilon}\right)$ then $W_2(p^{(n)}, p^*) \leq \epsilon$

Compare with Durmus & Moulines '16 (Overdamped)

If $n \geq O\left(\frac{d}{\epsilon^2}\right)$ then $W_2(p^{(n)}, p^*) \leq \epsilon$

Proof Idea: Reflection Coupling

Tricky to prove continuous-time process contracts. Consider two processes,

$$\begin{aligned} dx_t &= -\nabla U(x_t)dt + \sqrt{2} dB_t^x \\ dy_t &= -\nabla U(y_t)dt + \sqrt{2} dB_t^y \end{aligned}$$

where $x_0 \sim p_0$ and $y_0 \sim p^*$. Couple these through Brownian motion

$$dB_t^y = \left[I_{d \times d} - \frac{2 \cdot (x_t - y_t)(x_t - y_t)^\top}{\|x_t - y_t\|_2^2} \right] dB_t^x$$

“reflection along line separating the two processes”

Reduction to One Dimension

By Itô's Lemma we can monitor the evolution of the separation distance

$$d|x_t - y_t|_2 = - \underbrace{\left\langle \frac{x_t - y_t}{|x_t - y_t|_2}, \nabla U(x_t) - \nabla U(y_t) \right\rangle}_{\text{'Drift'}} dt + 2\sqrt{2} dB_t^1 \quad \text{'1-d random walk'}$$

Two cases are possible

1. If $|x_t - y_t|_2 \leq R$ then we have strong convexity; the drift helps.
2. If $|x_t - y_t|_2 \geq R$ then the drift hurts us, but Brownian motion helps stick*

Rates not exponential in d as we have a 1- d random walk

*Under a clever choice of Lyapunov function.

Part VI: Acceleration and Sampling

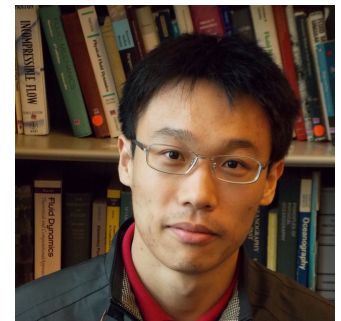
With Yi-An Ma, Niladri Chatterji, and Xiang Cheng

Acceleration of SDEs

- *The underdamped Langevin stochastic differential equation is Nesterov acceleration on the manifold of probability distributions, with respect to the KL divergence (Ma, et al., to appear)*

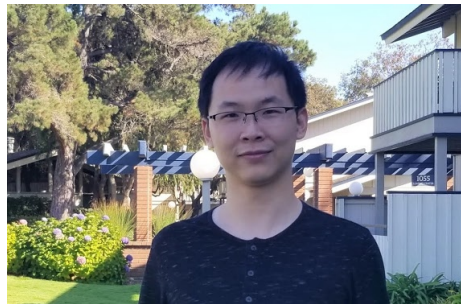
Sampling vs. Optimization: The Tortoise and the Hare

- Folk knowledge: Sampling is slow, while optimization is fast
 - but sampling provides **inferences**, while optimization only provides **point estimates**
- But there hasn't been a clear theoretical analysis that establishes this folk knowledge as true
- Is it really true?

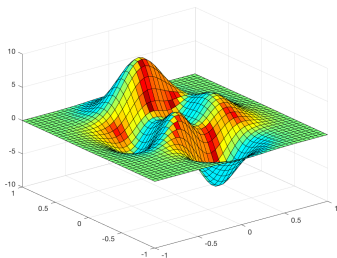


Part V: Population Risk and Empirical Risk

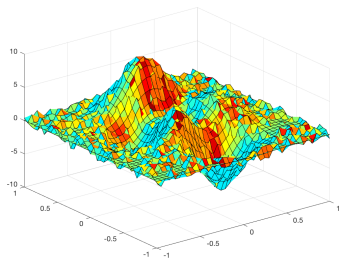
with Chi Jin and Lydia Liu



Population Risk vs Empirical Risk



Well-behaved population risk



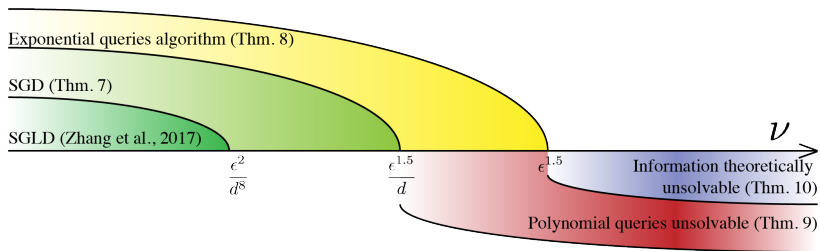
⇒ rough empirical risk

- ▶ Even when R is smooth, \hat{R}_n can be **non-smooth** and may even have many **additional local minima** (ReLU deep networks).
- ▶ Typically $\|R - \hat{R}_n\|_\infty \leq O(1/\sqrt{n})$ by empirical process results.

Can we find local min of R given only access to the function value \hat{R}_n ?

Our Contribution

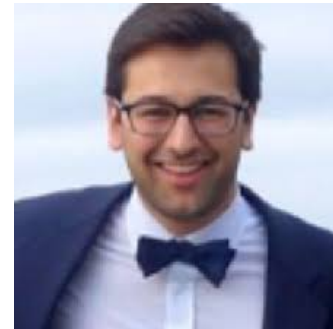
Our answer: **Yes!** Our **SGD** approach finds ϵ -SOSP of F if $\nu \leq \epsilon^{1.5}/d$, which is **optimal among all polynomial queries algorithms**.



Complete characterization of error ν vs accuracy ϵ and dimension d .

Part VII: Market Design Meets Gradient-Based Learning

with Lydia Liu, Horia Mania and Eric Mazumdar



Competing Bandits in Matching Markets



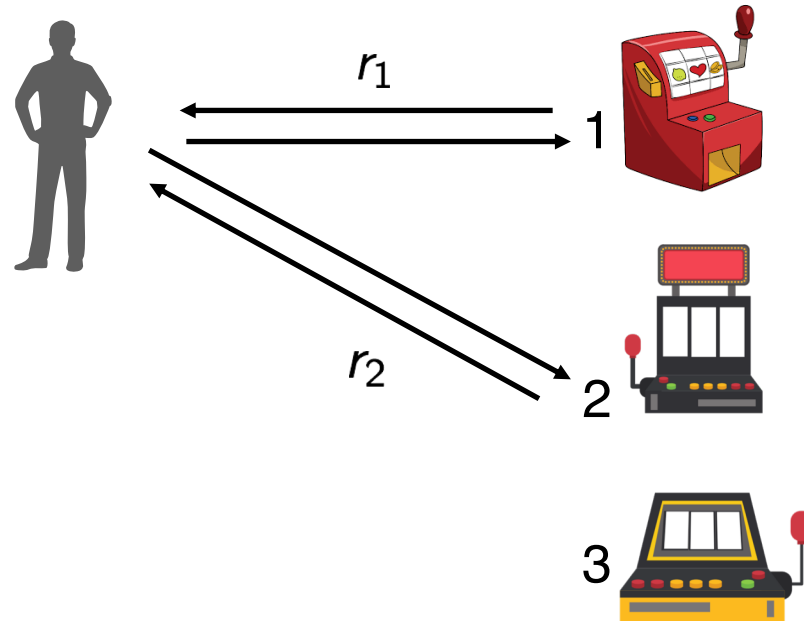
Lydia Liu



Horia Mania

Multi-Armed Bandits

- MABs offer a natural platform to understand exploration / exploitation trade-offs



Matching Markets

The two sides of the market must be matched. But many markets have constraints: capacity, preferences, etc.

Examples:

- Residents and hospitals
- High school admissions
- Restaurants and costumers
- Labor markets
- House allocations with existing tenants
- Many others

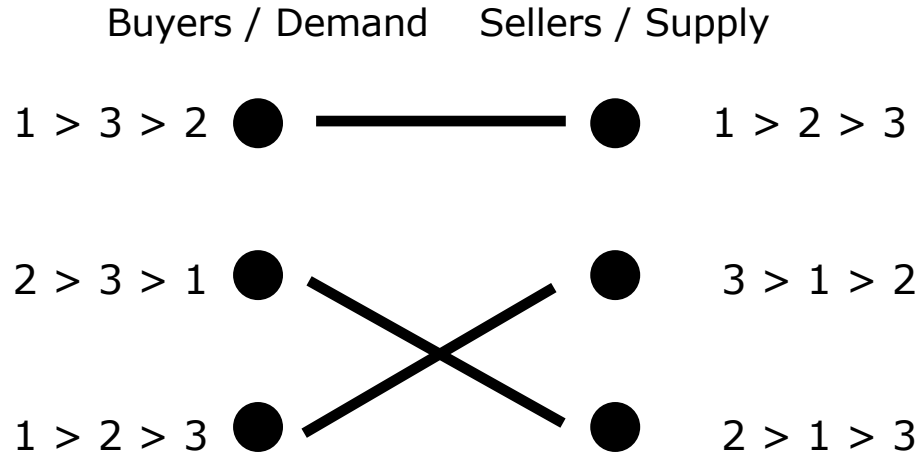
Buyers / Demand

Sellers / Supply



Matching Markets

Suppose we have a market in which the participants have preferences:

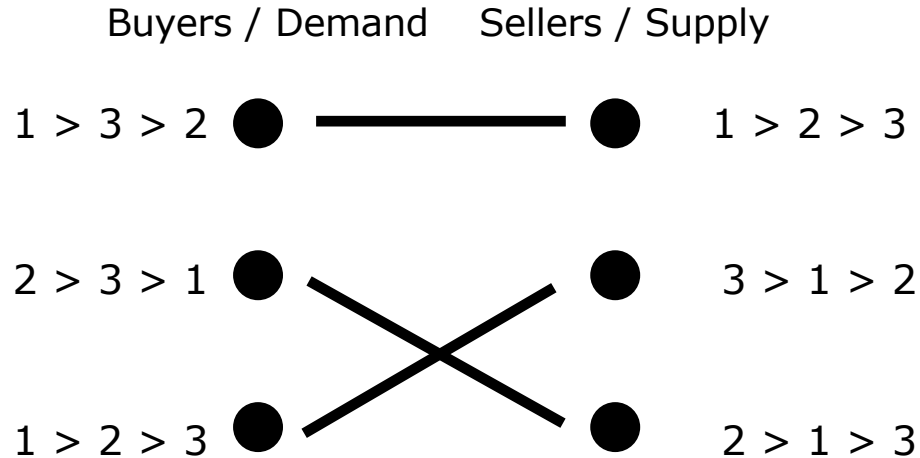


We want to find an equilibrium: no two participants would prefer to be matched with each other over their current match.

Such a matching is called stable.

Matching Markets

Suppose we have a market in which the participants have preferences:



Gale and Shapley introduced this problem in 1962 and proposed a celebrated algorithm that always finds a stable match

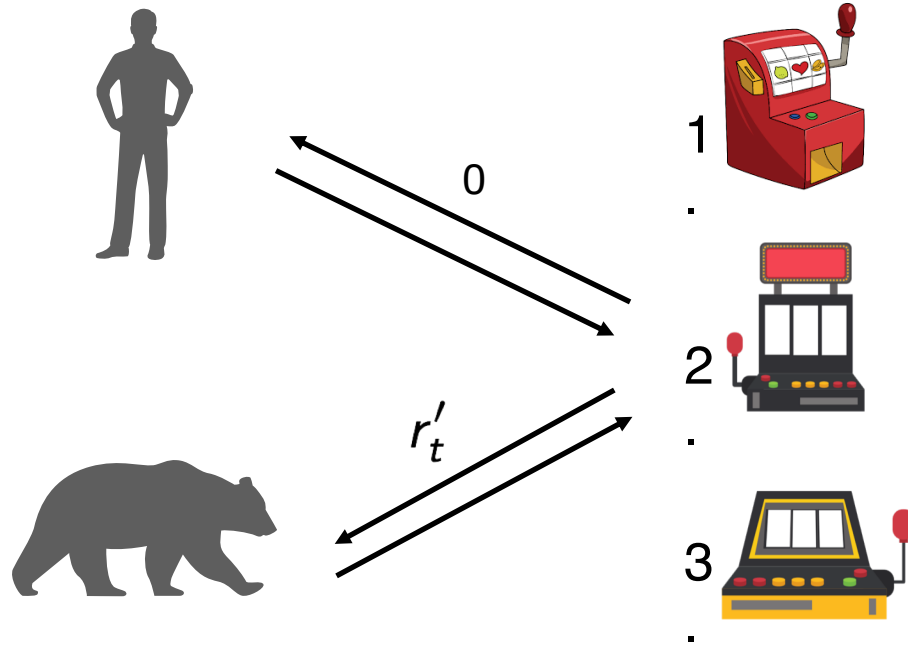
In this algorithm one side of the market iteratively makes proposals to the other side

Matching Markets Meet Learning

What if the participants in the market do not know their preferences a priori, but observe noisy utilities through repeated interactions?

Now the participants have an exploration/exploitation problem, in the context of other participants

Competing Agents



Bandit Markets

- We conceive of a **bandit market**: agents on one side, arms on the other side.

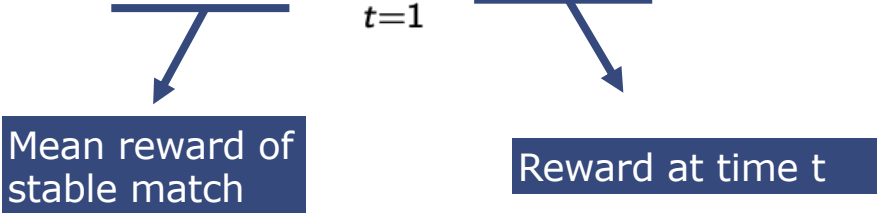
Agents get noisy rewards when they pull arms.

Arms have preferences over agents (these preferences can also express agents' skill levels)

When multiple agents pull the same arm only the most preferred agent gets a reward.

Bandit Markets

Then it is natural to define the regret of agent i up to time n as:

$$R_i(n) = \underbrace{n\mu_i(m(i))}_{\text{Mean reward of stable match}} - \sum_{t=1}^n \underbrace{\mathbb{E}X_{i,m_t}(t)}_{\text{Reward at time } t}$$


Minimizing this regret is natural. It says that agents should expect rewards as good as their stable match in hindsight.

If there are multiple stable matches, a bit more care is needed. See our paper.


Regret-Minimizing Algorithm

Gale-Shapley upper confidence bounds (GS-UCB):

- Agents rank arms according to upper confidence bounds for the mean rewards.
- Agents submit rankings to a matching platform.
- The platform uses these rankings to run the Gale-Shapley algorithm to match agents and arms.
- Agents receive rewards and update upper confidence bounds.
- Repeat.

Theorem

Theorem (informal): If there are N agents and K arms and GS-UCB is run, the regret of agent i satisfies

$$R_i(n) = \mathcal{O} \left(\frac{NK \log(n)}{\Delta^2} \right)$$


Reward gap of possibly other agents.

- In other words, if the bear decides to explore more, the human might have higher regret.
- See paper for refinements of this bound and further discussion of exploration-exploitation trade-offs in this setting.
- Finally, we note that GS-UCB is incentive compatible. No single agent has an incentive to deviate from the method.

Finding Nash Equilibria (and only Nash Equilibria) with gradients



Eric Mazumdar

Zero-Sum Games

Game over a shared function f that one player seeks to minimize and the other seeks to maximize.

$$\min_{x_1 \in \mathbb{R}^{d_1}} f(x_1, x_2) \quad \min_{x_2 \in \mathbb{R}^{d_2}} -f(x_1, x_2)$$

$$\omega(x) = \begin{bmatrix} \nabla_{x_1} f(x_1, x_2) \\ -\nabla_{x_2} f(x_1, x_2) \end{bmatrix} \quad J(x) = D\omega(x) = \begin{bmatrix} D_{11}^2 f(x_1, x_2) & D_{21}^2 f(x_1, x_2) \\ -D_{12}^2 f(x_1, x_2) & -D_{22}^2 f(x_1, x_2) \end{bmatrix}$$

Why zero-sum games?

- A lot of recent interest in finding the local Nash equilibria of zero-sum continuous games.
 - e.g. Adversarial Learning, training GANs, robust reinforcement learning

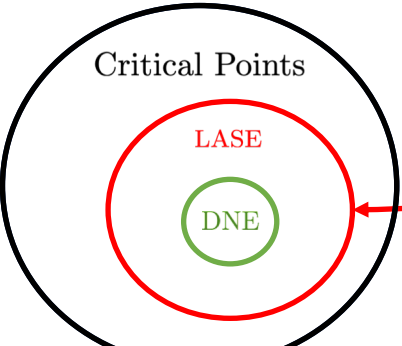
In each of these settings the goal is find local Nash equilibria of the game.

Issues with gradient-play in 2-player zero-sum games

Simultaneous Gradient descent (simGD) has two main issues:

- Convergence to limit cycles.
- Convergence to non-Nash fixed points.

Properties of Gradient Dynamics in Games			
Game Type	Avoid a subset of the DNE	Converge to Limit Cycles	Converge to Non-Nash LASE
Zero-Sum Game	—	✓	✓

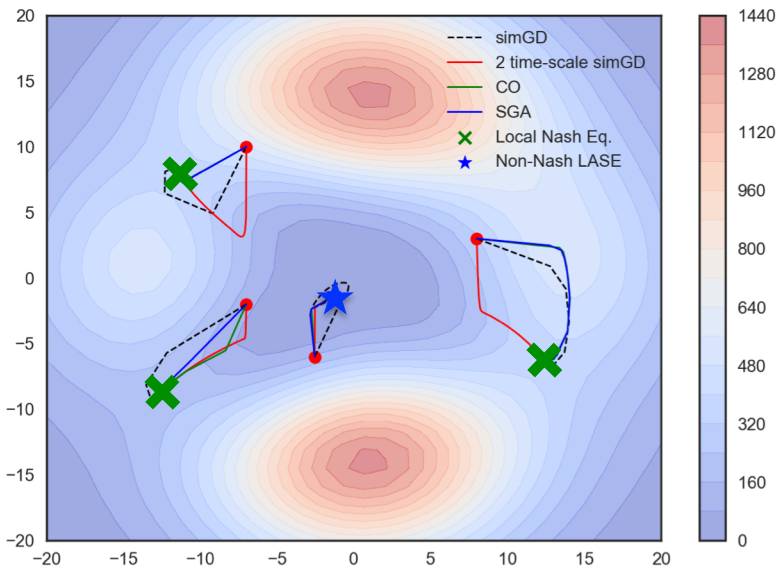


Not all locally asymptotically stable eq. are differential Nash eq.

These can be common!

- Every saddle point of the function that does not satisfy the specific conditions of a local Nash eq. is a candidate non-Nash LASE.

Recent algorithms do not solve these problems



Algorithm	Avoid some local Nash eq.	Converge to non-Nash eq.	Oscillate around eq.	Converge to limit cycles
Gradient play	—	✓	✓	✓
Symplectic Gradient Adjustment	—	✓	✓	✓
Consensus Optimization	—	✓	✓	✓

$$f(x_1, x_2) = e^{-0.01(x_1^2+x_2^2)} [(x_2 + 0.3x_1^2)^2 + (x_1 + 0.5x_2^2)^2]$$

Algorithm	Update Rule
Simultaneous Gradient Descent (simGD)	$x^+ = x - \gamma\omega(x)$
Two-timescale gradient descent	$x_1^+ = x_1 - \gamma_{1,t}\nabla_{x_1}f(x_1, x_2)$ $x_2^+ = x_2 + \gamma_{2,t}\nabla_{x_2}f(x_1, x_2)$
Consensus Optimization (CO)	$x^+ = x - \gamma(\omega(x) + J(x)^T\omega(x))$
Symplectic Gradient Adjustment (SGA)	$x^+ = x - \gamma(\omega(x) + (J^T(x) - J(x))\omega(x))$

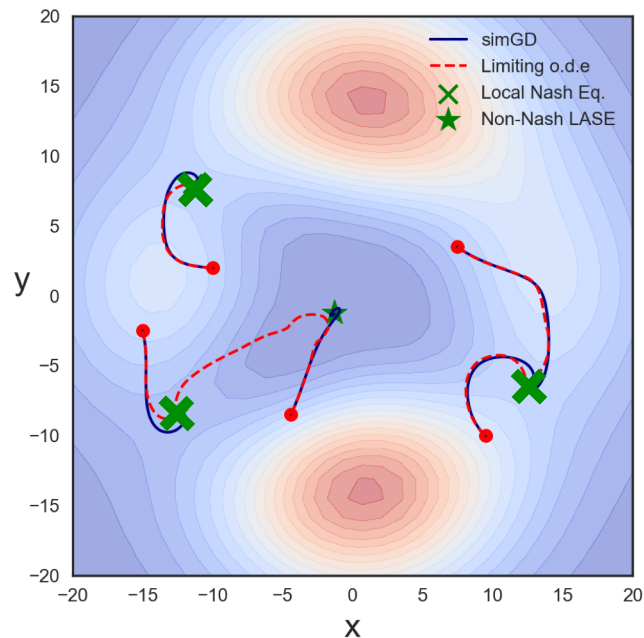
New algorithm for gradient-based learning in zero-sum games:

Local Symplectic Surgery:

$$x_{t+1} = x_t - \alpha_t (\omega(x_t) + J^T(x_t)v_t)$$

$$v_{t+1} = v_t - \beta_t (J^T(x_t)J(x_t)v_t - J^T(x_t)\omega(x_t))$$

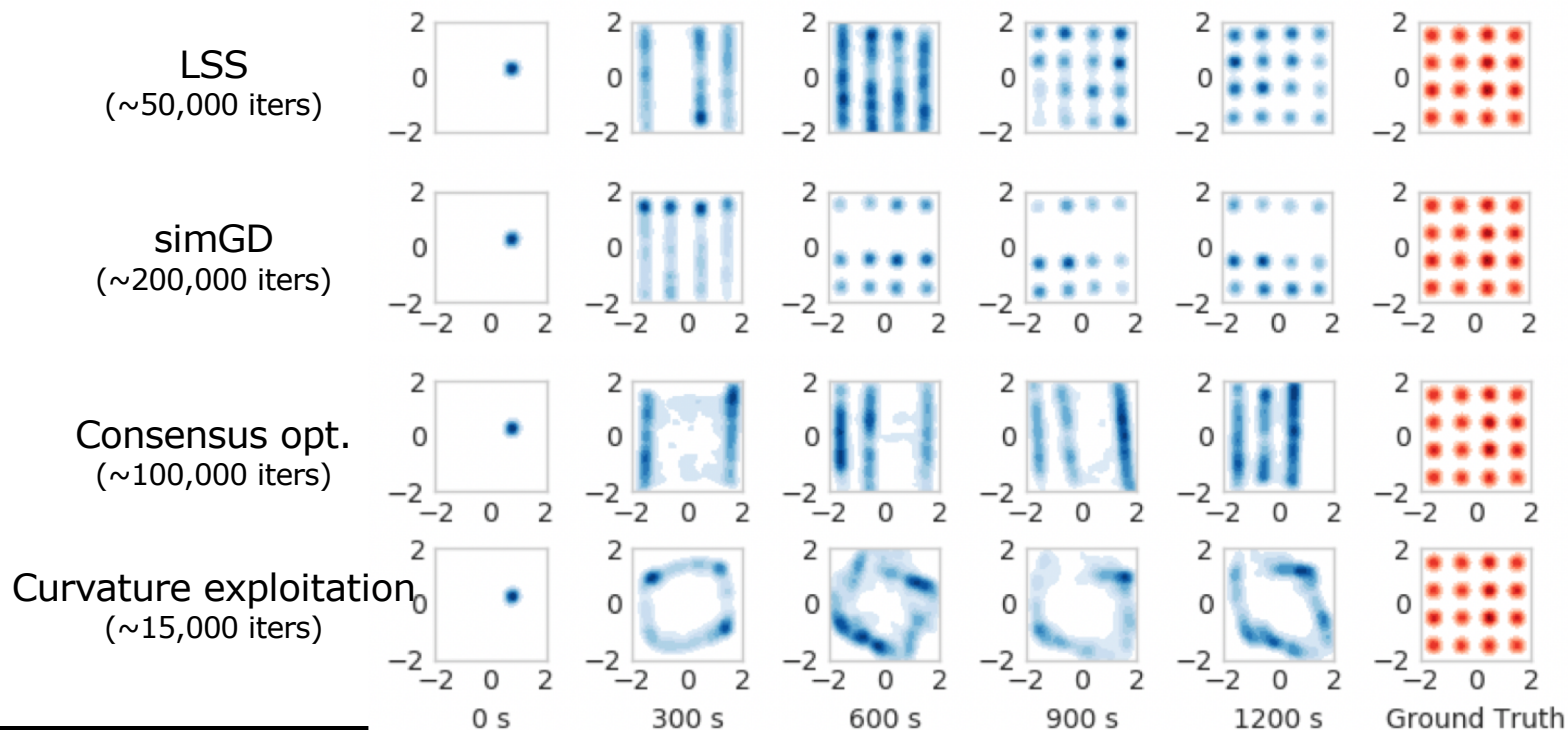
Local Symplectic Elimination	—	—	—	✓



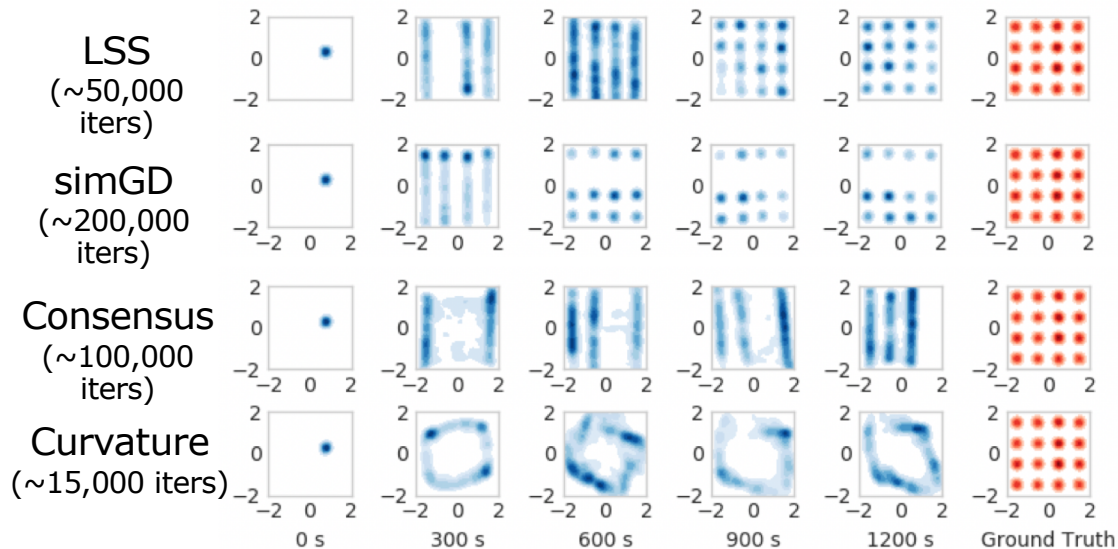
By cancelling out the symplectic part of the vector field around critical points, the only equilibria to which this method can converge are the local Nash equilibria of the game.

GAN experiments on a test for mode-collapse

Ground truth is a mixture of 16 Gaussians used to test for mode collapse, with covariance $0.005 I_2$
Generator and discriminator are Tanh neural networks with 4 hidden layers of 100 neurons each.



LSS seems to converge to better solutions



From an initialization where simGD quickly converges to an incorrect distribution, LSS recovers the ground truth.

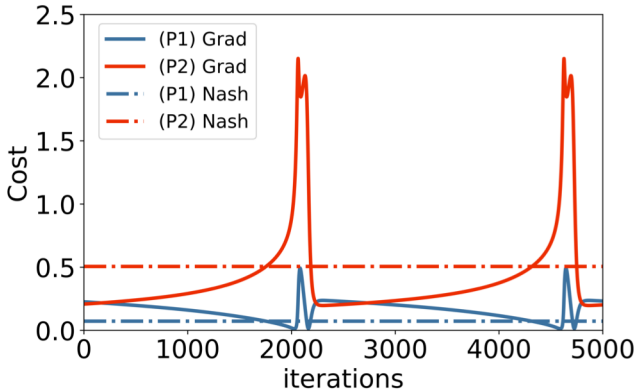
- Consensus optimization, without careful hyper-parameter tuning gives rise to new non-Nash equilibria, which results in worse performance.
- Local curvature exploitation is hard to implement in high dimensional settings **due to the need to find eigenvalue/eigenvector pairs of the diagonal blocks of J at each iteration**

This approach can be extended to general non-cooperative games:

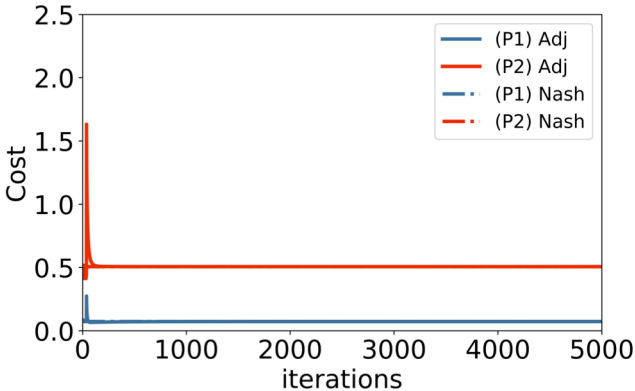
The algorithm can be extended to general-sum games where it converges in LQ games where gradient descent cycles.

$$x_{t+1} = x_t - \alpha \left(\omega(x_t) - e^{-\gamma \|\omega(x_t)\|^2} A(x_t) J^{-1}(x_t) \omega(x_t) \right)$$

Matrix of off-diagonal blocks of J



Simultaneous Gradient Play



Our Algorithm

What is Local Optimality in Nonconvex-Nonconcave Minimax Optimization?

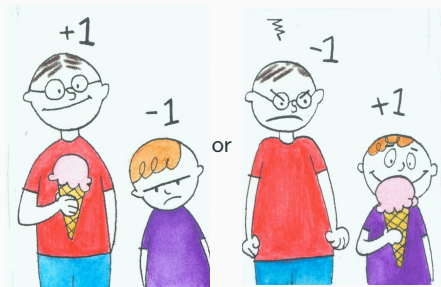
Chi Jin¹, Praneeth Netrapalli², Michael I. Jordan¹

¹University of California, Berkeley. ²Microsoft Research, India.



Minmax Optimization

Multi-agent decision making:



Framework:

$$\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^d} f(x, y)$$

where f is **nonconvex** in x and **nonconcave** in y .

Applications

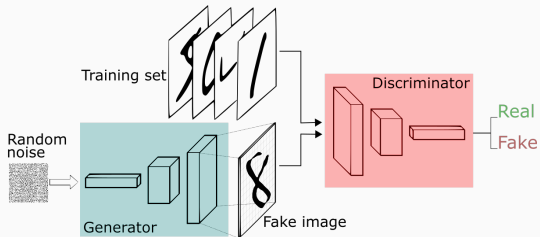


Adversarial Training

Applications



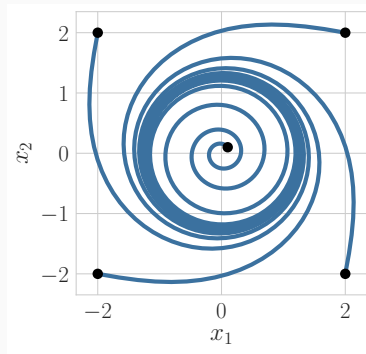
Adversarial Training



Generative Adversarial Network (GAN)

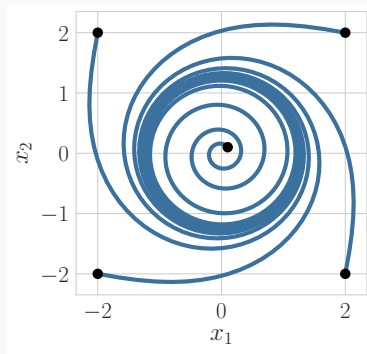
Gradient Descent Ascent (GDA):

$$\begin{cases} \mathbf{x}_{t+1} = \mathbf{x}_t - \eta_x \nabla f(\mathbf{x}_t). \\ \mathbf{y}_{t+1} = \mathbf{y}_t + \eta_y \nabla f(\mathbf{y}_t). \end{cases}$$



Gradient Descent Ascent (GDA):

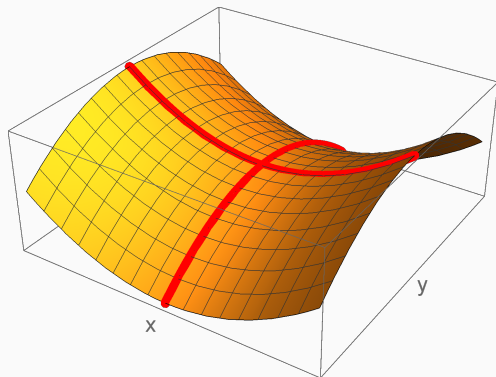
$$\begin{cases} \mathbf{x}_{t+1} = \mathbf{x}_t - \eta_x \nabla f(\mathbf{x}_t). \\ \mathbf{y}_{t+1} = \mathbf{y}_t + \eta_y \nabla f(\mathbf{y}_t). \end{cases}$$



Fundamental questions:

- What “optimal” points should we find?
- What are the points that GDA converges to? (if it converges)

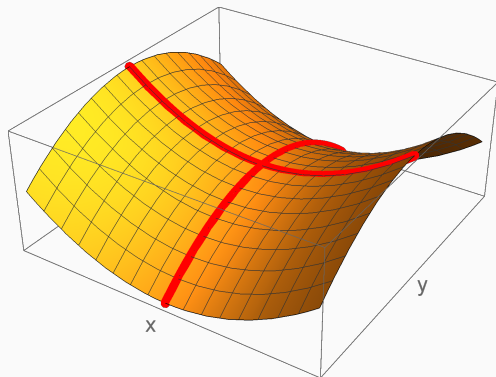
Nash Equilibrium



Point (x^*, y^*) is a **Nash equilibrium** if:

- y^* is a maximum of $f(x^*, \cdot)$; x^* is a minimum of $f(\cdot, y^*)$.

Nash Equilibrium



Point $(\mathbf{x}^*, \mathbf{y}^*)$ is a **Nash equilibrium** if:

- \mathbf{y}^* is a maximum of $f(\mathbf{x}^*, \cdot)$; \mathbf{x}^* is a minimum of $f(\cdot, \mathbf{y}^*)$.

Convex-concave case: GDA converges to a Nash equilibrium.

Local Nash Equilibrium

Nonconvex-nonconcave case: It's NP-hard to find Nash equilibrium.

Local Nash Equilibrium

Nonconvex-nonconcave case: It's NP-hard to find Nash equilibrium.

Find a **local Nash equilibrium**—point $(\mathbf{x}^*, \mathbf{y}^*)$ such that

- \mathbf{y}^* is a **local** maximum of $f(\mathbf{x}^*, \cdot)$; \mathbf{x}^* is a **local** minimum of $f(\cdot, \mathbf{y}^*)$.

Local Nash Equilibrium

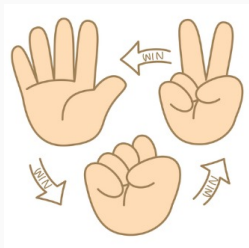
Nonconvex-nonconcave case: It's NP-hard to find Nash equilibrium.

Find a **local Nash equilibrium**—point $(\mathbf{x}^*, \mathbf{y}^*)$ such that

- \mathbf{y}^* is a **local** maximum of $f(\mathbf{x}^*, \cdot)$; \mathbf{x}^* is a **local** minimum of $f(\cdot, \mathbf{y}^*)$.

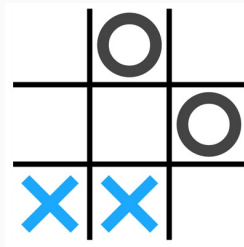
Are local Nash equilibria what we want in GAN/adversarial training?

Simultaneous vs. Sequential



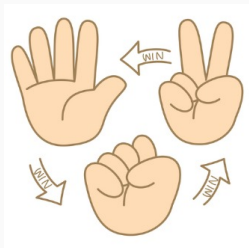
simultaneous

vs.



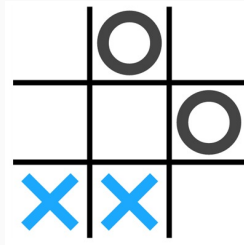
sequential

Simultaneous vs. Sequential



simultaneous

vs.

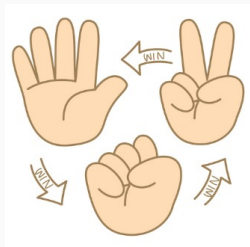


sequential

Nash equilibria come from **simultaneous** games—both act simultaneously.

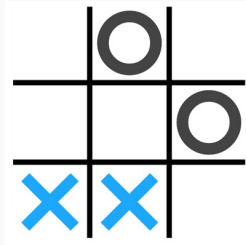
GAN/adversarial training are **sequential** games—one leader, one follower.

Simultaneous vs. Sequential



simultaneous

vs.



sequential

Nash equilibria come from **simultaneous** games—both act simultaneously.

GAN/adversarial training are **sequential** games—one leader, one follower.

For nonconvex-nonconcave f , which player acts first is crucial, since in general

$$\min_{\mathbf{x} \in \mathbb{R}^d} \max_{\mathbf{y} \in \mathbb{R}^d} f(\mathbf{x}, \mathbf{y}) \neq \max_{\mathbf{y} \in \mathbb{R}^d} \min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}, \mathbf{y})$$

Minimax Point (Stackelberg Equilibrium)

Point $(\mathbf{x}^*, \mathbf{y}^*)$ is a **Minimax Point** (or **Stackelberg Equilibrium**) if:

- \mathbf{y}^* is a maximum of $f(\mathbf{x}^*, \cdot)$;
- \mathbf{x}^* is a minimum of $\phi(\cdot)$ where $\phi(\mathbf{x}) = \max_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$.

[Leader always prepares for the best follower.]

Minimax points always exist even for nonconvex-nonconcave functions.

A New Notion of Local Optimality

[This Work] Point $(\mathbf{x}^*, \mathbf{y}^*)$ is a **local minimax point** if:

- \mathbf{y}^* is a **local** maximum of $f(\mathbf{x}^*, \cdot)$;
- $\exists \epsilon_0 > 0$, so that \mathbf{x}^* is a **local** minimum of $g_\epsilon(\cdot)$ for any $\epsilon \leq \epsilon_0$,
where $g_\epsilon(\mathbf{x}) = \max_{\mathbf{y}: \|\mathbf{y} - \mathbf{y}^*\| \leq \epsilon} f(\mathbf{x}, \mathbf{y})$.

A New Notion of Local Optimality

[This Work] Point $(\mathbf{x}^*, \mathbf{y}^*)$ is a **local minimax point** if:

- \mathbf{y}^* is a **local** maximum of $f(\mathbf{x}^*, \cdot)$;
 - $\exists \epsilon_0 > 0$, so that \mathbf{x}^* is a **local** minimum of $g_\epsilon(\cdot)$ for any $\epsilon \leq \epsilon_0$,
where $g_\epsilon(\mathbf{x}) = \max_{\mathbf{y}: \|\mathbf{y} - \mathbf{y}^*\| \leq \epsilon} f(\mathbf{x}, \mathbf{y})$.
-
- ▶ First proper definition of local optimality for **sequential games**.
 - ▶ Local minimax points enjoy various natural and good properties.

[DP18, MR18]: The stable limit points of GDA **need not to be local Nash!**

Limit Points of GDA

[DP18, MR18]: The stable limit points of GDA **need not to be local Nash!**

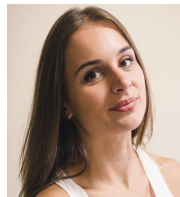
Theorem [This Work]

When learning rate $\eta_y/\eta_x \rightarrow \infty$, the stable limit points of GDA **are exactly local minimax points** up to some degenerate points.



Multiple Coupled Decisions Over Time

- Given a possibly infinite sequence of decisions over time can we guarantee **anytime control** of the **false-discovery rate (FDR)** in a fully asynchronous, online fashion?



Tijana
Zrnic



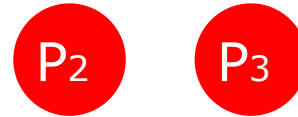
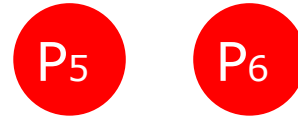
Aaditya
Ramdas

Foster-Stine '08
Aharoni-Rosset '14
Javanmard-Montanari '16

True nulls

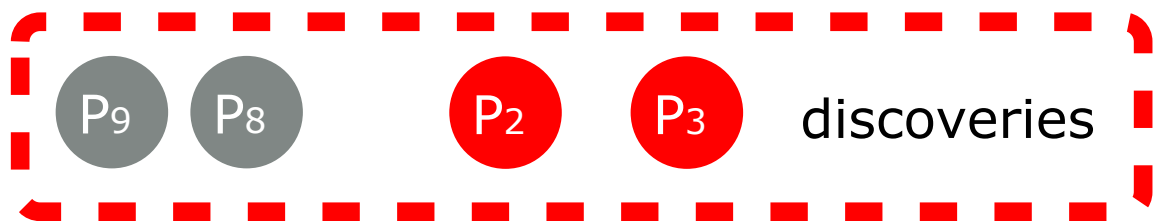
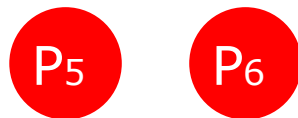


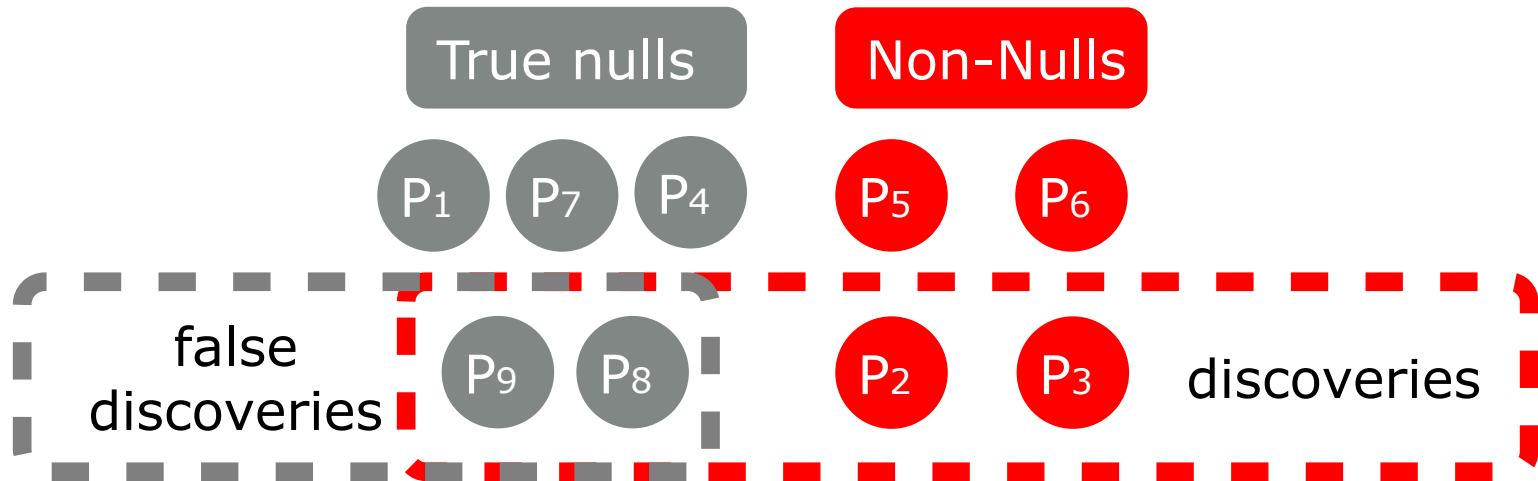
Non-Nulls



True nulls

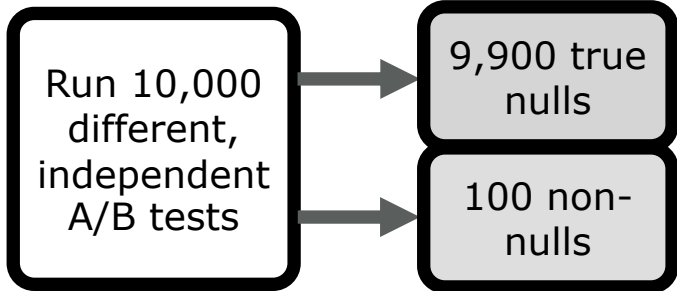
Non-Nulls



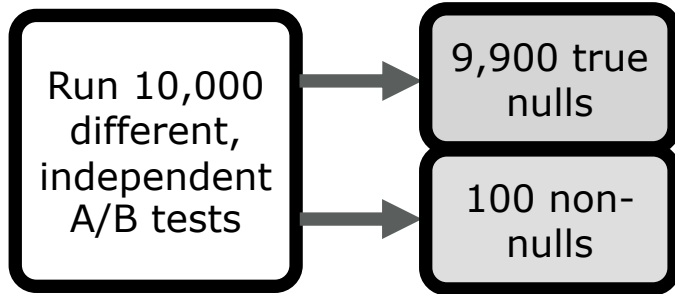


- False discovery proportion $FDP = \frac{\# \text{ false discoveries}}{\# \text{ discoveries}}$
- Want low false discovery rate $FDR = \mathbb{E}[FDP]$
- Want high Power = $\mathbb{E} \left[\frac{\# \text{ true discoveries}}{\# \text{ non-nulls}} \right]$

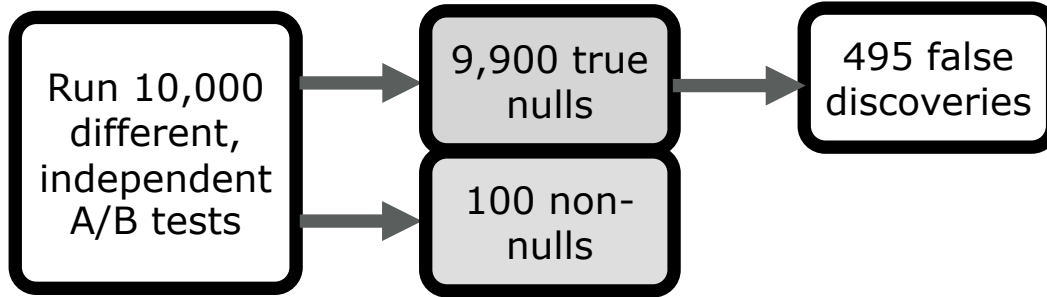
Run 10,000
different,
independent
A/B tests



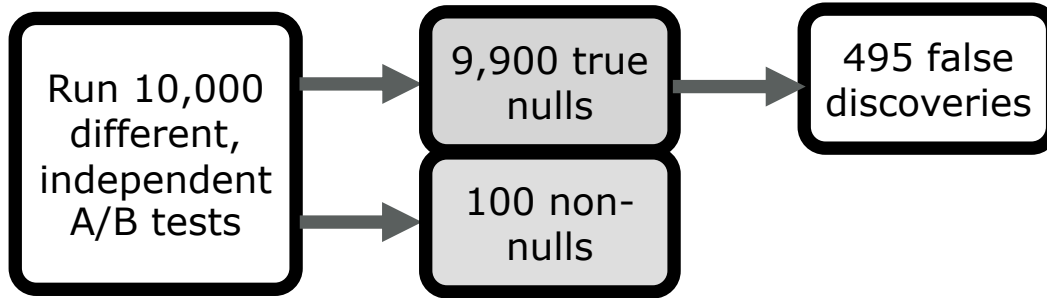
type-1 error rate (per test) = 0.05



type-1 error rate (per test) = 0.05

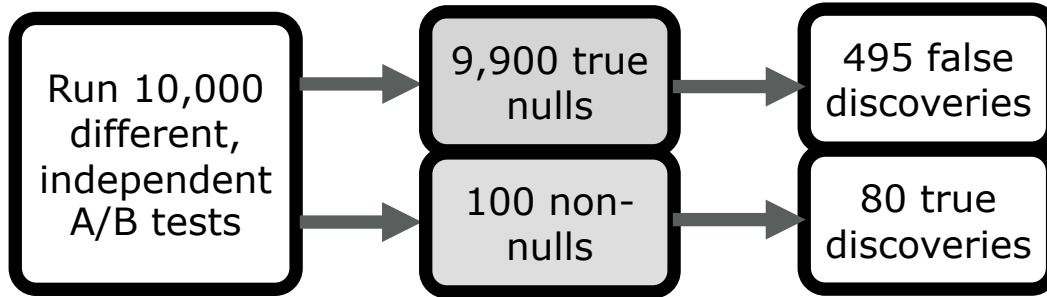


type-1 error rate (per test) = 0.05



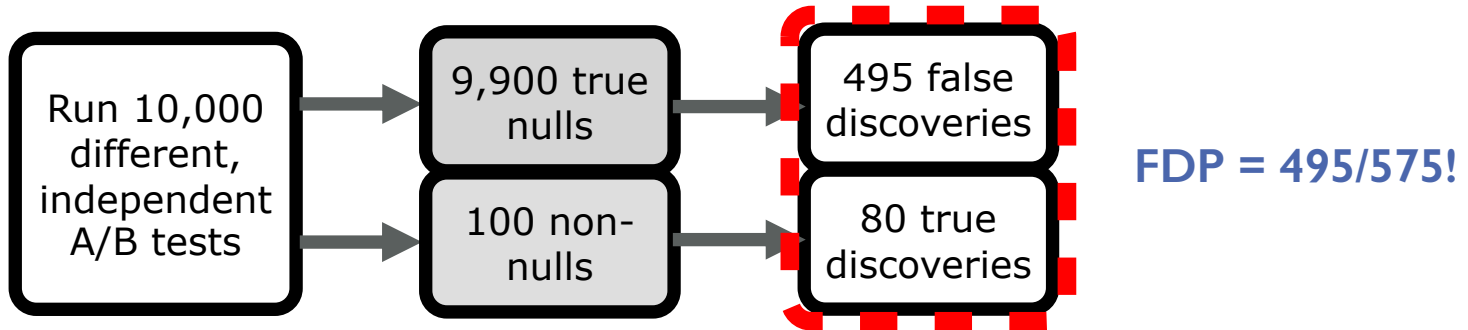
power (per test) = 0.80

type-1 error rate (per test) = 0.05



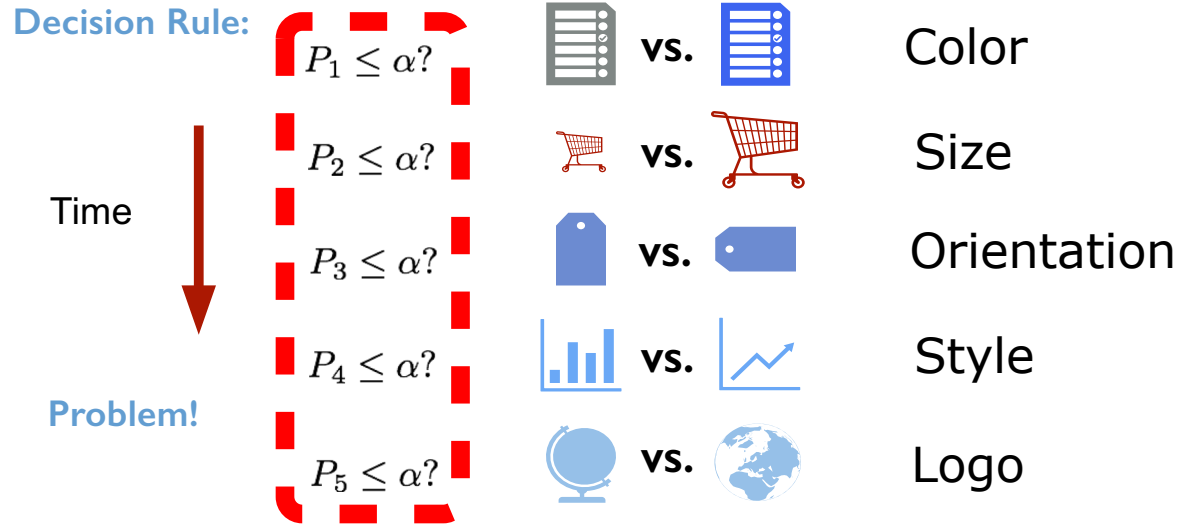
power (per test) = 0.80

type-1 error rate (per test) = 0.05

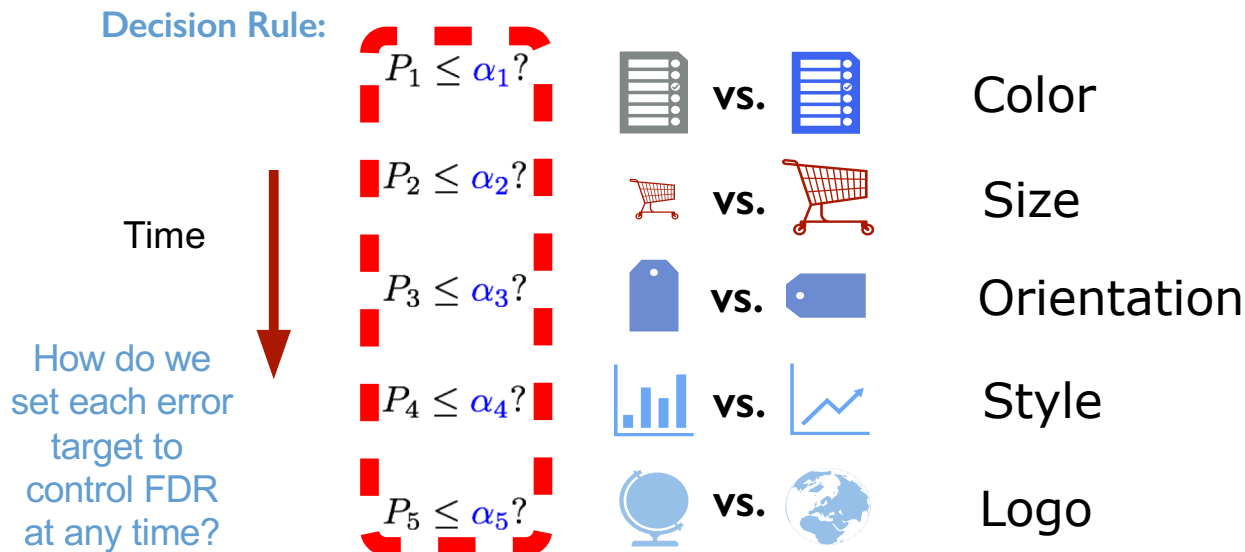


power (per test) = 0.80

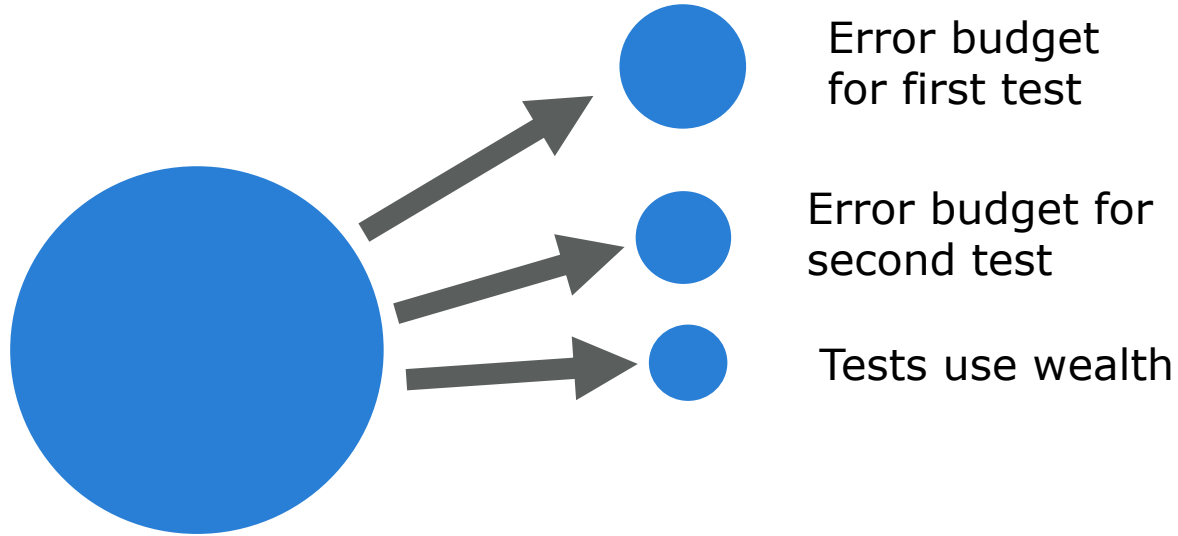
Many enterprises run thousands of different (independent) A/B tests over time



What we will do instead:

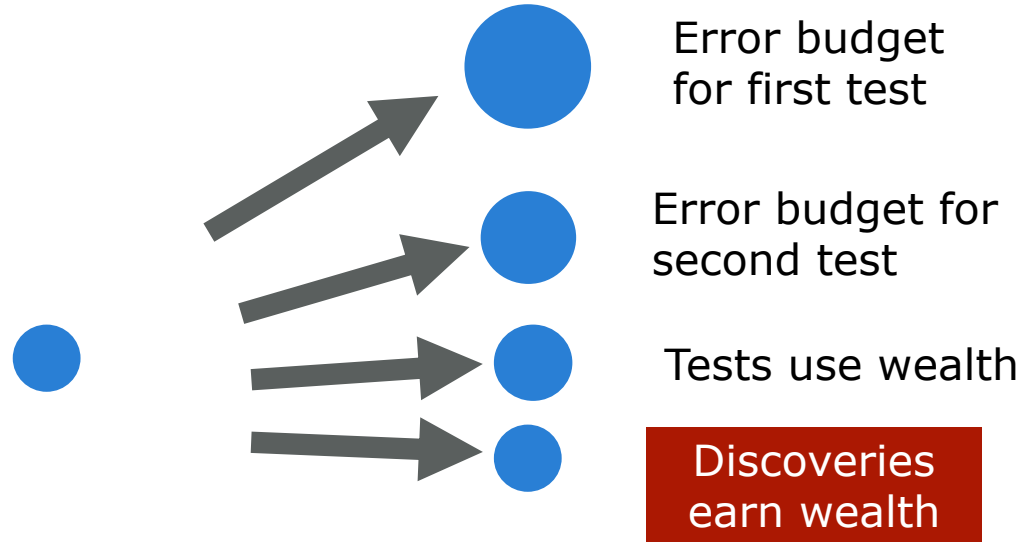


Online FDR control : high-level picture



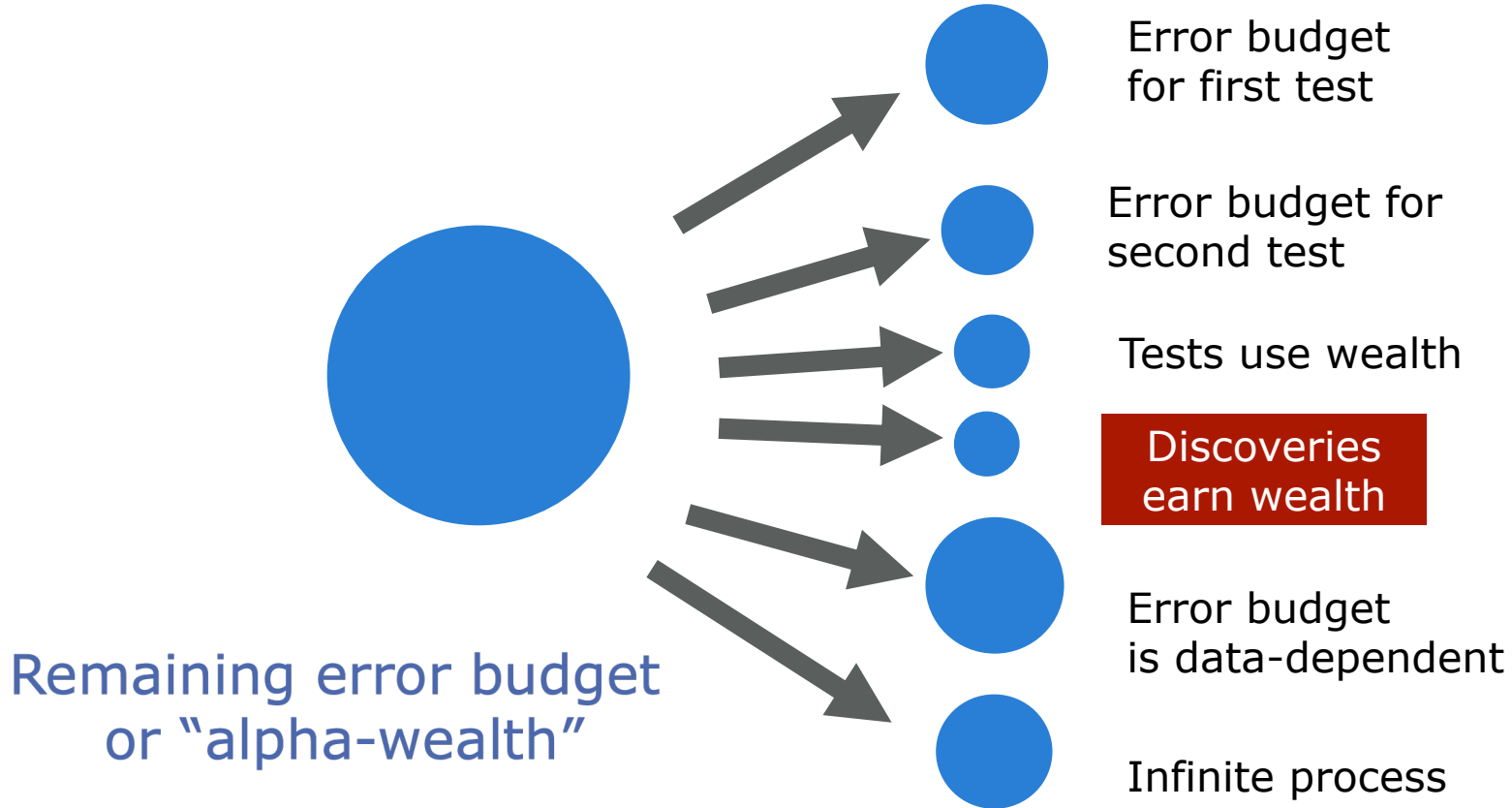
Remaining error budget
or "alpha-wealth"

Online FDR control : high-level picture



Remaining error budget
or "alpha-wealth"

Online FDR control : high-level picture



Executive Summary

- ML (AI) has come of age
- But it is far from being a solid engineering discipline that can yield robust, scalable solutions to modern data-analytic problems
- There are many hard problems involving uncertainty, inference, decision-making, robustness and scale that are far from being solved
 - not to mention economic, social and legal issues