Continuous and Discrete Dynamics For Online Learning and Convex Optimization

Walid Krichene

Thesis committee:
Alex Bayen Peter Bartlett Nikhil Srivastava

Electrical Engineering and Computer Sciences, UC Berkeley

August 18, 2016

Introduction: Continuous and discrete time dynamics

Continuous time dynamics $\leftrightarrow$ Discrete time dynamics

Example: gradient descent for convex optimization

$$\text{minimize}_{x\in\mathbb{R}^n} \quad f(x) \quad \text{(convex differentiable)}$$

	Continuous	Discrete
Dynamics	$\dot{X}(t) = -\nabla f(X(t))$	$x^{(k+1)} - x^{(k)} = -s\nabla f(x^{(k)})$
Lyapunov function	$\|X(t) - x^\star\|^2$	$\|x^{(k)} - x^\star\|^2$
Convergence rate	$f(X(t)) - f^\star = \mathcal{O}(1/t)$	$f(x^{(k)}) - f^\star = \mathcal{O}(1/k)$

Introduction

A dynamical systems approach to online learning and convex optimization

- Design dynamics for online learning and optimization in continuous time.
- Discretize to get algorithms.

Introduction

A dynamical systems approach to online learning and convex optimization

- Design dynamics for online learning and optimization in continuous time.
- Discretize to get algorithms.

Why continuous time?

1. Simple analysis.
2. Provides insight into the discrete process (can lead to new heuristics).
3. Streamlines design of new methods.

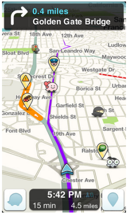
Outline

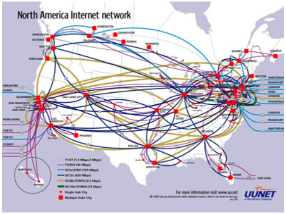
Outline

Online learning

Sequential decision problems:

- Ubiquitous in Cyber-Physical Systems (CPS)
- Routing (transportation, communication)
- Power networks
- Real-time bidding in online advertising





Distributed learning in games

Online Learning Model (decision maker k, action set $\mathcal{A}_k$)
1: **for** $t \in \mathbb{N}$ **do**
2: Play action $a \sim x_k^{(t)} \in \Delta^{\mathcal{A}_k}$
3: Discover loss vector $\ell_k^{(t)}$
4: Update $x_k^{(t+1)} = u_k\left(x_k^{(t)}, \ell_k^{(t)}\right)$
5: **end for**

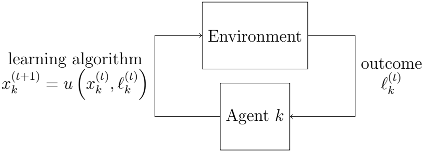


Figure: Sequential decision problem.

Distributed learning in games

Online Learning Model (decision maker k, action set $\mathcal{A}_k$)

for $t \in \mathbb{N}$ **do**
  Play action $a \sim x_k^{(t)} \in \Delta^{\mathcal{A}_k}$
  Discover loss vector $\ell_k^{(t)}$
  Update $x_k^{(t+1)} = u_k\left(x_k^{(t)}, \ell_k^{(t)}\right)$
end for

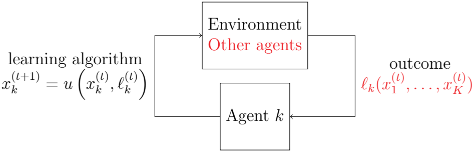


Figure: Coupled sequential decision problems.

Distributed learning in games

Online Learning Model (decision maker k, action set $\mathcal{A}_k$)

for $t \in \mathbb{N}$ **do**
  Play action $a \sim x_k^{(t)} \in \Delta^{\mathcal{A}_k}$
  Discover loss vector $\ell_k^{(t)}$
  Update $x_k^{(t+1)} = u_k\left(x_k^{(t)}, \ell_k^{(t)}\right)$
end for

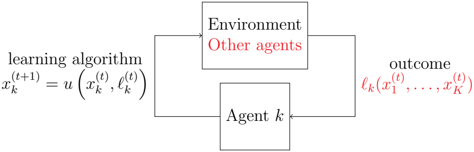


Figure: Coupled sequential decision problems.

- Game theory point of view:
  - Equilibria: a good description of system efficiency at steady-sate.
- Systems rarely operate at equilibrium.
- Online learning point of view:
  1. A prescriptive model: How do we drive system to eq.
  2. A descriptive model: How would players behave in the game.

Distributed learning in games

Online Learning Model (decision maker k, action set $\mathcal{A}_k$)

for $t \in \mathbb{N}$ **do**
    Play action $a \sim x_k^{(t)} \in \Delta^{\mathcal{A}_k}$
    Discover loss vector $\ell_k^{(t)}$
    Update $x_k^{(t+1)} = u_k \left(x_k^{(t)}, \ell_k^{(t)} \right)$
end for

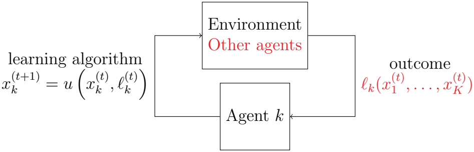


Figure: Coupled sequential decision problems.

- Game theory point of view:
  - Equilibria: a good description of system efficiency at steady-sate.
- Systems rarely operate at equilibrium.
- Online learning point of view:
  1. A prescriptive model: How do we drive system to eq.
  2. A descriptive model: How would players behave in the game.

Goals

- Define classes of algorithms for which we can prove convergence.
- Robustness to stochastic perturbations.
- Heterogeneous learning (different agents use different algorithms).
- Convergence rates.

A brief review

Discrete time:

- Hannan consistency: [7]
- Hedge algorithm for two-player games: [6]
- Regret based algorithms: [8]
- Online learning in games: [5]

Continuous time:

- Evolution in populations: [22]
- Replicator dynamics in evolutionary game theory [24]
- No-regret dynamics for two player games [8]

---

[7]J. Hannan. Approximation to bayes risk in repeated plays.
Contributions to the Theory of Games, 3:97–139, 1957

[6]Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights.
Games and Economic Behavior, 29(1):79–103, 1999

[8]S. Hart and A. Mas-Colell. A general class of adaptive strategies.
Journal of Economic Theory, 98(1):26 – 54, 2001

[5]N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games.*
Cambridge University Press, 2006

[22]W. H. Sandholm. *Population games and evolutionary dynamics.*
Economic learning and social evolution. Cambridge, Mass. MIT Press, 2010

[24]J. W. Weibull. *Evolutionary game theory.*
MIT press, 1997

[8]S. Hart and A. Mas-Colell. A general class of adaptive strategies.
Journal of Economic Theory, 98(1):26 – 54, 2001

Nonatomic, convex potential games

Notation:

$$x = (x_1, \dots, x_K) \in \Delta^{\mathcal{A}_1} \times \cdots \times \Delta^{\mathcal{A}_K} \qquad \ell(x) = (\ell_1(x), \dots, \ell_K(x))$$

Nonatomic, convex potential game

There exists a convex differentiable function f such that:

$$\ell(x) = \nabla f(x)$$

Nonatomic, convex potential games

Notation:

$$x = (x_1, \dots, x_K) \in \Delta^{\mathcal{A}_1} \times \dots \times \Delta^{\mathcal{A}_K} \qquad \ell(x) = (\ell_1(x), \dots, \ell_K(x))$$

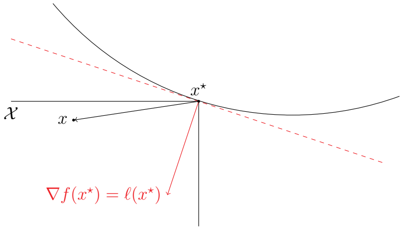
Nonatomic, convex potential game

There exists a convex differentiable function f such that:

$$\ell(x) = \nabla f(x)$$

Nash equilibria $\mathcal{X}^\star$

$x^\star$ is a Nash equilibrium	$\Leftrightarrow$	$x^\star$ is a minimizer of f
Nash condition		first order optimality
$\forall x,\ \langle \ell(x^\star), x\rangle \geq \langle \ell(x^\star), x^\star\rangle$		$\forall x,\ \langle \nabla f(x^\star), x - x^\star\rangle \geq 0$



Example: routing game

Online Learning Model. Action set $\mathcal{A}_k$: paths from o_k to d_k.

for $t \in \mathbb{N}$ **do**
  Play $a \sim x_k^{(t)} \in \Delta^{\mathcal{A}_k}$
  Discover $\ell_k^{(t)}$
  Update $x_k^{(t+1)} = u_k\left(x_k^{(t)}, \ell_k^{(t)}\right)$
end for

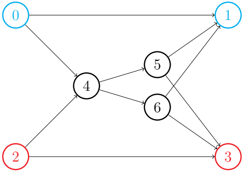


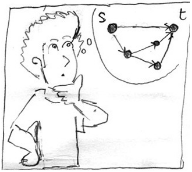
Figure: Routing game

Example: routing game

Online Learning Model. Action set $\mathcal{A}_k$: paths from o_k to d_k.

for $t \in \mathbb{N}$ **do**
  Play $a \sim x_k^{(t)} \in \Delta^{\mathcal{A}_k}$
  Discover $\ell_k^{(t)}$
  Update $x_k^{(t+1)} = u_k\left(x_k^{(t)}, \ell_k^{(t)}\right)$
end for

$x_{\mathbf{1}}^{(t)} \in \Delta^{\mathcal{A}_{\mathbf{1}}}$



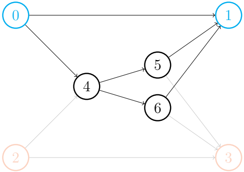


Figure: Routing game

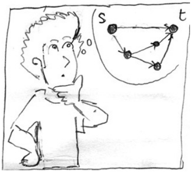
Example: routing game

Online Learning Model. Action set $\mathcal{A}_k$: paths from o_k to d_k.

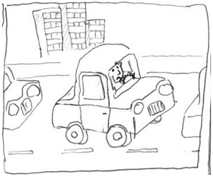
---

for $t \in \mathbb{N}$ **do**
  Play $a \sim x_k^{(t)} \in \Delta^{\mathcal{A}_k}$
  Discover $\ell_k^{(t)}$
  Update $x_k^{(t+1)} = u_k\left(x_k^{(t)}, \ell_k^{(t)}\right)$
end for

$x_1^{(t)} \in \Delta^{\mathcal{A}_1}$



Sample $a \sim x_1^{(t)}$



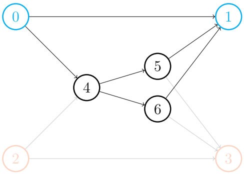


Figure: Routing game

Example: routing game

Online Learning Model. Action set $\mathcal{A}_k$: paths from o_k to d_k.

for $t \in \mathbb{N}$ **do**
    Play $a \sim x_k^{(t)} \in \Delta^{\mathcal{A}_k}$
    Discover $\ell_k^{(t)}$
    Update $x_k^{(t+1)} = u_k\left(x_k^{(t)}, \ell_k^{(t)}\right)$
end for

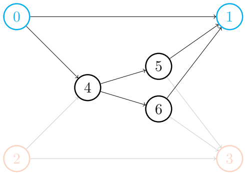
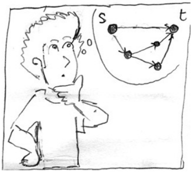
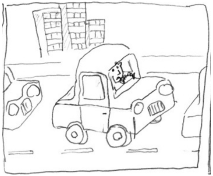


Figure: Routing game

$x_{\mathbf{1}}^{(t)} \in \Delta^{\mathcal{A}_{\mathbf{1}}}$



Sample $a \sim x_{\mathbf{1}}^{(t)}$



Discover $\ell_{\mathbf{1}}^{(t)}$

Example: routing game

Online Learning Model. Action set $\mathcal{A}_k$: paths from o_k to d_k.

for $t \in \mathbb{N}$ **do**
    Play $a \sim x_k^{(t)} \in \Delta^{\mathcal{A}_k}$
    Discover $\ell_k^{(t)}$
    Update $x_k^{(t+1)} = u_k\left(x_k^{(t)}, \ell_k^{(t)}\right)$
end for

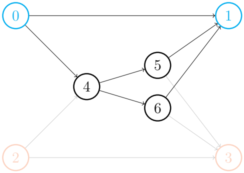


Figure: Routing game

$x_{\mathbf{1}}^{(t)} \in \Delta^{\mathcal{A}_{\mathbf{1}}}$ | Sample $a \sim x_{\mathbf{1}}^{(t)}$ | Discover $\ell_{\mathbf{1}}^{(t)}$ | Update $x_{\mathbf{1}}^{(t+\mathbf{1})}$

The Hedge algorithm

Hedge algorithm

for $t \in \mathbb{N}$ **do**
  Play $a \sim x_k^{(t)}$
  Discover $\ell_k^{(t)}$
  Update $x_{k,a}^{(t+1)} \propto x_{k,a}^{(t)} e^{-\eta_t \ell_{k,a}^{(t)}}$
end for

[5]N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games.*
Cambridge University Press, 2006

[1]S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications.
Theory of Computing, 8(1):121–164, 2012

[9]J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors.
Information and Computation, 132(1):1 – 63, 1997

[2]A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization.
Oper. Res. Lett., 31(3):167–175, May 2003

[4]L. E. Blume. The statistical mechanics of strategic interaction.
Games and Economic Behavior, 5(3):387 – 424, 1993

[15]J. R. Marden and J. S. Shamma. Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation.

The Hedge algorithm

Hedge algorithm

1: **for** $t \in \mathbb{N}$ **do**
2: Play $a \sim x_k^{(t)}$
3: Discover $\ell_k^{(t)}$
4: Update $x_{k,a}^{(t+1)} \propto x_{k,a}^{(t)} e^{-\eta_t \ell_{k,a}^{(t)}}$
5: **end for**

- Exponentially weighted average forecaster [5].
- Multiplicative weights update [1].
- Exponentiated gradient descent [9].
- Entropic descent [2].
- Log-linear learning [4], [15].

[5]N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games.*
Cambridge University Press, 2006

[1]S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications.
Theory of Computing, 8(1):121–164, 2012

[9]J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors.
Information and Computation, 132(1):1 – 63, 1997

[2]A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization.
Oper. Res. Lett., 31(3):167–175, May 2003

[4]L. E. Blume. The statistical mechanics of strategic interaction.
Games and Economic Behavior, 5(3):387 – 424, 1993

[15]J. R. Marden and J. S. Shamma. Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation.

Replicator ODE

Idea

- Take continuous-time limit of Hedge.
- Study convergence of ODE.
- View learning dynamics as a discretization of an ODE.
- Relate convergence of discrete algorithm to convergence of ODE.

[24] J. W. Weibull. *Evolutionary game theory.* MIT press, 1997

Replicator ODE

Idea

- Take continuous-time limit of Hedge.
- Study convergence of ODE.
- View learning dynamics as a discretization of an ODE.
- Relate convergence of discrete algorithm to convergence of ODE.

In Hedge $x_a^{(t+1)} \propto x_a^{(t)} e^{-\eta_t \ell_a^{(t)}}$, take $\eta_t \to 0$. Get replicator equation [24].

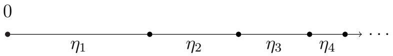


Figure: Underlying continuous time

Dynamics	$\dot{X}_a = X_a \left(\langle \ell(X), X \rangle - \ell_a(X) \right)$
Lyapunov function	$D_{\mathbf{KL}}(x^\star, X(t))$

[24] J. W. Weibull. *Evolutionary game theory.* MIT press, 1997

Replicator ODE

Idea

- Take continuous-time limit of Hedge.
- Study convergence of ODE.
- View learning dynamics as a discretization of an ODE.
- Relate convergence of discrete algorithm to convergence of ODE.

In Hedge $x_a^{(t+1)} \propto x_a^{(t)} e^{-\eta_t \ell_a^{(t)}}$, take $\eta_t \to 0$. Get replicator equation [24].

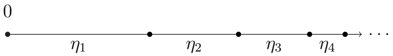


Figure: Underlying continuous time

Dynamics	$\dot{X}_a = X_a \left(\langle \ell(X), X \rangle - \ell_a(X) \right)$
Lyapunov function	$D_{\mathbf{KL}}(x^\star, X(t))$
	$t(f(X(t)) - f^\star) + D_{\mathbf{KL}}(x^\star, X(t))$
Convergence rate	$f(X(t)) - f^\star = \mathcal{O}(1/t)$

[24] J. W. Weibull. *Evolutionary game theory.* MIT press, 1997

AREP dynamics: Approximate REPlicator

$$\dot{X}_a = X_a \left(\langle \ell(X), X \rangle - \ell_a(X) \right)$$

Discrete approximation of the replicator ODE

$$\frac{x_a^{(t+1)} - x_a^{(t)}}{\eta_t} = x_a^{(t)} \left(\left\langle \ell(x^{(t)}), x^{(t)} \right\rangle - \ell_a(x^{(t)}) \right) + U_a^{(t+1)}$$

[3] M. Benaïm. Dynamics of stochastic approximation algorithms.
In *Séminaire de probabilités XXXIII*, pages 1–68. Springer, 1999

AREP dynamics: Approximate REPlicator

$$\dot{X}_a = X_a \left(\langle \ell(X), X \rangle - \ell_a(X) \right)$$

Discrete approximation of the replicator ODE

$$\frac{x_a^{(t+1)} - x_a^{(t)}}{\eta_t} = x_a^{(t)} \left(\left\langle \ell(x^{(t)}), x^{(t)} \right\rangle - \ell_a(x^{(t)}) \right) + U_a^{(t+1)}$$

- η_t discretization time steps.

[3] M. Benaïm. Dynamics of stochastic approximation algorithms.
In *Séminaire de probabilités XXXIII*, pages 1–68. Springer, 1999

AREP dynamics: Approximate REPlicator

$$\dot{X}_a = X_a \left(\langle \ell(X), X \rangle - \ell_a(X) \right)$$

Discrete approximation of the replicator ODE

$$\frac{x_a^{(t+1)} - x_a^{(t)}}{\eta_t} = x_a^{(t)} \left(\left\langle \ell(x^{(t)}), x^{(t)} \right\rangle - \ell_a(x^{(t)}) \right) + U_a^{(t+1)}$$

- η_t discretization time steps.
- $(U^{(t)})_{t \geq 1}$ perturbations that satisfy for all $T > 0$,
  $\lim_{\tau_1 \to \infty} \max_{\tau_2 : \sum_{t=\tau_1}^{\tau_2} \eta_t < T} \left\| \sum_{t=\tau_1}^{\tau_2} \eta_t U^{(t+1)} \right\| = 0$

(a sufficient condition is that $\exists q \geq 2$: $\sup_\tau \mathbb{E} \| U^{(\tau)} \|^q < \infty$ and $\sum_\tau \eta_\tau^{1+\frac{q}{2}} < \infty$)

[3] M. Benaïm. Dynamics of stochastic approximation algorithms.
In *Séminaire de probabilités XXXIII*, pages 1–68. Springer, 1999

AREP dynamics: Approximate REPlicator

$$\dot{X}_a = X_a \left(\langle \ell(X), X \rangle - \ell_a(X) \right)$$

Discrete approximation of the replicator ODE

$$\frac{x_a^{(t+1)} - x_a^{(t)}}{\eta_t} = x_a^{(t)} \left(\left\langle \ell(x^{(t)}), x^{(t)} \right\rangle - \ell_a(x^{(t)}) \right) + U_a^{(t+1)}$$

- η_t discretization time steps.
- $(U^{(t)})_{t \geq 1}$ perturbations that satisfy for all $T > 0$, $\lim_{\tau_1 \to \infty} \max_{\tau_2 : \sum_{t=\tau_1}^{\tau_2} \eta_t < T} \left\| \sum_{t=\tau_1}^{\tau_2} \eta_t U^{(t+1)} \right\| = 0$

(a sufficient condition is that $\exists q \geq 2$: $\sup_\tau \mathbb{E} \| U^{(\tau)} \|^q < \infty$ and $\sum_\tau \eta_\tau^{1 + \frac{q}{2}} < \infty$)

Examples

Hedge, REP, (stochastic and deterministic).

---

[3] M. Benaïm. Dynamics of stochastic approximation algorithms. In *Séminaire de probabilités XXXIII*, pages 1–68. Springer, 1999

Asymptotic Pseudo Trajectory

Sufficient conditions for $x^{(t)}$ to be an asymptotic pseudo trajectory of the ODE flow.

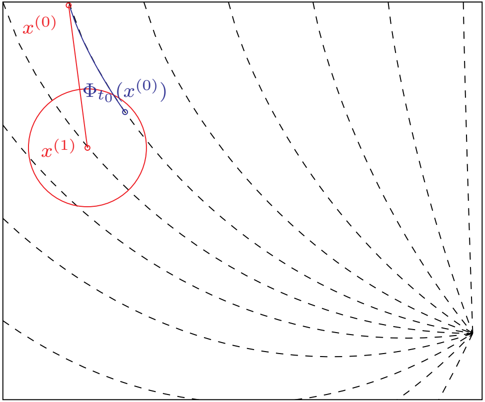


Figure: Asymptotic Pseudo Trajectory

Asymptotic Pseudo Trajectory

Sufficient conditions for $x^{(t)}$ to be an asymptotic pseudo trajectory of the ODE flow.

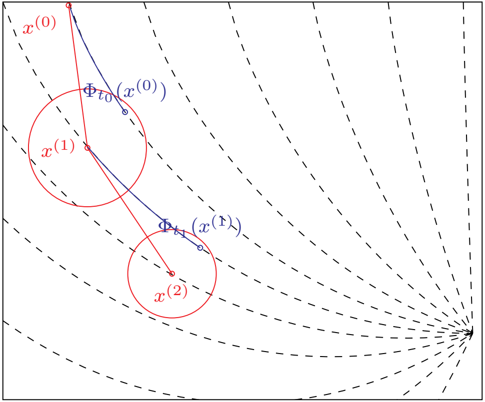


Figure: Asymptotic Pseudo Trajectory

Asymptotic Pseudo Trajectory

Sufficient conditions for $x^{(t)}$ to be an asymptotic pseudo trajectory of the ODE flow.

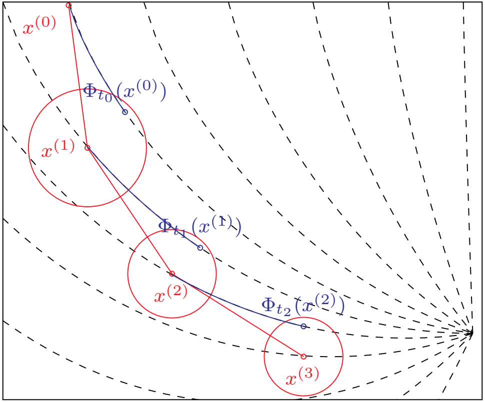


Figure: Asymptotic Pseudo Trajectory

Asymptotic Pseudo Trajectory

Sufficient conditions for $x^{(t)}$ to be an asymptotic pseudo trajectory of the ODE flow.

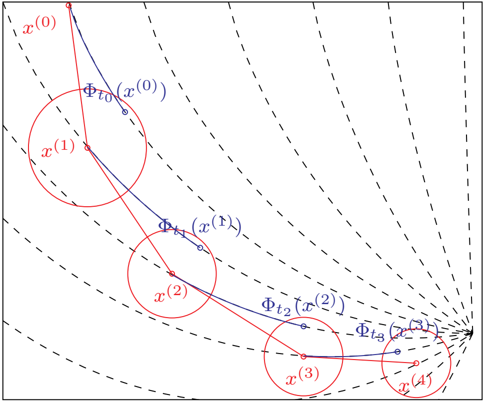


Figure: Asymptotic Pseudo Trajectory

Asymptotic Pseudo Trajectory

Figure: Discrete (Hedge) and continuous (Replicator) trajectories

Convergence to Nash equilibria

Theorem [12]

In convex potential games, under AREP updates, if $\eta_t \downarrow 0$ and $\sum \eta_t = \infty$, then

$$x^{(t)} \to \mathcal{X}^\star \text{ a.s.}$$

[10]S. Krichene, W. Krichene, R. Dong, and A. Bayen. Convergence of heterogeneous distributed learning in stochastic routing games.
In *53rd Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2015

[12] W. Krichene, B. Drighès, and A. Bayen. Learning nash equilibria in congestion games.
SIAM Journal on Control and Optimization (SICON), 2015

Convergence to Nash equilibria

Theorem [12]

In convex potential games, under AREP updates, if $\eta_t \downarrow 0$ and $\sum \eta_t = \infty$, then

$$x^{(t)} \to \mathcal{X}^\star \text{ a.s.}$$

- Affine interpolation of $x^{(t)}$ is an asymptotic pseudo trajectory of ODE.
- Use f as a Lyapunov function.
- However, No convergence rates.
- In order to derive convergence rates, can study specific dynamics. E.g. mirror descent dynamics [10].

---

[10]S. Krichene, W. Krichene, R. Dong, and A. Bayen. Convergence of heterogeneous distributed learning in stochastic routing games.
In *53rd Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2015

[12] W. Krichene, B. Drighès, and A. Bayen. Learning nash equilibria in congestion games.
SIAM Journal on Control and Optimization (SICON), 2015

Numerical example

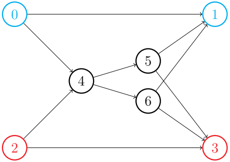
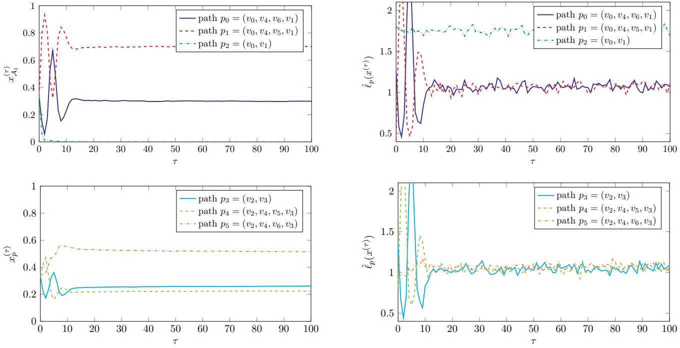


Figure: Example with strongly convex potential.

- Centered Gaussian noise on edges.
- Population 1: Hedge with (η_t^1)
- Population 2: Hedge with (η_t^2)

Mass distributions $x_k^{(t)}$ | Path losses $\ell_k(x^{(t)})$

Population 1 | Population 2



Numerical example

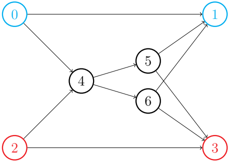


Figure: Example with strongly convex potential.

- Centered Gaussian noise on edges.
- Population 1: Hedge with (η^1_t)
- Population 2: Hedge with (η^2_t)

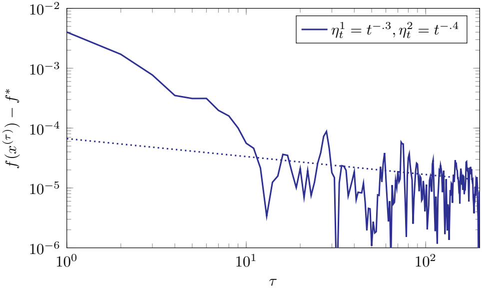
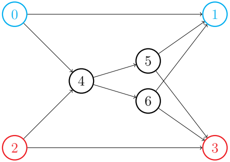


Figure: Potential values.

For $\eta^k_t = \frac{\theta_k}{t^{\alpha_k}}$, $\alpha_k \in (0,1)$, $\mathbb{E}\left[f(x^{(t)})\right] - f^\star = O\left(\sum_k \frac{\log t}{t^{\min(\alpha_k, 1-\alpha_k)}}\right)$

Numerical example



- Centered Gaussian noise on edges.
- Population 1: Hedge with (η_t^1)
- Population 2: Hedge with (η_t^2)

Figure: Example with strongly convex potential.

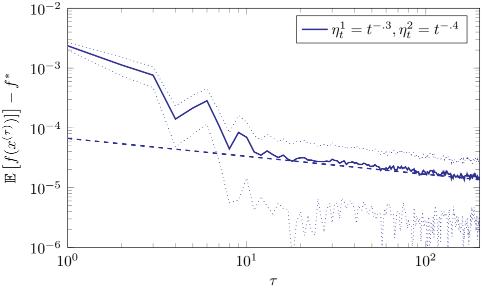


Figure: Potential values.

For $\eta_t^k = \frac{\theta_k}{t^{\alpha_k}}$, $\alpha_k \in (0, 1)$, $\mathbb{E}\left[f(x^{(t)})\right] - f^\star = O\left(\sum_k \frac{\log t}{t^{\min(\alpha_k, 1-\alpha_k)}}\right)$

Outline

First-order optimization

Constrained convex optimization

$$\begin{aligned} &\text{minimize} && f(x) \text{ (convex, } \nabla f \text{ Lipschitz)} \\ &\text{subject to} && x \in \mathcal{X} \text{ (closed convex)} \end{aligned}$$

Examples:

- Cost function
- Machine learning: loss function measures discrepancy of model and training data set $\{(\xi_i, y_i)\}$

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} \ell(g_x(\xi_i), y_i) + R(x)$$

  - $x \in \mathbb{R}^n$: parameter vector
  - $\xi_i \in \mathbb{R}^n$: feature vector
  - $y_i \in \mathbb{R}$: output

First order methods?

- Dimensionality n and size m of data sets: Higher order methods prohibitively expensive.
- First-order: can evaluate $f(x)$ and $\nabla f(x)$.

First-order optimization: from continuous to discrete time

Gradient descent	$\mathcal{O}(1/k)$
Mirror descent [16] Dual Averaging [19]	$\mathcal{O}(1/k)$
Nesterov's accelerated method [18, 17]	$\mathcal{O}(1/k^2)$

Unified approach to derive these algorithms

- Design ODE in continuous time using Lyapunov argument.
- Discretize.

---

[16]A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley-Interscience series in discrete mathematics. Wiley, 1983

[19]Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, 2009

[18]Y. Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). *Soviet Mathematics Doklady*, 27(2):372–376, 1983

[17]Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005

From Gradient Descent to Mirror Descent

	Gradient descent	Mirror descent
Dynamics	$\dot{X}(t) = -\nabla f(X(t))$	$\begin{cases} \dot{Z}(t) = -\nabla f(X(t)) \\ X(t) = \nabla \psi^*(Z(t)) \end{cases}$
Lyapunov function	$\frac{1}{2}\|X(t) - x^\star\|^2$	$D_{\psi^*}(z^\star, Z(t))$
Convergence rate	$f(X(t)) - f^\star = \mathcal{O}(1/t)$	$f(X(t)) - f^\star = \mathcal{O}(1/t)$

[16]A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience series in discrete mathematics. Wiley, 1983

From Gradient Descent to Mirror Descent

	Gradient descent	Mirror descent
Dynamics	$\dot{X}(t) = -\nabla f(X(t))$	$\begin{cases} \dot{Z}(t) = -\nabla f(X(t)) \\ X(t) = \nabla \psi^*(Z(t)) \end{cases}$
Lyapunov function	$\frac{1}{2}\lVert X(t) - x^\star \rVert^2$	$D_{\psi^*}(z^\star, Z(t))$
Convergence rate	$f(X(t)) - f^\star = \mathcal{O}(1/t)$	$f(X(t)) - f^\star = \mathcal{O}(1/t)$

Nemirovski and Yudin [16]

1. Start from Bregman divergence on the dual space

$$D_{\psi^*}(Z, z^\star) = \psi^*(Z) - \psi^*(z^\star) - \langle \nabla \psi^*(z^\star), Z - z^\star \rangle$$

2. Design dynamics to make it a Lyapunov function.

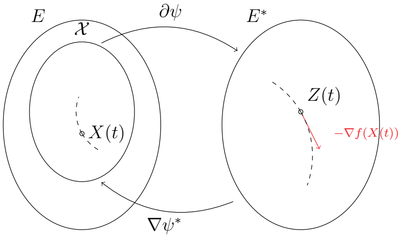


Figure: Illustration of Mirror Descent

[16]A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience series in discrete mathematics. Wiley, 1983

Mirror operator $\nabla\psi^*$

ψ^* is defined and differentiable on E^*, $\nabla\psi^*$ maps E^* to $\mathcal{X}$.

Sufficient condition

$\psi : \mathcal{X} \to \mathbb{R}$ is convex, closed, (essentially) strongly convex, such that epi f contains no non-vertical half-lines.

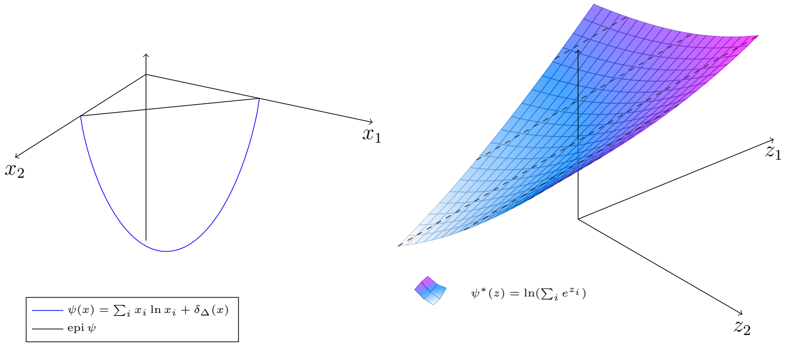


Figure: Example of dual distance generating functions ψ and ψ^*.

[21]R. Rockafellar. *Convex Analysis.*
Princeton University Press, 1970

An ODE interpretation of Nesterov's method

Su et al. [23]: ODE interpretation of Nesterov's method for unconstrained problems. Parameter $r \geq 2$.

	Unconstrained Nesterov
Dynamics	$\ddot{X} + \frac{r+1}{t}\dot{X} + \nabla f(X) = 0$
Lyapunov function	$\mathcal{E}(t) := \frac{t^2}{r^2}(f(X) - f^\star) + \frac{1}{2}\|X + \frac{t}{r}\dot{X} - x^\star\|_2^2$
Convergence rate	$f(X(t)) - f^\star = \mathcal{O}(1/t^2)$

Convergence rate

$$f(X(t)) - f^\star \leq \frac{r^2}{t^2}\mathcal{E}(t) \leq \frac{r^2}{t^2}\mathcal{E}(0) = \frac{r^2}{t^2}\|x_0 - x^\star\|^2$$

[23]W. Su, S. Boyd, and E. Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights.
In *NIPS*, 2014

Accelerated Mirror Descent in continuous time

We start from a Lyapunov function [11]

$$V(X, Z, t) = \frac{t^2}{r^2}(f(X) - f^\star) + D_{\psi^*}(Z, z^\star)$$

$Z \in E^*$, $z^\star$ its value at equilibrium.

[11]W. Krichene, A. Bayen, and P. Bartlett. Accelerated mirror descent in continuous and discrete time. In *29th Annual Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2015

Accelerated Mirror Descent in continuous time

We start from a Lyapunov function [11]

$$V(X, Z, t) = \frac{t^2}{r^2}(f(X) - f^\star) + D_{\psi^*}(Z, z^\star)$$

$Z \in E^*$, $z^\star$ its value at equilibrium.

	AMD (proximal Nesterov)
Dynamics	$\begin{cases} \dot{Z} = -\frac{t}{r}\nabla f(X), \\ \dot{X} = \frac{r}{t}(\nabla\psi^*(Z) - X), \end{cases}$
Lyapunov function	$\frac{t^2}{r^2}(f(X(t)) - f^\star) + D_{\psi^*}(Z(t), z^\star)$
Convergence rate	$f(X(t)) - f^\star = \mathcal{O}(1/t^2)$

[11]W. Krichene, A. Bayen, and P. Bartlett. Accelerated mirror descent in continuous and discrete time. In *29th Annual Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2015

Accelerated Mirror Descent in continuous time

We start from a Lyapunov function [11]

$$V(X, Z, t) = \frac{t^2}{r^2}(f(X) - f^\star) + D_{\psi^*}(Z, z^\star)$$

$Z \in E^*$, $z^\star$ its value at equilibrium.

	AMD (proximal Nesterov)
Dynamics	$\begin{cases} \dot{Z} = -\frac{t}{r}\nabla f(X), \\ \dot{X} = \frac{r}{t}(\nabla\psi^*(Z) - X), \end{cases}$
Lyapunov function	$\frac{t^2}{r^2}(f(X(t)) - f^\star) + D_{\psi^*}(Z(t), z^\star)$
Convergence rate	$f(X(t)) - f^\star = \mathcal{O}(1/t^2)$

Existence, uniqueness and viability of the solution

Suppose ∇f and $\nabla\psi^*$ are Lipschitz. Then the AMD ODE has a unique solution defined on $[0, +\infty)$, and $X(t)$ remains in $\mathcal{X}$.

[11]W. Krichene, A. Bayen, and P. Bartlett. Accelerated mirror descent in continuous and discrete time. In *29th Annual Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2015

Damped oscillator interpretation

Damped nonlinear oscillator

Accelerated mirror descent ODE is equivalent to

$$\ddot{X} + \frac{r+1}{t}\dot{X} = -\nabla^2 \psi^*(Z) \nabla f(X)$$

Damped oscillator interpretation

Damped nonlinear oscillator

Accelerated mirror descent ODE is equivalent to

$$\ddot{X} + \frac{r+1}{t}\dot{X} = -\nabla^2\psi^*(Z)\nabla f(X)$$

- Special case: $\ddot{X} + \frac{r+1}{t}\dot{X} = -\nabla f(X)$
- $\frac{r+1}{t}\dot{X}$: vanishing friction term.
- $\nabla^2\psi^*(Z)$: transforms the potential field to keep trajectory inside $\mathcal{X}$.

Effect of the parameter r

$$\ddot{X} + \frac{r+1}{t}\dot{X} = -\nabla^2 \psi^*(Z) \nabla f(X)$$

Figure: Effect of the parameter $r \in [2, 50]$.

Effect of $\nabla^2\psi^*(Z)$

$$\ddot{X} + \frac{r+1}{t}\dot{X} = -\nabla^2\psi^*(Z)\nabla f(X)$$

Figure: Flow field $x \mapsto \nabla^2\psi^*(Z(t))\nabla f(x)$, along the solution trajectory Z

Averaging Interpretation

$$\begin{cases} \dot{Z} = -\frac{t}{r}\nabla f(X), \\ \dot{X} = \frac{r}{t}(\nabla\psi^*(Z) - X), \end{cases}$$

Equivalent to

$$\begin{cases} \dot{Z} = -\frac{t}{r}\nabla f(X), \\ X(t) = \frac{\int_0^t w(\tau)\nabla\psi^*(Z(\tau))d\tau}{\int_0^t w(\tau)d\tau}, \\ (w(\tau) = \tau^{r-1}) \end{cases}$$

Averaging Interpretation

$$\begin{cases} \dot{Z} = -\frac{t}{r}\nabla f(X), \\ \dot{X} = \frac{r}{t}(\nabla\psi^*(Z) - X), \end{cases}$$

Equivalent to

$$\begin{cases} \dot{Z} = -\frac{t}{r}\nabla f(X), \\ X(t) = \frac{\int_0^t w(\tau)\nabla\psi^*(Z(\tau))d\tau}{\int_0^t w(\tau)d\tau}, \\ (w(\tau) = \tau^{r-1}) \end{cases}$$

AMD with generalized averaging

$$\text{AMD}_{w,\eta} \begin{cases} \dot{Z} = -\eta(t)\nabla f(X), \\ X(t) = \frac{\int_0^t w(\tau)\nabla\psi^*(Z(\tau))d\tau}{\int_0^t w(\tau)d\tau} \\ \nabla\psi^*(z_0) = x_0. \end{cases}$$

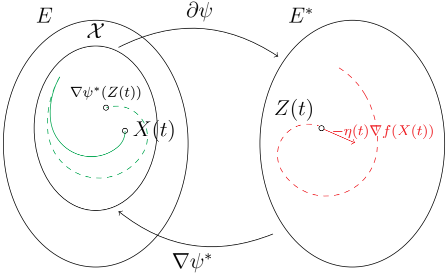


Figure: Averaging interpretation: Z evolves in E^*, X is a weighted average of the mirrored trajectory $\nabla\psi^*(Z)$.

Example: accelerated entropic descent on the simplex

Suppose the feasible set is the probability simplex $\mathcal{X} = \Delta = \{x \in \mathbb{R}^n_+ : \sum_i x_i = 1\}$.

$$\psi(x) = \sum_i x_i \ln x_i + \delta(x|\Delta), \qquad \psi^*(z) = \ln \sum_i e^{z_i}, \qquad \nabla \psi^*(z)_i = \frac{e^{z_i}}{\sum_i e^{z_i}},$$

Accelerated replicator ODE

$$\begin{cases} \dot{\check{Z}}_i = \check{Z}_i \left(\langle \check{Z}, \nabla f(X) \rangle - \nabla_i f(X) \right) \\ X = \frac{\int_0^t \tau^{r-1} \check{Z}(\tau) d\tau}{\int_0^t \tau^{r-1} d\tau} \end{cases}$$

Replicator:

$$\dot{X}_i = X_i \left(\langle X, \nabla f(X) \rangle - \nabla_i f(X) \right)$$

Numerical Example

Figure: Accelerated entropic descent on a quadratic on the simplex.

Generalized Averaging

Dynamics	$\text{AMD}_{w,\eta} \begin{cases} \dot{Z} = -\eta(t)\nabla f(X), \\ X(t) = \frac{\int_0^t w(\tau)\nabla\psi^*(Z(\tau))d\tau}{\int_0^t w(\tau)d\tau} \end{cases}$
Lyapunov function	$\mathcal{E}_r(t) := r(t)(f(X(t)) - f^\star) + D_{\psi^*}(Z(t), z^\star)$
Convergence rate	$f(X(t)) - f^\star = \mathcal{O}(1/r(t))$

[13]W. Krichene, A. Bayen, and P. Bartlett. Adaptive averaging in accelerated descent dynamics. In *30th Annual Conference on Neural Information Processing Systems (NIPS), in review*, 2016

Generalized Averaging

Dynamics	$\mathrm{AMD}_{w,\eta} \begin{cases} \dot{Z} = -\eta(t)\nabla f(X), \\ X(t) = \frac{\int_0^t w(\tau)\nabla\psi^*(Z(\tau))d\tau}{\int_0^t w(\tau)d\tau} \end{cases}$
Lyapunov function	$\mathcal{E}_r(t) := r(t)(f(X(t)) - f^\star) + D_{\psi^*}(Z(t), z^\star)$
Convergence rate	$f(X(t)) - f^\star = \mathcal{O}(1/r(t))$

Derivative of energy function

$$\frac{d}{dt}\mathcal{E}_r(t) \leq (f(X) - f^\star)(r' - \eta) + \left\langle \nabla f(X), \dot{X} \right\rangle \left(r - \frac{\eta}{a}\right)$$

$a(t) = w(t)/\int_0^t w(\tau)d\tau$, i.e. $w(t) = \frac{a(t)}{a(0)} \int_0^t a(\tau)d\tau$.

[13]W. Krichene, A. Bayen, and P. Bartlett. Adaptive averaging in accelerated descent dynamics. In *30th Annual Conference on Neural Information Processing Systems (NIPS), in review*, 2016

Generalized Averaging

Dynamics	$\text{AMD}_{w,\eta} \begin{cases} \dot{Z} = -\eta(t)\nabla f(X), \\ X(t) = \frac{\int_0^t w(\tau)\nabla\psi^*(Z(\tau))d\tau}{\int_0^t w(\tau)d\tau} \end{cases}$
Lyapunov function	$\mathcal{E}_r(t) := r(t)(f(X(t)) - f^\star) + D_{\psi^*}(Z(t), z^\star)$
Convergence rate	$f(X(t)) - f^\star = \mathcal{O}(1/r(t))$

Derivative of energy function

$$\frac{d}{dt}\mathcal{E}_r(t) \le (f(X) - f^\star)(r' - \eta) + \left\langle \nabla f(X), \dot{X} \right\rangle (r - \frac{\eta}{a})$$

$a(t) = w(t)/\int_0^t w(\tau)d\tau$, i.e. $w(t) = \frac{a(t)}{a(0)} \int_0^t a(\tau)d\tau$.

Convergence rate

If $a(t) = \frac{\eta(t)}{r(t)}$ and $\eta(t) \ge r'(t)$, then $\mathcal{E}_r$ is a Lyapunov function for $\text{AMD}_{w,\eta}$ and

$$f(X(t)) - f^\star \le \frac{\mathcal{E}_r(t_0)}{r(t)}$$

[13]W. Krichene, A. Bayen, and P. Bartlett. Adaptive averaging in accelerated descent dynamics. In *30th Annual Conference on Neural Information Processing Systems (NIPS), in review*, 2016

Adaptive Averaging

$$\frac{d}{dt}\mathcal{E}_r(t) \leq (f(X) - f^\star)(r' - \eta) + \left\langle \nabla f(X), \dot{X} \right\rangle (r - \frac{\eta}{a})$$

- We set $a(t) = \frac{\eta(t)}{r(t)}$ to cancel last term.
- Instead,

Adaptive Averaging

$$\begin{cases} a(t) = \frac{\eta(t)}{r(t)} & \text{if } \left\langle \nabla f(X), \dot{X} \right\rangle > 0 \\ a(t) \text{ constant} & \text{otherwise.} \end{cases}$$

Discrete AMD algorithm in the quadratic case.

Accelerated mirror descent in discrete time

1: Initialize $\tilde{x}^{(0)} = x_0$, $\check{z}^{(0)} = x_0$
2: **for** $k \in \mathbb{N}$ **do**
3: $\quad \check{z}^{(k+1)} = \arg\min_{\check{z} \in \mathcal{X}} \frac{\beta ks}{r^2} \langle \nabla f(x^{(k)}), \check{z} \rangle + D_\psi(\check{z}, x^{(k)})$
4: $\quad \tilde{x}^{(k+1)} = \arg\min_{\tilde{x} \in \mathcal{X}} \gamma s \langle \nabla f(x^{(k)}), \tilde{x} \rangle + R(\tilde{x}, x^{(k)})$
5: $\quad x^{(k+1)} = \lambda_k \check{z}^{(k+1)} + (1 - \lambda_k)\tilde{x}^{(k+1)}$, with $\lambda_k = \frac{\sqrt{s}a_k}{1+\sqrt{s}a_k}$.
6: $\quad a_k = \frac{\beta}{k\sqrt{s}}$
7: **end for**

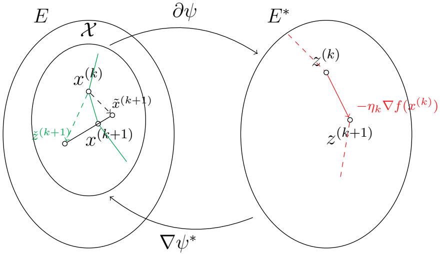


Figure: Illustration of the discrete AMD algorithm.

Discrete AMD algorithm in the quadratic case.

Accelerated mirror descent in discrete time

Initialize $\tilde{x}^{(0)} = x_0$, $\check{z}^{(0)} = x_0$

for $k \in \mathbb{N}$ **do**

$\quad \check{z}^{(k+1)} = \arg\min_{\check{z} \in \mathcal{X}} \frac{\beta k s}{r^2} \left\langle \nabla f(x^{(k)}), \check{z} \right\rangle + D_\psi(\check{z}, x^{(k)})$

$\quad \tilde{x}^{(k+1)} = \arg\min_{\tilde{x} \in \mathcal{X}} \gamma s \left\langle \nabla f(x^{(k)}), \tilde{x} \right\rangle + R(\tilde{x}, x^{(k)})$

$\quad x^{(k+1)} = \lambda_k \check{z}^{(k+1)} + (1 - \lambda_k)\tilde{x}^{(k+1)}$, with $\lambda_k = \frac{\sqrt{s} a_k}{1 + \sqrt{s} a_k}$.

$$a_k = \begin{cases} \frac{\beta}{k\sqrt{s}} & \text{if } f(\tilde{x}^{(k+1)}) - f(\tilde{x}^{(k)}) > 0 \\ a_{k-1} & \text{otherwise} \end{cases}$$

end for

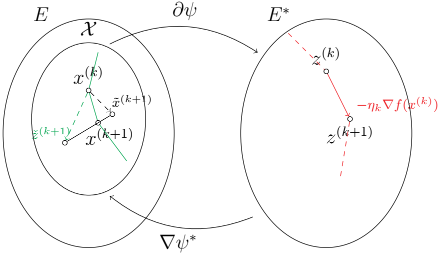


Figure: Illustration of the discrete AMD algorithm.

Illustration of Adaptive Averaging

Figure: Illustration of adaptive averaging

Convergence rate

Convergence rate

If $\gamma \geq \frac{\beta \beta^{\mathbf{max}} L_f L_{\psi^*}}{r^2}$ and $s \leq \frac{\ell_R}{2 L_f \gamma}$, then under AMD (both adaptive and non-adaptive),

$$f(\tilde{x}^{(k)}) - f^\star \leq C/k^2,$$

where $C = D_{\psi^*}(z_0, z^\star) + \frac{s}{r^2}(f(x_0) - f^\star)$.

Proof: $\tilde{E}^{(k)} = V(\tilde{x}^{(k)}, z^{(k)}, k\sqrt{s})$ is a Lyapunov sequence.

Heuristics

Gradient Restart [20]	Speed Restart [23]	Adaptive Averaging [13]
Damped non-linear oscillator $\ddot{X} + \frac{r+1}{t}\dot{X} + \nabla f(X) = 0$ Restart when $\left\langle \nabla f(X), \dot{X} \right\rangle > 0$ Restart when moving in bad direction	Damped non-linear oscillator $\ddot{X} + \frac{r+1}{t}\dot{X} + \nabla f(X) = 0$ Restart when $\frac{d}{dt}\|\dot{X}\| < 0$ Restart when progress is slowing	Generalized Averaging $\begin{cases} \dot{Z} = -\eta(t)\nabla f(X), \\ X(t) = \frac{\int_0^t w(\tau)\nabla\psi^*(Z(\tau))d\tau}{\int_0^t w(\tau)d\tau} \end{cases}$ $a(t) = \frac{\eta(t)}{r(t)}$ if $\left\langle \nabla f(X), \dot{X} \right\rangle > 0$, constant otherwise. Increase weights on good portions of trajectory

[20]B. O'Donoghue and E. Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715–732, 2015

[23]W. Su, S. Boyd, and E. Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *NIPS*, 2014

[13]W. Krichene, A. Bayen, and P. Bartlett. Adaptive averaging in accelerated descent dynamics. In *30th Annual Conference on Neural Information Processing Systems (NIPS), in review*, 2016

Comparison of Heuristics

Figure: Comparison of the adaptive averaging and restarting heuristics

Higher order methods

Figure: Adaptive averaging for quadratic and cubic accelerated methods.

Summary / Extensions

Dynamical systems approach to online learning and optimization

- Design / analyze dynamics in continuous-time.
- Discretize.

[14]A. Lew, J. E. Marsden, M. Ortiz, and M. West. Variational time integrators. *International Journal for Numerical Methods in Engineering*, 60(1):153–212, 2004

[25]A. Wibisono, A. C. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization. *CoRR*, abs/1603.04245, 2016

Summary / Extensions

Dynamical systems approach to online learning and optimization

- Design / analyze dynamics in continuous-time.
- Discretize.

Contributions

- Online learning algorithms as stochastic approximation of the replicator ODE.
- (Estimation and control under Hedge dynamics: not covered in this talk).
- Unifying framework for design of accelerated methods for first-order optimization.
- Averaging interpretation and heuristics.

[14]A. Lew, J. E. Marsden, M. Ortiz, and M. West. Variational time integrators.
International Journal for Numerical Methods in Engineering, 60(1):153–212, 2004

[25]A. Wibisono, A. C. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization.
CoRR, abs/1603.04245, 2016

Summary / Extensions

Dynamical systems approach to online learning and optimization

- Design / analyze dynamics in continuous-time.
- Discretize.

Contributions

- Online learning algorithms as stochastic approximation of the replicator ODE.
- (Estimation and control under Hedge dynamics: not covered in this talk).
- Unifying framework for design of accelerated methods for first-order optimization.
- Averaging interpretation and heuristics.

Possible extensions

- ODE for monotone operators.
- Use variational integrators [14] to discretize the ODE.
    - Discretize dynamics while preserving natural energy of mechanical system.
    - Discretize Hamilton's critical action principle instead of ODE.
    - Combine with Wibisono et al.'s Lagrangian interpretation of AMD dynamics [25].

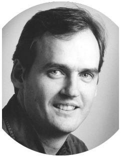
---

[14]A. Lew, J. E. Marsden, M. Ortiz, and M. West. Variational time integrators. *International Journal for Numerical Methods in Engineering*, 60(1):153–212, 2004

[25]A. Wibisono, A. C. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization. *CoRR*, abs/1603.04245, 2016

Acknowledgements

Alex Bayen

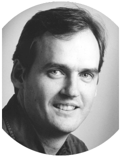


Peter Bartlett

Nikhil Srivastava

Acknowledgements

Alex Bayen



Peter Bartlett

Nikhil Srivastava

Laurent El Ghaoui

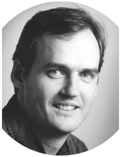
Claire Tomlin

Shankar Sastry

Satish Rao

Acknowledgements


Alex Bayen


Peter Bartlett


Nikhil Srivastava


Laurent El Ghaoui


Claire Tomlin


Shankar Sastry


Satish Rao


Benjamin Drighès


Milena Suarez


Syrine Krichene


Kiet Lam


Chedly Bourguiba

Thank you!

`eecs.berkeley.edu/~walid/`

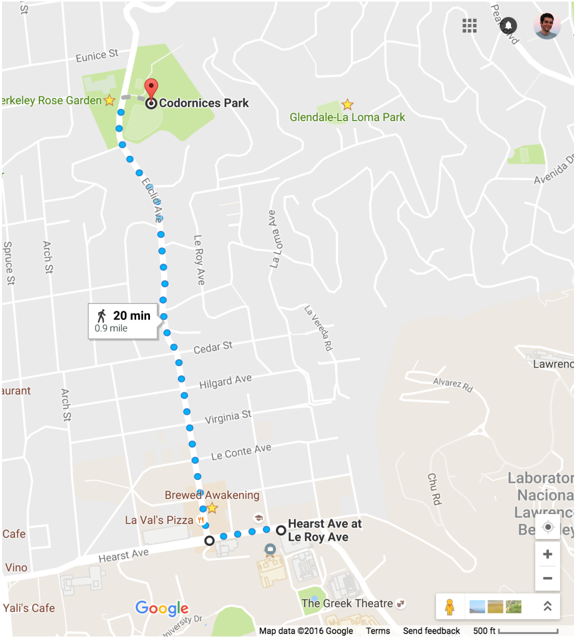


Figure: Picnic in 1 hour!

References I

[1] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[2] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, May 2003.

[3] M. Benaïm. Dynamics of stochastic approximation algorithms. In *Séminaire de probabilités XXXIII*, pages 1–68. Springer, 1999.

[4] L. E. Blume. The statistical mechanics of strategic interaction. *Games and Economic Behavior*, 5(3):387 – 424, 1993.

[5] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[6] Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1):79–103, 1999.

[7] J. Hannan. Approximation to bayes risk in repeated plays. *Contributions to the Theory of Games*, 3:97–139, 1957.

[8] S. Hart and A. Mas-Colell. A general class of adaptive strategies. *Journal of Economic Theory*, 98(1):26 – 54, 2001.

[9] J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1 – 63, 1997.

References II

[10] S. Krichene, W. Krichene, R. Dong, and A. Bayen. Convergence of heterogeneous distributed learning in stochastic routing games. In *53rd Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2015.

[11] W. Krichene, A. Bayen, and P. Bartlett. Accelerated mirror descent in continuous and discrete time. In *29th Annual Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2015.

[12] W. Krichene, B. Drighès, and A. Bayen. Learning nash equilibria in congestion games. *SIAM Journal on Control and Optimization (SICON)*, 2015.

[13] W. Krichene, A. Bayen, and P. Bartlett. Adaptive averaging in accelerated descent dynamics. In *30th Annual Conference on Neural Information Processing Systems (NIPS), in review*, 2016.

[14] A. Lew, J. E. Marsden, M. Ortiz, and M. West. Variational time integrators. *International Journal for Numerical Methods in Engineering*, 60(1):153–212, 2004.

[15] J. R. Marden and J. S. Shamma. Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation. *Games and Economic Behavior*, 75(2):788 – 808, 2012. ISSN 0899-8256.

[16] A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience series in discrete mathematics. Wiley, 1983.

[17] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

References III

[18] Y. Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[19] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, 2009.

[20] B. O'Donoghue and E. Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715–732, 2015.

[21] R. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[22] W. H. Sandholm. *Population games and evolutionary dynamics*. Economic learning and social evolution. Cambridge, Mass. MIT Press, 2010.

[23] W. Su, S. Boyd, and E. Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *NIPS*, 2014.

[24] J. W. Weibull. *Evolutionary game theory*. MIT press, 1997.

[25] A. Wibisono, A. C. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization. *CoRR*, abs/1603.04245, 2016.