PDE continuum limits for prediction with expert advice

Jeff Calder

School of Mathematics
University of Minnesota

Nonlinear Analysis Seminar
Rutger's University
March 31, 2021

Joint work with Nadejda Drenska (UMN) and Charlie Smart (Chicago)

This research was supported by the National Science Foundation and the Alfred P. Sloan Foudation.

Outline

Outline

Two Player Games and PDEs

There is a long history connecting two player games and PDEs

- Differential Games (Isaacs Equation)

Two Player Games and PDEs

There is a long history connecting two player games and PDEs

- Differential Games (Isaacs Equation)
- Kohn-Serfaty game for curvature motion [Kohn & Serfaty, 2006]
  - Fully nonlinear parabolic equations [Kohn & Serfaty, 2010]

Two Player Games and PDEs

There is a long history connecting two player games and PDEs

- Differential Games (Isaacs Equation)
- Kohn-Serfaty game for curvature motion [Kohn & Serfaty, 2006]
    - Fully nonlinear parabolic equations [Kohn & Serfaty, 2010]
- Stochastic Tug-of-War games for the p-Laplacian (inlcuding $p = \infty$)
    - [Peres & Scheffield, 2008]
    - [Peres, Schramm, Scheffield, Wilson, 2009]
    - [Manfredi, Parviainen, Rossi, 2010, 2012]
    - [Armstrong & Smart, 2012]
    - [Lewicka, Manfredi, 2014, 2017]
    - Applications to machine learning [Calder 2018] [Slepčev & Thorpe, 2019]

Two Player Games and PDEs

There is a long history connecting two player games and PDEs

- Differential Games (Isaacs Equation)

- Kohn-Serfaty game for curvature motion [Kohn & Serfaty, 2006]
  - Fully nonlinear parabolic equations [Kohn & Serfaty, 2010]

- Stochastic Tug-of-War games for the p-Laplacian (inlcuding $p = \infty$)
  - [Peres & Scheffield, 2008]
  - [Peres, Schramm, Scheffield, Wilson, 2009]
  - [Manfredi, Parviainen, Rossi, 2010, 2012]
  - [Armstrong & Smart, 2012]
  - [Lewicka, Manfredi, 2014, 2017]
  - Applications to machine learning [Calder 2018] [Slepčev & Thorpe, 2019]

- Convex Hull Peeling and the affine flow [Calder & Smart, 2020]

Two Player Games and PDEs

There is a long history connecting two player games and PDEs

- Differential Games (Isaacs Equation)
- Kohn-Serfaty game for curvature motion [Kohn & Serfaty, 2006]
  - Fully nonlinear parabolic equations [Kohn & Serfaty, 2010]
- Stochastic Tug-of-War games for the p-Laplacian (inlcuding $p = \infty$)
  - [Peres & Scheffield, 2008]
  - [Peres, Schramm, Scheffield, Wilson, 2009]
  - [Manfredi, Parviainen, Rossi, 2010, 2012]
  - [Armstrong & Smart, 2012]
  - [Lewicka, Manfredi, 2014, 2017]
  - Applications to machine learning [Calder 2018] [Slepčev & Thorpe, 2019]
- Convex Hull Peeling and the affine flow [Calder & Smart, 2020]
- Prediction from expert advice [Kohn & Drenska, 2020] [Drenska & Calder, 2020]
  - Generalization of the Kohn-Serfaty game

Kohn-Serfaty Game

The game is played in a convex domain $\Omega \subset \mathbb{R}^2$ starting at $x_0 \in \Omega$ and involves a small parameter $\varepsilon > 0$. The rules of the game are

1. Paul chooses a direction vector $v_k \in \mathbb{S}^1$.
2. Carol moves the token from x_k to $x_{k+1} = x_0 \pm \sqrt{2}\varepsilon v_k$.

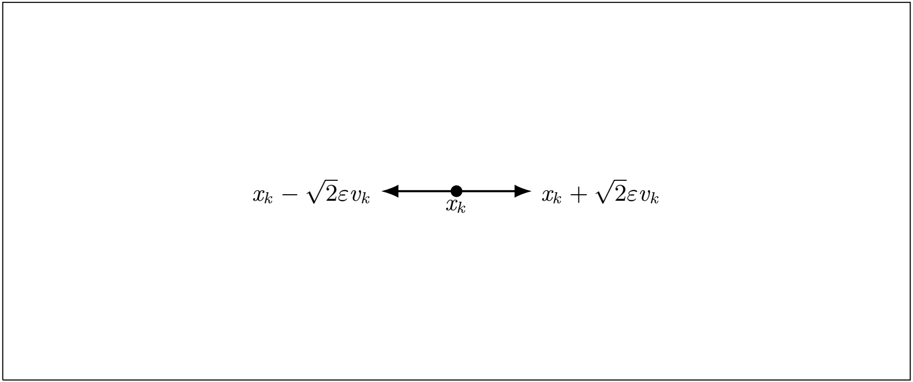
Paul wants to escape Ω and Carol wants to obstruct.

Kohn-Serfaty Game

The game is played in a convex domain $\Omega \subset \mathbb{R}^2$ starting at $x_0 \in \Omega$ and involves a small parameter $\varepsilon > 0$. The rules of the game are

1. Paul chooses a direction vector $v_k \in \mathbb{S}^1$.
2. Carol moves the token from x_k to $x_{k+1} = x_0 \pm \sqrt{2}\varepsilon v_k$.

Paul wants to escape Ω and Carol wants to obstruct.

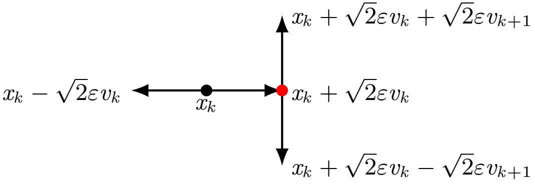


Kohn-Serfaty Game

The game is played in a convex domain $\Omega \subset \mathbb{R}^2$ starting at $x_0 \in \Omega$ and involves a small parameter $\varepsilon > 0$. The rules of the game are

1. Paul chooses a direction vector $v_k \in \mathbb{S}^1$.
2. Carol moves the token from x_k to $x_{k+1} = x_0 \pm \sqrt{2}\varepsilon v_k$.

Paul wants to escape Ω and Carol wants to obstruct.



Kohn-Serfaty Game

Let us define

$$u_\varepsilon(x_0) = \varepsilon^2(\text{Number of steps for Paul to escape } \Omega)$$

given that both players play optimally and the game starts at x_0. The value function u satisfies the dynamic programming principle

$$u_\varepsilon(x) = \varepsilon^2 + \min_{|v|=1} \max_{b=\pm 1} u_\varepsilon(x + \sqrt{2}\varepsilon bv).$$

Kohn-Serfaty Game

Let us define

$$u_\varepsilon(x_0) = \varepsilon^2(\text{Number of steps for Paul to escape } \Omega)$$

given that both players play optimally and the game starts at x_0. The value function u satisfies the dynamic programming principle

$$u_\varepsilon(x) = \varepsilon^2 + \min_{|v|=1} \max_{b=\pm 1} u_\varepsilon(x + \sqrt{2}\varepsilon bv).$$

We assume $u_\varepsilon \approx u$ where u is smooth and Taylor expand to obtain

$$u(x) \approx \varepsilon^2 + \min_{|v|=1} \max_{b=\pm 1} \left\{ u(x) + \sqrt{2}\varepsilon b \nabla u(x)^T v + \varepsilon^2 v^T \nabla^2 u(x) v \right\}.$$

Kohn-Serfaty Game

Let us define

$$u_\varepsilon(x_0) = \varepsilon^2(\text{Number of steps for Paul to escape } \Omega)$$

given that both players play optimally and the game starts at x_0. The value function u satisfies the dynamic programming principle

$$u_\varepsilon(x) = \varepsilon^2 + \min_{|v|=1} \max_{b=\pm 1} u_\varepsilon(x + \sqrt{2}\varepsilon bv).$$

We assume $u_\varepsilon \approx u$ where u is smooth and Taylor expand to obtain

$$u(x) \approx \varepsilon^2 + \min_{|v|=1} \max_{b=\pm 1} \left\{ u(x) + \sqrt{2}\varepsilon b \nabla u(x)^T v + \varepsilon^2 v^T \nabla^2 u(x) v \right\}.$$

Paul should choose $v = \nabla^\perp u / |\nabla u|$, where $\nabla^\perp u = (-u_{x_2}, u_{x_1})$, yielding

$$0 \approx 1 + \frac{(\nabla^\perp u)^T}{|\nabla u|} \nabla^2 u \frac{\nabla^\perp u}{|\nabla u|}$$

Kohn-Serfaty Game

Let us define

$$u_\varepsilon(x_0) = \varepsilon^2(\text{Number of steps for Paul to escape } \Omega)$$

given that both players play optimally and the game starts at x_0. The value function u satisfies the dynamic programming principle

$$u_\varepsilon(x) = \varepsilon^2 + \min_{|v|=1} \max_{b=\pm 1} u_\varepsilon(x + \sqrt{2}\varepsilon bv).$$

We assume $u_\varepsilon \approx u$ where u is smooth and Taylor expand to obtain

$$u(x) \approx \varepsilon^2 + \min_{|v|=1} \max_{b=\pm 1} \left\{ u(x) + \sqrt{2}\varepsilon b \nabla u(x)^T v + \varepsilon^2 v^T \nabla^2 u(x) v \right\}.$$

Paul should choose $v = \nabla^\perp u / |\nabla u|$, where $\nabla^\perp u = (-u_{x_2}, u_{x_1})$, yielding

$$0 \approx 1 + \frac{(\nabla^\perp u)^T}{|\nabla u|} \nabla^2 u \frac{\nabla^\perp u}{|\nabla u|} = 1 + |\nabla u| \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right).$$

Kohn-Serfaty Game

Kohn & Serfaty showed that $u_\varepsilon \to u$ as $\varepsilon \to 0$ where u is the viscosity solution of

$$(1) \qquad \begin{cases} -|\nabla u| \text{div}\left(\frac{\nabla u}{|\nabla u|}\right) = 1 & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$

Kohn-Serfaty Game

Kohn & Serfaty showed that $u_\varepsilon \to u$ as $\varepsilon \to 0$ where u is the viscosity solution of

$$\begin{cases} -|\nabla u| \text{div}\left(\frac{\nabla u}{|\nabla u|}\right) = 1 & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{cases} \tag{1}$$

- This is the level-set equation for motion by mean curvature of the level sets of u.

Kohn-Serfaty Game

Kohn & Serfaty showed that $u_\varepsilon \to u$ as $\varepsilon \to 0$ where u is the viscosity solution of

$$(1) \qquad \begin{cases} -|\nabla u| \text{div}\left(\dfrac{\nabla u}{|\nabla u|}\right) = 1 & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$

- This is the level-set equation for motion by mean curvature of the level sets of u.
- The number of steps for Paul to escape concides in the limit as $\varepsilon \to 0$ with the arrival time for the boundary evolving under curvature motion.

Kohn-Serfaty Game

Kohn & Serfaty showed that $u_\varepsilon \to u$ as $\varepsilon \to 0$ where u is the viscosity solution of

(1)
$$\begin{cases} -|\nabla u| \text{div}\left(\dfrac{\nabla u}{|\nabla u|}\right) = 1 & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$

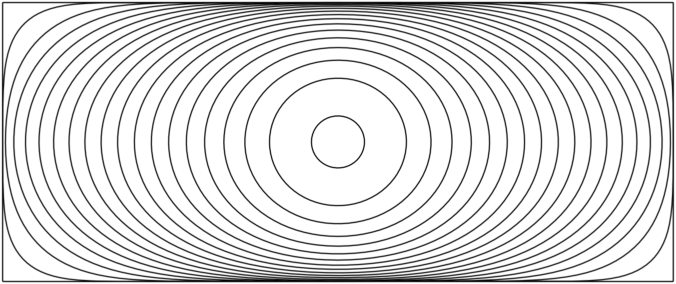
- This is the level-set equation for motion by mean curvature of the level sets of u.
- The number of steps for Paul to escape concides in the limit as $\varepsilon \to 0$ with the arrival time for the boundary evolving under curvature motion.
- Paul's asymptotically optimal strategy to choose v tangent to level sets of u.

Kohn-Serfaty Game

Kohn & Serfaty showed that $u_\varepsilon \to u$ as $\varepsilon \to 0$ where u is the viscosity solution of

(1) $$\begin{cases} -|\nabla u|\text{div}\left(\frac{\nabla u}{|\nabla u|}\right) = 1 & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$

- This is the level-set equation for motion by mean curvature of the level sets of u.
- The number of steps for Paul to escape concides in the limit as $\varepsilon \to 0$ with the arrival time for the boundary evolving under curvature motion.
- Paul's asymptotically optimal strategy to choose v tangent to level sets of u.



Playing the Kohn-Serfaty game on a point cloud

Players: Paul and Carol
State space: $\mathcal{X} := \{X_1, \ldots, X_n\}$

Playing the Kohn-Serfaty game on a point cloud

Players: Paul and Carol
State space: $\mathcal{X} := \{X_1, \ldots, X_n\}$

Paul's goal: Reach vertex of convex hull
Carol's goal: Obstruct Paul

Playing the Kohn-Serfaty game on a point cloud

Players: Paul and Carol

State space: $\mathcal{X} := \{X_1, \ldots, X_n\}$

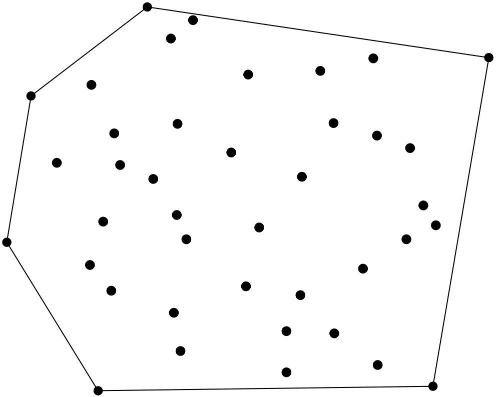
Paul's goal: Reach vertex of convex hull

Carol's goal: Obstruct Paul

Rules of the game: Token starts at $x^0 \in \mathcal{X}$ and is moved according to:

1. Paul picks $v \in \mathbb{S}^{d-1}$
2. Carol moves token to any $x^{k+1} \in \mathcal{X}$ satisfying

$$(x^{k+1} - x^k)^T v > 0.$$



Playing the Kohn-Serfaty game on a point cloud

Players: Paul and Carol

State space: $\mathcal{X} := \{X_1, \ldots, X_n\}$

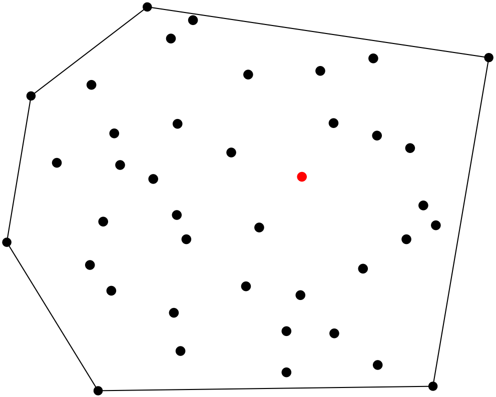
Paul's goal: Reach vertex of convex hull

Carol's goal: Obstruct Paul

Rules of the game: Token starts at $x^0 \in \mathcal{X}$ and is moved according to:

1. Paul picks $v \in \mathbb{S}^{d-1}$
2. Carol moves token to any $x^{k+1} \in \mathcal{X}$ satisfying

$$(x^{k+1} - x^k)^T v > 0.$$



Playing the Kohn-Serfaty game on a point cloud

Players: Paul and Carol

State space: $\mathcal{X} := \{X_1, \ldots, X_n\}$

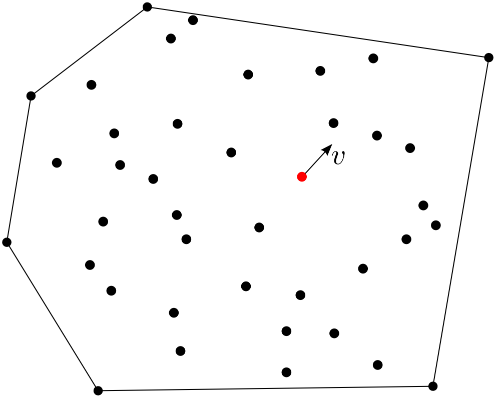
Paul's goal: Reach vertex of convex hull

Carol's goal: Obstruct Paul

Rules of the game: Token starts at $x^0 \in \mathcal{X}$ and is moved according to:

1. Paul picks $v \in \mathbb{S}^{d-1}$
2. Carol moves token to any $x^{k+1} \in \mathcal{X}$ satisfying

$$(x^{k+1} - x^k)^T v > 0.$$



Playing the Kohn-Serfaty game on a point cloud

Players: Paul and Carol

State space: $\mathcal{X} := \{X_1, \ldots, X_n\}$

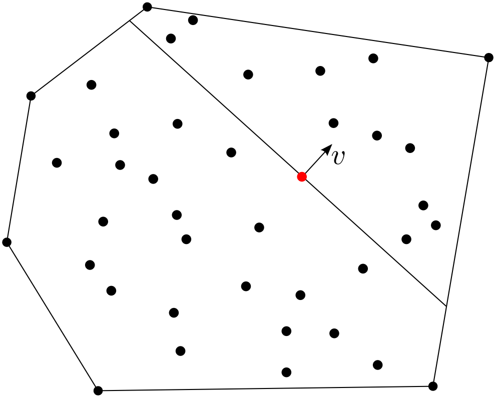
Paul's goal: Reach vertex of convex hull

Carol's goal: Obstruct Paul

Rules of the game: Token starts at $x^0 \in \mathcal{X}$ and is moved according to:

1. Paul picks $v \in \mathbb{S}^{d-1}$
2. Carol moves token to any $x^{k+1} \in \mathcal{X}$ satisfying

$$(x^{k+1} - x^k)^T v > 0.$$



Playing the Kohn-Serfaty game on a point cloud

Players: Paul and Carol

State space: $\mathcal{X} := \{X_1, \dots, X_n\}$

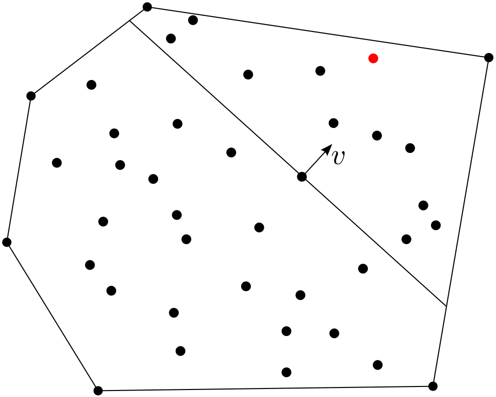
Paul's goal: Reach vertex of convex hull

Carol's goal: Obstruct Paul

Rules of the game: Token starts at $x^0 \in \mathcal{X}$ and is moved according to:

1. Paul picks $v \in \mathbb{S}^{d-1}$
2. Carol moves token to any $x^{k+1} \in \mathcal{X}$ satisfying

$$(x^{k+1} - x^k)^T v > 0.$$



Playing the Kohn-Serfaty game on a point cloud

Players: Paul and Carol

State space: $\mathcal{X} := \{X_1, \ldots, X_n\}$

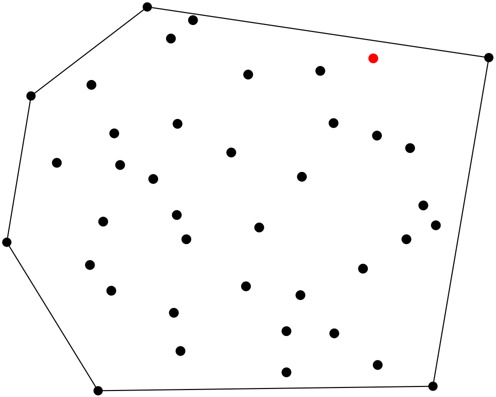
Paul's goal: Reach vertex of convex hull

Carol's goal: Obstruct Paul

Rules of the game: Token starts at $x^0 \in \mathcal{X}$ and is moved according to:

1. Paul picks $v \in \mathbb{S}^{d-1}$
2. Carol moves token to any $x^{k+1} \in \mathcal{X}$ satisfying

$$(x^{k+1} - x^k)^T v > 0.$$



Playing the Kohn-Serfaty game on a point cloud

Players: Paul and Carol

State space: $\mathcal{X} := \{X_1, \ldots, X_n\}$

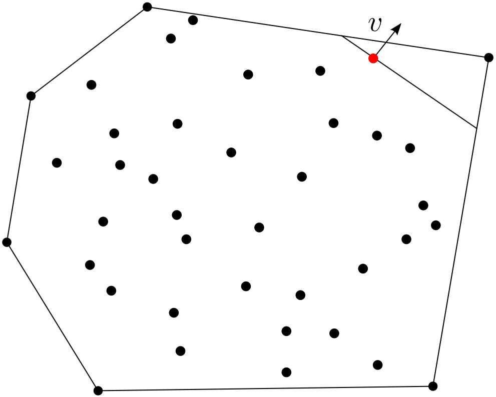
Paul's goal: Reach vertex of convex hull

Carol's goal: Obstruct Paul

Rules of the game: Token starts at $x^0 \in \mathcal{X}$ and is moved according to:

1. Paul picks $v \in \mathbb{S}^{d-1}$
2. Carol moves token to any $x^{k+1} \in \mathcal{X}$ satisfying

$$(x^{k+1} - x^k)^T v > 0.$$



Playing the Kohn-Serfaty game on a point cloud

Players: Paul and Carol

State space: $\mathcal{X} := \{X_1, \ldots, X_n\}$

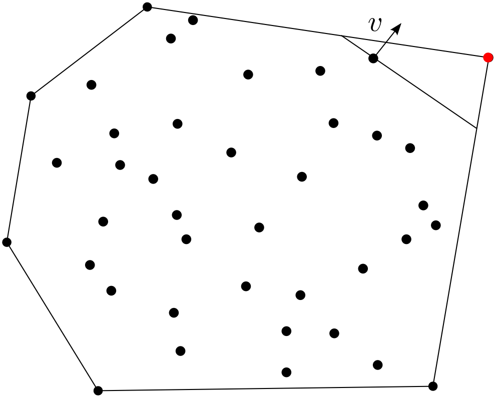
Paul's goal: Reach vertex of convex hull

Carol's goal: Obstruct Paul

Rules of the game: Token starts at $x^0 \in \mathcal{X}$ and is moved according to:

1. Paul picks $v \in \mathbb{S}^{d-1}$
2. Carol moves token to any $x^{k+1} \in \mathcal{X}$ satisfying

$$(x^{k+1} - x^k)^T v > 0.$$

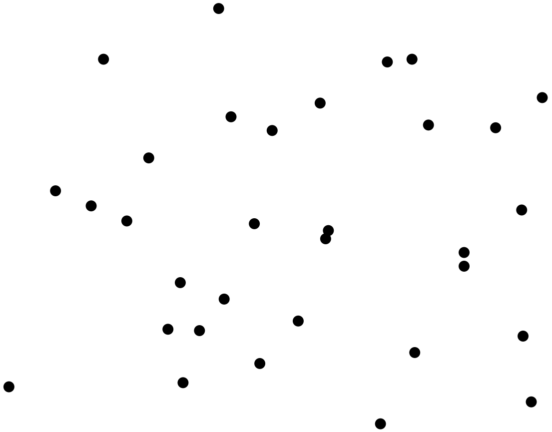


Convex hull peeling

- Introduced by Barnet 1976 as a notion of multivariate median.
- Used in robust statistics, machine learning, mathcing of point clouds, fingerprint identification, etc.

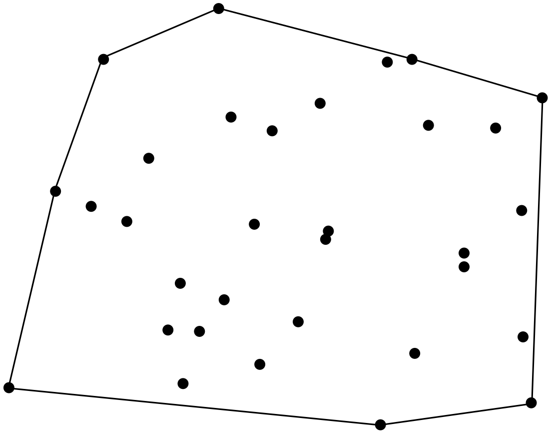
Convex hull peeling

- Introduced by Barnet 1976 as a notion of multivariate median.
- Used in robust statistics, machine learning, mathcing of point clouds, fingerprint identification, etc.



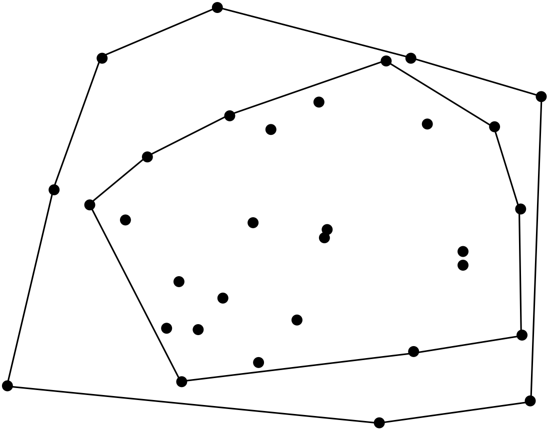
Convex hull peeling

- Introduced by Barnet 1976 as a notion of multivariate median.
- Used in robust statistics, machine learning, mathcing of point clouds, fingerprint identification, etc.



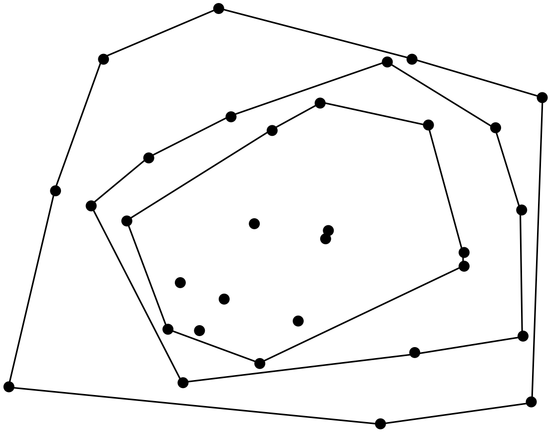
Convex hull peeling

- Introduced by Barnet 1976 as a notion of multivariate median.
- Used in robust statistics, machine learning, mathcing of point clouds, fingerprint identification, etc.



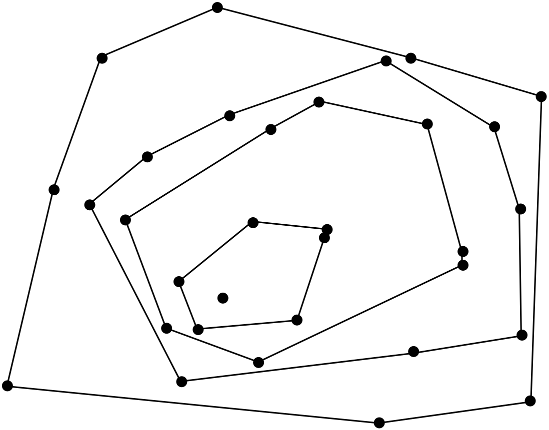
Convex hull peeling

- Introduced by Barnet 1976 as a notion of multivariate median.
- Used in robust statistics, machine learning, mathcing of point clouds, fingerprint identification, etc.



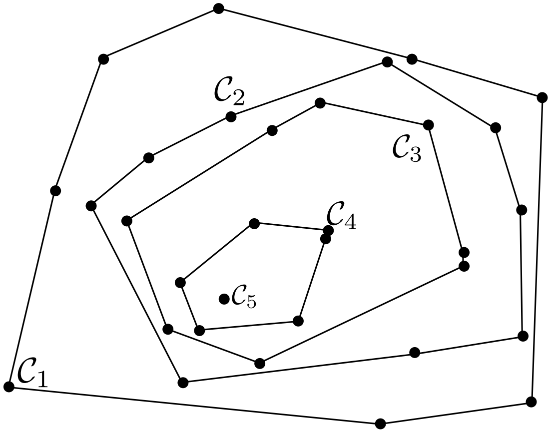
Convex hull peeling

- Introduced by Barnet 1976 as a notion of multivariate median.
- Used in robust statistics, machine learning, mathcing of point clouds, fingerprint identification, etc.



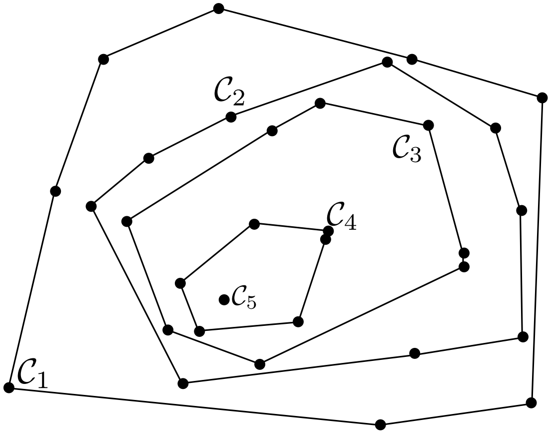
Convex hull peeling

- Introduced by Barnet 1976 as a notion of multivariate median.
- Used in robust statistics, machine learning, mathcing of point clouds, fingerprint identification, etc.



Convex hull peeling

- Introduced by Barnet 1976 as a notion of multivariate median.
- Used in robust statistics, machine learning, mathcing of point clouds, fingerprint identification, etc.

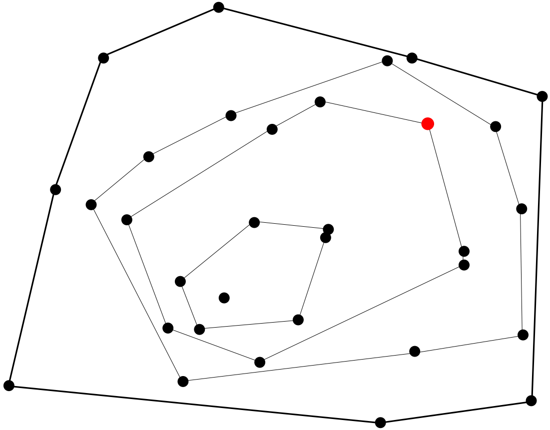


Convex hull peeling median := Centroid of final layer

Optimal strategies come from Convex Hull Peeling

Paul's optimal choice: Any halfspace supporting current convex layer

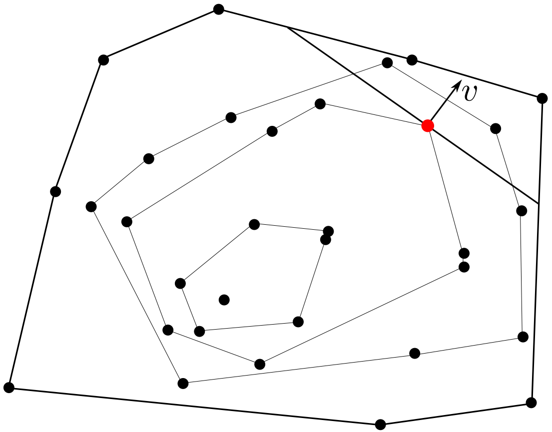
Carol's optimal choice: Any point on the previous convex layer



Optimal strategies come from Convex Hull Peeling

Paul's optimal choice: Any halfspace supporting current convex layer

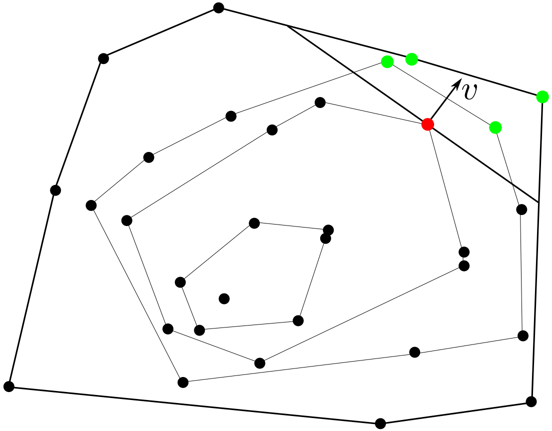
Carol's optimal choice: Any point on the previous convex layer



Optimal strategies come from Convex Hull Peeling

Paul's optimal choice: Any halfspace supporting current convex layer

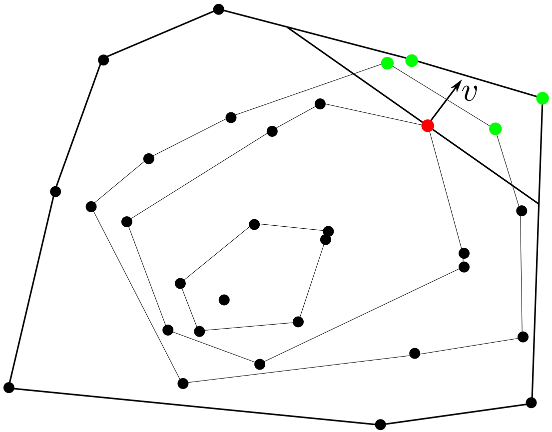
Carol's optimal choice: Any point on the previous convex layer



Optimal strategies come from Convex Hull Peeling

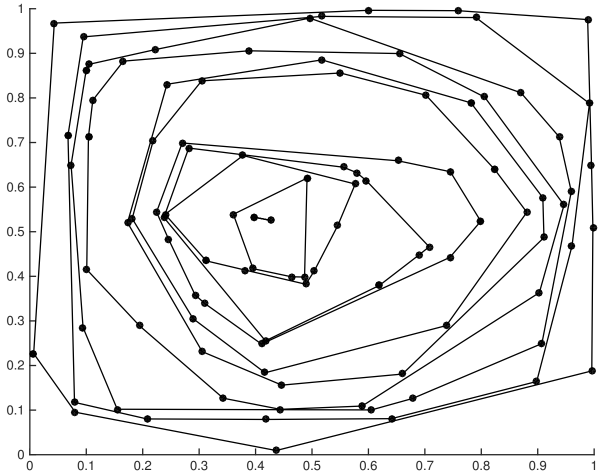
Paul's optimal choice: Any halfspace supporting current convex layer

Carol's optimal choice: Any point on the previous convex layer



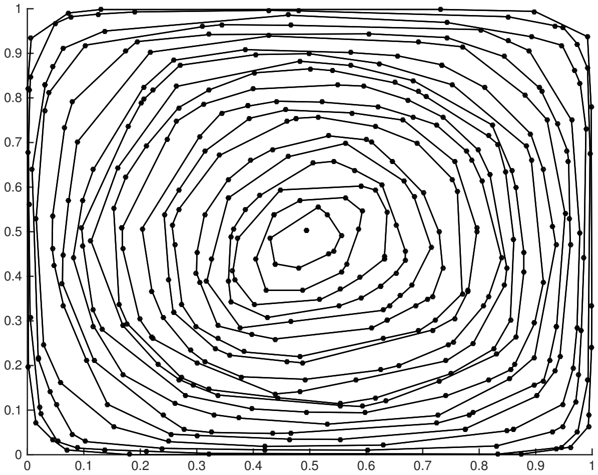
Value function $= U_n(x^0) =$ Convex depth function.

Convex hull peeling: Demo - Uniform distribution



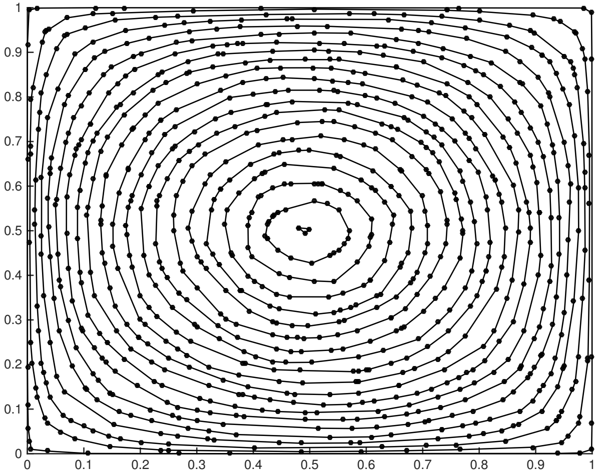
$n = 10^2$ points

Convex hull peeling: Demo - Uniform distribution



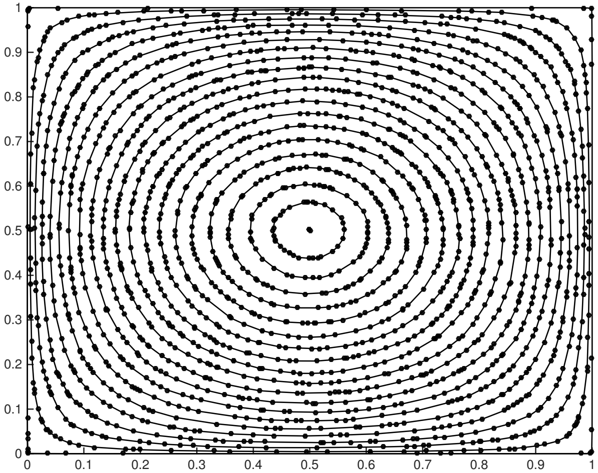
$n = 10^3$ points

Convex hull peeling: Demo - Uniform distribution



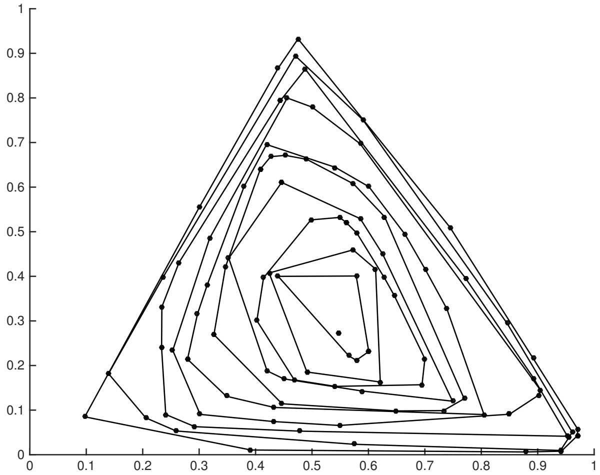
$n = 10^4$ points

Convex hull peeling: Demo - Uniform distribution



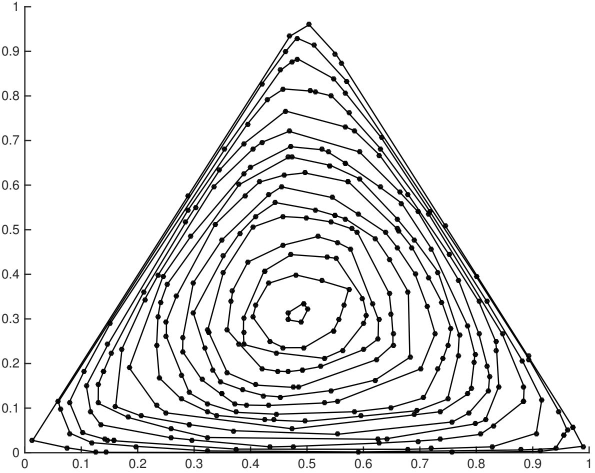
$n = 10^5$ points

Convex hull peeling: Demo - Triangle distribution



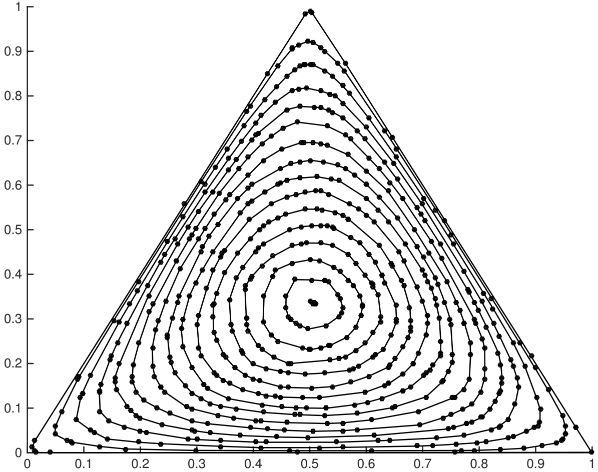
$n = 10^2$ points

Convex hull peeling: Demo - Triangle distribution



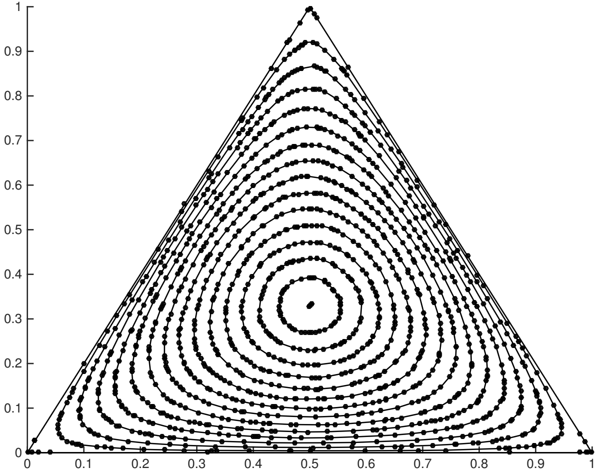
$n = 10^3$ points

Convex hull peeling: Demo - Triangle distribution



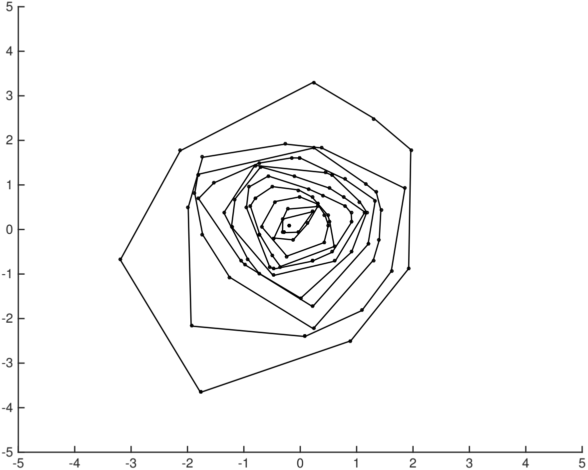
$n = 10^4$ points

Convex hull peeling: Demo - Triangle distribution



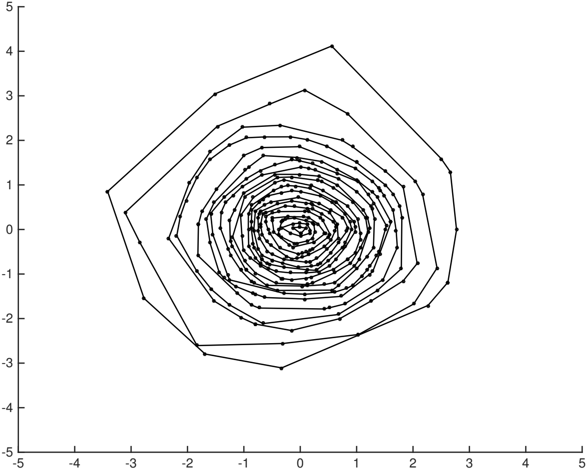
$n = 10^5$ points

Convex hull peeling: Demo - Gaussian distribution



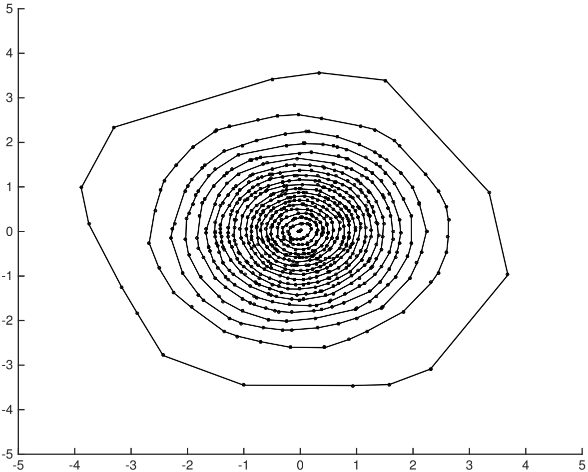
$n = 10^2$ points

Convex hull peeling: Demo - Gaussian distribution



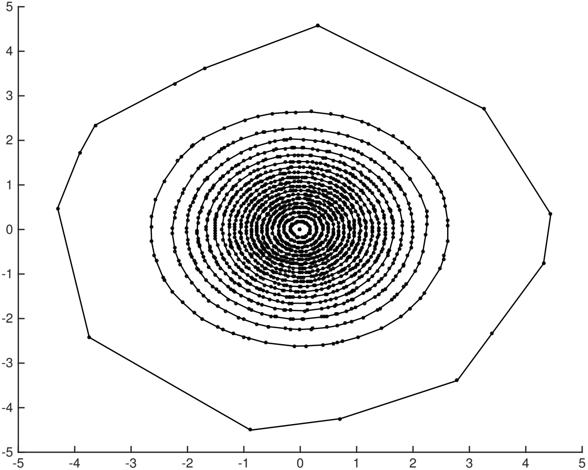
$n = 10^3$ points

Convex hull peeling: Demo - Gaussian distribution



$n = 10^4$ points

Convex hull peeling: Demo - Gaussian distribution

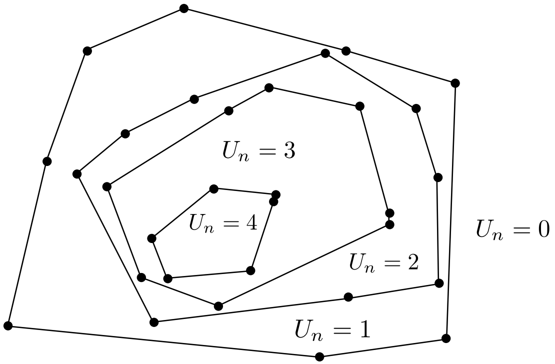


$n = 10^5$ points

A PDE continuum limit for convex hull peeling

Let $X_1, \ldots, X_n$ be i.i.d. with a continuous density ρ on a convex set $\Omega \subset \mathbb{R}^d$.

Let U_n be the function that 'counts' the associated convex layers.



Partial differential equation (PDE) continuum limit

Theorem (Calder & Smart, 2020)

There exists a universal constant α_d *such that with probability one*

$$n^{-\frac{2}{d+1}} U_n \longrightarrow \alpha_d u \quad \textit{uniformly on } \Omega,$$

where $u \in C(\overline{\Omega})$ *is the unique viscosity solution of*

$$\begin{cases} \nabla u^T \mathrm{cof}(-\nabla^2 u) \nabla u = \rho^2 & \textit{in } \Omega \\ u = 0 & \textit{on } \partial\Omega. \end{cases} \tag{2}$$

Partial differential equation (PDE) continuum limit

Theorem (Calder & Smart, 2020)

There exists a universal constant α_d *such that with probability one*

$$n^{-\frac{2}{d+1}}\, U_n \longrightarrow \alpha_d u \quad \textit{uniformly on } \Omega,$$

where $u \in C(\overline{\Omega})$ *is the unique viscosity solution of*

$$\begin{cases} \nabla u^T \mathrm{cof}(-\nabla^2 u)\nabla u = \rho^2 & \textit{in } \Omega \\ u = 0 & \textit{on } \partial\Omega. \end{cases} \tag{2}$$

This is just motion by a power of Gauss curvature

$$\frac{dS}{dt} = \rho^{-2/(d+1)} \kappa_G^{1/(d+1)} \mathbf{n}.$$

Known as affine invariant curvature motion when $\rho \equiv 1$.

Theorem (Calder & Smart, 2020)

There exists a universal constant α_d *such that with probability one*

$$n^{-\frac{2}{d+1}} U_n \longrightarrow \alpha_d u \quad \textit{uniformly on } \Omega,$$

where $u \in C(\overline{\Omega})$ *is the unique viscosity solution of*

$$\begin{cases} \nabla u^T \mathrm{cof}(-\nabla^2 u)\nabla u = \rho^2 & \textit{in } \Omega \\ u = 0 & \textit{on } \partial\Omega. \end{cases} \tag{3}$$

U_n satisfies a dynamic programming principle arising from the two player game

$$U_n(x) = \inf_{p \in \mathbb{R}^d \setminus \{0\}} \sup_{p^T(y-x)>0} [\mathbb{1}_{\{X_1,\dots,X_n\}}(y) + U_n(y)].$$

- Proof requires more than Taylor expansion and reading off the optimal strategies.
- Involves analyzing the scaling limit of the game after a large number of steps (locally), which has connections to stochsatic growth models.

Calder, J., and Smart, C.K. The limit shape of convex hull peeling. Duke Mathematical Journal, 169.11 (2020): 2079-2124.

A PDE continuum limit for convex hull peeling

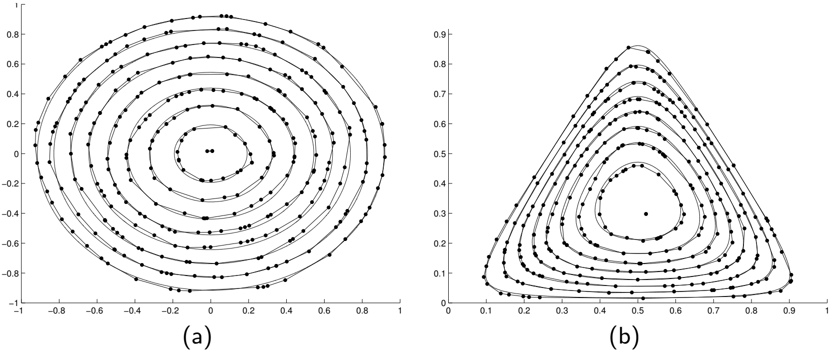
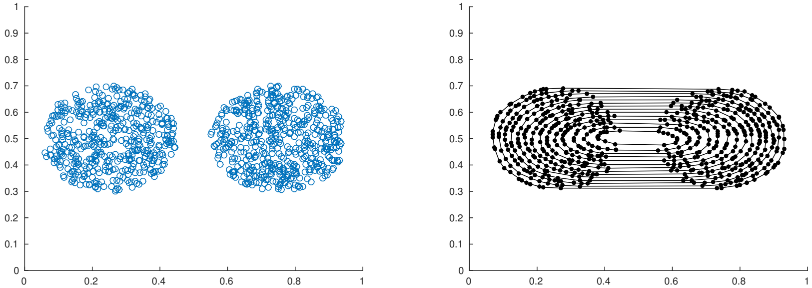


Figure: Convex layers vs continuum limit for $n = 5 \times 10^3$.

A nonconvex example

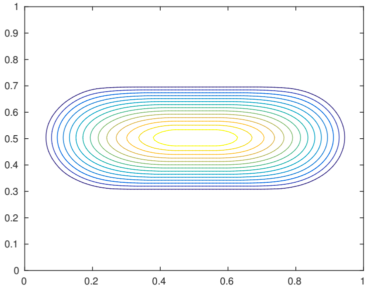


(a) Samples

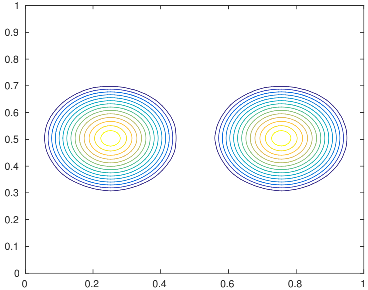
(b) Convex layers

Figure: Convex layers corresponding to disjoint clusters.

A nonconvex example



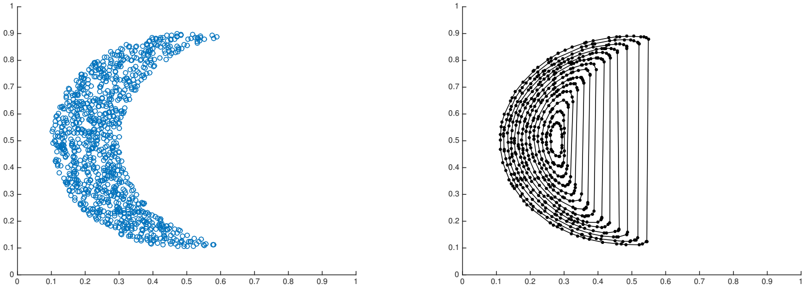
(a) One solution



(b) Another solution

Figure: Two different solutions continuum PDE.

The halfmoon

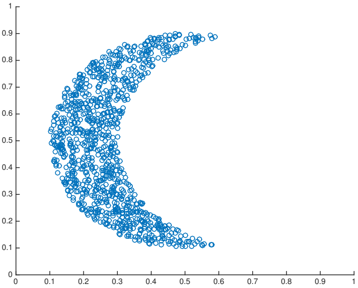


(a) Samples

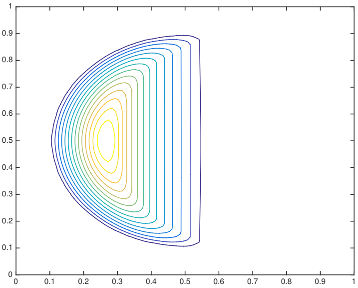
(b) Convex layers

Figure: Convex layers corresponding to the halfmoon distribution.

The halfmoon



(a) Samples



(b) PDE

Figure: Solution of PDE for the halfmoon example.

Outline

Prediction with expert advice

- One of the oldest online machine learning problems [Cover, 1966].
- We are given a stream of data $b_1, b_2, b_3, \ldots$.
- A pool of "experts" makes predictions about future values b_k.
- The player must use the expert advice to make their own prediction.
- The player's performance is measured by regret

$$\text{Regret to expert } i := \text{Expert } i\text{'s performance} - \text{Player's performance.}$$

Prediction with expert advice

Key feature: Worst case analysis.

Prediction with expert advice

Key feature: Worst case analysis.

- No modeling assumptions made on the data stream $b_1, b_2, b_3, \ldots$.

Prediction with expert advice

Key feature: Worst case analysis.

- No modeling assumptions made on the data stream $b_1, b_2, b_3, \ldots$.
- The data stream (environment) is assumed to be controlled by an adversary.

Prediction with expert advice

Key feature: Worst case analysis.

- No modeling assumptions made on the data stream $b_1, b_2, b_3, \ldots$.
- The data stream (environment) is assumed to be controlled by an adversary.
- Yields two player zero-sum games with minimax optimal strategies.

Prediction with expert advice

Key feature: Worst case analysis.

- No modeling assumptions made on the data stream $b_1, b_2, b_3, \ldots$.
- The data stream (environment) is assumed to be controlled by an adversary.
- Yields two player zero-sum games with minimax optimal strategies.

Applications: Financial math, weather prediction, click prediction,. . .

Example: Weather prediction

Goal: Each morning predict whether it will rain or not.

Example: Weather prediction

Goal: Each morning predict whether it will rain or not.

Possible Experts:

1. The Weather Network
2. AccuWeather
3. Weather Underground
4. Your own deep neural network
5. It will rain today if it rained yesterday
6. It always rains
7. It never rains
8. Toss a coin
9. Red sky in the morning

Previous work

2 constant experts:

- Optimal strategies [Cover, 1966]

Previous work

2 constant experts:

- Optimal strategies [Cover, 1966]

Multiplicative weights algorithm (MWA):

- [Littlestone and Warmuth, 1994, Vovk, 1990]
- Also called weighted majority algorithm.

Previous work

2 constant experts:

- Optimal strategies [Cover, 1966]

Multiplicative weights algorithm (MWA):

- [Littlestone and Warmuth, 1994, Vovk, 1990]
- Also called weighted majority algorithm.
- Provably optimal as $n, T \to \infty$ [Cesa-Bianchi and Lugosi, 2006].

Previous work

2 constant experts:

- Optimal strategies [Cover, 1966]

Multiplicative weights algorithm (MWA):

- [Littlestone and Warmuth, 1994, Vovk, 1990]
- Also called weighted majority algorithm.
- Provably optimal as $n, T \to \infty$ [Cesa-Bianchi and Lugosi, 2006].
- For finite number of experts n, MWA is not optimal.

Previous work

2 constant experts:

- Optimal strategies [Cover, 1966]

Multiplicative weights algorithm (MWA):

- [Littlestone and Warmuth, 1994, Vovk, 1990]
- Also called weighted majority algorithm.
- Provably optimal as $n, T \to \infty$ [Cesa-Bianchi and Lugosi, 2006].
- For finite number of experts n, MWA is not optimal.

Optimal strategies:

- $n = 2, 3$ experts [Gravin et al., 2016, Abbasi et al., 2017].
- $n = 4$ experts [Bayraktar et al., 2019]

Previous work

2 constant experts:

- Optimal strategies [Cover, 1966]

Multiplicative weights algorithm (MWA):

- [Littlestone and Warmuth, 1994, Vovk, 1990]
- Also called weighted majority algorithm.
- Provably optimal as $n, T \to \infty$ [Cesa-Bianchi and Lugosi, 2006].
- For finite number of experts n, MWA is not optimal.

Optimal strategies:

- $n = 2, 3$ experts [Gravin et al., 2016, Abbasi et al., 2017].
- $n = 4$ experts [Bayraktar et al., 2019]
- Connection to PDEs for $n \geq 2$ experts
  - [Zhu, 2014, Drenska, 2017, Drenska and Kohn, 2019b]

Problem setup: History dependent experts

- Daily stock price movements $b_1, b_2, b_3, \ldots, b_k, \ldots$ with $b_k \in \mathcal{B} := \{-1, 1\}$.

Problem setup: History dependent experts

- Daily stock price movements $b_1, b_2, b_3, \ldots, b_k, \ldots$ with $b_k \in \mathcal{B} := \{-1, 1\}$.
- We have n experts predicting b_i based on d-days of history

$$m^i := (b_{i-d}, b_{i-d+1}, \ldots, b_{i-1}) \in \mathcal{B}^d.$$

Problem setup: History dependent experts

- Daily stock price movements $b_1, b_2, b_3, \ldots, b_k, \ldots$ with $b_k \in \mathcal{B} := \{-1, 1\}$.
- We have n experts predicting b_i based on d-days of history

$$m^i := (b_{i-d}, b_{i-d+1}, \ldots, b_{i-1}) \in \mathcal{B}^d.$$

- The expert predictions are publicly available algorithms

$$q_1, \ldots, q_n : \mathcal{B}^d \to [-1, 1],$$

and we write $q = (q_1, \ldots, q_n)$.

Problem setup: History dependent experts

- Daily stock price movements $b_1, b_2, b_3, \ldots, b_k, \ldots$ with $b_k \in \mathcal{B} := \{-1, 1\}$.
- We have n experts predicting b_i based on d-days of history

$$m^i := (b_{i-d}, b_{i-d+1}, \ldots, b_{i-1}) \in \mathcal{B}^d.$$

- The expert predictions are publicly available algorithms

$$q_1, \ldots, q_n : \mathcal{B}^d \to [-1, 1],$$

  and we write $q = (q_1, \ldots, q_n)$.

- Rules of the game: For $i = 1$ up to N

  1. The investor views $q(m^i)$ and decides on an investment $f_i \in [-1, 1]$.

Problem setup: History dependent experts

- Daily stock price movements $b_1, b_2, b_3, \ldots, b_k, \ldots$ with $b_k \in \mathcal{B} := \{-1, 1\}$.
- We have n experts predicting b_i based on d-days of history

$$m^i := (b_{i-d}, b_{i-d+1}, \ldots, b_{i-1}) \in \mathcal{B}^d.$$

- The expert predictions are publicly available algorithms

$$q_1, \ldots, q_n : \mathcal{B}^d \to [-1, 1],$$

  and we write $q = (q_1, \ldots, q_n)$.

- Rules of the game: For $i = 1$ up to N

  1. The investor views $q(m^i)$ and decides on an investment $f_i \in [-1, 1]$.
  2. The market chooses $b_i \in \mathcal{B}$.

Problem setup: History dependent experts

- Daily stock price movements $b_1, b_2, b_3, \ldots, b_k, \ldots$ with $b_k \in \mathcal{B} := \{-1, 1\}$.
- We have n experts predicting b_i based on d-days of history

$$m^i := (b_{i-d}, b_{i-d+1}, \ldots, b_{i-1}) \in \mathcal{B}^d.$$

- The expert predictions are publicly available algorithms

$$q_1, \ldots, q_n : \mathcal{B}^d \to [-1, 1],$$

  and we write $q = (q_1, \ldots, q_n)$.

- Rules of the game: For $i = 1$ up to N
  1. The investor views $q(m^i)$ and decides on an investment $f_i \in [-1, 1]$.
  2. The market chooses $b_i \in \mathcal{B}$.
  3. Investor accumulates regret $q_j(m^i)b_i - f_i b_i$ with respect to expert j.

Problem setup: History dependent experts

- After N steps of the game, the accumulated regret is

$$R_N := \sum_{i=1}^{N} b_i(q(m^i) - f_i \mathbb{1}), \qquad \mathbb{1} = (1, \ldots, 1).$$

Problem setup: History dependent experts

- After N steps of the game, the accumulated regret is

$$R_N := \sum_{i=1}^{N} b_i(q(m^i) - f_i \mathbb{1}), \qquad \mathbb{1} = (1, \ldots, 1).$$

- Objective: Given a payoff function $g : \mathbb{R}^n \to \mathbb{R}$
  - Market's goal is to maximize $g(R_N)$.
  - Investor's goal is to minimize $g(R_N)$.

Problem setup: History dependent experts

- After N steps of the game, the accumulated regret is

$$R_N := \sum_{i=1}^{N} b_i(q(m^i) - f_i \mathbb{1}), \qquad \mathbb{1} = (1, \ldots, 1).$$

- Objective: Given a payoff function $g : \mathbb{R}^n \to \mathbb{R}$
  - Market's goal is to maximize $g(R_N)$.
  - Investor's goal is to minimize $g(R_N)$.
- Common choice for payoff is

$$g(x) = \max\{x_1, x_2, \ldots, x_n\},$$

where x_i = regret with respect to expert i.

Drenska, N., and Kohn R.V. A PDE approach to the prediction of a binary sequence with advice from two history-dependent experts. arXiv preprint:2007.12732 (2020).

Problem setup: History dependent experts

- Notation: For $m = (m_1, \ldots, m_d) \in \mathcal{B}^d$ and $b \in \mathcal{B}$ we denote

$$m|b := (m_2, m_3, \ldots, m_d, b) \in \mathcal{B}^d.$$

The history transition is $m^{i+1} = m^i|b_i$.

Problem setup: History dependent experts

- Notation: For $m = (m_1, \ldots, m_d) \in \mathcal{B}^d$ and $b \in \mathcal{B}$ we denote

$$m|b := (m_2, m_3, \ldots, m_d, b) \in \mathcal{B}^d.$$

The history transition is $m^{i+1} = m^i|b_i$.

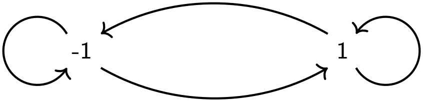
Definition (Value function)

Let $g : \mathbb{R}^n \to \mathbb{R}$. Given $N \in \mathbb{N}$, $m \in \mathcal{B}^d$, and $1 \leq \ell \leq N$, the value function $V_N(x, \ell; m)$ is defined by $V_N(x, \ell; m) = g(x)$ for $\ell = N$, and

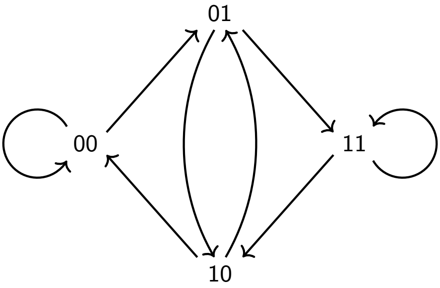
$$(4) \qquad V_N(x, \ell; m) = \min_{|f_\ell| \leq 1} \max_{b_\ell = \pm 1} \cdots \min_{|f_{N-1}| \leq 1} \max_{b_{N-1} = \pm 1} g\left(x + \sum_{i=\ell}^{N-1} b_i(q(m^i) - f_i \mathbb{1})\right)$$

for $1 \leq \ell \leq N - 1$, where $m^\ell = m$ and $m^{i+1} = m^i|b_i$ for $i = \ell, \ldots, N-1$.

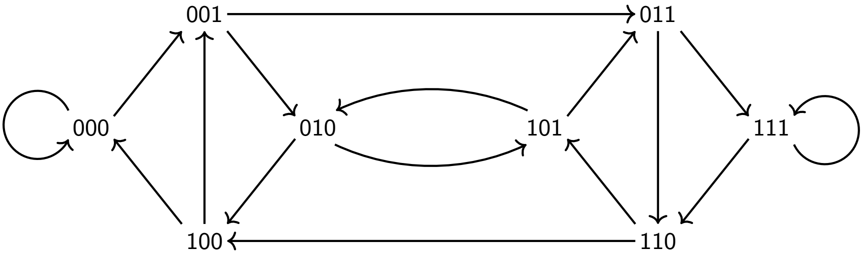
De Bruijn graph $d = 1$



De Bruijn graph $d = 2$



De Bruijn graph $d = 3$



Assumptions

- For $T > 0, N \in \mathbb{N}$, define $\varepsilon > 0$ by $T = \varepsilon^2 N$ and set

$$u_N(x,t;m) := \frac{1}{\sqrt{N}} V_N(\sqrt{N}x, \lceil Nt \rceil; m),$$

Assumptions

- For $T > 0, N \in \mathbb{N}$, define $\varepsilon > 0$ by $T = \varepsilon^2 N$ and set

$$u_N(x,t;m) := \frac{1}{\sqrt{N}} V_N(\sqrt{N}x, \lceil Nt \rceil; m),$$

- We assume $g \in C^4(\mathbb{R}^n)$ with uniformly bounded derivatives of order up to 4 over $\mathbb{R}^n$, there exists $\theta > 0$ such that

$$\nabla g(x)^T \mathbb{1} \geq \theta \quad \text{for all } x \in \mathbb{R}^n, \tag{5}$$

and that g is positively 1-homogeneous, that is

$$g(sx) = sg(x) \ \text{ for all } x \in \mathbb{R}^n, s > 0. \tag{6}$$

Assumptions

- For $T > 0, N \in \mathbb{N}$, define $\varepsilon > 0$ by $T = \varepsilon^2 N$ and set

$$u_N(x, t; m) := \frac{1}{\sqrt{N}} V_N(\sqrt{N}x, \lceil Nt \rceil; m),$$

- We assume $g \in C^4(\mathbb{R}^n)$ with uniformly bounded derivatives of order up to 4 over $\mathbb{R}^n$, there exists $\theta > 0$ such that

$$\nabla g(x)^T \mathbb{1} \geq \theta \quad \text{for all } x \in \mathbb{R}^n, \tag{5}$$

  and that g is positively 1-homogeneous, that is

$$g(sx) = sg(x) \ \text{ for all } x \in \mathbb{R}^n, s > 0. \tag{6}$$

- We also assume the expert strategies $q = (q_1, \ldots, q_n)$ satisfy

$$q : \mathcal{B}^d \to [-\mu, \mu]^n \quad \text{for some } \mu \in (0, 1). \tag{7}$$

Our main result

Let u be the viscosity solution of

$$\begin{cases} u_t + \frac{1}{2^{d+1}} \sum_{m \in \mathcal{B}^d} \eta(m)^T \nabla^2 u \, \eta(m) = 0, & \text{in } \mathbb{R}^n \times (0,1) \\ u = g, & \text{on } \mathbb{R}^n \times \{t = 1\}, \end{cases} \tag{8}$$

where

$$\eta(m) = q(m) - \frac{\nabla u^T q(m)}{\nabla u^T \mathbb{1}} \mathbb{1}. \tag{9}$$

Theorem (Drenska & Calder, 2020)

There exists $C_1, C_2 > 0$, depending on u, n and θ, such that

$$|u_N(x,t;m) - u(x,t)| \leq C_1 d(1 - t + \varepsilon)\varepsilon \tag{10}$$

holds for all $N \geq C_2 d^2/\mu^2$, $(x,t) \in \mathbb{R}^n \times [0,1]$ and $m \in \mathcal{B}^d$, where $\varepsilon = N^{-1/2}$.

Optimal strategies

An $O(\varepsilon)$ asymptotically optimal investor strategy is

$$f^* = \frac{\nabla u^T q}{\nabla u^T \mathbb{1}} + \frac{\varepsilon}{2}\left(\frac{\mathcal{H}(m_+) - \mathcal{H}(m_-)}{\nabla u^T \mathbb{1}}\right),$$

where $\mathcal{H}$ satisfies the graph Poisson equation

$$\Delta_{\mathcal{B}^d}\mathcal{H} = h - \frac{1}{2^d}\sum_{m\in\mathcal{B}^d} h(m)$$

where

$$\Delta_{\mathcal{B}^d}\mathcal{H}(m) = \mathcal{H}(m) - \frac{1}{2}\mathcal{H}(m_+) - \frac{1}{2}\mathcal{H}(m_-),$$

and

$$h(m) = \frac{1}{2}\eta(m)^T\nabla^2 u\,\eta(m) \quad \text{and} \quad \eta(m) = q(m) - \frac{\nabla u^T q(m)}{\nabla u^T \mathbb{1}}\mathbb{1}.$$

Optimal strategies

An $O(\varepsilon)$ asymptotically optimal investor strategy is

$$f^* = \frac{\nabla u^T q}{\nabla u^T \mathbb{1}} + \frac{\varepsilon}{2}\left(\frac{\mathcal{H}(m_+) - \mathcal{H}(m_-)}{\nabla u^T \mathbb{1}}\right),$$

where $\mathcal{H}$ satisfies the graph Poisson equation

$$\Delta_{\mathcal{B}^d}\mathcal{H} = h - \frac{1}{2^d}\sum_{m\in\mathcal{B}^d} h(m)$$

where

$$\Delta_{\mathcal{B}^d}\mathcal{H}(m) = \mathcal{H}(m) - \frac{1}{2}\mathcal{H}(m_+) - \frac{1}{2}\mathcal{H}(m_-),$$

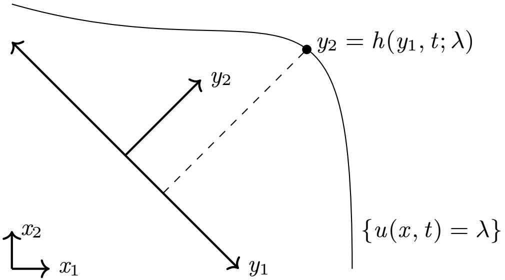
and

$$h(m) = \frac{1}{2}\eta(m)^T\nabla^2 u\,\eta(m) \quad \text{and} \quad \eta(m) = q(m) - \frac{\nabla u^T q(m)}{\nabla u^T \mathbb{1}}\mathbb{1}.$$

An asymptotically optimal market strategy is

$$b^* = \mathrm{sign}(f^* - f),$$

Underlying linear heat equation

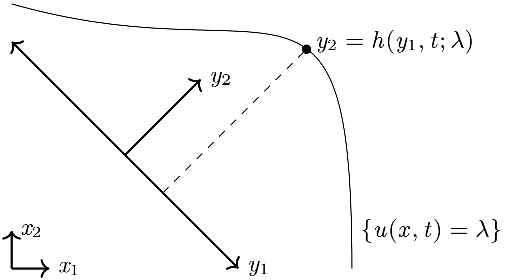


Change coordinates so $y_n = x_1 + \cdots + x_n$, $y_i = x_i - x_n$ and define h by

$$v(y_1, \ldots, y_{n-1}, h(y_1, \ldots, y_{n-1}, t; \lambda), t) = \lambda,$$

where $v(y,t) = u(x,t)$.

Underlying linear heat equation



Change coordinates so $y_n = x_1 + \cdots + x_n$, $y_i = x_i - x_n$ and define h by

$$v(y_1, \ldots, y_{n-1}, h(y_1, \ldots, y_{n-1}, t; \lambda), t) = \lambda,$$

where $v(y,t) = u(x,t)$. We find h satisfies a linear heat equation

$$h_t + \frac{1}{2^{d+1}} \sum_{m \in \{-1,1\}^d} r(m)^T \nabla^2 h \, r(m) = 0, \tag{11}$$

where $r_i(m) := q_i(m) - q_n(m)$. The condition $g \in C^4$ ensures u is smooth.

Dynamic programming principle (DPP)

Recall the value function

$$V_N(x,\ell;m) = \min_{|f_\ell|\leq 1} \max_{b_\ell=\pm 1} \cdots \min_{|f_{N-1}|\leq 1} \max_{b_{N-1}=\pm 1} g\left(x + \sum_{i=\ell}^{N-1} b_i(q(m^i) - f_i\mathbb{1})\right)$$

Dynamic programming principle (DPP)

Recall the value function

$$V_N(x,\ell;m) = \min_{|f_\ell|\leq 1}\max_{b_\ell=\pm 1}\cdots\min_{|f_{N-1}|\leq 1}\max_{b_{N-1}=\pm 1} g\left(x + \sum_{i=\ell}^{N-1} b_i(q(m^i) - f_i\mathbb{1})\right)$$

Proposition (1-Step Dynamic Programming Principle)

For $\ell \leq N-1$ *and* $m \in \{-1,1\}^d$

(12) $$V_N(x,\ell;m) = \min_{|f|\leq 1}\max_{b=\pm 1} V_N(x + b(q(m) - f\mathbb{1}), \ell+1; m|b).$$

Dynamic programming principle (DPP)

Recall the value function

$$V_N(x,\ell;m) = \min_{|f_\ell|\leq 1} \max_{b_\ell=\pm 1} \cdots \min_{|f_{N-1}|\leq 1} \max_{b_{N-1}=\pm 1} g\left(x + \sum_{i=\ell}^{N-1} b_i(q(m^i) - f_i\mathbb{1})\right)$$

Proposition (1-Step Dynamic Programming Principle)

For $\ell \leq N-1$ *and* $m \in \{-1,1\}^d$

$$V_N(x,\ell;m) = \min_{|f|\leq 1} \max_{b=\pm 1} V_N(x + b(q(m) - f\mathbb{1}), \ell+1; m|b). \tag{12}$$

Note: The DPP is a coupled set of 2^d equations.

Dynamic programming principle

Let us assume that

$$u_N(x,t;m) = \frac{1}{\sqrt{N}} V_N(\sqrt{N}x, \lceil Nt \rceil; m) \approx u(x,t),$$

for some $u \in C^3$.

Dynamic programming principle

Let us assume that

$$u_N(x,t;m) = \frac{1}{\sqrt{N}} V_N(\sqrt{N}x, \lceil Nt \rceil; m) \approx u(x,t),$$

for some $u \in C^3$. With $\varepsilon = N^{-1/2}$, the dynamic programming principle (DPP) becomes

$$u(x,t) \quad = \quad \min_{|f|\leq 1} \max_{b=\pm 1} u(x + \varepsilon b(q(m) - f\mathbb{1}), t + \varepsilon^2)$$

Dynamic programming principle

Let us assume that

$$u_N(x,t;m) = \frac{1}{\sqrt{N}} V_N(\sqrt{N}x, \lceil Nt \rceil; m) \approx u(x,t),$$

for some $u \in C^3$. With $\varepsilon = N^{-1/2}$, the dynamic programming principle (DPP) becomes

$$\begin{aligned} u(x,t) &= \min_{|f|\le 1} \max_{b=\pm 1} u(x + \varepsilon b(q(m) - f\mathbb{1}), t + \varepsilon^2) \\ &= \min_{|f|\le 1} \max_{b=\pm 1} \Big\{ u(x,t) + \varepsilon^2 u_t + \varepsilon b \nabla u^T (q(m) - f\mathbb{1}) \\ &\qquad + \frac{\varepsilon^2}{2} (q(m) - f\mathbb{1})^T \nabla^2 u \, (q(m) - f\mathbb{1}) \Big\} + O(\varepsilon^3) \end{aligned}$$

Dynamic programming principle

Let us assume that

$$u_N(x,t;m) = \frac{1}{\sqrt{N}} V_N(\sqrt{N}x, \lceil Nt \rceil; m) \approx u(x,t),$$

for some $u \in C^3$. With $\varepsilon = N^{-1/2}$, the dynamic programming principle (DPP) becomes

$$\begin{aligned} u(x,t) &= \min_{|f|\leq 1} \max_{b=\pm 1} u(x + \varepsilon b(q(m) - f\mathbb{1}), t + \varepsilon^2) \\ &= \min_{|f|\leq 1} \max_{b=\pm 1} \Big\{ u(x,t) + \varepsilon^2 u_t + \varepsilon b \nabla u^T (q(m) - f\mathbb{1}) \\ &\qquad + \frac{\varepsilon^2}{2} (q(m) - f\mathbb{1})^T \nabla^2 u \, (q(m) - f\mathbb{1}) \Big\} + O(\varepsilon^3) \end{aligned}$$

$$u_t + \min_{|f|\leq 1} \max_{b=\pm 1} \Big\{ \varepsilon^{-1} b \nabla u^T (q(m) - f\mathbb{1}) + \frac{1}{2} (q(m) - f\mathbb{1})^T \nabla^2 u \, (q(m) - f\mathbb{1}) \Big\} = O(\varepsilon).$$

Dynamic programming principle

Let us assume that

$$u_N(x,t;m) = \frac{1}{\sqrt{N}} V_N(\sqrt{N}x, \lceil Nt \rceil; m) \approx u(x,t),$$

for some $u \in C^3$. With $\varepsilon = N^{-1/2}$, the dynamic programming principle (DPP) becomes

$$\begin{aligned} u(x,t) &= \min_{|f|\le 1} \max_{b=\pm 1} u(x + \varepsilon b(q(m) - f\mathbb{1}), t + \varepsilon^2) \\ &= \min_{|f|\le 1} \max_{b=\pm 1} \Big\{ u(x,t) + \varepsilon^2 u_t + \varepsilon b \nabla u^T (q(m) - f\mathbb{1}) \\ &\qquad + \frac{\varepsilon^2}{2} (q(m) - f\mathbb{1})^T \nabla^2 u\, (q(m) - f\mathbb{1}) \Big\} + O(\varepsilon^3) \end{aligned}$$

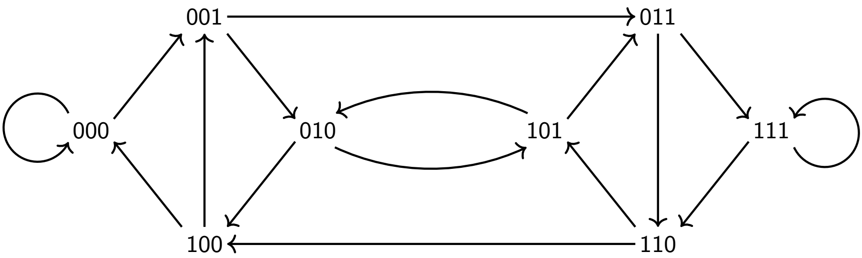
$$u_t + \min_{|f|\le 1} \max_{b=\pm 1} \Big\{ \varepsilon^{-1} b \nabla u^T (q(m) - f\mathbb{1}) + \frac{1}{2} (q(m) - f\mathbb{1})^T \nabla^2 u\, (q(m) - f\mathbb{1}) \Big\} = O(\varepsilon).$$

Investor (player) may wish to choose f to cancel out ε^{-1} term:

$$f = \frac{\nabla u^T q(m)}{\nabla u^T \mathbb{1}} \quad \text{and} \quad \boxed{u_t + \frac{1}{2} \eta(m)^T \nabla^2 u\, \eta(m) = O(\varepsilon),}$$

where $\eta(m) = q(m) - \frac{\nabla u^T q(m)}{\nabla u^T \mathbb{1}} \mathbb{1}$.

De Bruijn graph $d = 3$



Dynamic programming principle

Let us assume that

$$u_N(x,t;m) = \frac{1}{\sqrt{N}} V_N(\sqrt{N}x, \lceil Nt \rceil; m) \approx u(x,t),$$

for some $u \in C^3$. With $\varepsilon = N^{-1/2}$, the dynamic programming principle (DPP) becomes

$$\begin{aligned} u(x,t) &= \min_{|f|\leq 1} \max_{b=\pm 1} u(x+\varepsilon b(q(m)-f\mathbb{1}), t+\varepsilon^2) \\ &= \min_{|f|\leq 1} \max_{b=\pm 1} \Big\{ u(x,t) + \varepsilon^2 u_t + \varepsilon b \nabla u^T (q(m)-f\mathbb{1}) \\ &\qquad + \frac{\varepsilon^2}{2}(q(m)-f\mathbb{1})^T \nabla^2 u\,(q(m)-f\mathbb{1}) \Big\} + O(\varepsilon^3) \end{aligned}$$

$$u_t + \min_{|f|\leq 1} \max_{b=\pm 1} \Big\{ \varepsilon^{-1} b \nabla u^T (q(m)-f\mathbb{1}) + \frac{1}{2}(q(m)-f\mathbb{1})^T \nabla^2 u\,(q(m)-f\mathbb{1}) \Big\} = O(\varepsilon).$$

Investor (player) may wish to choose f to cancel out ε^{-1} term:

$$f = \frac{\nabla u^T q(m) + \varepsilon f^{\#}(m)}{\nabla u^T \mathbb{1}} \quad \text{and} \quad \boxed{u_t + \frac{1}{2}\eta(m)^T \nabla^2 u\,\eta(m) - b f^{\#}(m) = O(\varepsilon),}$$

where $\eta(m) = q(m) - \frac{\nabla u^T q(m)}{\nabla u^T \mathbb{1}}\mathbb{1}$. [Drenska and Kohn, 2019a]

k-step Dynamic Programming Principle

Proposition (Dynamic Programming Principle)

For any $N \geq 1$, $x \in \mathbb{R}^n$, $m \in \mathcal{B}^d$, $k \geq 1$ and $\ell \leq N-k$ it holds that

$$V_N(x,\ell;m) = \min_{|f_1|\leq 1} \max_{b_1=\pm 1} \cdots \min_{|f_k|\leq 1} \max_{b_k=\pm 1} V_N\left(x + \sum_{i=1}^{k} b_i(q(m^i) - \mathbb{1} f_i), \ell + k; m^{k+1}\right),$$

where $m^1 = m$ and $m^{i+1} = m^i | b_i$ for $i = 1, \ldots, k$.

k-step Dynamic Programming Principle

Proposition (Dynamic Programming Principle)

For any $N \geq 1$, $x \in \mathbb{R}^n$, $m \in \mathcal{B}^d$, $k \geq 1$ *and* $\ell \leq N - k$ *it holds that*

$$V_N(x, \ell; m) = \min_{|f_1| \leq 1} \max_{b_1 = \pm 1} \cdots \min_{|f_k| \leq 1} \max_{b_k = \pm 1} V_N\left(x + \sum_{i=1}^{k} b_i(q(m^i) - \mathbb{1} f_i), \ell + k; m^{k+1}\right),$$

where $m^1 = m$ *and* $m^{i+1} = m^i | b_i$ *for* $i = 1, \ldots, k$.

The equivalent DPP for u_N is

$$u_N(x, t; m) = \min_{|f_1| \leq 1} \max_{b_1 = \pm 1} \cdots \min_{|f_k| \leq 1} \max_{b_k = \pm 1} u_N\left(x + \varepsilon \sum_{i=1}^{k} b_i(q(m^i) - \mathbb{1} f_i), t + \varepsilon^2 k; m^{k+1}\right).$$

The local problem

Assume $u_N(x, t; m) \approx u(x, t)$ for smooth u.

The local problem

Assume $u_N(x,t;m) \approx u(x,t)$ for smooth u. Then

$$u(x,t) = \min_{|f_1|\leq 1} \max_{b_1=\pm 1} \cdots \min_{|f_k|\leq 1} \max_{b_k=\pm 1} \left\{ u(x+\varepsilon\Delta x, t+k\varepsilon^2) \right\}$$

The local problem

Assume $u_N(x,t;m) \approx u(x,t)$ for smooth u. Then

$$\begin{aligned} u(x,t) &= \min_{|f_1|\le 1} \max_{b_1=\pm 1} \cdots \min_{|f_k|\le 1} \max_{b_k=\pm 1} \left\{ u(x+\varepsilon\Delta x, t+k\varepsilon^2) \right\} \\ &\approx \min_{|f_1|\le 1} \max_{b_1=\pm 1} \cdots \min_{|f_k|\le 1} \max_{b_k=\pm 1} \left\{ u + k\varepsilon^2 u_t + \varepsilon \nabla u^T \Delta x + \frac{\varepsilon^2}{2} \Delta x^T \nabla^2 u \Delta x \right\}, \end{aligned}$$

The local problem

Assume $u_N(x,t;m) \approx u(x,t)$ for smooth u. Then

$$\begin{aligned} u(x,t) &= \min_{|f_1|\le 1} \max_{b_1=\pm 1} \cdots \min_{|f_k|\le 1} \max_{b_k=\pm 1} \left\{ u(x+\varepsilon\Delta x, t+k\varepsilon^2) \right\} \\ &\approx \min_{|f_1|\le 1} \max_{b_1=\pm 1} \cdots \min_{|f_k|\le 1} \max_{b_k=\pm 1} \left\{ u + k\varepsilon^2 u_t + \varepsilon \nabla u^T \Delta x + \frac{\varepsilon^2}{2} \Delta x^T \nabla^2 u \Delta x \right\}, \end{aligned}$$

and so

$$u_t + \frac{1}{k} \min_{|f_1|\le 1} \max_{b_1=\pm 1} \cdots \min_{|f_k|\le 1} \max_{b_k=\pm 1} \left\{ \varepsilon^{-1} \nabla u^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 u \Delta x \right\} \approx 0.$$

The local problem

Assume $u_N(x,t;m) \approx u(x,t)$ for smooth u. Then

$$u(x,t) = \min_{|f_1|\le 1} \max_{b_1=\pm 1} \cdots \min_{|f_k|\le 1} \max_{b_k=\pm 1} \left\{ u(x+\varepsilon\Delta x, t+k\varepsilon^2) \right\}$$
$$\approx \min_{|f_1|\le 1} \max_{b_1=\pm 1} \cdots \min_{|f_k|\le 1} \max_{b_k=\pm 1} \left\{ u + k\varepsilon^2 u_t + \varepsilon \nabla u^T \Delta x + \frac{\varepsilon^2}{2}\Delta x^T \nabla^2 u \Delta x \right\},$$

and so

$$u_t + \frac{1}{k} \min_{|f_1|\le 1} \max_{b_1=\pm 1} \cdots \min_{|f_k|\le 1} \max_{b_k=\pm 1} \left\{ \varepsilon^{-1} \nabla u^T \Delta x + \frac{1}{2}\Delta x^T \nabla^2 u \Delta x \right\} \approx 0.$$

Definition (Local Problem)

The local problem is defined by

$$\mathcal{L}(\varepsilon, k, X, p, m) := \min_{|f_1|\le 1} \max_{b_1=\pm 1} \cdots \min_{|f_k|\le 1} \max_{b_k=\pm 1} \left\{ \varepsilon^{-1} p^T \Delta x + \frac{1}{2}\Delta x^T X \Delta x \right\}$$

where $m_1 = m$, $m_{i+1} = m_i | b_i$, and $\Delta x := \sum_{i=1}^k b_i(q(m_i) - \mathbb{1} f_i)$.

The local problem

Theorem (Local problem)

Let $X \in \mathbb{S}(n)$, $p \in (0,\infty)^n$, $m \in \mathcal{B}^d$, $k \geq d+1$, $\varepsilon > 0$, *and set* $\gamma_p = \min_{1\leq i\leq n} p_i$. *Then there exists* $C, c > 0$, *depending only on* n, *such that whenever* $\|X\|k\varepsilon \leq c\,\vartheta_q\gamma_p$ *we have*

$$\left|\frac{1}{k}\mathcal{L}_{k,\varepsilon}(X,p,m) - \frac{1}{2^{d+1}}\sum_{m\in\mathcal{B}^d}\eta(m)^T X\eta(m)\right| \leq C\|X\|\left(\frac{d}{k} + \|X\|\gamma_p^{-1}k\varepsilon\right). \tag{13}$$

Drenska, N., and Calder J. Online Prediction With History-Dependent Experts: The General Case. To appear in Communications on Pure and Applied Mathematics (CPAM), (2021).

Back to the dynamic programming principle

With $\varepsilon = N^{-1/2}$, the dynamic programming principle (DPP) becomes

$$u_t + \min_{|f|\leq 1} \max_{b=\pm 1} \left\{ \varepsilon^{-1} b \nabla u^T (q(m) - f\mathbb{1}) + \frac{1}{2}(q(m) - f\mathbb{1})^T \nabla^2 u \, (q(m) - f\mathbb{1}) \right\} = O(\varepsilon).$$

Investor (player) can choose a strategy of the form

$$f = \frac{\nabla u^T q(m) + \frac{\varepsilon}{2} f^{\#}(m)}{\nabla u^T \mathbb{1}} \quad \text{and} \quad \boxed{u_t + h(m) - \frac{b(m)}{2} f^{\#}(m) = O(\varepsilon),}$$

where $\eta(m) = q(m) - \frac{\nabla u^T q(m)}{\nabla u^T \mathbb{1}} \mathbb{1}$ and $h(m) = \frac{1}{2}\eta(m)^T \nabla^2 u \, \eta(m)$.

Back to the dynamic programming principle

With $\varepsilon = N^{-1/2}$, the dynamic programming principle (DPP) becomes

$$u_t + \min_{|f|\leq 1} \max_{b=\pm 1} \left\{ \varepsilon^{-1} b \nabla u^T (q(m) - f\mathbb{1}) + \frac{1}{2}(q(m) - f\mathbb{1})^T \nabla^2 u \, (q(m) - f\mathbb{1}) \right\} = O(\varepsilon).$$

Investor (player) can choose a strategy of the form

$$f = \frac{\nabla u^T q(m) + \frac{\varepsilon}{2} f^{\#}(m)}{\nabla u^T \mathbb{1}} \quad \text{and} \quad \boxed{u_t + h(m) - \frac{b(m)}{2} f^{\#}(m) = O(\varepsilon),}$$

where $\eta(m) = q(m) - \frac{\nabla u^T q(m)}{\nabla u^T \mathbb{1}} \mathbb{1}$ and $h(m) = \frac{1}{2}\eta(m)^T \nabla^2 u \, \eta(m)$.

Question: How to choose $f^{\#}(m)$ so the equation averages out to

$$u_t + (h)_{\mathcal{B}^d} = 0 \quad \text{where } (h)_{\mathcal{B}^d} := \frac{1}{2^d} \sum_{m \in \mathcal{B}^d} h(m)$$

over many steps?

Optimal investor strategy

Why not choose $f^{\#}(m)$ so that

$$h(m) - \frac{b(m)}{2} f^{\#}(m) = (h)_{\mathcal{B}^d}?$$

Optimal investor strategy

Why not choose $f^{\#}(m)$ so that

$$h(m) - \frac{b(m)}{2} f^{\#}(m) = (h)_{\mathcal{B}^d}?$$

This would violate the rules, since $f^{\#} = \frac{2}{b(m)}(h(m) - (h))$ depends on b.

Optimal investor strategy

It turns out a small correction on this choice is possible. We choose $f^{\#}(m)$ to satisfy

$$h(m) - \frac{b(m)}{2} f^{\#}(m) = (h)_{\mathcal{B}^d} + \mathcal{H}(m) - \mathcal{H}(m|b(m)),$$

for a potential $\mathcal{H}$ to be determined.

Optimal investor strategy

It turns out a small correction on this choice is possible. We choose $f^{\#}(m)$ to satisfy

$$h(m) - \frac{b(m)}{2} f^{\#}(m) = (h)_{\mathcal{B}^d} + \mathcal{H}(m) - \mathcal{H}(m|b(m)),$$

for a potential $\mathcal{H}$ to be determined. Solving for $f^{\#}$ we have

$$f^{\#} = 2b\left[h(m) - (h)_{\mathcal{B}^d} + \mathcal{H}(m|b) - \mathcal{H}(m)\right].$$

Optimal investor strategy

It turns out a small correction on this choice is possible. We choose $f^{\#}(m)$ to satisfy

$$h(m) - \frac{b(m)}{2} f^{\#}(m) = (h)_{\mathcal{B}^d} + \mathcal{H}(m) - \mathcal{H}(m|b(m)),$$

for a potential $\mathcal{H}$ to be determined. Solving for $f^{\#}$ we have

$$f^{\#} = 2b\left[h(m) - (h)_{\mathcal{B}^d} + \mathcal{H}(m|b) - \mathcal{H}(m)\right].$$

Introducing the De Bruijn graph Laplacian

$$\Delta_{\mathcal{B}^d}\mathcal{H}(m) = \mathcal{H}(m) - \frac{1}{2}\mathcal{H}(m_+) - \frac{1}{2}\mathcal{H}(m_-),$$

where $m_\pm = m|\pm 1$, we can write

$$f^{\#} = 2b\Big[h(m) - (h)_{\mathcal{B}^d} - \Delta_{\mathcal{B}^d}\mathcal{H}(m)\Big] + b\left(\mathcal{H}(m|b) - \mathcal{H}(m|-b)\right).$$

Optimal investor strategy

It turns out a small correction on this choice is possible. We choose $f^{\#}(m)$ to satisfy

$$h(m) - \frac{b(m)}{2} f^{\#}(m) = (h)_{\mathcal{B}^d} + \mathcal{H}(m) - \mathcal{H}(m|b(m)),$$

for a potential $\mathcal{H}$ to be determined. Solving for $f^{\#}$ we have

$$f^{\#} = 2b\left[h(m) - (h)_{\mathcal{B}^d} + \mathcal{H}(m|b) - \mathcal{H}(m)\right].$$

Introducing the De Bruijn graph Laplacian

$$\Delta_{\mathcal{B}^d}\mathcal{H}(m) = \mathcal{H}(m) - \frac{1}{2}\mathcal{H}(m_+) - \frac{1}{2}\mathcal{H}(m_-),$$

where $m_\pm = m|\pm 1$, we can write

$$f^{\#} = 2b\Big[h(m) - (h)_{\mathcal{B}^d} - \Delta_{\mathcal{B}^d}\mathcal{H}(m)\Big] + b\left(\mathcal{H}(m|b) - \mathcal{H}(m|-b)\right).$$

If $\Delta_{\mathcal{B}^d}\mathcal{H}(m) = h(m) - (h)_{\mathcal{B}^d}$ then

$$f^{\#} = b\left(\mathcal{H}(m|b) - \mathcal{H}(m|-b)\right) = \mathcal{H}(m_+) - \mathcal{H}(m_-).$$

Poisson equation

The equation

$$\Delta_{\mathcal{B}^d}\mathcal{H} = h - (h)_{\mathcal{B}^d}$$

is a Poisson equation over the De Bruijn graph.

Poisson equation

The equation

$$\Delta_{\mathcal{B}^d}\mathcal{H} = h - (h)_{\mathcal{B}^d}$$

is a Poisson equation over the De Bruijn graph. The solution is given by

$$\mathcal{H}(m) = h(m) + \sum_{\ell=1}^{d-1} \frac{1}{2^\ell} \sum_{s\in\mathcal{B}^\ell} h(m|s).$$

Poisson equation

The equation

$$\Delta_{\mathcal{B}^d}\mathcal{H} = h - (h)_{\mathcal{B}^d}$$

is a Poisson equation over the De Bruijn graph. The solution is given by

$$\mathcal{H}(m) = h(m) + \sum_{\ell=1}^{d-1} \frac{1}{2^\ell} \sum_{s\in\mathcal{B}^\ell} h(m|s).$$

The solution is unique up to an additive constant, and the optimal strategy

$$f^{\#} = \mathcal{H}(m_+) - \mathcal{H}(m_-)$$

is clearly independent of this constant.

Poisson equation

The equation

$$\Delta_{\mathcal{B}^d}\mathcal{H} = h - (h)_{\mathcal{B}^d}$$

is a Poisson equation over the De Bruijn graph. The solution is given by

$$\mathcal{H}(m) = h(m) + \sum_{\ell=1}^{d-1} \frac{1}{2^\ell} \sum_{s\in\mathcal{B}^\ell} h(m|s).$$

The solution is unique up to an additive constant, and the optimal strategy

$$f^{\#} = \mathcal{H}(m_+) - \mathcal{H}(m_-)$$

is clearly independent of this constant.

It is possible to extend these ideas slightly to other directed graphs.

Calder, J., and Drenska, N. Asymptotically optimal strategies for online prediction with history-dependent experts. Journal of Fourier Analysis and Applications 27.2 (2021): 1-20.

Outline

Future work

1. Numerical schemes for solving the PDE and computing optimal strategies.

Future work

1. Numerical schemes for solving the PDE and computing optimal strategies.
2. Generalizations to other games (e.g., Markov Decision Processes in adversarial settings).

Future work

1. Numerical schemes for solving the PDE and computing optimal strategies.
2. Generalizations to other games (e.g., Markov Decision Processes in adversarial settings).
3. Prediction with mixed (randomized) strategies.

References:

Drenska, N., and Calder J. Online Prediction With History-Dependent Experts: The General Case. To appear in Communications on Pure and Applied Mathematics (CPAM), (2021).

Calder, J., and Drenska, N. Asymptotically optimal strategies for online prediction with history-dependent experts. Journal of Fourier Analysis and Applications 27.2 (2021): 1-20.

Outline

Abbasi, Y., Bartlett, P. L., and Gabillon, V. (2017).
Near minimax optimal players for the finite-time 3-expert prediction problem.
In *Advances in Neural Information Processing Systems*, pages 3033–3042.

Bayraktar, E., Ekren, I., and Zhang, Y. (2019).
On the asymptotic optimality of the comb strategy for prediction with expert advice.
arXiv preprint arXiv:1902.02368.

Cesa-Bianchi, N. and Lugosi, G. (2006).
Prediction, learning, and games.
Cambridge university press.

Cover, T. M. (1966).
Behavior of sequential predictors of binary sequences.
Technical report, STANFORD UNIV CALIF STANFORD ELECTRONICS LABS.

Drenska, N. (2017).
A PDE Approach to a Prediction Problem Involving Randomized Strategies.
PhD thesis, New York University.

Drenska, N. and Kohn, R. V. (2019a).
A pde approach to the stock prediction problem with two history-dependent experts.
Preprint.

Drenska, N. and Kohn, R. V. (2019b).
Prediction with expert advice: a pde perspective.
arXiv preprint arXiv:1904.11401.

Gravin, N., Peres, Y., and Sivan, B. (2016).
Towards optimal algorithms for prediction with expert advice.
In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 528–547. SIAM.

Littlestone, N. and Warmuth, M. K. (1994).
The weighted majority algorithm.
Information and computation, 108(2):212–261.

Vovk, V. G. (1990).
Aggregating strategies.
Proc. of Computational Learning Theory, 1990.

Zhu, K. (2014).
Two problems in applications of PDE.
PhD thesis, New York University.