Machine Learning: Social Values, Data Efficiency, and Beyond Prediction

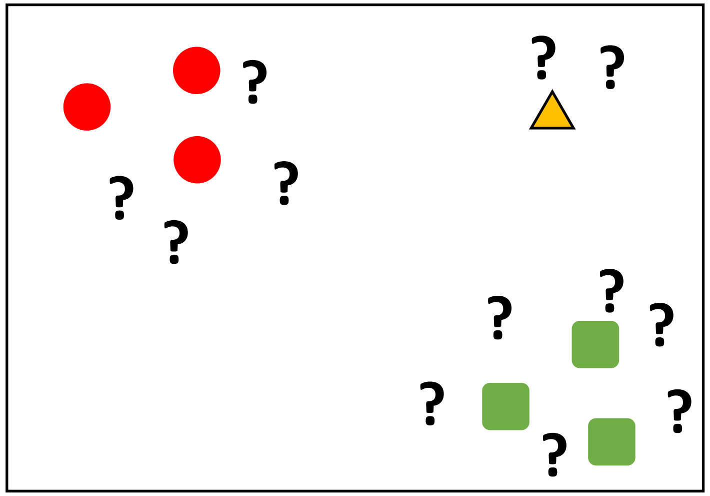
Travis Dick

Jan. 29, 2019

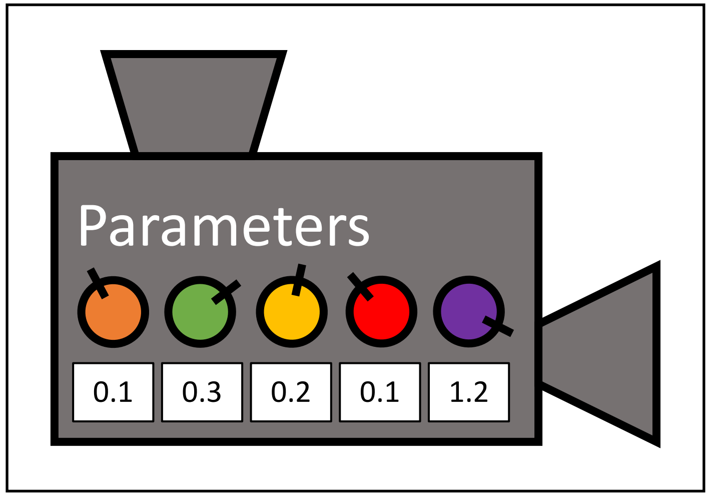
The Goal of my Research

Extend the theory and practice of machine learning to accommodate new requirements emerging from its use in real-world contexts.

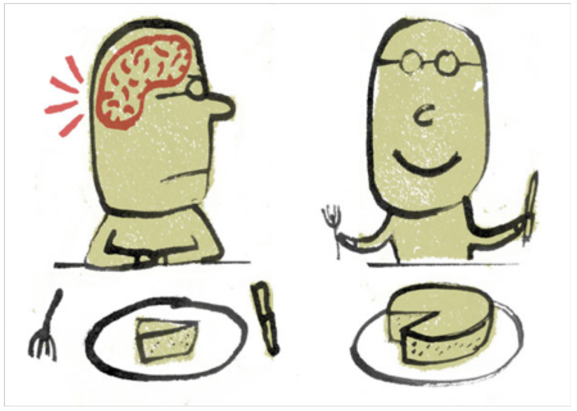
Data Efficiency



Beyond Prediction

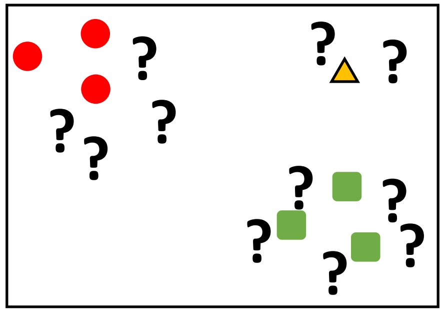


Social Values



The Goal of My Research

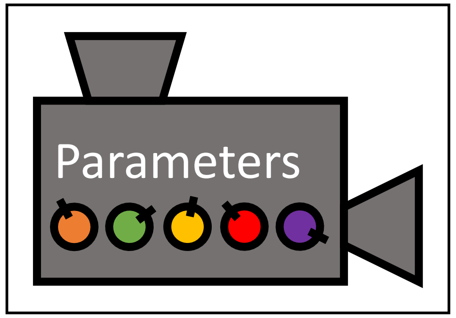
Data Efficiency



- There is disparity in the cost and availability of different types of data.
- Need to make the best use of cheap and abundant data.
- We focus on using unlabeled data in multi-class classification.

[Balcan, Dick, Mansour; AAAI 2017]

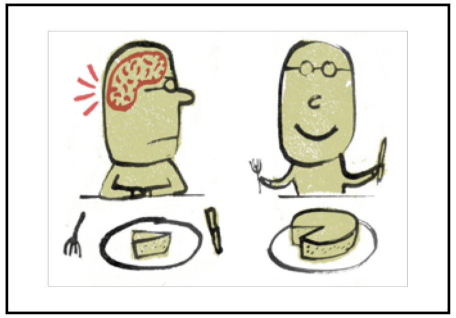
Beyond Prediction



- Many learners learn to make predictions (e.g., medical diagnosis).
- Important settings where learner's output is not a prediction rule.
- We'll look at algorithm configuration / parameter tuning.

[Balcan, Dick, Vitercik; FOCS 2018]
[Balcan, Dick, Vitercik; ICML 2018]
[Balcan, Dick, White; NeurIPS 2018]

Social Values



- Learning systems now regularly interact with us.
- Learning from personal data, making predictions about our behavior.
- We want these systems to uphold our social values (e.g. privacy, fairness)

[Balcan, Dick, Liang, Mou, Zhang; ICML 2017]
[Balcan, Dick, Procaccia, Noothigattu; 2019]

Outline

1. Online and Private Optimization of Piecewise Lipschitz Functions

Joint with Nina Balcan and Ellen Vitercik

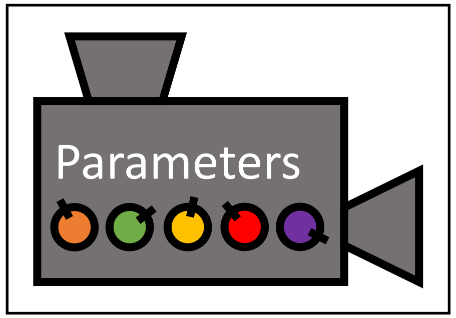
My related work on **algorithm configuration**.

Joint with Nina Balcan, Colin White, and Ellen Vitercik

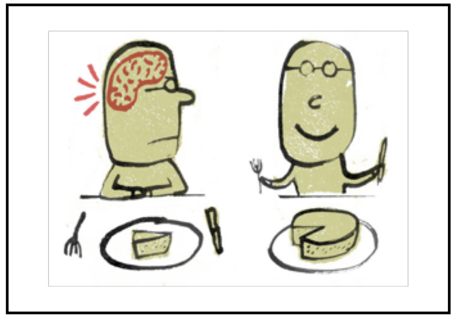
2. Envy-free Classification

Joint with Nina Balcan, Ariel Procaccia, Ritesh Noothigattu

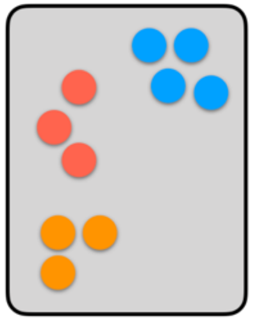
Beyond Prediction



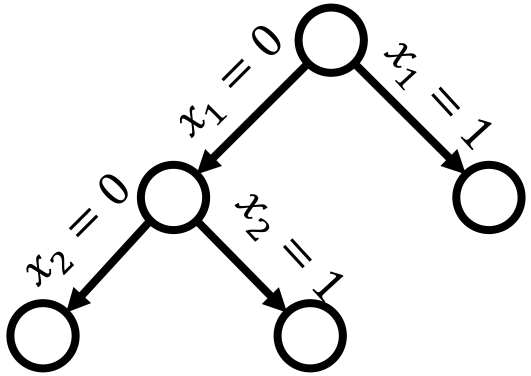
Social Values



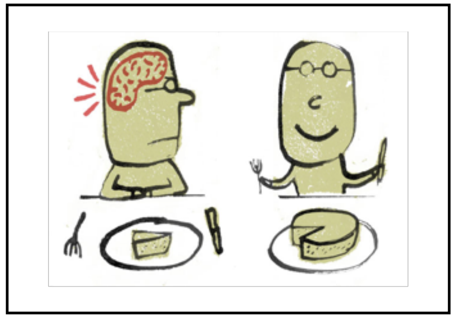
Clustering



Branch & Bound

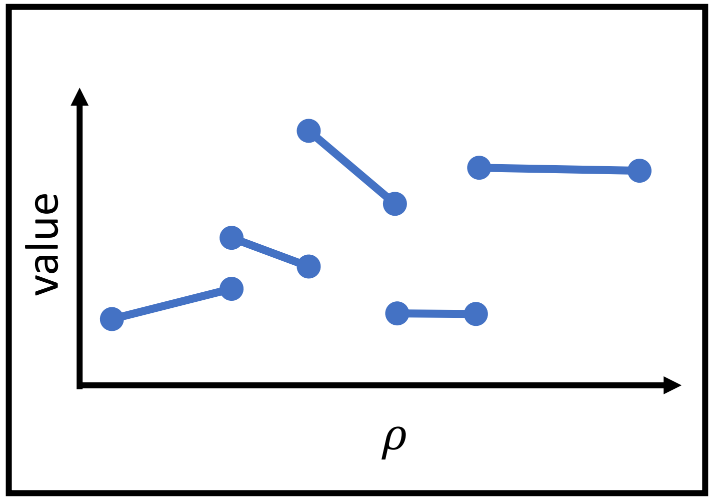


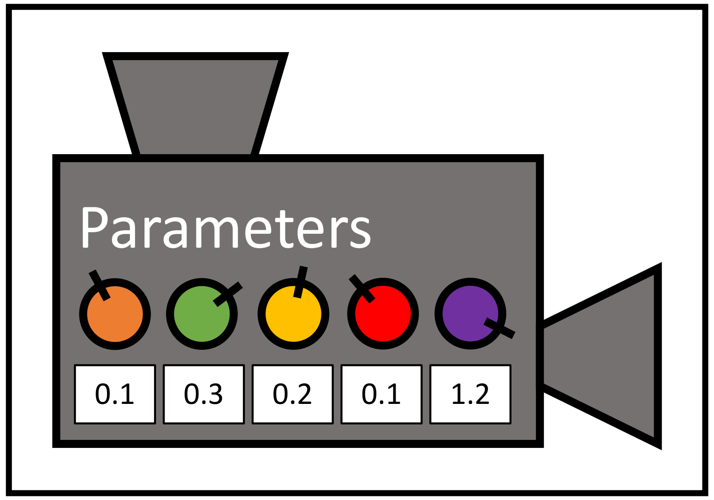
Social Values



Part 1:
Piecewise Lipschitz Optimization

Joint work with Nina Balcan and Ellen Vitercik [FOCS 2018]

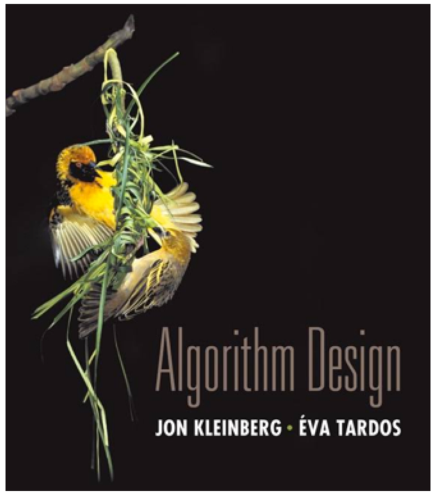


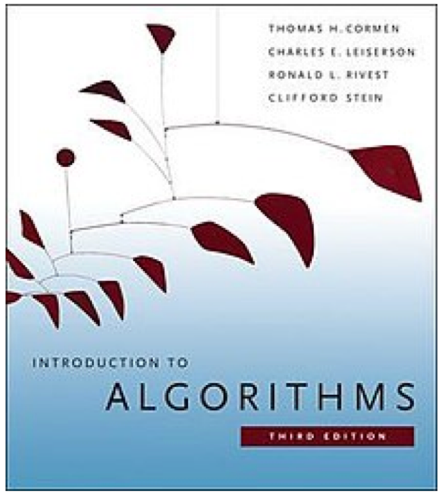


Motivation: Data-driven Algorithm Configuration

Classic Algorithm Design: Design algorithms for the worst-case.

- Some domains have always-efficient optimal algorithms
- Many important domains do not:
    - Clustering, subset selection, auction design, etc.





Data Driven Algorithm Design: Use learning and data to design/fine-tune algorithms

- Suitable when we will repeatedly solve problems from the same domain.
- We will see that *piecewise Lipschitz optimization* arises naturally in this setting.

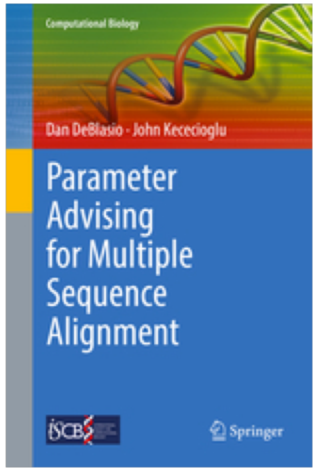
Motivation: Data-driven Algorithm Configuration

Data Driven Algorithm Design Approach:

- Design a large parameterized family of methods.
- Different methods will work better for different settings.
- Learn the best method/parameters for a specific application.

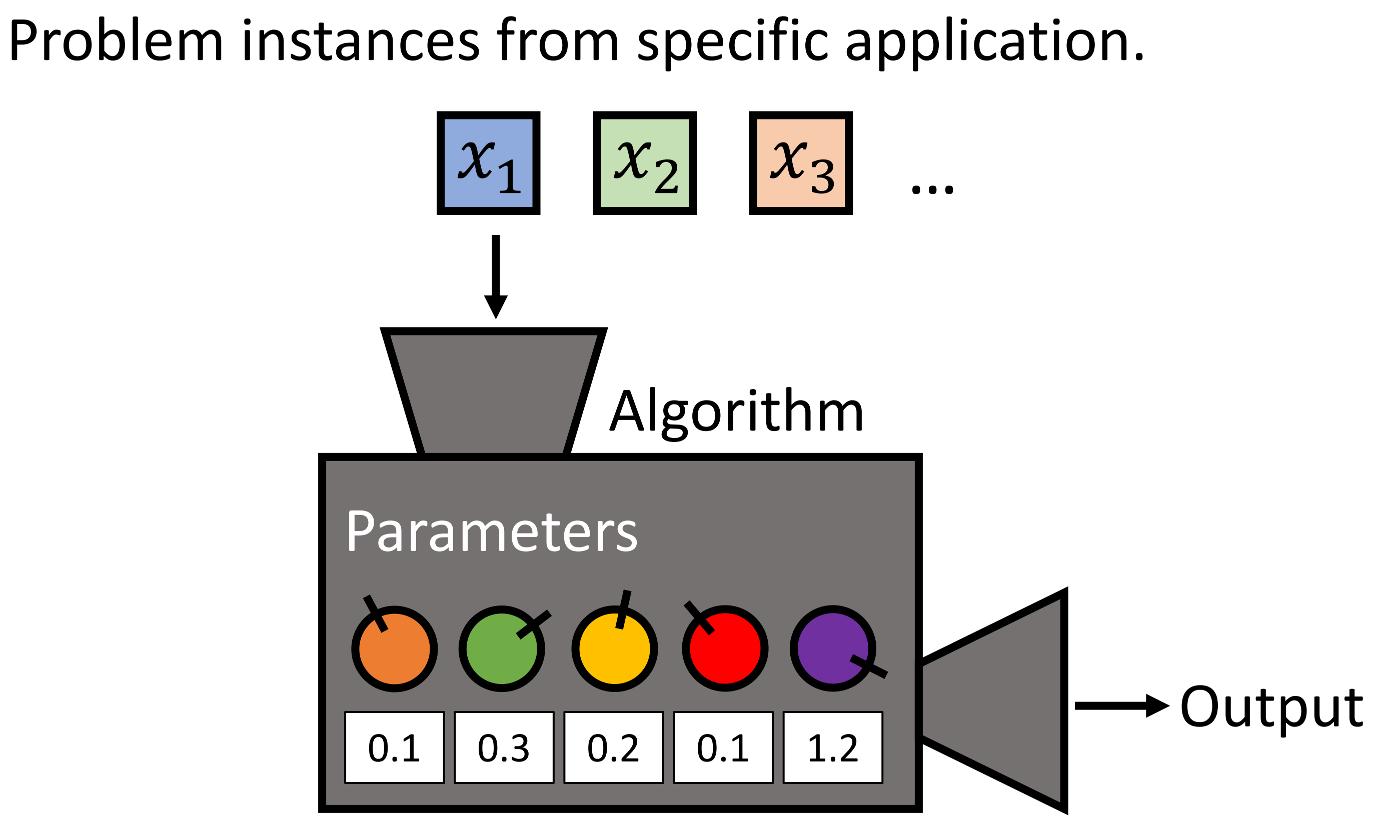
Prior Work is Mostly Empirical:

- Artificial Intelligence: E.g. [Xu-Hutter-Hoos-LeytonBrown, JAIR 2008]
- Computational Biology: E.g. [DeBlasio-Kececioglu, 2018]
- Game Theory: E.g. [Likhodedov and Sandholm, 2004]


Computational Biology
Dan DeBlasio · John Kececioglu
Parameter Advising for Multiple Sequence Alignment
ISCB
Springer

This Talk: Procedures with formal guarantees

Data-Driven Algorithm Configuration



Goal:

- Automatically find the best parameters for a specific application domain.
- Algorithm is run repeatedly, historic instances are training data.
- Want provable guarantees for online and private settings.

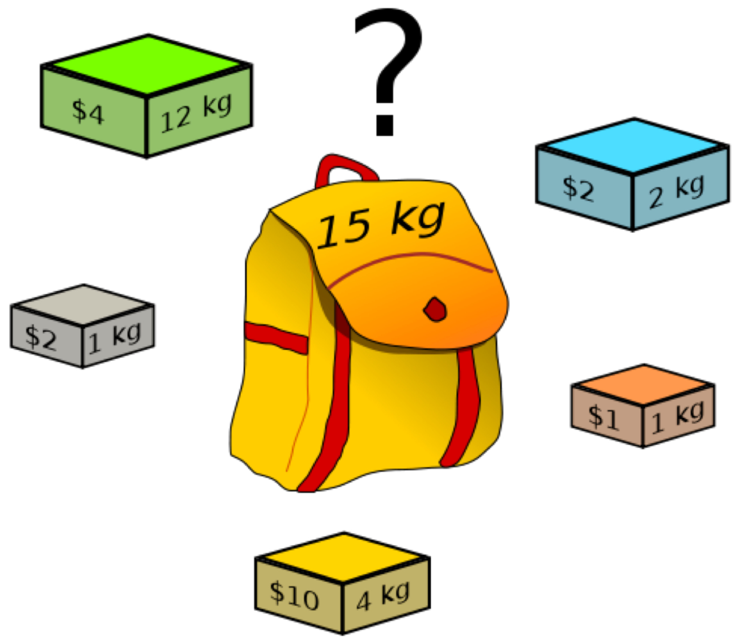
Example: Greedy Knapsack Algorithm

[Gupta & Roughgarden '16]

Problem Instance:

- Given n items
- item i has value v_i and size s_i
- a knapsack with capacity K

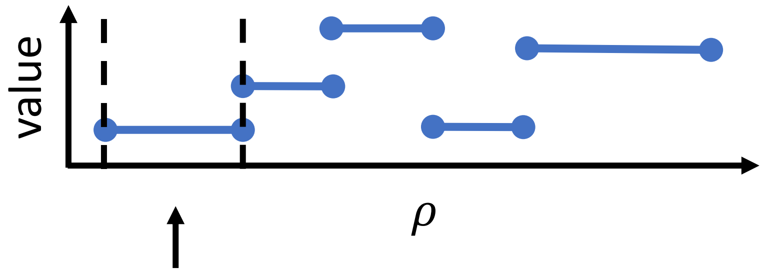
Find the most valuable subset of items that fits.



Algorithm: (Parameter $\rho \geq 0$)

Add items in decreasing order of $\text{score}_\rho(i) = v_i / s_i^\rho$.

Observation: For one instance, total value is piecewise constant in ρ.



Order of items fixed.

Online Piecewise Lipschitz Optimization

More generally: utility is a piecewise Lipschitz function of parameters.

Learning protocol:

For each round $t = 1, \dots, T$:

1. Learner chooses point $\rho_t \in C \subset R^d$.
2. Adversary chooses piecewise L-Lipschitz function $u_t: C \to R$.
3. Learner gets reward $u_t(\rho_t)$
4. **Full information:** Learner observes entire function u_t
5. **Bandit information:** Learner only observes the scalar $u_t(\rho_t)$

Goal: Minimize regret $= \max_{\rho \in C} \sum_{t=1}^{T} u_t(\rho) - \sum_{t=1}^{T} u_t(\rho_t)$.

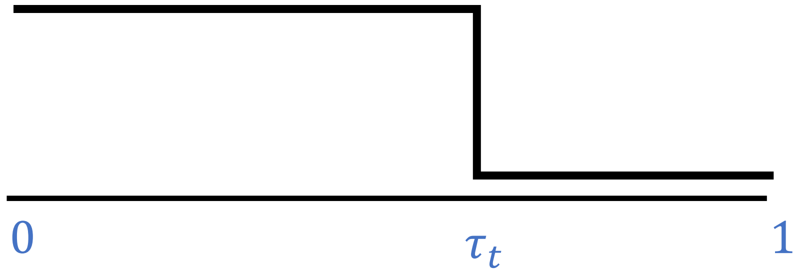
Meaningful Learning: Regret sublinear in $T \rightarrow$ optimal average per round utility

Main Challenge: Utility functions have discontinuities – can't use existing techniques.

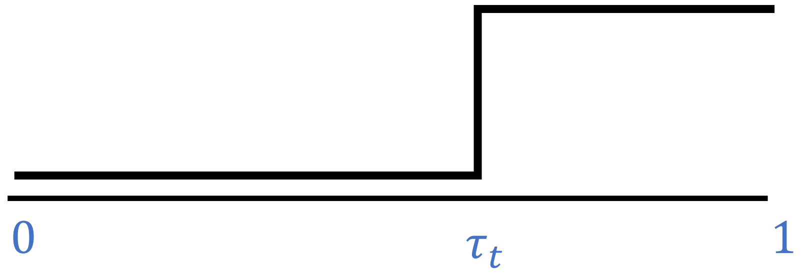
A Mean Adversary

Fact: There exists an adversary choosing piecewise constant functions from $[0,1]$ to $[0,1]$ such that **every** full information online algorithm has **linear regret**.

At round t, adversary chooses a threshold τ_t and flips a coin to choose either



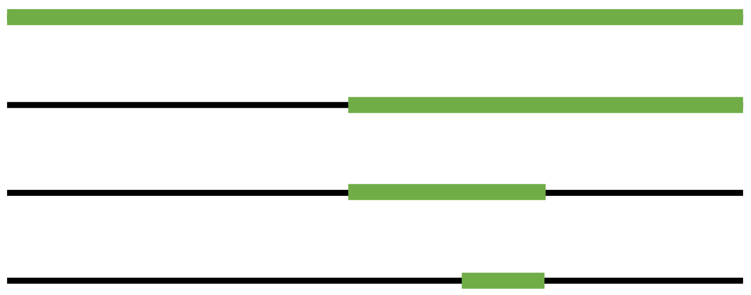
or



Every learner has expected utility of ½ per round → expected total utility $T/2$.

Let $G_t = \{\rho \in [0,1] : u_s(\rho) = 1 \text{ for all } 1 \le s \le t\}$

Set τ_t to be midpoint of G_{t-1} → $\max_T U_T(\rho) = T$.

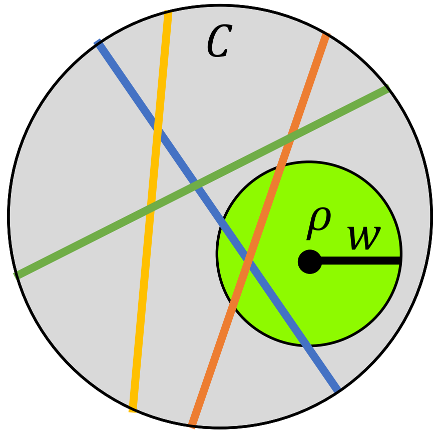


Regret = $T/2$.

Dispersion

The mean adversary concentrated discontinuities near ρ^*. Even very near points had low utility!

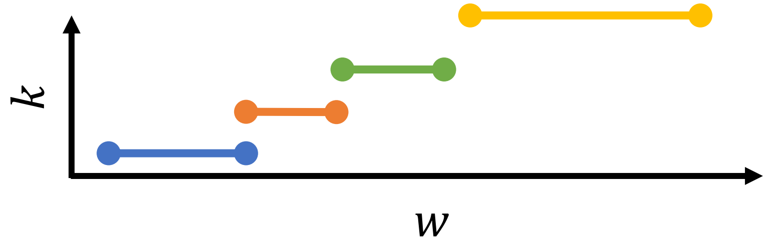
Def: Functions $u_1(\cdot), \dots, u_T(\cdot)$ are $(\boldsymbol{w}, \boldsymbol{k})$***-dispersed at point*** $\boldsymbol{\rho}$ if the ℓ_2-ball $B(\rho, w)$ contains discontinuities for at most k of $u_1 \dots, u_T$.



- 4 functions u_1, u_2, u_3, u_4
- Each colored line is a discontinuity of one function.
- Ball of radius w about ρ contains 2 discontinuities.

→ $(w, 2)$-dispersed at ρ.

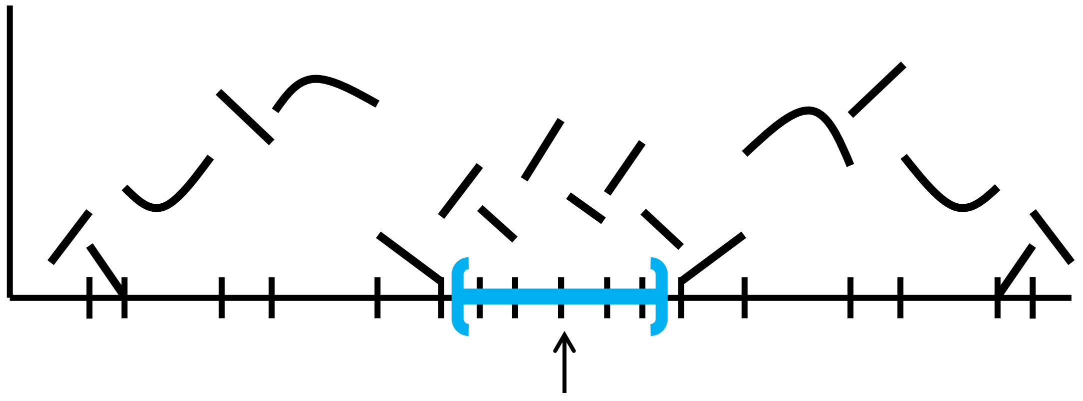
Functions will satisfy a range of dispersion parameters:



The Sum of Disperse Functions

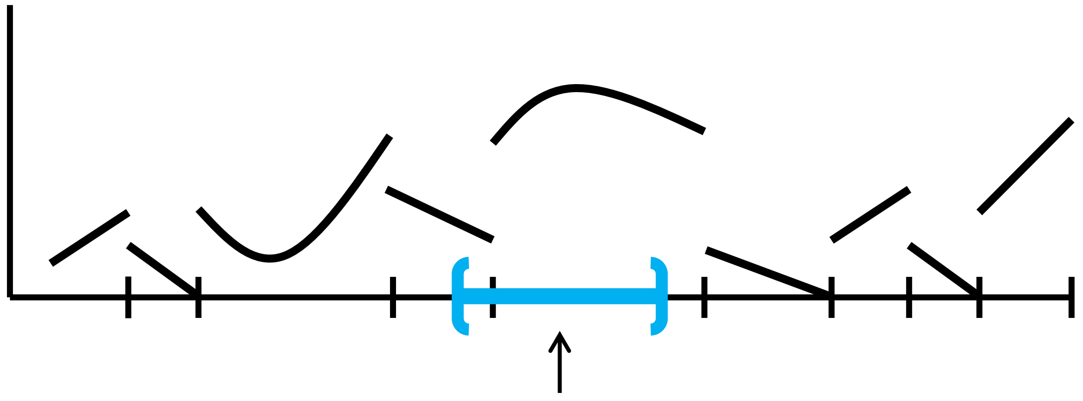
Let $u_1, \dots, u_T$ be PWL functions and plot their sum $\sum_t u_t$

Not disperse



Many boundaries within interval

Disperse



Few boundaries within interval

Full Information Regret Bounds

We analyze the classic Exponentially Weighted Forecaster *[Cesa-Bianchi & Lugosi '06]*

Algorithm: (Parameter $\lambda > 0$)
At round t, sample ρ_t from $p_t(\rho) \propto \exp(\lambda \sum_{s=1}^{t-1} u_s(\rho))$.

Theorem: If $u_1, \dots, u_T : C \to [0,1]$ are piecewise L-Lipschitz and (w,k)-dispersed at ρ^*,

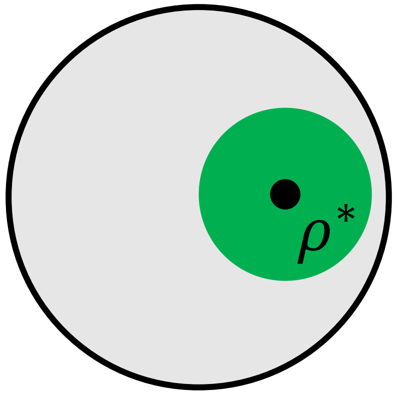
EWF has regret $O\left(\sqrt{Td \log \frac{1}{w}} + TLw + k\right)$.

When is this a good bound? For $w = 1/(L\sqrt{T})$ and $k = \tilde{O}(\sqrt{T})$ regret is $\tilde{O}(\sqrt{Td})$.

(We essentially achieve these values in all of our applications)

Intuition:

- The ball $B(\rho^*, w)$ has utility at least $OPT - TLw - k$.
- EWF can compete with $B(\rho^*, w)$ when its volume is non-negligible.



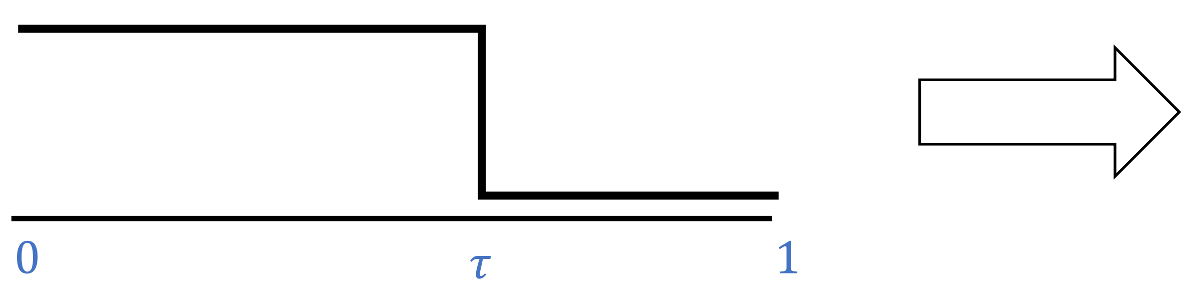
Note: Don't need to know (w,k) to run algorithm.

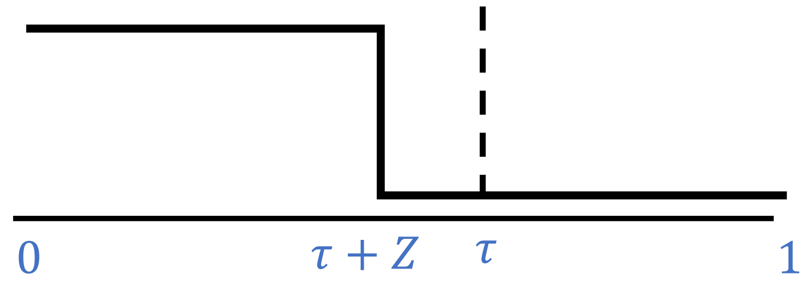
Smoothed Adversaries and Dispersion

Consider any adversary chooses threshold functions $u_1, \dots, u_T: [0,1] \to [0,1]$:

Location $\tau \in [0,1]$
Orientation $s \in \{\pm 1\}$

Location τ corrupted by adding $Z \sim N(0, \sigma^2)$.



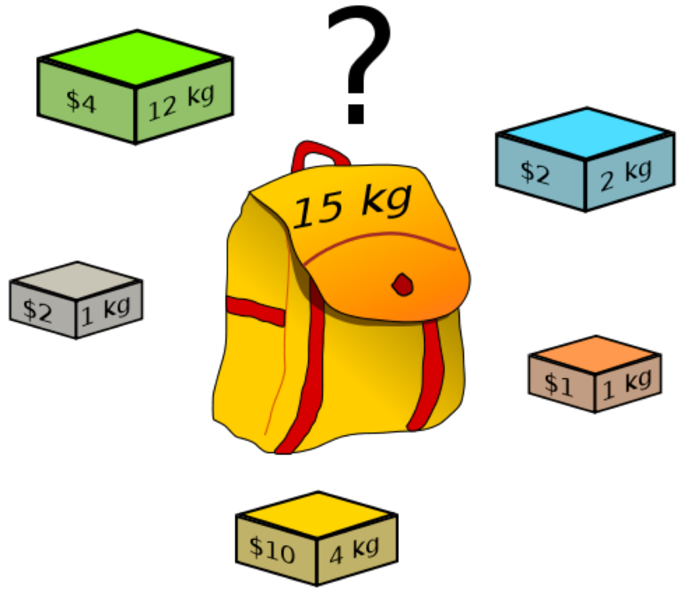


Lemma: For any $w > 0$, the functions $u_1, \dots, u_T$ are (w, k)-dispersed for $k = \tilde{O}\left(\frac{Tw}{\sigma} + \sqrt{T}\right)$ w.h.p. For any $\alpha \geq \frac{1}{2}$, we can take $w = T^{\alpha - 1}\sigma$ and $k = \tilde{O}(T^{\alpha})$.

Fix any interval $I = [a, a + w]$.
Expected number of discontinuities in I is at most $T \cdot w / (\sigma\sqrt{2\pi})$.
Uniform convergence → all width w intervals have $k = \tilde{O}(Tw/\sigma + \sqrt{T})$ discontinuities w.h.p.

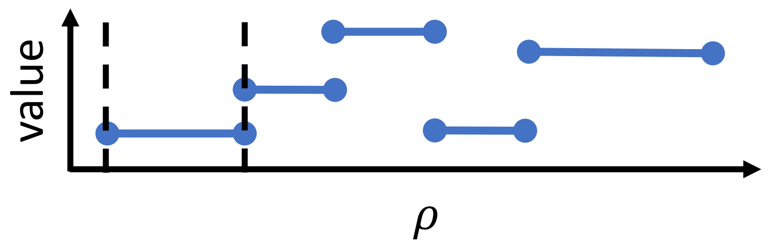
Smoothed Adversaries and Dispersion

More generally: adversary is unable to precisely pick some problem parameters (e.g. item values in knapsack).



Challenges:

- Each utility function has multiple dependent discontinuities.
- Distribution of discontinuity location depends on setting.
- How do we generalize to multiple dimensions?



$$\rho = \frac{\ln(v_i/v_j)}{\ln(s_i/s_j)}$$

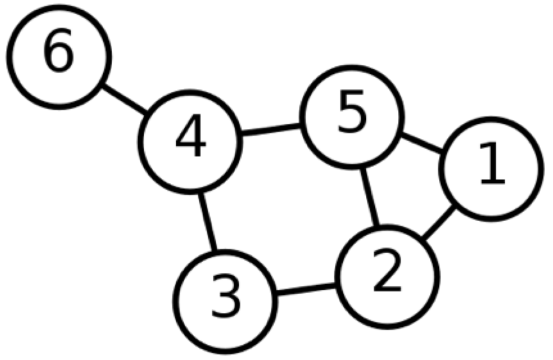
Dispersion decouples problem-specific smoothness arguments from regret bounds and private utility guarantees.

More Results from Dispersion

Bandit online optimization of piecewise Lipschitz functions:

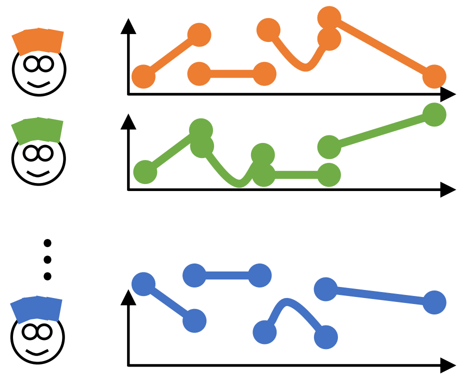
- Learner only observes the scalar $u_t(\rho_t)$ each round.
- Dispersion with $w \approx 1/T^{1/(d+2)}$ and $k \approx T^{\frac{d+1}{d+2}}$ implies regret $\tilde{O}\left(T^{\frac{d+1}{d+2}}\left(\sqrt{d3^d} + L\right)\right)$.
- Optimal exponent on T – other factors may be improvable.

Verify Dispersion for Algorithm Configuration



- Typically, *smoothed* adversaries choose dispersed functions.
- Greedy algorithms for knapsack and max-weight independent set.
- SDP rounding schemes for integer quadratic programs.

Differentially Private Optimization:



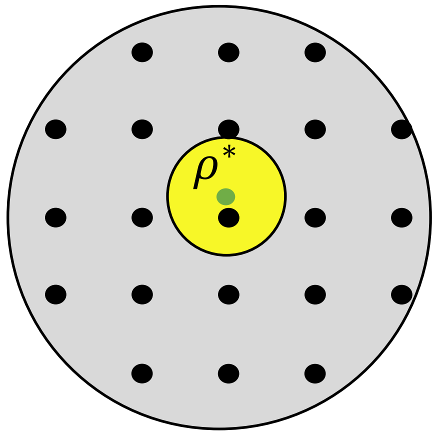
Goal: Approximately maximize the average with strong privacy guarantees for each individual.

- No algorithm has non-trivial utility in the worst-case.
- Matching upper and lower bounds from dispersion.

Future Work

Tidying Up and Easier Tools
- Current definition is a bit complicated with many “gotchas”.
- Working on getting crisp and intuitive definitions and better tools for verifying dispersion in practice.

Better (Semi)Bandit Algorithms
- Current algorithm discretizes $\mathcal{C}$ using a w-net.
  - Exponential running time in the dimension.
  - Does not adapt to optimal (w, k)-parameters.
- Exploit partial feedback (often observe an entire piece of u_t)



More Applications
- Derive dispersion for new algorithms (e.g., Initialization of Lloyd’s method)
- Applications beyond algorithm configuration

Other Algorithm Configuration Work

Branch & Bound for Mixed Integer Programming
[Balcan, Dick, Vitercik; ICML 2018]

Initialization Procedures for Center-Based Clustering
[Balcan, Dick, White; NeurIPS 2018]

Interpolating between Single, Complete, and Average Linkage
[Balcan, Dick, Lang; 2019]

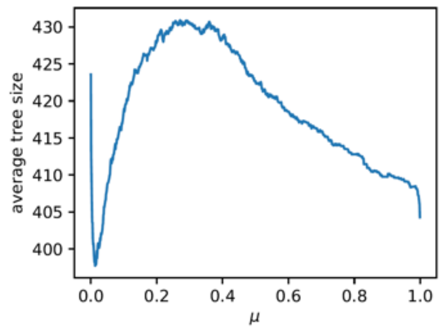
[Balcan, Nagarajan, Vitercik, White; COLT 2017]

- Uniform convergence guarantees in the i.i.d. setting (problems drawn from a distribution)
- Large-scale experiments carried out on 256 vCPUs on AWS EC2.

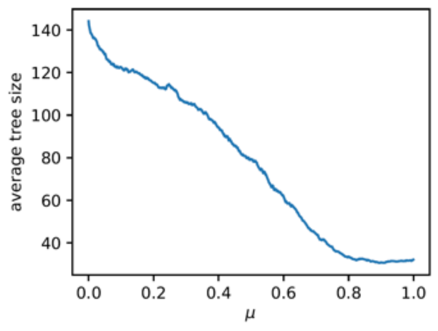
- E.g., for B&B we modify IBM's CPLEX using callbacks to change the "variable selection policy" and implement our tuning algorithm.

CPLEX Tuning Results:

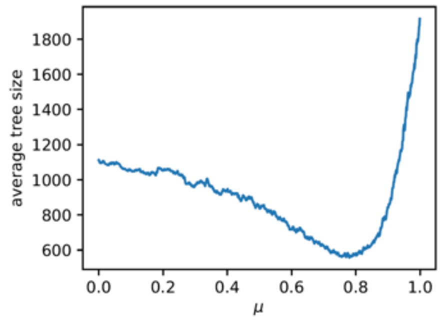
- X-axis is B&B parameter.
- Y-axis is tree-size (≈ running time).
- Best parameters vary significantly across tasks.



Linear Separators



Clustering

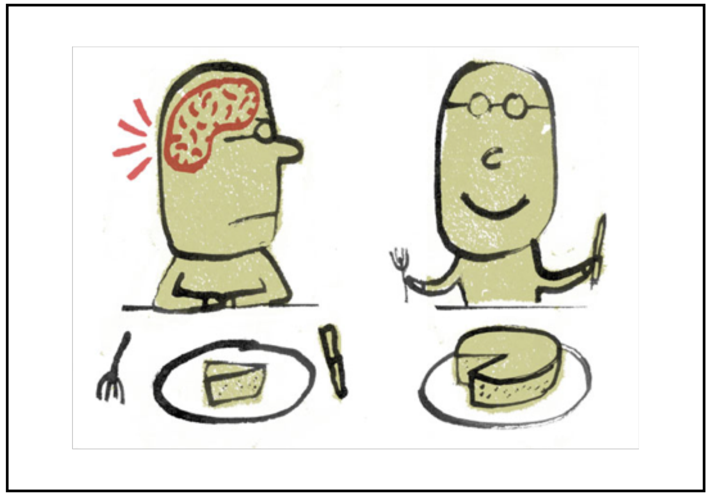


CATS Arbitrary

Part 2: Envy-free Classification

Joint work with Nina Balcan, Ariel Procaccia, and Ritesh Noothigattu [Ongoing]

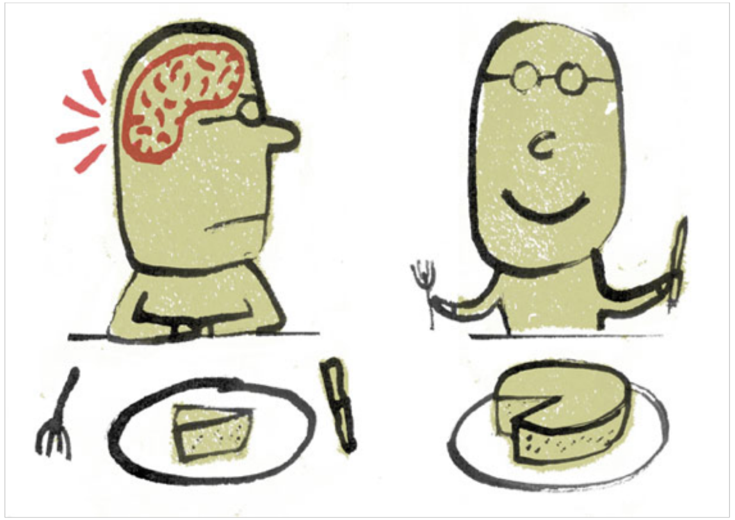
Social Values



Envy-freeness

Envy-freeness is the gold standard of fairness for fair division.

E.g.



cake cutting

chore assignment

Idea: Assignment is envy free if no one prefers the assignment of anyone else over their own.

Envy-Freeness in Machine Learning

Idea: A classifier is envy-free if no individual prefers the prediction/assignment of another.

Key Properties:

1. Provides fairness guarantees to individuals.

Group Fairness

Individual Fairness

2. It is suitable when there are many outcomes and heterogeneous preferences.

Advertisement Preferences

Job Listings

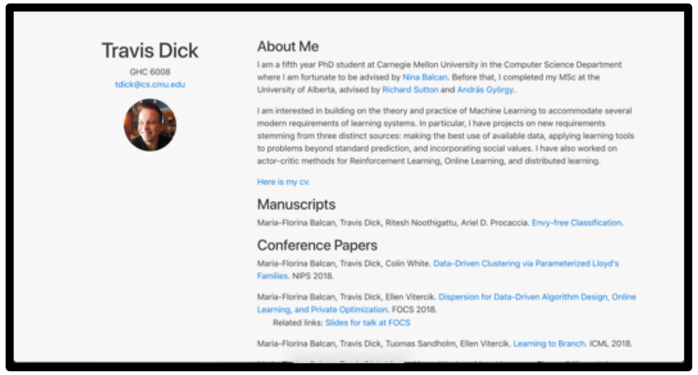
vs

Being released on bail.

Getting a loan.

Problem Setting

- We want to use ML to learn to assign outcomes to individuals.
- For example, deciding which advertisements to show in order to maximize revenue.



Travis Dick

GHC 6008

tdick@cs.cmu.edu

About Me

I am a fifth year PhD student at Carnegie Mellon University in the Computer Science Department where I am fortunate to be advised by Nina Balcan. Before that, I completed my MSc at the University of Alberta, advised by Richard Sutton and András György.

I am interested in building on the theory and practice of Machine Learning to accommodate several modern requirements of learning systems. In particular, I have projects on new requirements stemming from three distinct sources: making the best use of available data, applying learning tools to problems beyond standard prediction, and incorporating social values. I have also worked on actor-critic methods for Reinforcement Learning, Online Learning, and distributed learning.

Here is my cv.

Manuscripts

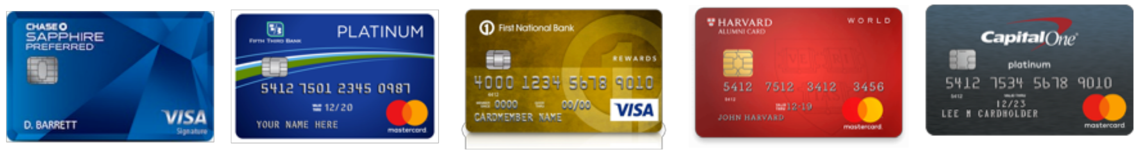
Maria-Florina Balcan, Travis Dick, Ritesh Noothigattu, Ariel D. Procaccia. Envy-free Classification.

Conference Papers

Maria-Florina Balcan, Travis Dick, Colin White. Data-Driven Clustering via Parameterized Lloyd's Families. NIPS 2018.

Maria-Florina Balcan, Travis Dick, Ellen Vitercik. Dispersion for Data-Driven Algorithm Design, Online Learning, and Private Optimization. FOCS 2018.
Related links: Slides for talk at FOCS

Maria-Florina Balcan, Travis Dick, Tuomas Sandholm, Ellen Vitercik. Learning to Branch. ICML 2018.



- Space $\mathcal{X}$ of individuals, space $\mathcal{Y}$ of outcomes.
- Assignment rule is a classifier $h: \mathcal{X} \to \mathcal{Y}$.
- Two objectives:
  1. Minimize expected loss (e.g., maximize revenue with advertisements)
  2. Treat people fairly (what does this mean?)
- We import the notion of **envy freeness** from fair division.
  - Suitable when there are many outcomes and heterogeneous preferences.
  - Provides fairness guarantees to individuals, rather than groups.

Envy-free Classifiers

($\Delta(\mathcal{Y})$ is the set of distributions on $\mathcal{Y}$)

- Given a utility function $u: \mathcal{X} \times \mathcal{Y} \to [0,1]$ measuring individual preferences.
- A classifier $h: \mathcal{X} \to \Delta(\mathcal{Y})$ is (α, β)-envy free with respect to u and $\mathcal{P}$ if:

$$\Pr_{x,x' \sim \mathcal{P}}\big(u(x, h(x)) < u(x, h(x')) - \beta\big) \leq \alpha\,.$$

- $\alpha = \beta = 0$: every individual likes their assignment as much as anyone else's.
- $\alpha > 0$: A small number of pairs of individuals are allowed to be envious.
- $\beta > 0$: A small amount of envy is tolerated.

Example: Two individuals and two outcomes

Utilities:

	🍎	🍰
🟠	1	0
🟢	0	1

- (0,0)-EF requires $h($🟠, 🍎$) \geq h($🟢, 🍎$)$ and $h($🟢, 🍰$) \geq h($🟠, 🍰$)$.

Randomized Classifiers

- Envy-freeness is a very strong requirement on deterministic classifiers:

Lem: Every *deterministic* (0,0)-EF map is of the form $h(x) = \text{argmax}_{y \in Y'} u(x, y)$ where Y' is a subset of the classes Y (breaking ties arbitrarily).

- We consider randomized classifiers (that assign distributions to individuals) since they have a lot more freedom.
- Better tradeoffs between fairness and accuracy.
    - Can construct examples where best randomized (0,0)-EF classifier has arbitrary multiplicative improvement over best deterministic (0,0)-EF classifier.

Main Question: Can we learn EF classifiers?

- If a classifier is EF on a sample of individuals, does this imply that it will remain EF on the underlying distribution?

- Following classic learning theory results, we assume the learner chooses an assignment rule from some class $\mathcal{H} = \{h: \mathcal{X} \to \Delta(\mathcal{Y})\}$ of *low complexity*.

- Main challenge: no known complexity measure applicable for distribution-valued outputs.

  - VC-dimension works for binary outputs.
  - Pseudo-dimension and Rademacher complexity work for a single real value output.
  - Natarajan dimension works for multi-class outputs.

Random Mixtures of Deterministic Classifiers

- No existing natural notion of complexity for distribution-valued outputs…
- **Idea:** Many randomized classifiers can be expressed as mixtures of deterministic classifiers.

Def: A class $H = \{h: X \to \Delta(Y)\}$ is an $\boldsymbol{m}$**-mixture** of a deterministic class $G = \{g: X \to Y\}$ if for each $h \in H$ there exists m functions $g_1, \dots, g_m \in G$ and mixing weights $p_1, \dots, p_m \in \Delta_m$ such that $\Pr(h(x) = y) = \sum_i p_i 1\{g_i(x) = y\}$.

h randomly picks g_i with probability p_i and outputs its classification of x.

$$\begin{tikzcd}
 & g_1(x) \\
h(x) \arrow[ur, "p_1"] \arrow[dr, "p_m"'] & \vdots \\
 & g_m(x)
\end{tikzcd}$$

If G has low Natarajan dimension, and H is an m-mixture of G, then we should expect envy-freeness to generalize for H.

Envy-Freeness Generalizes for Random Mixtures

Thm: Let $H = \{h: X \to \Delta(Y)\}$ be an m-mixture of $G = \{g: X \to Y\}$ and suppose the Natarajan dimension of G is d. If $S = \{(x_1, x_1'), \dots, (x_n, x_n')\}$ is an iid sample of $n = \tilde{O}\left(\frac{m^2 d}{\gamma^2}\right)$ pairs from P, then w.h.p. every $h \in H$ that is (α, β)-EF on S will be $(\alpha + 7\gamma, \beta + 4\gamma)$-EF on P.

Idea:

- Approximate H by a finite class $\widehat{H}$ (the size will depend on m and d).
- Can get generalization for $\widehat{H}$ from Hoeffding + Union bound.
- For each h, let $\hat{h}$ be its approximation in $\widehat{H}$.
- The classifiers h and $\hat{h}$ behave similarly on P and sufficiently large samples.

$$\begin{tikzcd} h \text{ on } S \arrow[r] & \hat{h} \text{ on } S \arrow[d] \\ h \text{ on } P & \hat{h} \text{ on } P \arrow[l] \end{tikzcd}$$

- If h is EF on S then $\hat{h}$ is too.
- $\hat{h}$ belongs to a finite class, so $\hat{h}$ is also EF on P.
- If $\hat{h}$ is EF on P, then h is too.

EF parameters degrade along each edge.

Future Work

Improved Dependence on m (number of classifiers mixed):

- Current generalization bound is $\tilde{O}\left(\frac{dm^2}{\gamma^2}\right)$, but it should be $\tilde{O}\left(\frac{d}{\gamma^2}\right)$.

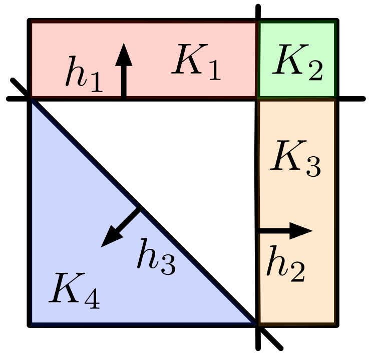
ERM Algorithms: Algorithms for efficiently finding low-envy low-loss classifiers.

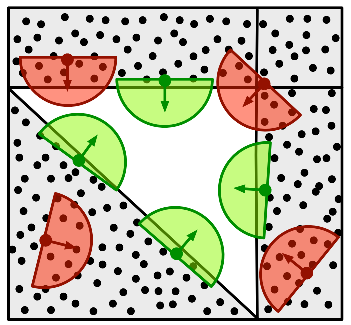
New Fairness Notions:

- I’m particularly interested in looking to the humanities and other disciplines where fairness has been important.

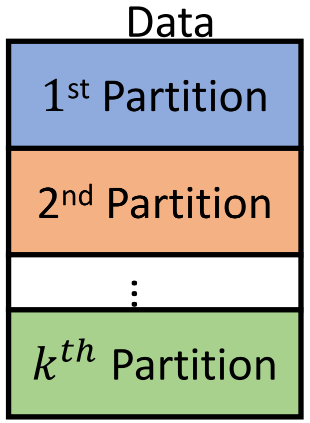
Other Work

- Label-efficient multi-class learning.
  [Balcan, Dick, Mansour; AAAI 2017]

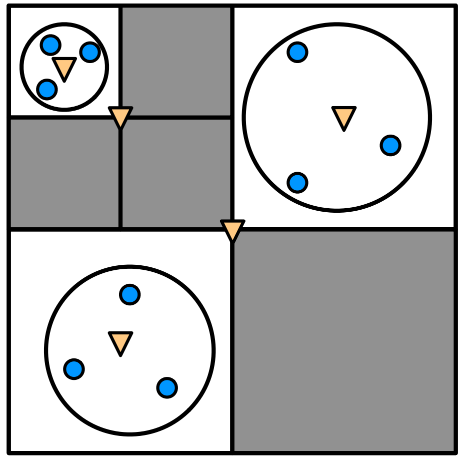




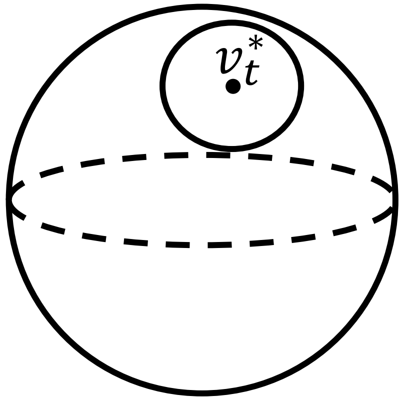
- Partitioning schemes for communication-efficient distributed learning.
  [Dick, Li, Pillutla, White, Balcan, Smola; AIStats 2017]



- Differentially private k-means and k-median clustering
  [Balcan, Dick, Liang, Mou, Zhang; ICML 2017]



- Differentially private covariance estimation
  [Amin, Dick, Kulesza, Medina, Vassilvitskii; 2018]



Thanks!