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

Travis Dick

Committee:

Maria-Florina Balcan (Chair)

Yishay Mansour (Tel Aviv University)

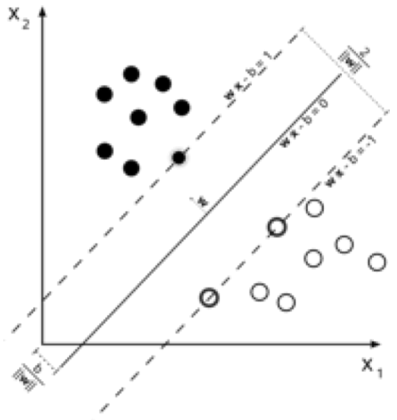
Tom Mitchell

Ariel Procaccia

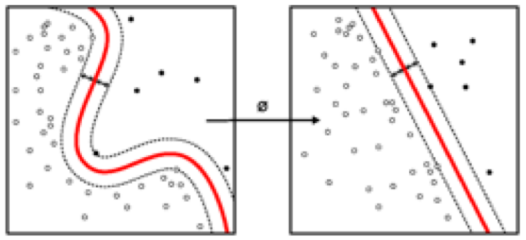
Modern Machine Learning

Classic machine learning: deep theory and powerful tools for learning to predict from data.

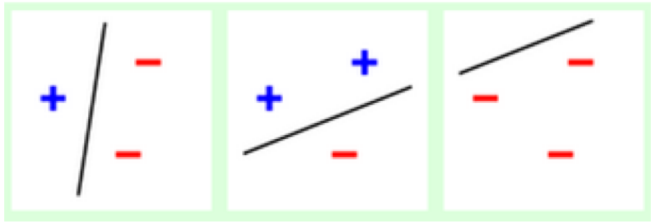
SVMs



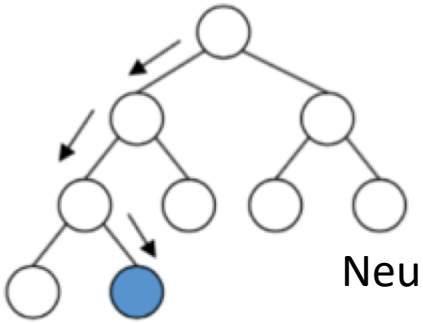
Kernel Methods



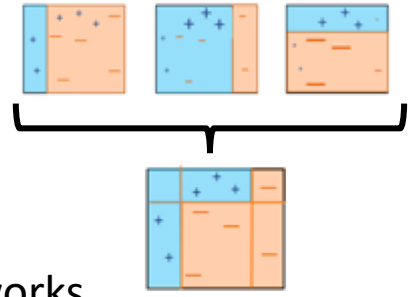
VC-Dimension



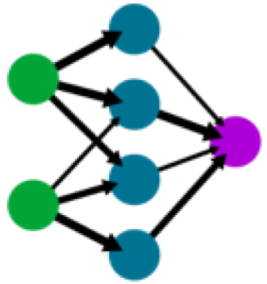
Decision Trees



Boosting



Neural Networks



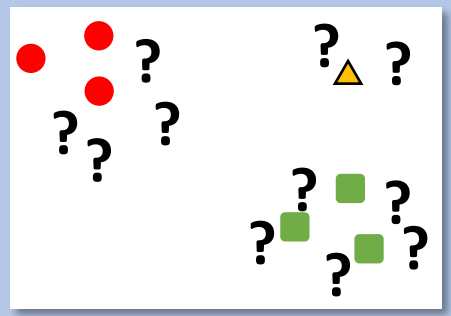
& much more

As ML is applied in the real world, we encounter new and interesting questions.

This thesis builds on the theory and practice of ML to accommodate modern ML requirements.

Three projects expanding the predictive possibilities of ML

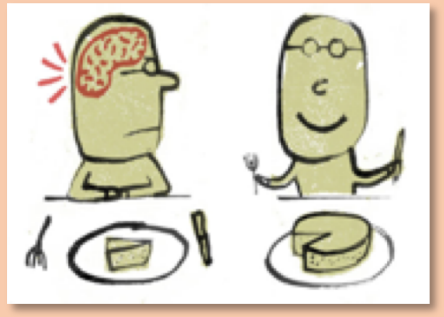
Improving our ability to make predictions with available data.



- Disparity in cost of data.
- Unlabeled data is very cheap, labeled examples are expensive.
- Design & analyze label-efficient multi-class learning algorithms.

[Balcan, Dick, Mansour AAAI 2017]

Making sure ML predictions don't violate our social values.



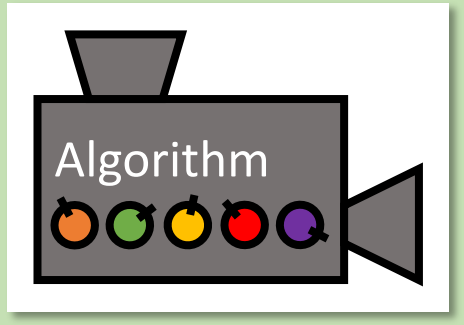
- Learning from private data.
- Making consequential predictions.
- We study envy-freeness as a notion fairness for ML.

[Balcan, Dick, Noothigattu, Procaccia, 2019]

- Differentially private learning.

[Balcan, Dick, Vitercik, FOCS 2018]

ML for learning forms of knowledge beyond prediction.



- Learning models motivated by data-driven alg. configuration.
- Using data to learn the best algorithm for specific application.
- Results for online learning with piecewise Lipschitz losses.

[Balcan, Dick, Vitercik, FOCS 2018]
[Balcan, Dick, Pegden, 2019]
[Balcan, Dick, Lang, 2019]

Label-Efficient Learning in Multiclass Problems

Joint work with Nina Balcan and Yishay Mansour [AAAI 2017]

Label Efficient Learning by Exploiting Multi-class Output Codes

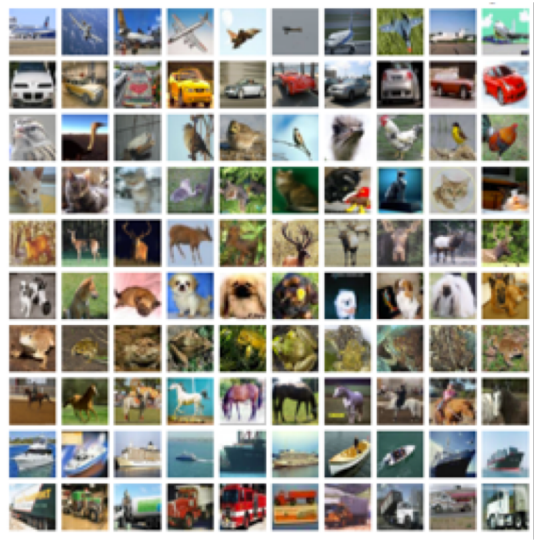
[Balcan, Dick, Mansour AAAI 2017]

Modern ML has huge amounts of raw data

Social Network Data

~6000 tweets per second

Images



Labeling data is expensive

Amazon SageMaker Ground Truth

Build highly accurate training datasets using machine learning and reduce data labeling costs by up to 70%

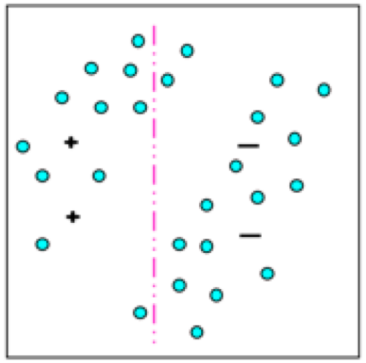
Charge ~$0.10 per text label.
~$0.84 per image segmentation.

Label Efficient Learning by Exploiting Multi-class Output Codes

[Balcan, Dick, Mansour AAAI 2017]

Long line of work on learning prediction rules from limited labeled data.

Semi-Supervised



[Blum & Mitchell '98; Balcan, Blum, Yang '04; Balcan & Blum '10, Chapelle et al. '10]

Active Learning

Learner

interaction

[Balcan et al. '06; Dasgupta '11; Hanneke 14'; Balcan & Urner '15]

Prior work primarily focused on binary classification.

- We prove guarantees for ***multiclass*** prediction problems.
- Exploit implicit assumptions of supervised learning algorithms.

Label Efficient Learning by Exploiting Multi-class Output Codes

[Balcan, Dick, Mansour AAAI 2017]

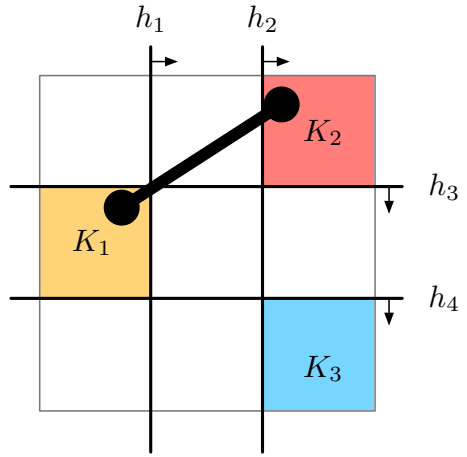
High Level Approach:

- Assume an output code classifier could learn from labeled data.
- This implies geometric structure in the underlying data.
- Exploit that structure to design label-efficient learners.

	Pet?	Fur?	Long neck?	Multiple lives?
cat	+1	+1	-1	+1
dog	+1	+1	-1	-1
penguin	-1	-1	-1	-1
giraffe	-1	+1	+1	-1

Lem: If ∃ consistent error correcting linear output code s.t. rows of the code matrix have hamming distance at least $d + 1$, then there is a margin between all pairs of classes.

- Careful clustering of the data has label-homogeneous clusters!
- Just need one label per cluster.



- Our other results significantly reduce the Hamming distance requirement.

Fairness in Machine Learning

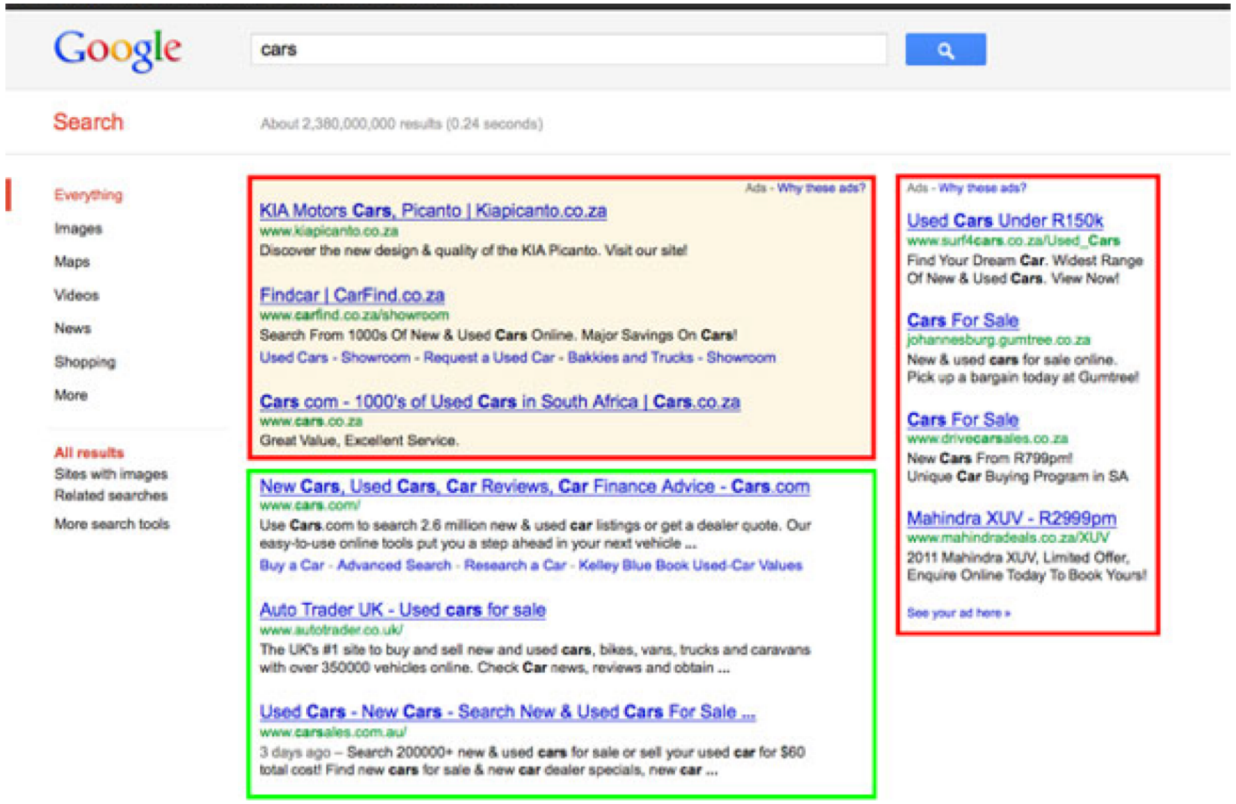
Joint work with Nina Balcan, Ritesh Noothigattu, and Ariel Procaccia

A New Approach to Individual Fairness: Envy-free Classification

[Balcan, Dick, Noothigattu, Procaccia, 2019]

ML is making predictions about *us*.
Want guarantees about fair treatment.

E.g., Advertisements shown by search engines.



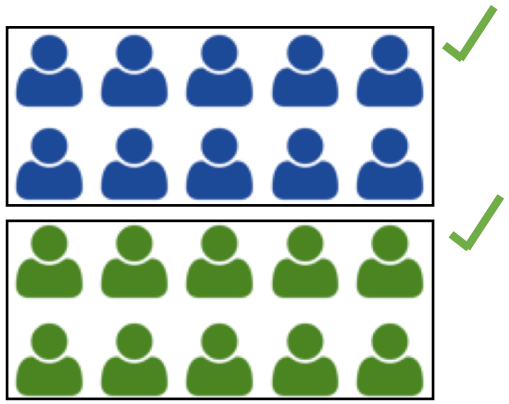
- Sweeney [‘13] found that searches for names showed ads suggestive of arrest records more often for some racial groups.
- Datta et al. [‘15] found different employment ads shown to men and women with identical search histories.

A New Approach to Individual Fairness: Envy-free Classification

[Balcan, Dick, Noothigattu, Procaccia, 2019]

Group Fairness:

Subgroups treated fairly *on average*.



[Luong *et al.* '11; Zemel *et al.* '13; Hardt *et al.* 16; Zafar *et al.* '16]

Individual Fairness:

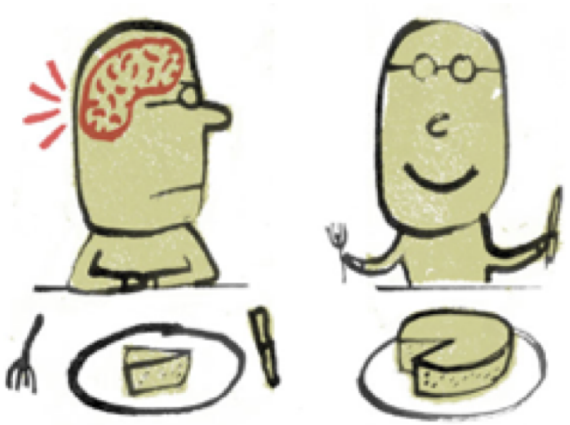
Fairness for every individual.

[Dwork *et al.* '12; Joseph *et al.* '16]

A New Approach to Individual Fairness: Envy-free Classification

[Balcan, Dick, Noothigattu, Procaccia, 2019]

- Import ***Envy-freeness*** from fair division [Foley '67; Varian '74].
- The gold standard for:
  - cake cutting [Robertson & Webb 98; Procaccia '13] ,
  - Rent division [Su '99; Gal *et al.* '17].



Def: Classifier h is envy-free when no individual prefers prediction made for another.

- Preferences can be estimated from data!
- Works with heterogeneous preferences.
- Main result: generalization guarantees

Thm: For family H of random classifiers, any $h \in H$ that is EF on sample of $O(D/\gamma^2)$ is (γ, γ)-approximately EF on the distribution.

H contains random mixtures of deterministic classifiers.

$D = m \cdot \mathrm{NDim}(G)$, m = mixture size, $\mathrm{NDim}(G)$ = complexity of fns being mixed.

New Learning Formulations Motivated by Data-driven Algorithm Configuration

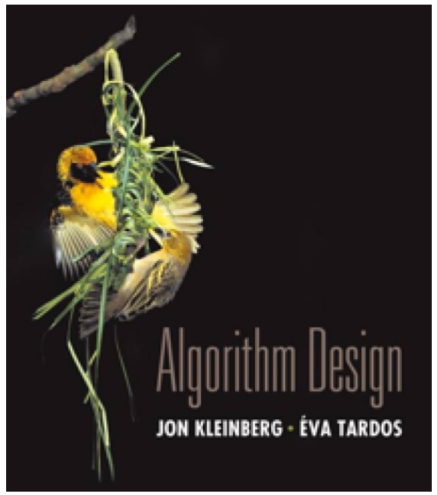
Based on joint work with

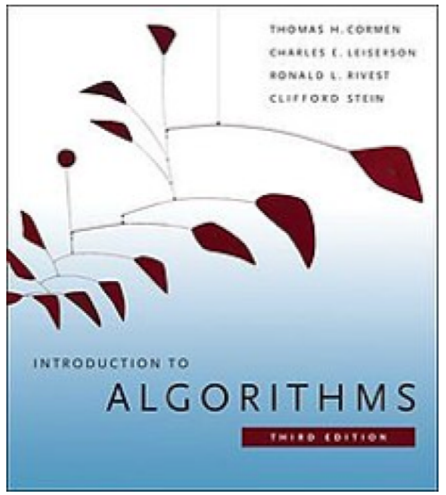
- Nina Balcan and Ellen Vitercik [FOCS 2018]
- Nina Balcan and Wesley Pegden [2019]
- Nina Balcan and Manuel Lang [2019]

Data-driven Algorithm Configuration

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

- Some domains have always-efficient optimal algorithms
- Many important domains are hard in worst case:
  - Clustering, subset selection, auction design, etc.
- "typical" applications aren't too hard.





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

- Repeatedly solve problems from the same application.
- Use historic problem instances to find the best algorithm.

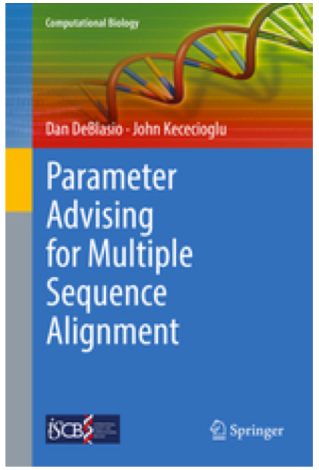
Data-driven Algorithm Configuration

Data Driven Algorithm Design Approach:

- Fix a parameterized family of algorithms.
- Different algs from family work better for different applications.
- Learn best alg/parameters from example problems.

Common in Practice:

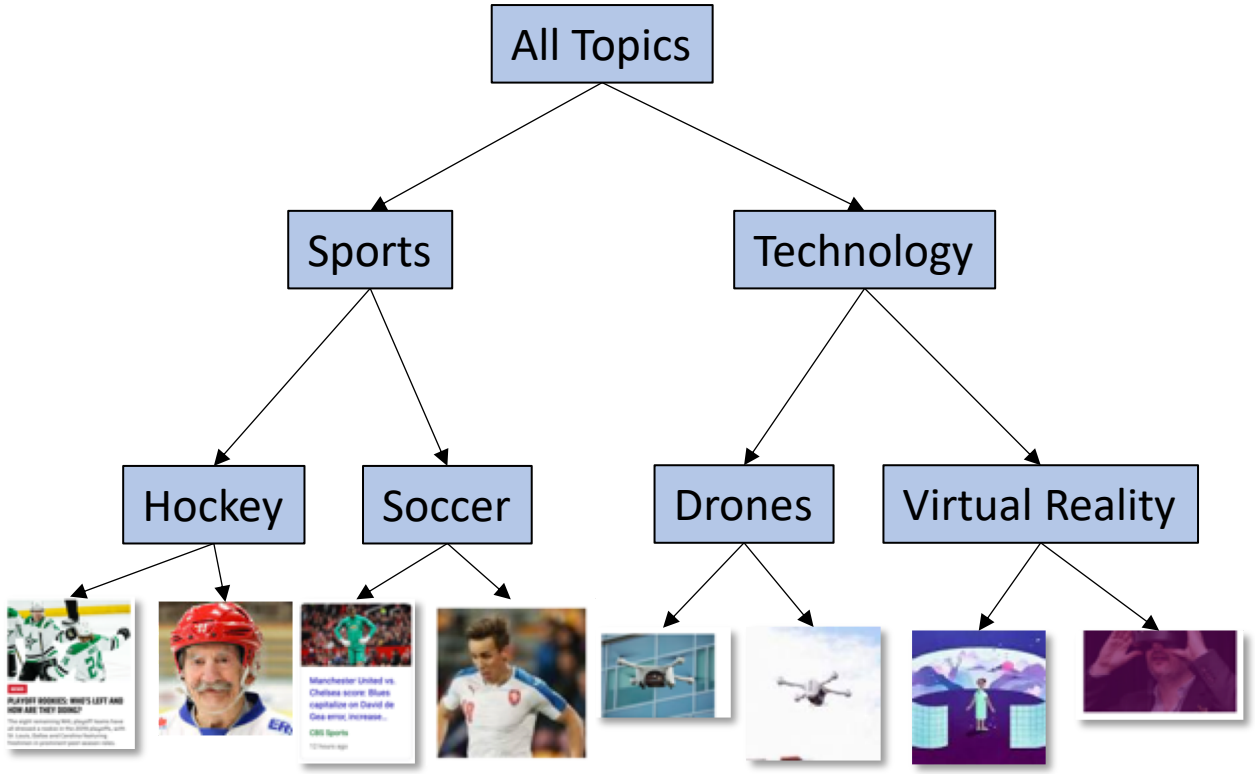
- Artificial Intelligence: E.g. *[Xu, Hutter, Hoos, Leyton-Brown, JAIR '08]*
- Computational Biology: E.g. *[DeBlasio, Kececioglu, '18]*
- Game Theory: E.g. *[Likhodedov & Sandholm, '04]*



Focus on empirical performance.

Example: Hierarchical Clustering

Given a collection of objects, organize them into a hierarchical clustering.
E.g., Clustering news articles by topic



Or clustering images by content.
Or vacation destinations by type of attraction.

Example: Hierarchical Clustering

Linkage based clustering

1. Start with each object in its own cluster
2. Repeatedly merge "closest" pair of clusters

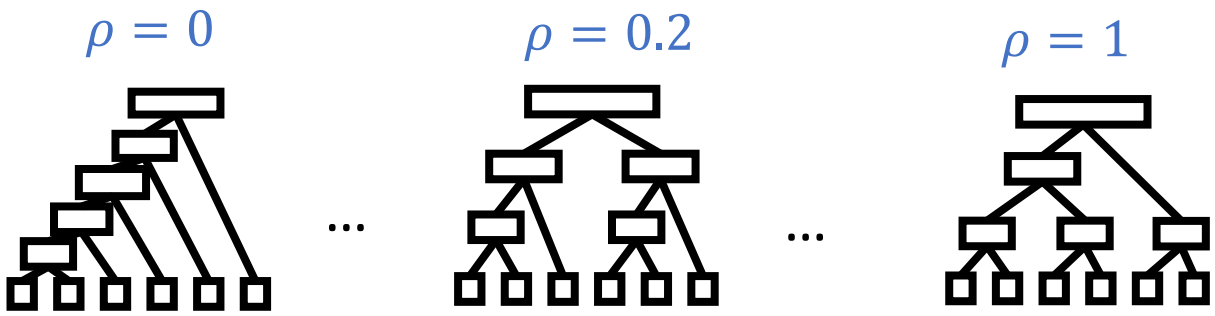
Different definitions of "closest" give different algorithms.

Single Linkage: $D_{\min}(A,B) = \min_{a\in A, b\in B} d(a,b)$

Complete Linkage: $D_{\max}(A,B) = \max_{a\in A, b\in B} d(a,b)$

ρ-Linkage: For $\rho \in [0,1]$ $D_\rho(A,B) = (1-\rho)D_{\min}(A,B) + \rho \cdot D_{\max}(A,B)$

[Balcan et al. '17]

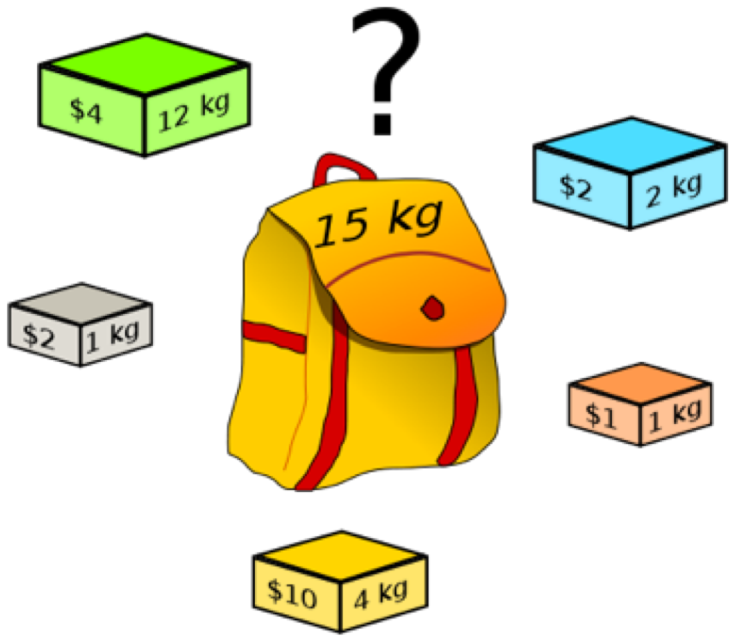


Example: Greedy Knapsack Algorithm

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 $\rho \geq 0$) *[Gupta & Roughgarden '16]*

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

ML Theory for Data Driven Algorithm Configuration

First ML-style guarantees are recent.

A PAC APPROACH TO APPLICATION-SPECIFIC ALGORITHM SELECTION*

RISHI GUPTA† AND TIM ROUGHGARDEN†

[ITCS '16, SICOMP'17]

Learning-Theoretic Foundations of Algorithm Configuration for Combinatorial Partitioning Problems*

Maria-Florina Balcan Vaishnavh Nagarajan Ellen Vitercik Colin White

[COLT '17]

Learning to Branch*

Maria-Florina Balcan Travis Dick Tuomas Sandholm Ellen Vitercik

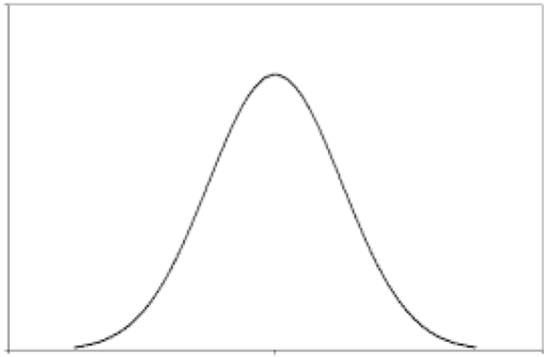
[ICML '18]

Data-Driven Clustering via Parameterized Lloyd's Families*

Maria-Florina Balcan Travis Dick Colin White

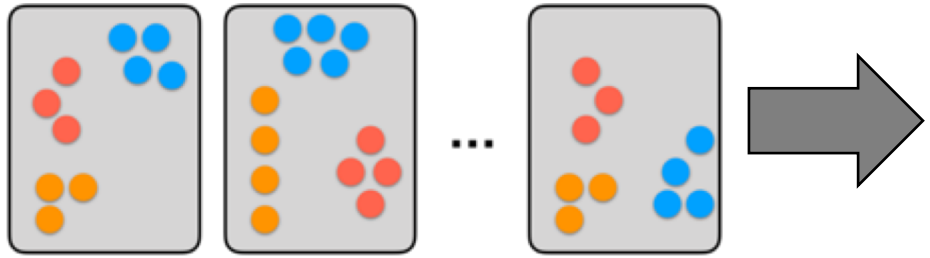
[NeurIPS '18]

Prior work focuses on the batch setting.



Application Specific Problem Distribution

Sample of i.i.d. problem instances

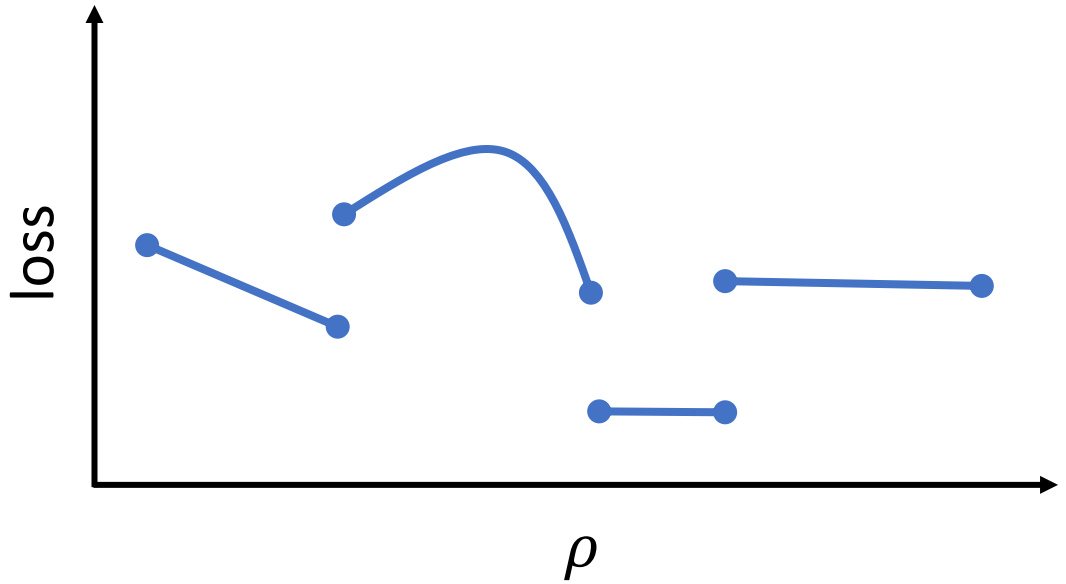


Choose algorithm parameter to minimize expected cost.

In this thesis: ML for data driven combinatorial alg. config in ***online*** and ***private*** settings.

Key Structure in Combinatorial Alg. Configuration

For one problem instance, cost/loss is piecewise Lipschitz fn of parameters.

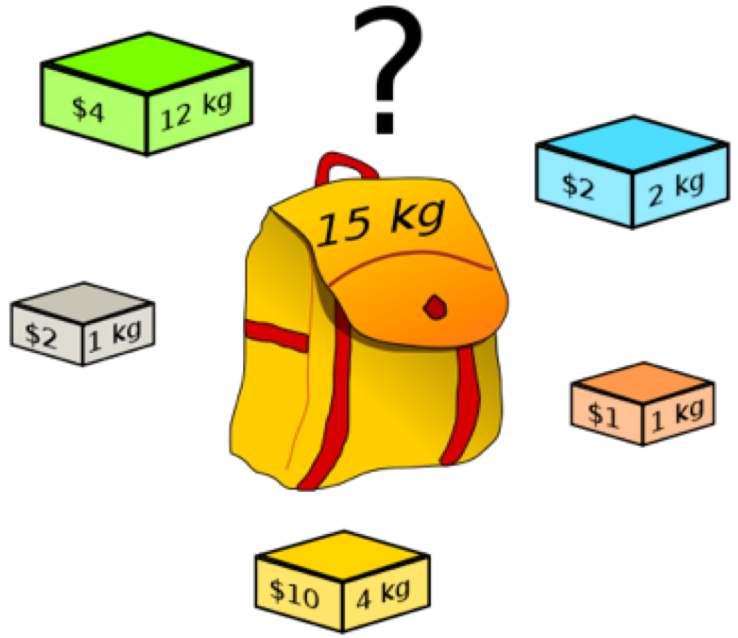


Piecewise Lipschitz Fns in Knapsack

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.



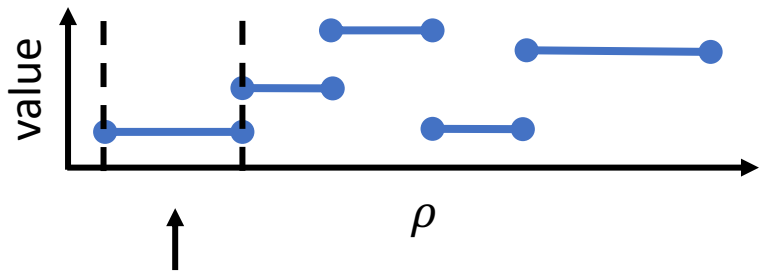
Algorithm: (Parameter $\rho \geq 0$)

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

[Gupta & Roughgarden '16]

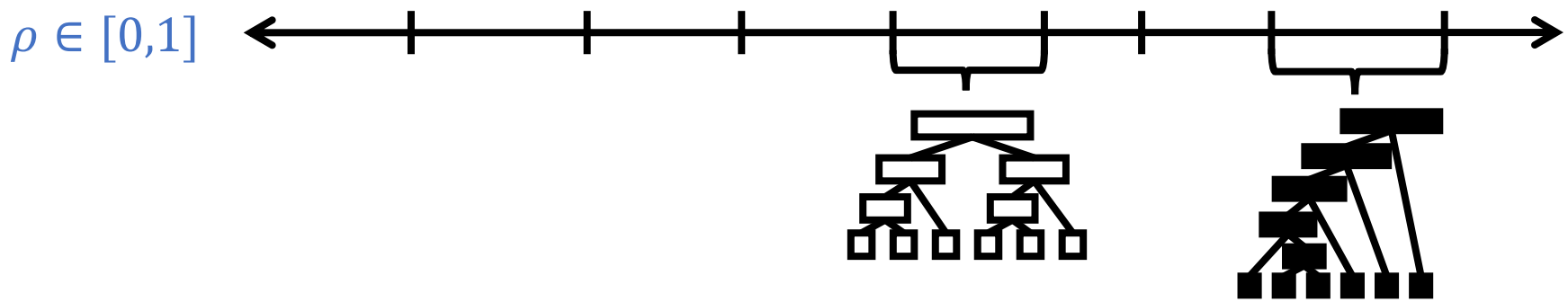
Lem: Value is piecewise constant in ρ.

- Only item order matters.
- Any item pair changes order at one value of ρ



Piecewise Lipschitz Fns in ρ-Linkage

- Balcan *et al.* [‘17] show output of ρ-Linkage is piecewise constant.
- Only distance ordering on possible merges matters.
- For any pair of pair of merges (A, A') and (B, B'), order changes only for one value of ρ.
- Distances between (A, A'), (B, B') depend on 8 points → $O(n^8)$ discontinuities.



Any loss that only depends on output is piecewise constant!

Online Learning with Piecewise Lipschitz Losses

Learning protocol:

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

1. Learner chooses $\rho_t \in \mathcal{C} \subset \mathbb{R}^d$.
2. Adversary chooses piecewise L-Lipschitz loss $\ell_t : \mathcal{C} \to \mathbb{R}$.
3. Learner incurs cost $\ell_t(\rho_t)$
4. Learner gets feedback on ℓ_t.

- Full-information: Observe entire loss function ℓ_t.
- Bandit: Observe just $\ell_t(\rho_t) \in \mathbb{R}$.
- Semi-bandit: Observe $\ell_t(\rho)$ for all ρ in subset $A_t \subset \mathcal{C}$.

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

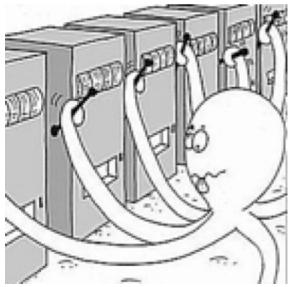
Meaningful Learning: Regret sublinear in T.

Why isn't this solved already?

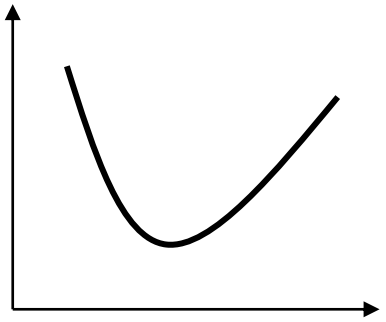
Existing bounds are for more structured settings *[e.g., Cesa-Bianchi & Lugosi '06, Bubeck '11]*

Prediction with (finite) expert advice.

Finite-armed Bandits

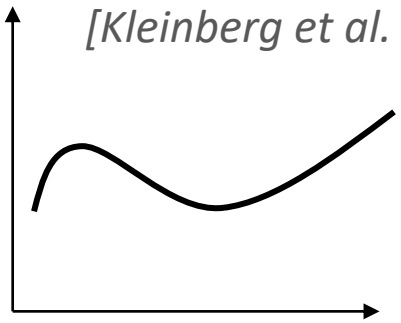


Convex Fns



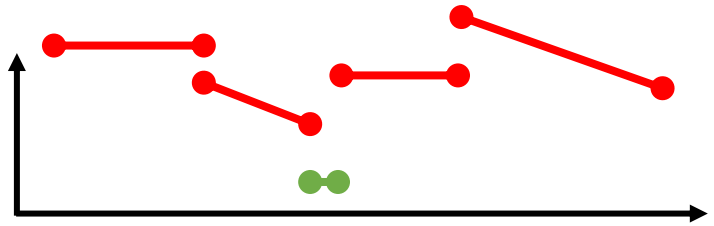
Globally Lipschitz Fns

[Kleinberg et al. '08]



Under full-info, Regret $\leq \tilde{O}(\sqrt{T} \times \text{problem dependent terms}).$

Main Challenge: When the losses have discontinuities, cannot achieve sublinear regret!



Simple adversary ensures $\Omega(T)$ regret. **Need additional structure!**

Dispersion

$\tilde{O}$ hides log terms and application-specific terms

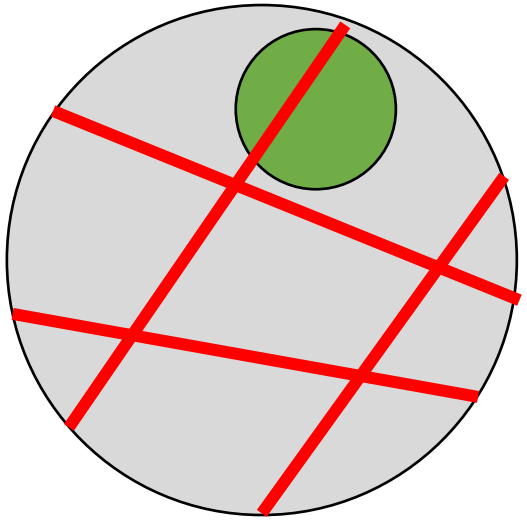
- Discontinuities only problematic if concentrated.
- We introduce *dispersion* for measuring concentration.
- Improved version of definition from Balcan, Dick, Vitercik [FOCS 2018].

Def: Function $\ell_1, \ell_2 \ldots$ are ***β*-dispersed** if $\forall$ times $T \in \mathbb{N}$ and $\forall$ radiuses $\epsilon \geq T^{-\beta}$, expected # of non-Lipschitz fns on worst ball of radius ϵ is $\tilde{O}(T\epsilon)$. I.e.,

[BDP '19]

$$\mathbb{E}\left[\max_{\rho} |\{1 \leq t \leq T : \ell_t \text{ not Lipschitz on } B(\rho, \epsilon)\}|\right] \leq \tilde{O}(T\epsilon)$$

"In balls of radius ϵ, encounter non-Lipschitz functions at rate $\tilde{O}(\epsilon)$"
(as long as ϵ is not too small)

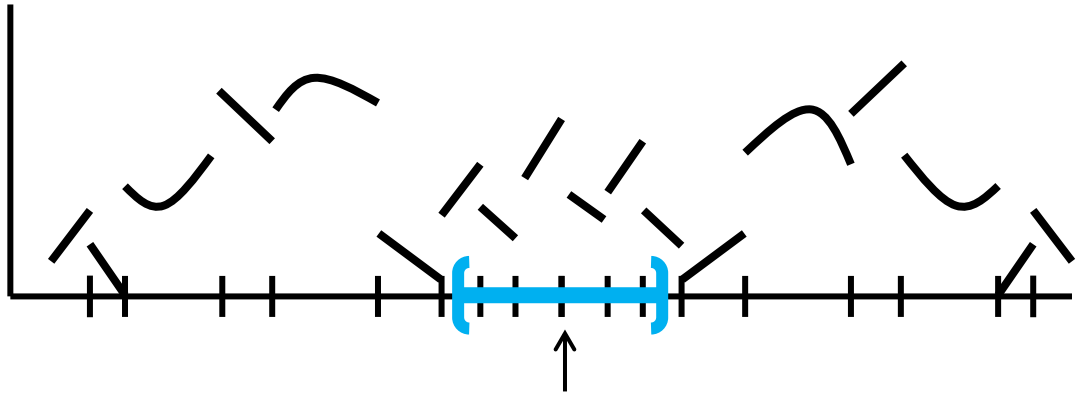


- Expectation over random losses (e.g., if smoothed)
- Larger β is stronger.

The Sum of Disperse Functions

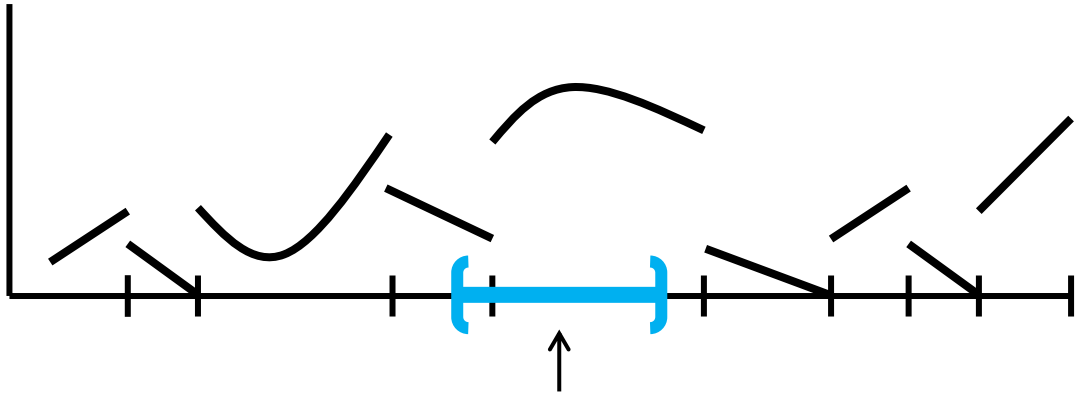
Let $\ell_1, \ell_2, \ldots$ be PWL functions and plot their sum $\sum_{t=1}^{T} \ell_t$

Not disperse



Many discontinuities within interval

Disperse



Few boundaries within interval

Dispersion for ρ-Linkage

Study *smoothed* adversaries – small amount of noise added to problems.

Thm: If $\mathcal{N}(0, \sigma)$ noise added to pairwise distances every round. Then after T rounds

- Expected non-Lipschitz loss fns on worst ϵ-interval is $O\left(T\epsilon n^8/\sigma^2 + \sqrt{T\log(Tn)}\right)$.
- β-dispersed for $\beta = 1/2$.

[BDP '19]

Intuition:

- Each loss fn has $O(n^8)$ discontinuities.
- Noise in distances → noise in discontinuity locations.
- Don't expect many to land in an interval of width ϵ.
- $\sqrt{T}$ term accounts for taking the *worst* interval.

Similar arguments hold for many other domains.

Full Information Regret Bounds

Continuous Multiplicative Weights *[Cesa-Bianchi & Lugosi '06]*

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

Thm: If $\ell_1, \ell_2, \ldots\ C \to [0,1]$ are piecewise L-Lipschitz and β-dispersed, then $\forall T$,
[BDV, FOCS 2018]

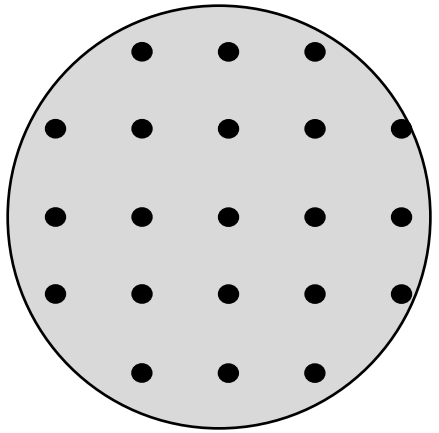
$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\rho_t) - \ell_t(\rho^*)\right] \leq \tilde{O}\left(\sqrt{Td} + T^{1-\beta}\right).$$

- $\tilde{O}(\sqrt{Td})$ optimal for *globally* Lipschitz functions.
- Regret due to discontinuities is $\tilde{O}(T^{1-\beta})$.
- Bound improves as β grows until $\beta = 1/2$.

Theorem: (ρ-Linkage) Add $\mathcal{N}(0, \sigma)$ to pairwise distances. Then after T rounds

$$\text{Regret} \leq O(\sqrt{T \log(nT/\sigma)})$$

Bandit-feedback Regret Bounds



Discretization Algorithm:

Algorithm: (Parameters $r > 0$, $\lambda > 0$
Let $\rho^1, \dots, \rho^N$ be an r-net for $\mathcal{C}$.
Use EXP3 to choose $\rho_t \in \{\rho^1, \dots, \rho^N\}$ *[Auer et al. '01]*

Thm: If $\ell_1, \ell_2, \dots : C \to [0,1]$ are piecewise L-Lipschitz and β-dispersed, then

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\rho_t) - \ell_t(\rho^*)\right] \leq \tilde{O}\left(T^{\frac{d+1}{d+2}}(3^d + L) + T^{1-\beta}\right).$$

[BDV, FOCS 2018]

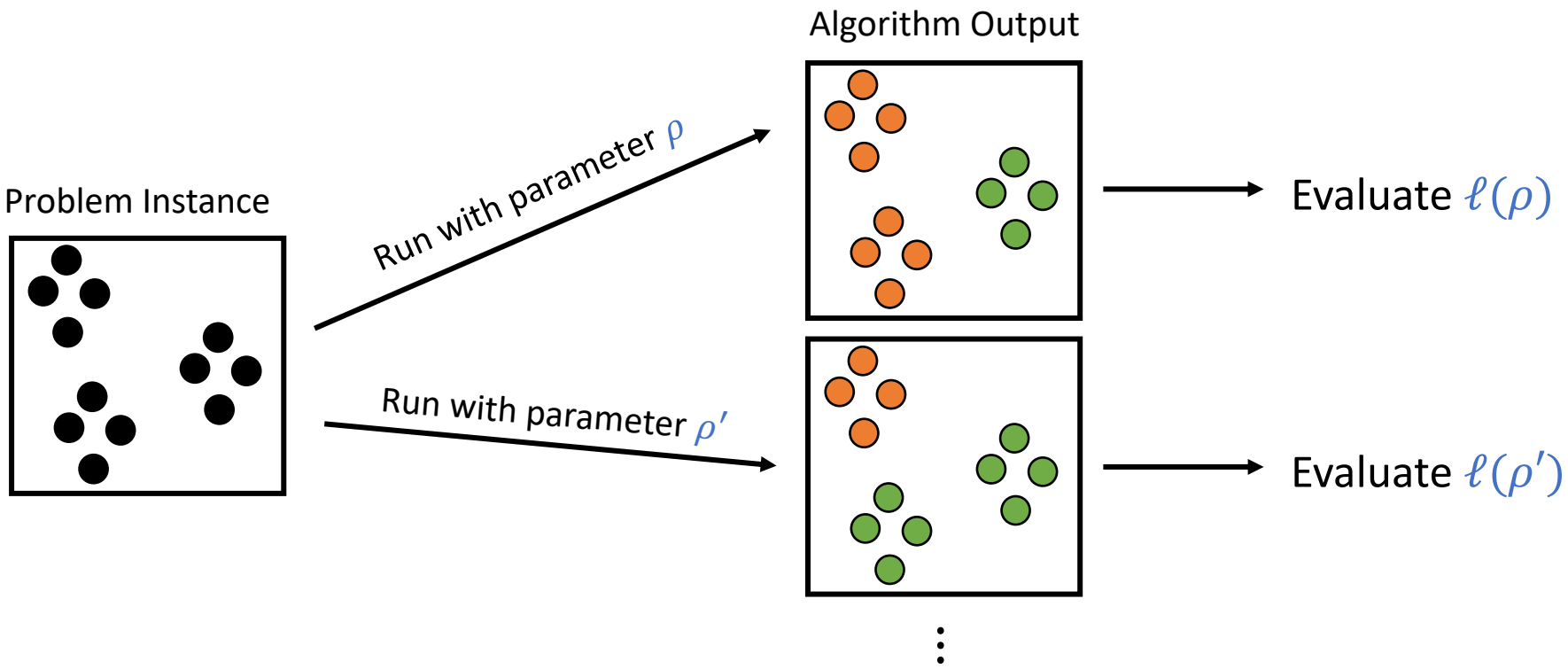
- $T^{\frac{d+1}{d+2}}$ is optimal exponent for *globally* Lipschitz functions. *[Kleinberg et al. '08]*
- Bound improves as β grows until $\beta = 1/(d+2)$.

Thm: (ρ-Linkage) Add $\mathcal{N}(0, \sigma)$ to pairwise distances. Then after T rounds

$$\text{Regret} \leq O(n^8 T^{2/3}/\sigma^2)$$

Feedback in Algorithm Configuration

Most domains provide full-information feedback.



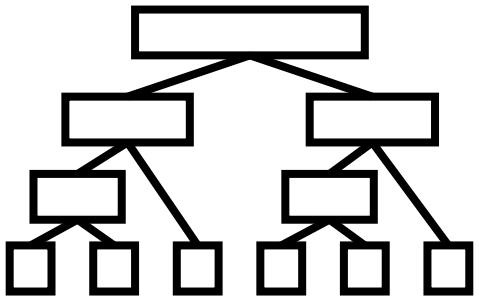
- Full-info feedback can be expensive: many runs of algorithm.
- Bandit feedback is more efficient, but has worse regret bounds.

- Next: we can sometimes get the best of both by exploiting extra structure *[BDP '19].*

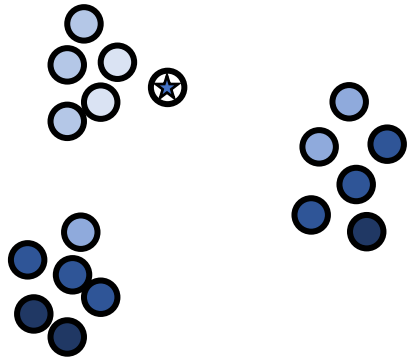
Extra Structure: Semi-bandit feedback

Key insight from many implementations:

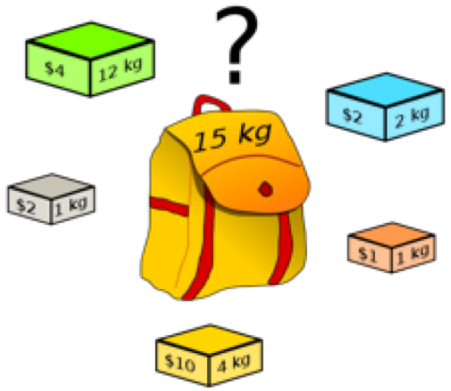
- Running algorithm once reveals loss for a range of parameters.
- Often an entire piece of piecewise Lipschitz loss!
- E.g.



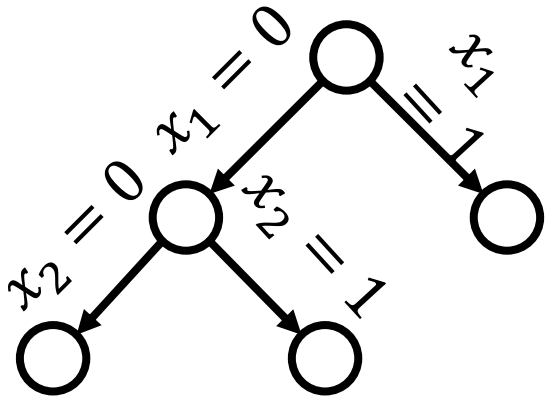
Linkage-based Clustering



Initialization for k-means clustering



Greedy Knapsack



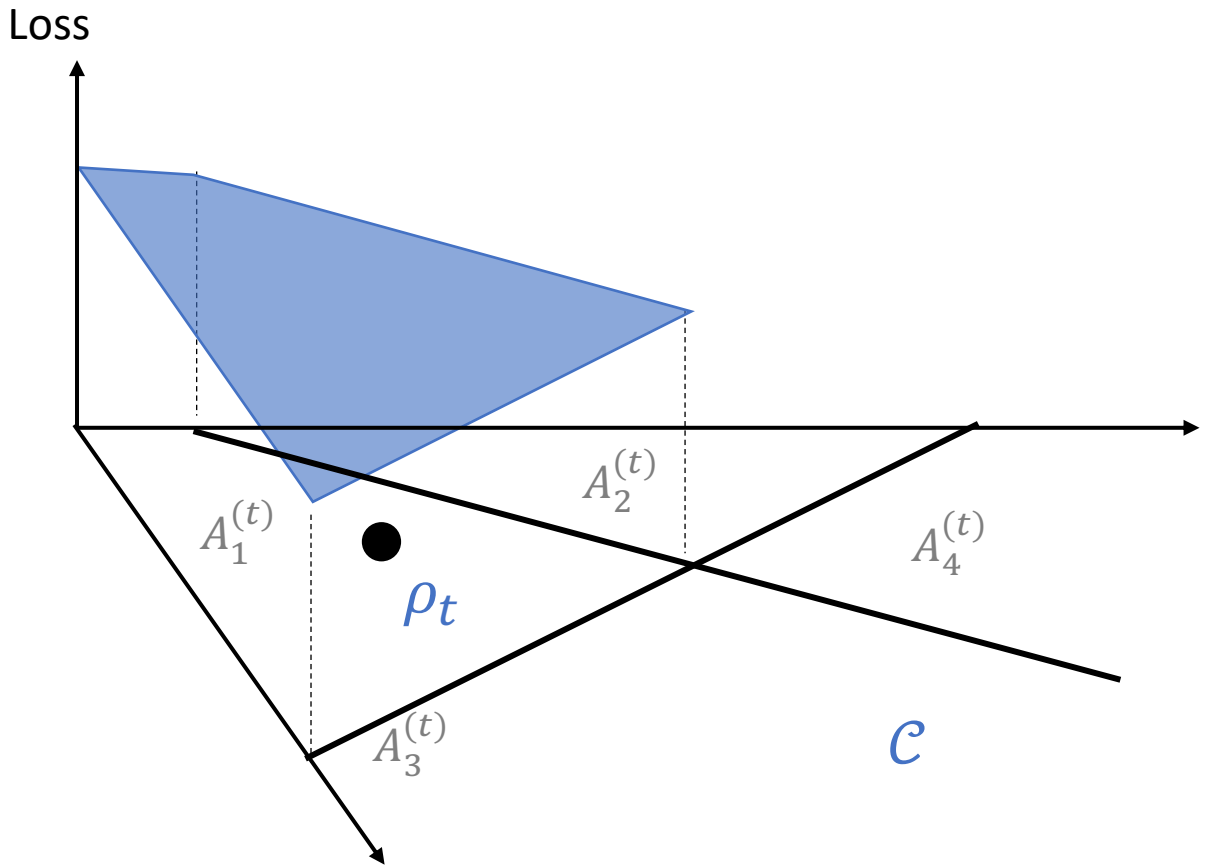
Variable selection in Branch and Bound

Semi-bandit Feedback

Def: *(Semi-bandit feedback)*

At time t, there is a partition $A_1^{(t)}, \dots, A_M^{(t)}$ of $\mathcal{C}$.

When learner plays $\rho_t \in A_i^{(t)}$ they observe $A_i^{(t)}$ and $\ell_t(\rho)$ for all $\rho \in A_i^{(t)}$.



Learning with Semi-bandit Feedback

Continuous version of EXP3-SET algorithm of [Alon, Cesa-Bianchi, Gentile, Mannor, Mansour, Shamir '17]

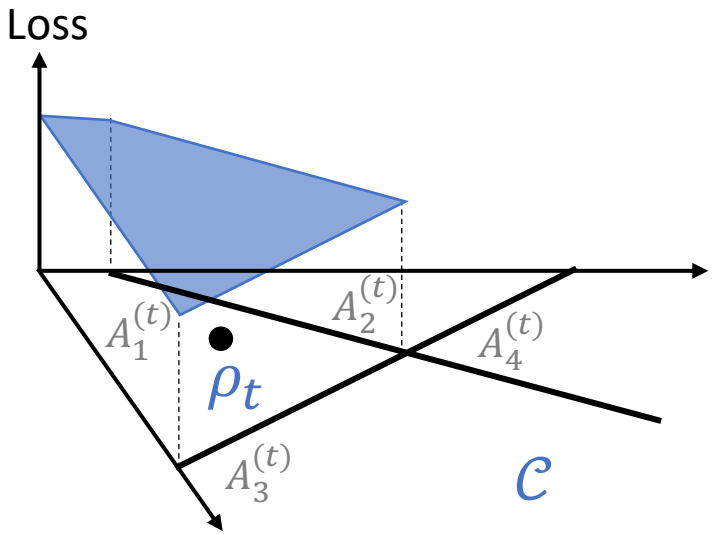
Algorithm:

Use importance weighting to estimate complete loss

$$\hat{\ell}_t(\rho) = \underbrace{\frac{\mathbb{I}\{\rho \in A_t\}}{p_t(A_t)}}\, \ell_t(\rho)$$

Scales loss by 1/(probability of observing it).

Run continuous multiplicative weights on $\hat{\ell}_1, \hat{\ell}_2, \ldots$



Thm: If $\ell_1, \ell_2, \ldots : C \to [0,1]$ are piecewise L-Lipschitz, β-dispersed, and $\leq M$ feedback sets, [BDP '19]

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\rho_t) - \ell_t(\rho^*)\right] \leq \tilde{O}\left(\sqrt{dTM} + T^{1-\beta}\right).$$

Linkage Clustering with Semi-bandit Feedback

- From a single run of ρ-linkage, we can get semi-bandit feedback.
- A bit of extra computation to keep track of nearest discontinuities to ρ.
- Number of feedback sets is $M = O(n^8)$.

Thm: (ρ-Linkage) Add $\mathcal{N}(0, \sigma)$ to pairwise distances. Then after T rounds EXP3-SET satisfies:

$$\text{Regret} \leq O\left(n^4\sqrt{T\log(Tn/\sigma)}\right)$$

- $\sqrt{T}$ regret, like full-info.
- Run algorithm once per round, like bandit feedback.

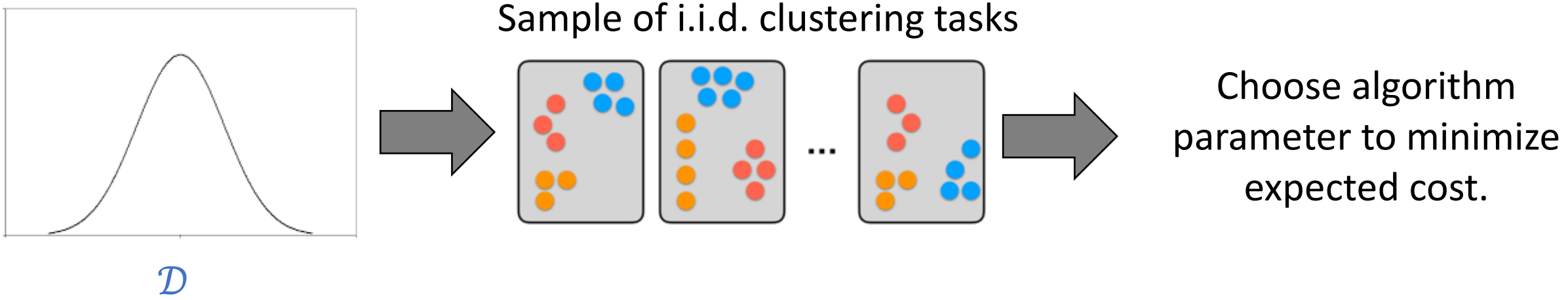
Algorithm Configuration Experiments

Joint work with Nina Balcan & Manuel Lang

Experiments in the Batch Setting

Batch setting:

- Distribution $\mathcal{D}$ over problems.
- Given i.i.d. sample from $\mathcal{D}$, find algorithm with lowest expected error on $\mathcal{D}$.



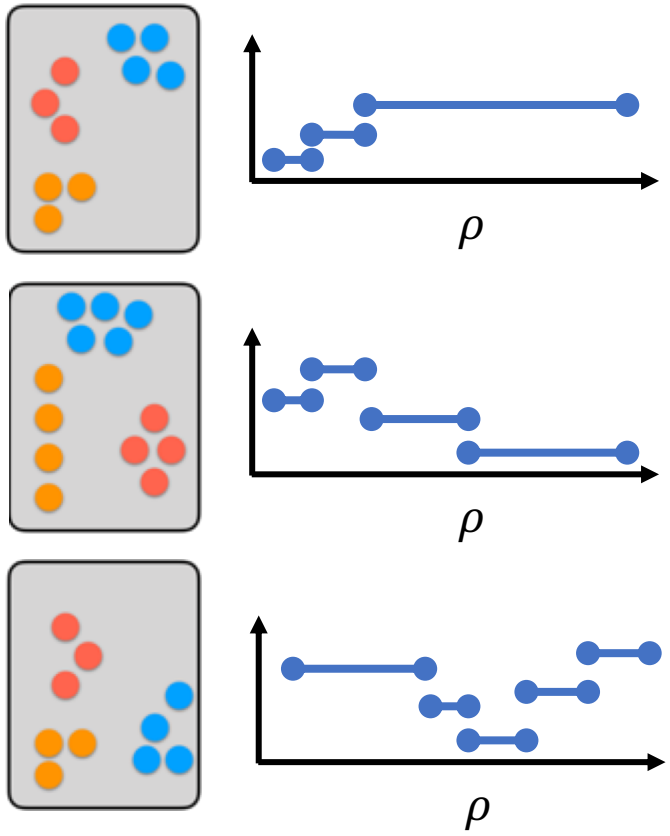
Two Questions:

1. Sample Complexity: How many samples do we need to see to find nearly optimal parameter.
2. Computational Complexity: How can we efficiently find the best parameter?

Empirical Risk Minimization for Linkage Clustering

Cluster distance fn: $D_\rho(A,B) = (1-\rho)D_{\min}(A,B) + \rho \cdot D_{\max}(A,B)$

For each instance, Hamming error is piecewise constant

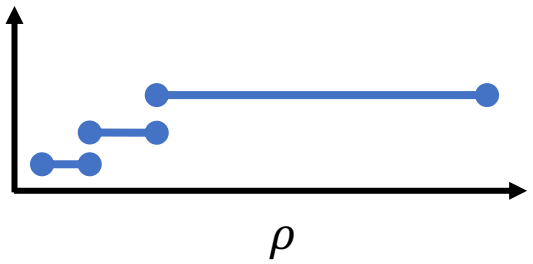


Algorithm:

1. Compute PWC loss for each instance.
   (i.e., discontinuities and values)
2. Average PWC losses
3. Iterate through pieces and output lowest loss.

Computing PWC Loss Functions

ρ



Each discontinuity determined by 8 points in the data.

- Prior work enumerates all $O(n^8)$ point subsets. [Balcan et al. '17]
- Gives a piecewise constant partition of parameters.
- Run the algorithm with one parameter from each constant interval.
- Total runtime: $O(n^{10} \log n)$ ($\mathrm{O}(n^8)$ runs of an $\mathrm{O}(n^2 \log n)$ time alg.)

Empirical Insight: Significantly fewer than $O(n^8)$ discontinuities in practice.

We exploit this to get faster running time.

Efficient PWC Loss Computation

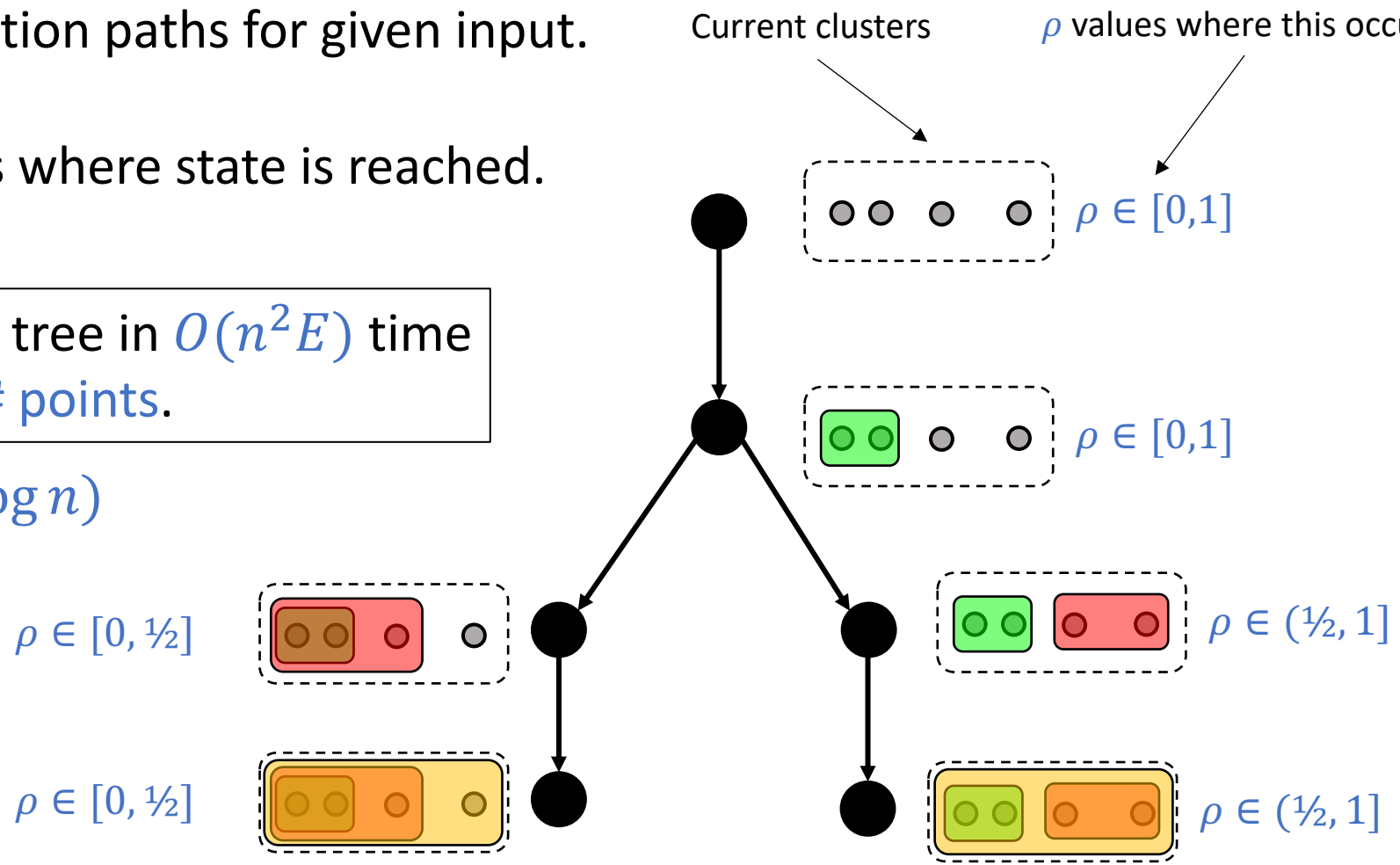
A more efficient alg. for enumerating all algorithm outputs.

Def: Execution Tree

- Represents all possible execution paths for given input.
- Nodes are algorithm states.
- Nodes labeled by parameters where state is reached.
- Leaves are possible outputs.

Thm: Can enumerate execution tree in $O(n^2 E)$ time
E = # of edges, n = # points.

Empirically, faster than $O(n^{10} \log n)$

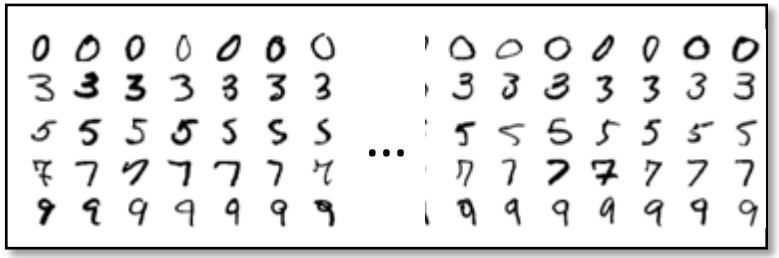


Clustering Subsets of MNIST

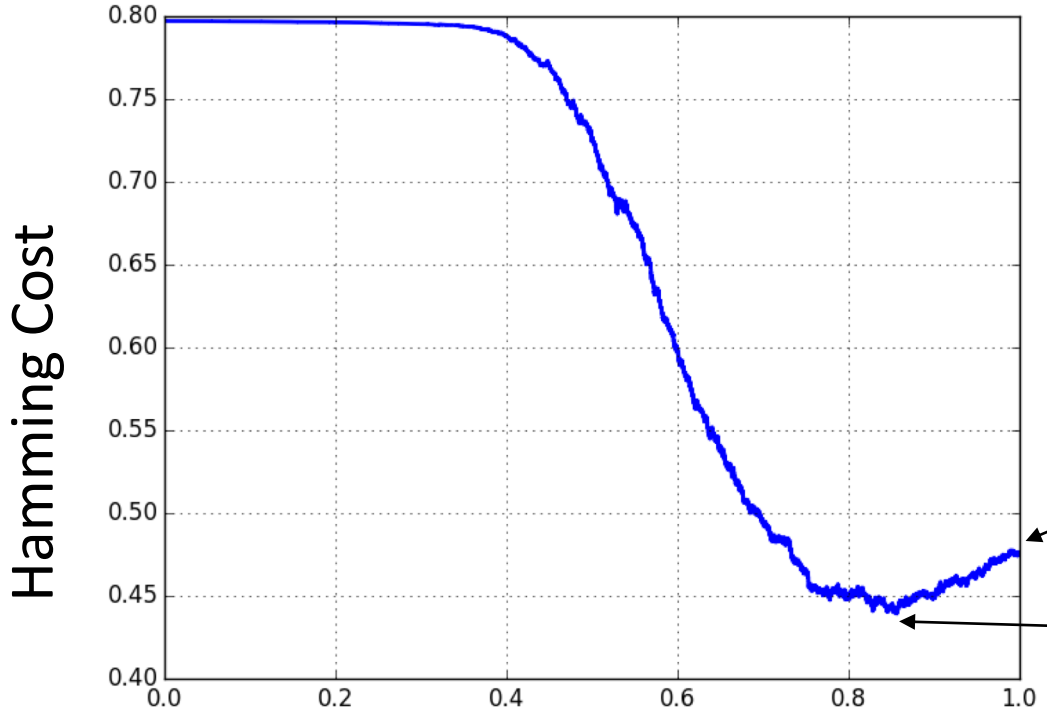
Instance Distribution

- Pick 5 random digits.
- Pick 200 random images per digit (1000 in total)
- Target clustering is given by the digit classification.

Example instance:



Average over $n = 500$ sampled instances



$\rho = 1$
Error = 47.6%

$\rho^* = 0.857$
Error = 44.1%
Improvement of 3.54%

Another Alg. Family: Learning the Best Distance Metric

- Suppose we have more than one way to measure distances between examples.
- E.g. Captioned images: both the caption and image tell us about similarity.

"Black Cat"

"Bobcat"

"Roaring Cat"

"Evacuator"

- Captions show similarity between felines, but do not separate the "bobcat".
- Images distinguish machinery from the animals.

Can we learn how to combine these metrics to get best clusterings?

Algorithm Family: Learning the Metric

- Two metrics d_0 and d_1.
- $\forall \beta \in [0,1]$, define

$$d_\beta(x, x') = (1-\beta)d_0(x, x') + \beta d_1(x, x').$$

- Algorithm with parameter β runs complete linkage using d_β metric.
- Learn value of β with lowest expected loss.

★ Execution tree gives efficient loss computation again!

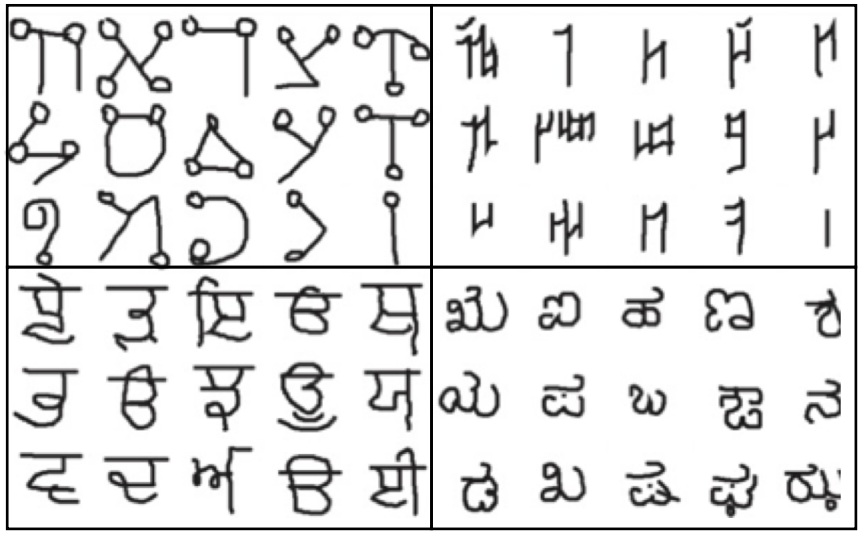
Advantages over other metric learning approaches:

- Optimize directly for alg. performance (instead of surrogate loss).
- Exact optimization procedures.
- Sample complexity guarantees.

Omniglot Results:

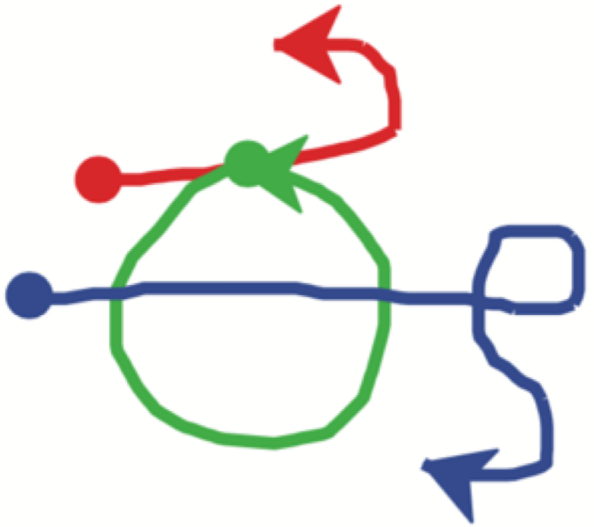
Dataset *[Lake, Salakhutdinov, Tenenbaum '15]*

- Standard meta-learning dataset.
- Written characters from 50 different alphabets.
- Each character has 20 examples.
  - Image of character
  - Stroke data (trajectory)



Instance Distribution

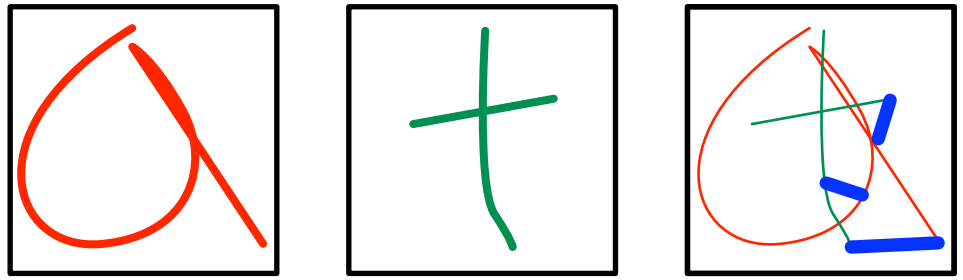
- Pick random alphabet.
- Pick between 5 and 10 characters.
- Use all 20 examples of chosen characters (100 – 200 points)
- Target clusters are characters



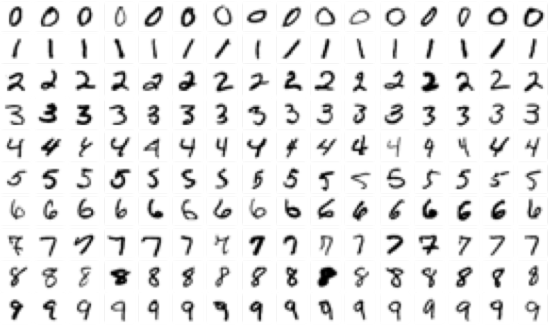
Omniglot Results:

”Stroke” Distance:

Avg. distance between pts and other stroke.



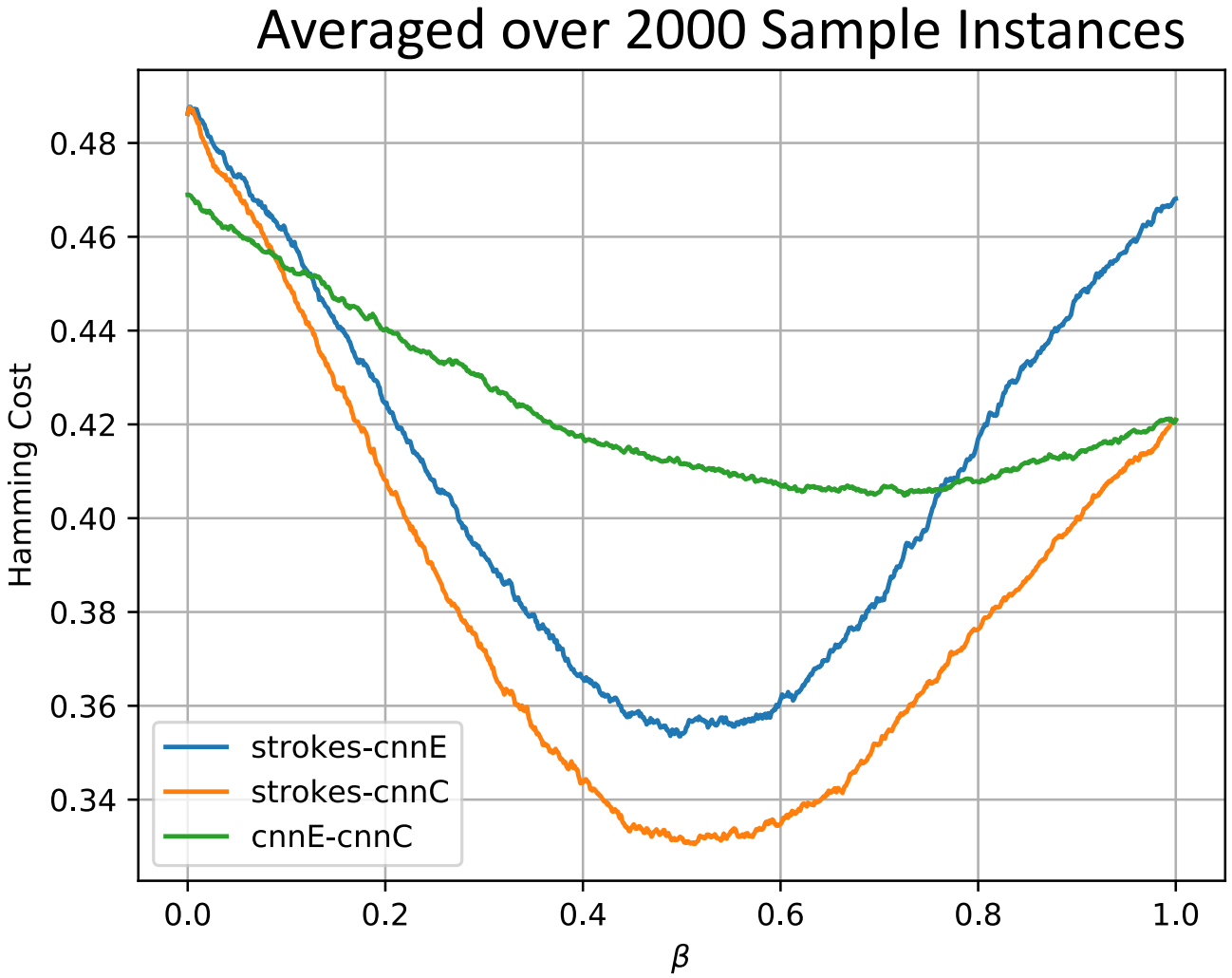
MNIST CNN Features:



Train CNN on MNIST.

Use feature embeddings for Omniglot.

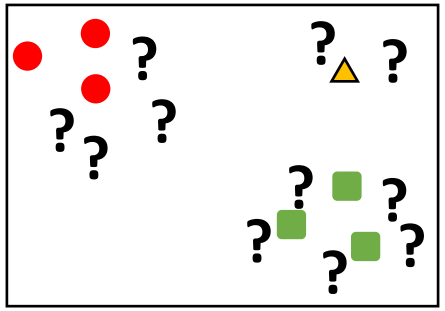
Euclidean / Cosine distance.



Conclusion

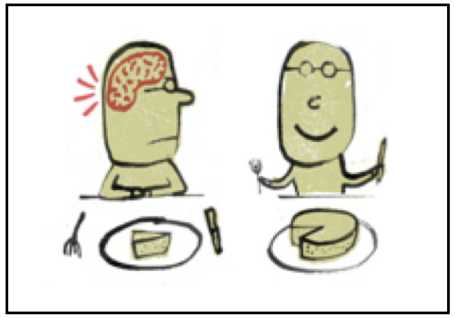
New theory and practice for modern machine learning.

Data Efficiency



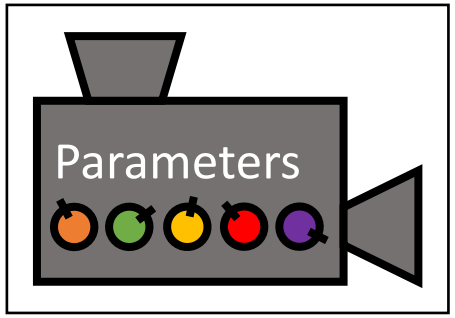
- Label efficient learning algorithms for multi-class problems.
- Exploit implicit assumptions of popular supervised algorithms.

Social Values



- Envy-freeness as a new notion of individual fairness in machine learning.
- Differentially private learning with piecewise Lipschitz losses.

Beyond Prediction



- Online learning formulations for Data-driven Algorithm Configuration.
- Boils down to online learning with piecewise Lipschitz losses.
- Dispersion-dependent regret bounds.

Thanks!