Data-driven Algorithm Design

Maria-Florina (Nina) Balcan
Carnegie Mellon University
Analysis and Design of Algorithms

Classic algo design: solve a worst case instance.

- Easy domains, have optimal poly time algos.
  E.g., sorting, shortest paths

- Most domains are hard.
  E.g., clustering, partitioning, subset selection, auction design, …

Data driven algo design: use learning & data for algo design.

- Suited when repeatedly solve instances of the same algo problem.
Data Driven Algorithm Design

Data driven algo design: use learning & data for algo design.

- Different methods work better in different settings.
- Large family of methods - what’s best in our application?

Prior work: largely empirical.

- Artificial Intelligence:
  [Horvitz-Ruan-Gomes-Kautz-Selman-Chickering, UAI 2001]
  [Xu-Hutter-Hoos-LeytonBrown, JAIR 2008]
- Computational Biology: E.g., [DeBlasio-Kececioglu, 2018]
- Game Theory: E.g., [Likhodedov and Sandholm, 2004]
Data Driven Algorithm Design

Data driven algo design: use learning & data for algo design.

- Different methods work better in different settings.
- Large family of methods - what’s best in our application?

Prior work: largely empirical.

Our Work: Data driven algos with formal guarantees.

- Several cases studies of widely used algo families.
- General principles (for distributional & online learning): push boundaries of algorithm design and machine learning.

Related in spirit to Hyperparameter tuning, AutoML, MetaLearning.
Structure of the Talk

• Data driven algo design as batch learning.
  • A formal framework.
  • Case studies: clustering, partitioning pbs, auction pbs.
  • General sample complexity theorem.
• Data driven algo design as online learning.
Example: Clustering Problems

Clustering: Given a set objects organize them into natural groups.

- E.g., cluster news articles, or web pages, or search results by topic.

- Or, cluster customers according to purchase history.

- Or, cluster images by who is in them.

Often need to solve such problems repeatedly.

- E.g., clustering news articles (Google news).
Clustering Problems

Clustering: Given a set objects (news articles, customer surveys, web pages, ...) organize them into natural groups.

Objective based clustering

$k$-means

**Input**: Set of objects $S, d$

**Output**: centers $\{c_1, c_2, ..., c_k\}$

To minimize $\sum_p \min_i d^2(p, c_i)$

Or minimize distance to ground-truth
Algorithm Design as Distributional Learning

**Goal:** given large family of algs, sample of typical instances from domain, find an algo that performs well on new instances from same domain.

[Gupea-Roughgarden, ITCS'16 & SICOMP'17]

**Large family** \( \mathcal{F} \) of algorithms

**Sample of i.i.d. typical inputs**

Facility location:

Clustering:
Sample Complexity of Algorithm Selection

**Goal:** given family of algos $F$, sample of typical instances from domain (unknown distr. $D$), find algo that performs well on new instances from $D$.

**Approach:** ERM, find $\hat{A}$ near optimal algorithm over the set of samples.

**Key Question:** Will $\hat{A}$ do well on future instances?

**Sample Complexity:** How large should our sample of typical instances be in order to guarantee good performance on new instances?
Statistical Learning Approach to AAD

Sample Complexity: How large should our sample of typical instances be in order to guarantee good performance on new instances?

\[ m = O(\text{dim}(F)/\epsilon^2) \] instances suffice to ensure generalizability

Challenge: “nearby” algs can have drastically different behavior.
Algorithm Design as Distributional Learning


Our results: New algorithm classes for a wide range of problems.

**Clustering:** Parametrized Linkage
[Balcan-Nagarajan-Vitercik-White, COLT 2017]
[Balcan-Dick-Lang, 2019]

**Parametrized Lloyds**
[Balcan-Dick-White, NeurIPS 2018]

Alignment pbs (e.g., string alignment): parametrized dynamic prog.

[Balcan-DeBlasio-Dick-Kingsford-Sandholm-Vitercik, 2019]
Our results: New algorithm classes for a wide range of problems.

- **Partitioning pbs via IQPs:** SDP + Rounding

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

  E.g., Max-Cut, Max-2SAT, Correlation Clustering

  \[
  \dim(F) = O(\log n)
  \]

- **MIPs:** Branch and Bound Techniques

  [Balcan-Dick-Sandholm-Vitercik, ICML’18]

  \[
  \text{Max } c \cdot x \\
  \text{s.t. } Ax = b \\
  x_i \in \{0,1\}, \forall i \in I
  \]

- **Automated mechanism design for revenue maximization**

  Parametrized VCG auctions, posted prices, lotteries.

  [Balcan-Sandholm-Vitercik, EC 2018]
Clustering: Linkage + Post-processing

Family of poly time 2-stage algorithms:

1. Greedy linkage-based algo to get hierarchy (tree) of clusters.

2. Fixed algo (e.g., DP or last k-merges) to select a good pruning.
Linkage Procedures for Hierarchical Clustering

**Bottom-Up (agglomerative)**

- Start with every point in its own cluster.
- Repeatedly merge the “closest” two clusters.

Different defs of “closest” give different algorithms.
Linkage Procedures for Hierarchical Clustering

Have a **distance** measure on pairs of objects.

\[ d(x, y) - \text{distance between } x \text{ and } y \]

E.g., # keywords in common, edit distance, etc

- **Single linkage**:  \( \text{dist}(A, B) = \min_{x \in A, x' \in B} \text{dist}(x, x') \)

- **Complete linkage**:  \( \text{dist}(A, B) = \max_{x \in A, x' \in B} \text{dist}(x, x') \)

- **Parametrized family, \( \alpha \)-weighted linkage**:  

\[
\text{dist}_\alpha(A, B) = (1 - \alpha) \min_{x \in A, x' \in B} d(x, x') + \alpha \max_{x \in A, x' \in B} d(x, x')
\]
Clustering: Linkage + Dynamic Programming

Our Results: \(\alpha\)-weighted linkage + Post-processing

- Pseudo-dimension is \(O(\log n)\), so small sample complexity.
- Given sample \(S\), find best algo from this family in poly time.

Key Technical Challenge: small changes to the parameters of the algo can lead to radical changes in the tree or clustering produced.

Problem: a single change to an early decision by the linkage algo, can snowball and produce large changes later on.
Claim: Pseudo-dim of $\alpha$-weighted linkage + Post-process is $O(\log n)$.

Key fact: If we fix a clustering instance of $n$ pts and vary $\alpha$, at most $O(n^8)$ switching points where behavior on that instance changes.

So, the cost function is piecewise-constant with at most $O(n^8)$ pieces.
Claim: Pseudo-dim of $\alpha$-weighted linkage + Post-process is $O(\log n)$.

**Key fact:** If we fix a clustering instance of $n$ pts and vary $\alpha$, at most $O(n^8)$ switching points where behavior on that instance changes.

Key idea:

- For a given $\alpha$, which will merge first, $N_1$ and $N_2$, or $N_3$ and $N_4$?
- Depends on which of $\alpha d(p, q) + (1 - \alpha)d(p', q')$ or $\alpha d(r, s) + (1 - \alpha)d(r', s')$ is smaller.
- An interval boundary an equality for 8 points, so $O(n^8)$ interval boundaries.
Claim: Pseudo-dim of $\alpha$-weighted linkage + Post-process is $O(\log n)$.

Key idea: For $m$ clustering instances of $n$ points, $O(mn^8)$ patterns.

- Pseudo-dim largest $m$ for which $2^m$ patterns achievable.
- So, solve for $2^m \leq m n^8$. Pseudo-dimension is $O(\log n)$. 
Clustering: Linkage + Dynamic Programming

**Claim:** Pseudo-dimension of \( \alpha \)-weighted linkage + DP is \( O(\log n) \), so small sample complexity.

For \( N = O(\log n / \varepsilon^2) \), w.h.p. expected performance cost of best \( \alpha \) over the sample is \( \varepsilon \)-close to optimal over the distribution.

**Algorithm**

- Solve for all \( \alpha \) intervals over the sample
  \[ \alpha \in \mathbb{R} \]
- Find the \( \alpha \) interval with the smallest empirical cost

**Claim:** Given sample \( S \), can find best algo from this family in poly time.
Learning Both Distance and Linkage Criteria

[Balcan-Dick-Lang, 2019]

- Often different types of distance metrics.
  - Captioned images, $d_0$ image info, $d_1$ caption info.
  - Handwritten images: $d_0$ pixel info (CNN embeddings), $d_1$ stroke info.
- Family of Metrics: Given $d_0$ and $d_1$, define
  \[
  d_\beta(x, x') = (1 - \beta) \cdot d_0(x, x') + \beta \cdot d_1(x, x')
  \]

Parametrized $(\alpha, \beta)$-weighted linkage ($\alpha$ interpolation between single and complete linkage and $\beta$ interpolation between two metrics):

\[
\text{dist}_\alpha(A, B; d_\beta) = (1 - \alpha) \min_{x \in A, x' \in B} d_\beta(x, x') + \alpha \max_{x \in A, x' \in B} d_\beta(x, x')
\]
Learning Both Distance and Linkage Criteria

Claim: Pseudo-dim. of \((\alpha, \beta)\)-weighted linkage is \(O(\log n)\).

Key fact: Fix instance of \(n\) pts; vary \(\alpha, \beta\), partition space with \(O(n^8)\) linear, quadratic equations s.t. within each region, get same cluster tree.

Key Idea:

1. \(O(n^4)\) linear sep. s.t. all \(\beta_1, \beta_2\) in same region, \(d_{\beta_1}\) and \(d_{\beta_2}\) agree on order of distances between all \(n\) pts.

   \[
   \text{Given } \beta, \text{ decision whether } d_{\beta}(a, b) \text{ greater than } d_{\beta}(a', b') \text{ depends on which } (1 - \beta)d_0(a, b) + \beta d_1(a, b) \text{ or } (1 - \beta)d_0(a', b') + \beta d_1(a', b') \text{ is greater}
   \]

2. Fix region, for sets \(A, B\), all \(\beta\) agree on \(a_1, b_1 = \arg\min_{a \in A, b \in B} d_{\beta}(a, b), a_2, b_2 = \arg\max_{a \in A, b \in B} d_{\beta}(a, b)\).

   So, \(\text{dist}_{\alpha}(A, B; d_{\beta})\) is a quadratic fn of \(\alpha, \beta\):

   \[
   \text{dist}_{\alpha}(A, B; d_{\beta}) = (1 - \alpha)[(1 - \beta)d_0(a_1, b_1) + \beta d_1(a_1, b_1)] + \alpha[(1 - \beta)d_0(a_2, b_2) + \beta d_1(a_2, b_2)]
   \]

   \(\text{Given } \alpha, \text{ decision to merge } A, B \text{ or } A', B' \text{ quadratic boundary, defined by } 8 \text{ pts.}\)
Clustering Subsets of Omniglot

Graph showing the relationship between Hamming Cost and Stroke Distance with MNIST Features. The graph indicates that 

- For \( \beta = 1 \), Error = 42.1%
- For the optimal value \( \beta^* = 0.514 \), Error = 33.0%

Improvement of 9.1% is observed.

\( \beta \) values and corresponding Error percentages are marked on the graph.
Partitioning Problems via IQPs

Our Results: SDP + s-linear rounding

Pseudo-dimension is $O(\log n)$, so small sample complexity.

Key idea: expected IQP objective value is piecewise quadratic in $\frac{1}{s}$ with $n$ boundaries.
Data-driven Mechanism Design

- **Similar ideas** for sample complex. of data-driven mechanism design for revenue maximization. [Balcan-Sandholm-Vitercik, EC’18]

- Pseudo-dim of \(\{\text{revenue}_M: M \in \mathcal{M}\}\) for multi-item multi-buyer settings:
  - Many families: second-price auctions with reserves, posted pricing, two-part tariffs, parametrized VCG auctions, etc.

- **Key insight:** dual function sufficiently structured.
  - For a fixed set of bids, revenue is piecewise linear fnc of parameters.

2nd-price auction with reserve

Posted price mechanisms

![Graph showing revenue for 2nd highest bid and highest bid with reserve](image)

![Graph showing revenue for posted price mechanisms with Price](image)
Want to prove that for all algorithm parameters $\alpha$:

$$\frac{1}{|S|} \sum_{I \in S} \text{cost}_\alpha(I) \text{ close to } \mathbb{E}[\text{cost}_\alpha(I)].$$

Function class whose complexity want to control: $\{\text{cost}_\alpha: \text{parameter } \alpha\}$.

Proof takes advantage of structure of dual class $\{\text{cost}_I: \text{instances } I\}$.

Theorem: Suppose for each $\text{cost}_I(\alpha)$ there are $\leq N$ boundary fns $f_1, f_2, \ldots \in F$ s. t. within each region defined by them, $\exists g \in G$ s. t. $\text{cost}_I(\alpha) = g(\alpha)$.

$$\text{Pdim}(\{\text{cost}_\alpha(I)\}) = O((d_F^* + d_G^*) \log (d_F^* + d_G^*) + d_F^* \log N)$$

$d_F^* = \text{VCdim of dual of } F$, $d_G^* = \text{Pdim of dual of } G$. 
Theorem: Suppose for each \( \text{cost}_1(\alpha) \) there are \( \leq N \) boundary fns \( f_1, f_2, \ldots \in F \) s.t. within each region defined by them, \( \exists g \in G \) s.t. \( \text{cost}_1(\alpha) = g(\alpha) \).

\[
P\text{dim}(\{\text{cost}_\alpha(I)\}) = O((d_{F^*} + d_{G^*}) \log(d_{F^*} + d_{G^*}) + d_{F^*} \log N)
\]

\( d_{F^*} = \text{VCdim of dual of } F, \quad d_{G^*} = \text{Pdim of dual of } G. \)
General Sample Complexity via Dual Classes

Theorem: Suppose for each $\text{cost}_1(\alpha)$ there are $\leq N$ boundary fns $f_1, f_2, \ldots \in F$ s.t. within each region defined by them, $\exists g \in G$ s.t.
$\text{cost}_1(\alpha) = g(\alpha)$.

$$\text{Pdim}(\{\text{cost}_\alpha(I)\}) = O((d_{F^*} + d_{G^*}) \log(d_{F^*} + d_{G^*}) + d_{F^*} \log N)$$

$d_{F^*} = \text{VCdim of dual of F}$, $d_{G^*} = \text{Pdim of dual of G}$.

**VCdim(F):** fix N pts. Bound # of labelings of these pts by $f \in F$ via Sauer’s lemma in terms of VCdim(F).

**VCdim(F*):** fix N fns, look at # regions. In the dual, a point labels a function, so direct correspondence between the shattering coefficient of the dual and the number of regions induced by these fns. Just use Sauer’s lemma in terms of VCdim(F*).
General Sample Complexity via Dual Classes

Theorem: Suppose for each \( \text{cost}_i(\alpha) \) there are \( \leq N \) boundary fns \( f_1, f_2, \ldots \in F \) s.t. within each region defined by them, \( \exists g \in G \) s.t. \( \text{cost}_i(\alpha) = g(\alpha) \).

\[
Pdim(\{\text{cost}_\alpha(I)\}) = O((d_F^* + d_G^*) \log(d_F^* + d_G^*) + d_F^* \log N)
\]

\[d_F^* = \text{VCdim of dual of } F, \quad d_G^* = Pdim \text{ of dual of } G.
\]

Proof:

- Fix \( D \) instances \( I_1, \ldots, I_D \) and \( D \) thresholds \( z_1, \ldots, z_D \). Bound \# sign patterns \( (\text{cost}_\alpha(I_1), \ldots, \text{cost}_\alpha(I_D)) \) ranging over all \( \alpha \). Equivalently, \( (\text{cost}_{I_1}(\alpha), \ldots, \text{cost}_{I_D}(\alpha)) \).

- Use VCdim of \( F^* \), bound \# of regions induced by \( \text{cost}_{I_1}(\alpha), \ldots, \text{cost}_{I_D}(\alpha) : (eND)^{d_F^*} \).

- On a region, exist \( g_{I_1}, \ldots, g_{I_D} \) s.t., \( (\text{cost}_{I_1}(\alpha), \ldots, \text{cost}_{I_D}(\alpha)) = (g_{I_1}(\alpha), \ldots, g_{I_D}(\alpha)), \) which equals \( (\alpha(g_{I_1}), \ldots, \alpha(g_{I_D})) \). These are fns in dual class of \( G \). Sauer's lemma on \( G^* \), bounds \# of sign patterns in that region by \( (eD)^{d_G^*} \).

- Combining, total of \( (eND)^{d_F^*}(eD)^{d_G^*} \). Set to \( 2^D \) and solve.
Online Algorithm Selection

- So far, batch setting: collection of typical instances given upfront.

- **Challenge**: scoring fns non-convex, with lots of discontinuities.

  Cannot use known techniques.

- Identify general properties (piecewise Lipschitz fns with dispersed discontinuities) sufficient for strong bounds.

  - Show these properties hold for many alg. selection pbs.
Dispersion, Sufficient Condition for No-Regret

\{u_1(\cdot), \ldots, u_T(\cdot)\} is \((w, k)\)-dispersed if any ball of radius \(w\) contains boundaries for at most \(k\) of the \(u_i\).
Summary and Discussion

- **Strong performance guarantees for data driven algorithm selection for combinatorial problems.**

- **Provide and exploit structural properties of dual class for good sample complexity and regret bounds.**

- **Learning theory: techniques of independent interest beyond algorithm selection.**
Discussion, Open Problems

- Analyze other widely used classes of algorithmic paradigms.
- Other learning models (e.g., one shot, domain adaptation, RL).
- Explore connections to program synthesis; automated algo design.
- Explore connections to Hyperparameter tuning, AutoML, MetaLearning.

Use our insights for pbs studied in these settings (e.g., tuning hyper-parameters in deep nets)