Efficient Optimization of Partition Scan Statistics via the Consecutive Partitions Property

Charles A. Pehlivanian, Ph.D.[1]
Daniel B. Neill, Ph.D.[2]

[1]New York, NY

[2]New York University

July 22, 2021

charles.a.pehlivanian@jpmchase.com
pehlivaniancharles@gmail.com

Summary

1 Partition scan statistics – moving from t = 2 to t>= 2, formulation of problem as combinatorial optimization over size t partitions

2 Derivation of partition scan statistics objective functions for t >= 2

- Risk partitioning objective
- Multiple clustering objective

3 Derivation of partition scan statistics objective functions for exponential familes

- Risk partitioning objective as f-divergence
- Multiple clustering objective as Bregman divergence

4 Solution methods, dynamic programming approach

5 Simulation results

6 Future directions

Partition Scan Statistics – Motivation

Existing approaches to pattern and event detection using scan statistics can be thought of as partitioning a dataset into 2 disjoint sets – the detected set and the remainder.

- [**Kulldorff, 1997**]: Spatial scan statistic, "population-based", splits data into a detected cluster with risk q_in, and remainder with risk q_out, q_in > q_out. Call this "risk partitioning".
- [**Neill, 2005**]: Expectation-based scan statistic, detected cluster has risk q_in > 1, remainder cluster has risk q = 1. Call this "cluster detection".

Our work expands both of these ideas (risk partitioning and cluster detection) to the case of more that two subsets of interest, while maintaining above results as special cases.

- Multiple risk partitioning
- Multiple cluster detection

Partition Scan Statistics – Motivation

The Kulldorff approach scans exhaustively over circular subregions of the data (O(n^2) in operations), although this does not provide flexibility for detecting irregularly-shaped clusters.

"Fast subset scanning" approaches starting with Neill (2012) allow irregular clusters by solving unconstrained optimization problem in linear time, the "fast subset scanning" approach.

Holds for objective functions satisfying the **LTSS** (**Linear Time Subset Scanning**) property. This property can be used as a building block for some constrained optimization problems – spatial cluster detection, pattern detection on graphs, etc.

LTSS approach: After sorting the data by an appropriate priority function, the top-performing subset is the set of top k elements, for some k. Just examine all the k's between 1 and n.

We show the LTSS property can be generalized to partitions of the data.

Partition Scan Statistics – Motivation

Consecutive Partitions Property (**CPP**): After sorting the data by an appropriate priority function, the optimal partitioning of the data consists of sets of consecutively-ranked items.

We will provide sufficient conditions for which CPP and similar properties hold.

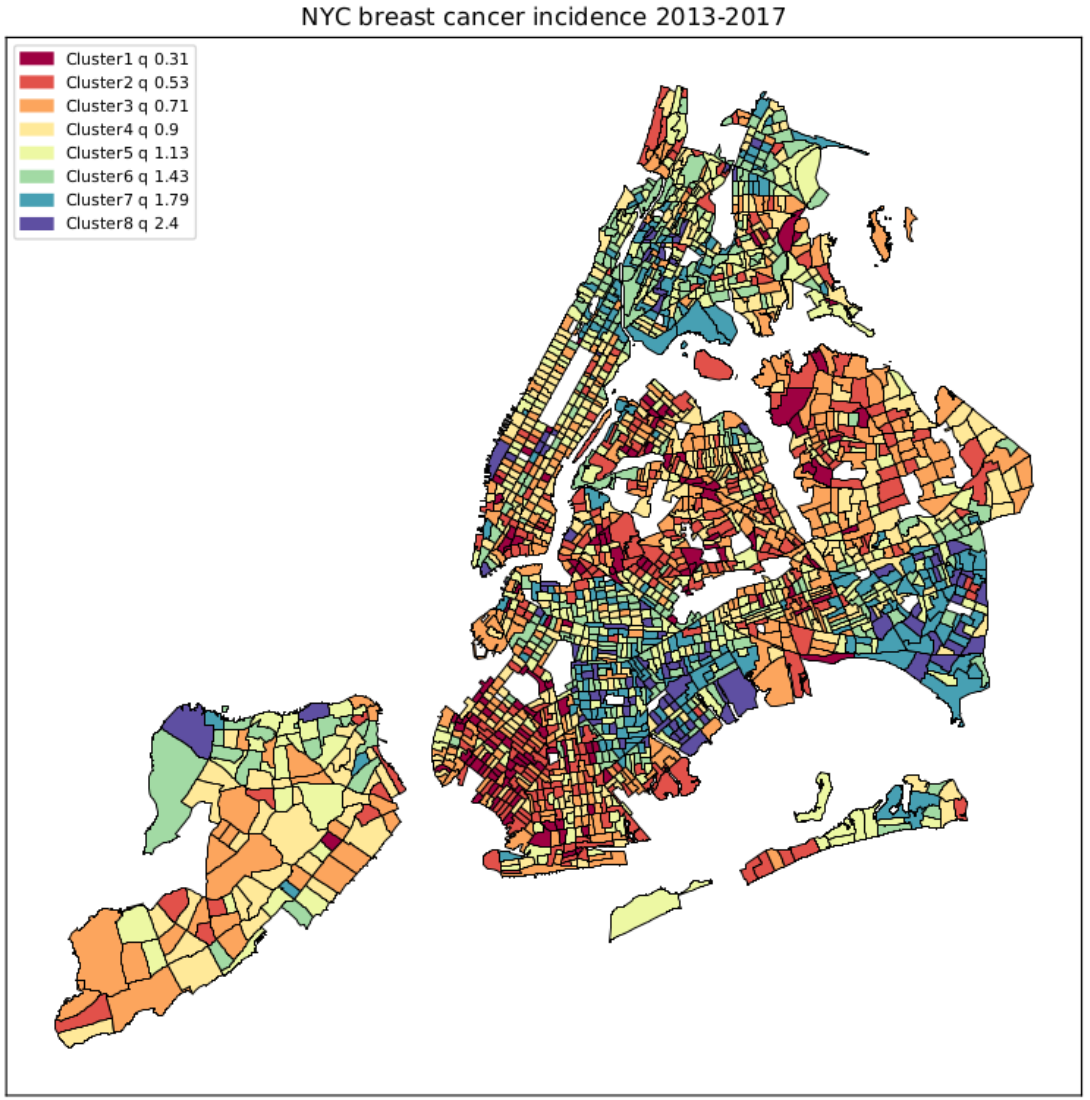
For n = number of data points, t = number of subsets in partition, the CPP translates to an $O(n^{(t-1)})$ algorithm for finding optimal partitions. We will explain a dynamic programming approach which runs in $(O((n^2)t))$ time, for all n regardless of t, allowing for consideration of large-scaled problems.

Partition Scan Statistics – Motivation

Bell_n_k(50, 8)
3.504173e+40

Bell_n_k(100, 8)
5.052108e+85
.
.
.

NYC census tract data:
n=2089, t=8.



Partition Scan Statistics – Preliminaries

Setup:

$n \in \mathbf{N}$, $\mathcal{V} = \{1, \ldots, n\}$,

Real sequences: $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_n\}$, $y_i > 0$, for all i.

Dataset: $\mathcal{D} = \mathcal{D}_{X,Y} = \{(x_i, y_i)\}_{i=1}^{n}$

Assume in what follows that $\mathcal{D}$ is ordered by the priority function $g(x, y) = \frac{x}{y}$.

Usual interpretation:

i : spatial region,

x_i : occurrences, counts, measurements,

y_i : baselines, expectations, population.

Goal

Find the optimal partitioning of $\mathcal{V}$ that maximizes a cumulative score on subsets.

Partition Scan Statistics – Risk Partitioning

The objective function F, a real-valued set function $F\colon 2^{\mathcal{V}} \to \mathbf{R}$, comes from an ambient *score* function f:

$f\colon \mathbf{R} \times \mathbf{R}^{+} \to \mathbf{R}$,

continuous on any cone $\mathcal{W}$ eminating from the origin,

$\lim_{(x,y)\in\mathcal{W}\to(0,0)} f(x,y) = 0$.

f can be constrained to a set function on $2^{\mathcal{V}}$ by

$$F(S) = f(\sum_{i\in S} x_i, \sum_{i\in S} y_i),\ S \in \mathcal{V}$$

The goal is then to find an optimal partition $\mathcal{P} = \{S_1, \ldots, S_t\}$

$$\mathcal{P}^* = \underset{\mathcal{P}=\{S_1,\ldots,S_t\}}{\operatorname{argmax}} \sum_{j=1\ldots t} F(S_j) = \underset{\mathcal{P}=\{S_1,\ldots,S_t\}}{\operatorname{argmax}} \sum_{j=1\ldots t} f\left(\sum_{i\in S_j} x_i, \sum_{i\in S_j} y_i\right) \quad (1)$$

For $S \in \mathcal{V}$ we call the point $p_S = (\sum_{i\in S} x_i, \sum_{i\in S} y_i) \in \mathbf{R}$ the *partition polytope point* for S.

Partition Scan Statistics – Risk Partitioning

Partition Scan Statistics - expressed as log likelihood ratios, parametric assumption: usually $x_i \sim \Psi = \{Poisson, Gaussian\}$

Risk Partitioning Objective:

$$H_0 : x_i \sim \Psi(\cdot; \mu = q_{all}\mu_i) \; \forall s_i,$$
$$H_1(\mathcal{P}) : x_i \sim \Psi(\cdot; \mu = q_j\mu_i) \; \forall s_i \in S_j,$$

where $\mathcal{P} = \{S_1, \ldots, S_t\}$ is a partitioning of $\mathcal{D}$ of size t. Then

$$F(P) = F(\{S_1, \ldots, S_t\}) = \log \frac{\max_{q_1,\ldots,q_t} \prod_{j=1\ldots t} \prod_{s_i \in S_j} \Pr(x_i \mid x_i \sim \psi(\cdot; \mu = q_j\mu_i))}{\max_{q_{all}} \prod_{s_i} \Pr(x_i \mid x_i \sim \psi(\cdot; \mu = q_{all}\mu_i))}.$$

- Generalizes Kulldorff spatial scan statistic,
- Seeks to partition the data optimally into equal risk buckets.

Partition Scan Statistics – Multiple Clustering

Multiple Cluster Objective:

$$H_0 : x_i \sim \Psi(\cdot; \mu = \mu_i)\ \forall s_i,$$
$$H_1(\mathcal{P}) : x_i \sim \Psi(\cdot; \mu = q_j \mu_i)\ , q_j > 1, \forall s_i \in S_j,$$

where $\mathcal{P} = \{S_1, \ldots, S_t\}$ is a partitioning of $\mathcal{D}$ of size t. Then

$$F_{clust}(P) = F_{clust}(\{S_1, \ldots, S_t\}) = \max_{q_2, \ldots, q_t > 1} \log \prod_{j=2 \ldots t} \prod_{s_i \in S_j} \frac{\Pr(x_i \mid x_i \sim \psi(\cdot; \mu = q_j \mu_i))}{\Pr(x_i \mid x_i \sim \psi(\cdot; \mu = \mu_i))}.$$

- Generalizes expectation-based scan statistic,
- Seeks to find the (at most) $t - 1$ highest scoring clusters in the data (empty clusters are allowed).

Partition Scan Statistics – Exponential Families

Partition scan functions can be calculated easily for distributions from separable exponential families

Assume Ψ is from a separable exponential family

$$p_\Psi(x) = \exp\left(\langle x, \theta\rangle - \Lambda(\theta)\right) p_\theta(x)$$

Here Λ is the *log partition function* or *cumulant function*, it is uniquely determined up to an additive constant.

Partition Scan Statistics – Risk Partitioning Score Function for Exponential Families

Risk Partitioning Objective:

$$F(P) = B_{all} D_f \left(\vec{C} \parallel \vec{B} \right),$$

where

$\phi_0 = \Lambda^*$, the *Legendre-Fenchel convex conjugate*,

$\vec{C}$ and $\vec{B}$ are the normalized count and baseline vectors $\frac{1}{C_{all}} \langle C_1, \ldots, C_t \rangle$ and $\frac{1}{B_{all}} \langle B_1, \ldots, B_t \rangle$,

D_f is the f-divergence, $D_f(P \parallel Q) = \sum_{j=1 \ldots t} Q_j f\left(\frac{P_j}{Q_j}\right)$

$f(q) = \phi_0 \left(q \frac{C_{all}}{B_{all}} \right) - \phi_0 \left(\frac{C_{all}}{B_{all}} \right)$.

Equivalently,

$$F(P) = \sum_{j=1}^{t} f(C_j, B_j) - f(C_{all}, B_{all}),$$

for $f(x, y) = y \phi_0 \left(\frac{x}{y} \right)$

Example: $\Psi \sim$ Poisson, $\phi_0(q) = q \log q$, $f(x, y) = x \log \frac{x}{y}$

Partition Scan Statistics – Multiple Clustering Score Function for Exponential Families

Multiple Clustering Objective:

$$F(P) = \sum_{j=2}^{t} f(C_j, B_j),$$

where

ϕ_0, C_j, B_j as above,

$$f(x,y) = \begin{cases} yD_{\phi_0}(\frac{x}{y}, 1), & x > y \\ 0, & \text{otherwise} \end{cases}$$

D_f is the Bregman divergence, $D_{\phi_0}(x,y) = \phi_0(x) - \phi_0(y) - \langle x - y, \nabla\phi_0(y)\rangle$

Example: $\Psi \sim$ Poisson, $\phi_0(q) = q \log q$, $f(x,y) = \begin{cases} x \log \frac{x}{y} + x - y, & x > y \\ 0, & \text{otherwise} \end{cases}$

Solving for the Maximal Partition

Solving Equation 1

$$\mathcal{P}^* = \underset{\mathcal{P}=\{S_1,\ldots,S_t\}}{\operatorname{argmax}} \sum_{j=1\ldots t} F(S_j) = \underset{\mathcal{P}=\{S_1,\ldots,S_t\}}{\operatorname{argmax}} \sum_{j=1\ldots t} f\left(\sum_{i\in S_j} x_i, \sum_{i\in S_j} y_i\right)$$

Recall F is obtained from the given f by restriction to partition polytope points;

$$F(S) = f(\sum_{j=1}^{t} x_i, \sum_{j=1}^{t} y_i) = f(p_S), \text{ where } p_S = (\sum_{j=1}^{t} x_i, \sum_{j=1}^{t} y_i), S \in \mathcal{V}$$

We will find conditions under which the following properties hold:

CPP ("Consecutive Parition Property"): For all D, for all $1 \leq t \leq |\mathcal{D}|$, the highest scoring partition in Equation 1 is consecutive.

WCPP ("Weak Consecutive Partition Property") For all D, for all $1 \leq t \leq |\mathcal{D}|$, the highest scoring partition of size $t' \leq t$ in Equation 1 is consecutive.

In this way we dramatically reduce the size of the feasible space...

Solving for the Maximal Partition – Notion of Consecutive Partitions

Define the subset types - for $\mathcal{V} = \{1, \ldots, 10\}$, e.g.

consecutive	$\{1, 2, 3, 4, 5\}$
consecutive splitting	$\{3, 4, 5, 6\}$
consecutive nonsplitting	$\{8, 9, 10\}$
ascending consecutive (nonsplitting)	$\{1, 2\}$
descending consecutive (nonsplitting)	$\{6, 7, 8, 9, 10\}$
singleton nonsplitting	$\{10\}$
singleton splitting	$\{4\}$

Define

$$\mathcal{U} = \mathcal{U}_n = \{S \subseteq \mathcal{V} : S \text{ is consecutive nonsplitting}\},$$
$$\underline{\mathcal{U}} = \mathcal{U}_n = \{S \subseteq \mathcal{V} \setminus \{\emptyset, \mathcal{V}\} : S \text{ is consecutive nonsplitting}\},$$
$$\underline{\mathcal{S}} = \underline{\mathcal{S}}_n = \{S \subseteq \mathcal{V} \setminus \{\emptyset, \mathcal{V}\} : (S, \mathcal{V} \setminus S) \text{ form a singleton splitting pair}\}$$

and for any partition $\mathcal{P} = \{S_1, \ldots, S_t\}$ define the embedding

$$\Pi(\mathcal{P}) = \{p_{S_1}, \ldots, p_{S_t}\}$$

that associates the partition with the set of partition polytope points of its subsets.

Solving for the Maximal Partition – Description of Partition Polytope

The *partition polytope* $\mathcal{C}$ and the *constrained partition polytope* $\underline{\mathcal{C}}$ are defined by the convex hulls $\mathcal{C} = \hat{\mathcal{P}}$ and $\underline{\mathcal{C}} = \hat{\underline{\mathcal{P}}}$ respectively, where

$$\mathcal{P} = \{p_S = (\sum_{i \in S} x_i, \sum_{i \in S} y_i) : S \subseteq \mathcal{V}\}$$

$$\underline{\mathcal{P}} = \{p_S = (\sum_{i \in S} x_i, \sum_{i \in S} y_i) : S \subseteq \mathcal{V}, S \neq \emptyset, S \neq \mathcal{V}\}$$

The extreme points $\mathcal{E}$, $\underline{\mathcal{E}}$ of $\mathcal{C}$, $\underline{\mathcal{C}}$ have a rigid structure as given in the following proposition:

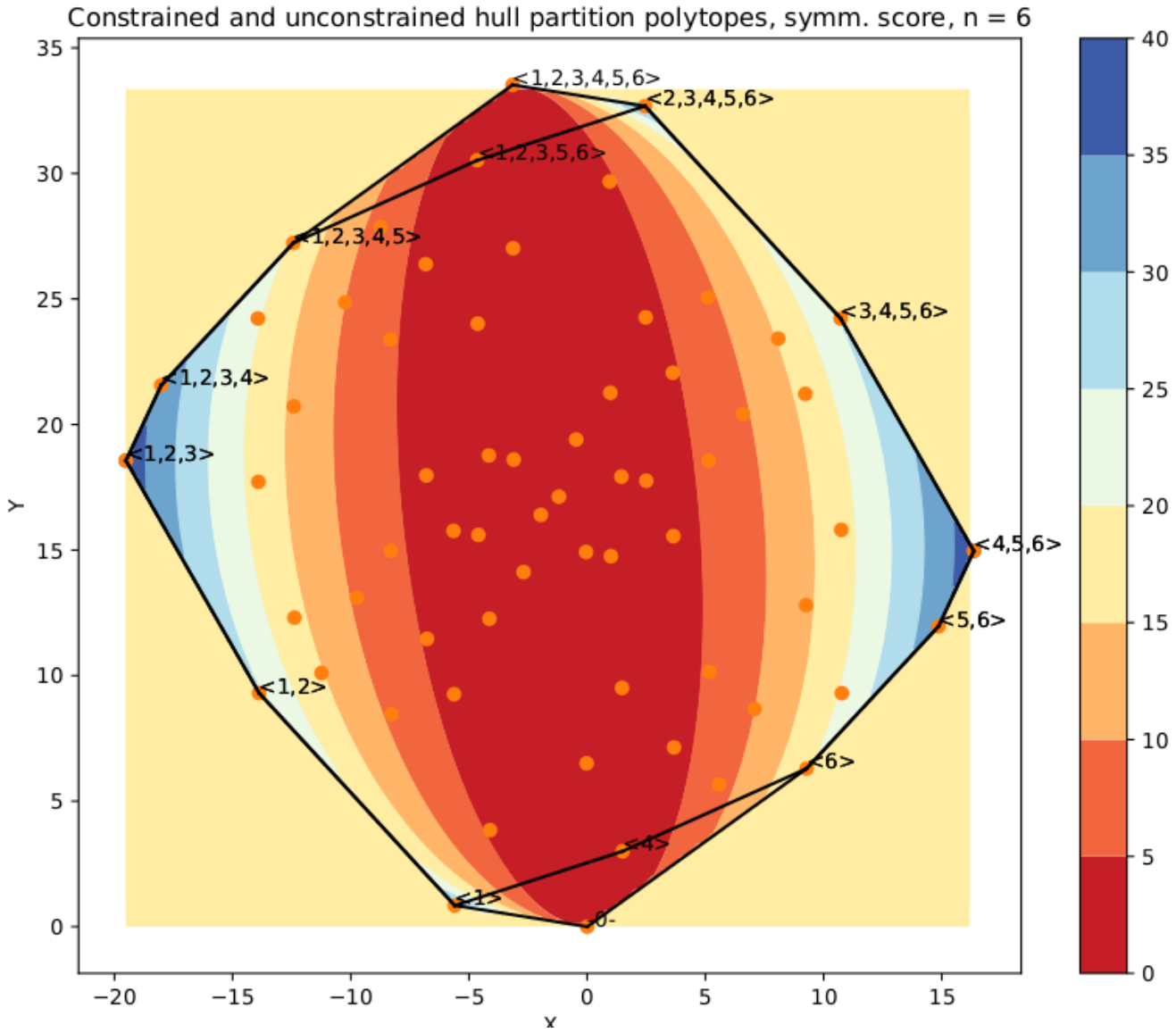
Proposition 1. *Let $\mathcal{D} = \{(x, y)\}$ be a dataset ordered by the standard priority function $g(x, y) = \frac{x}{y}$. Let $\mathcal{E}$ and $\underline{\mathcal{E}}$ be the sets of extreme points of the partition polytope $\mathcal{C}$ and the constrained partition polytope $\underline{\mathcal{C}}$ respectively. Then*

(i) $\mathcal{E} \subseteq \Pi(\mathcal{U})$,

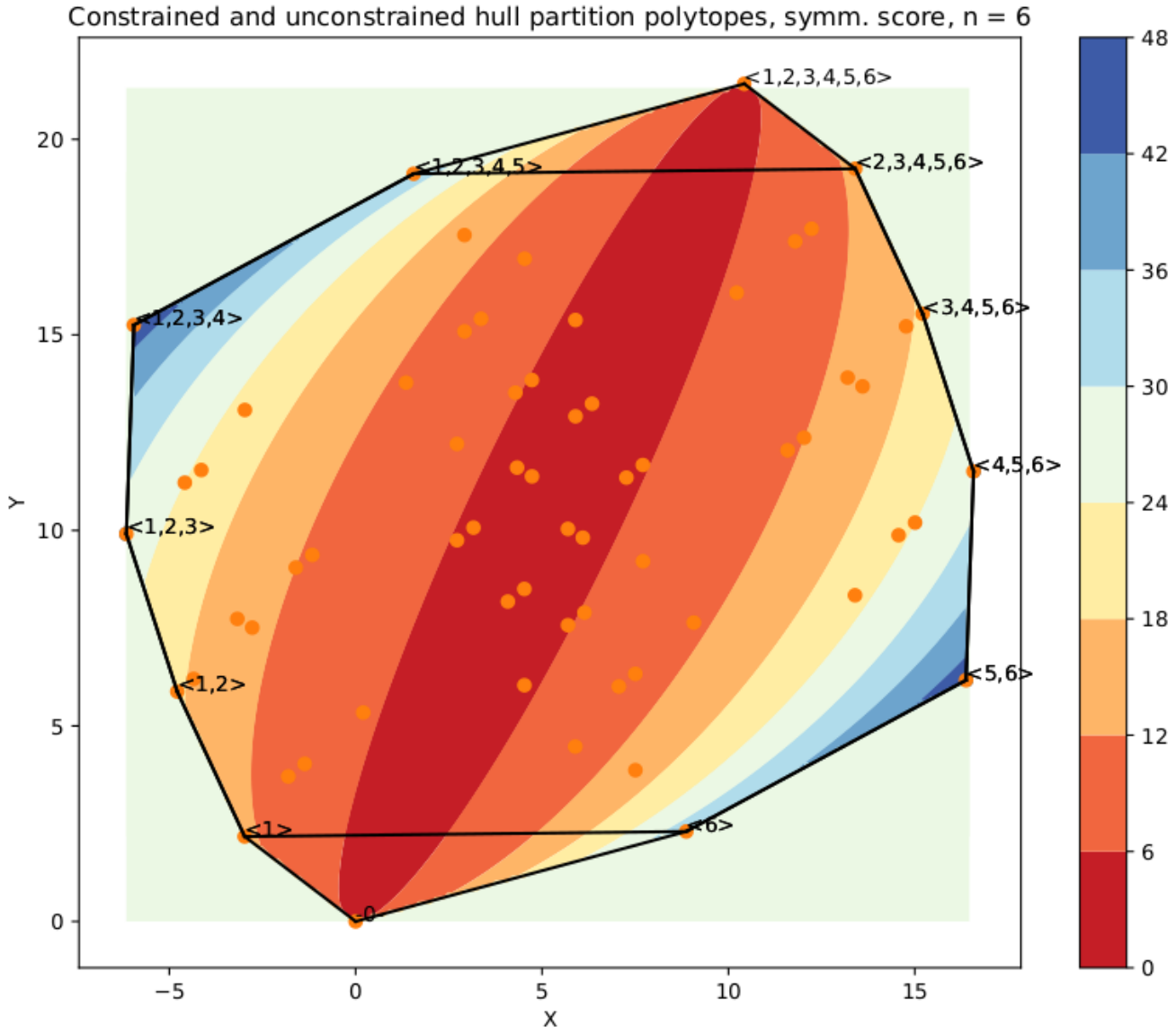
(ii) $\underline{\mathcal{E}} \subseteq \Pi(\underline{\mathcal{U}}) \cup \Pi(\underline{\mathcal{S}})$.

We use the interplay between $\underline{\mathcal{C}}$, $\mathcal{C}$ to solve the main maximization problem...

Solving for the Maximal Partition – Description of Partition Polytope



Solving for the Maximal Partition – Description of Partition Polytope



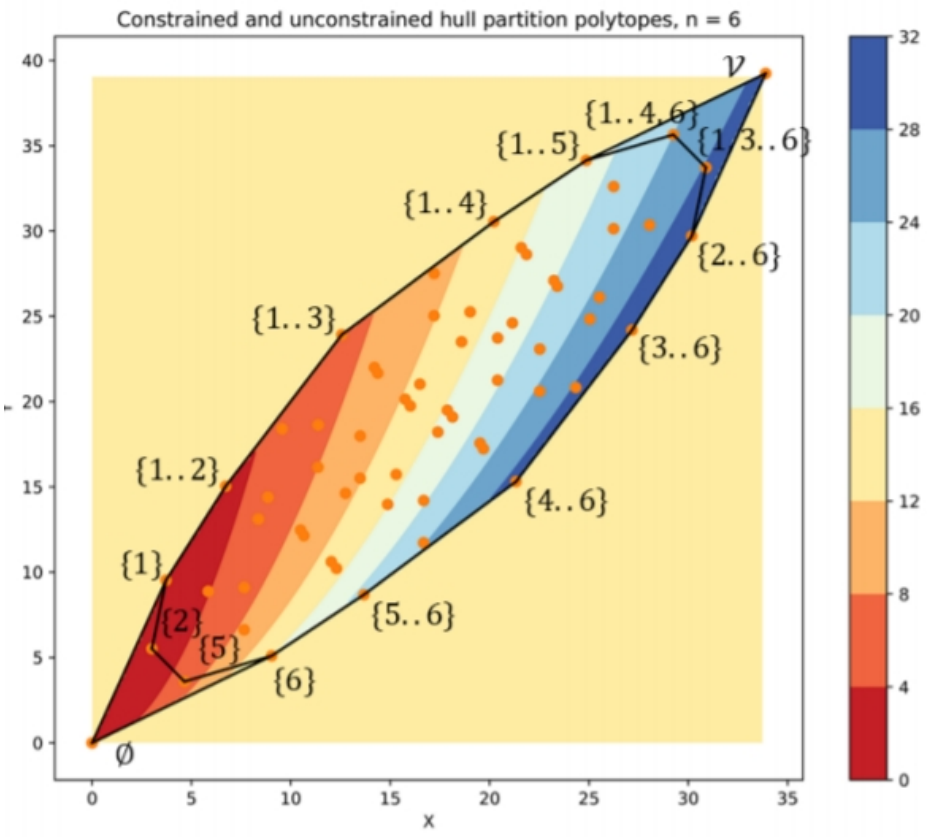
Solving for the Maximal Partition – Description of Partition Polytope

$$\mathcal{P}^* = \underset{\mathcal{P}=\{S_1,\ldots,S_t\}}{\operatorname{argmax}} \sum_{j=1\ldots t} F(S_j) = \underset{\mathcal{P}=\{S_1,\ldots,S_t\}}{\operatorname{argmax}} \sum_{j=1\ldots t} f\left(\sum_{i\in S_j} x_i, \sum_{i\in S_j} y_i\right)$$

$\overline{f}: \mathcal{C} \subseteq \mathbf{R}\times\mathbf{R}^+ \to \mathbf{R}$ by

$$\overline{f}(x,y) = \begin{cases} f(x,y) + f\left(C_x^n - x, C_y^n - y\right), & (x,y) \in \mathcal{C} \setminus \left\{(0,0), \left(C_x^n, C_y^n\right)\right\} \\ f\left(C_x^n, C_y^n\right), & (x,y) \in \left\{(0,0), \left(C_x^n, C_y^n\right)\right\} \end{cases}$$

where $C_x^n = \sum_{i=1\ldots n} x_i$, $C_y^n = \sum_{i=1\ldots n} y_i$.



Solving for the Maximal Partition - Solution

Upshot:

$$f \text{ convex, subadditive} \implies f \text{ satisfies } \mathbf{CPP}$$
$$f \text{ convex} \implies f \text{ satisfies } \mathbf{WCPP}$$

Furthermore, there is a dynamic programming approach for finding $\mathcal{P}^*$ that runs in $O(n^2 t)$ time.

Solving for the Maximal Partition – Dynamic Programming Solution

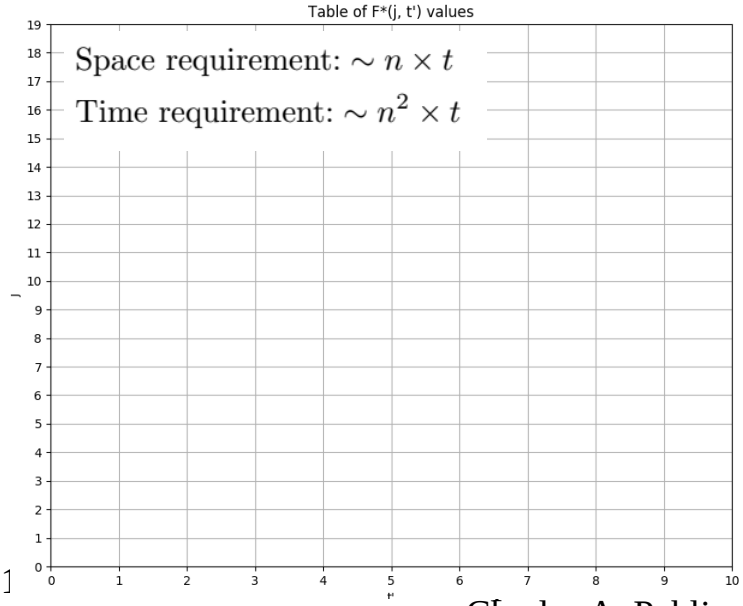
Given n, t, we wish to solve the optimization program in Eqution 1, armed with the knowledge that the solution is a consecutive partition (CPP holds).

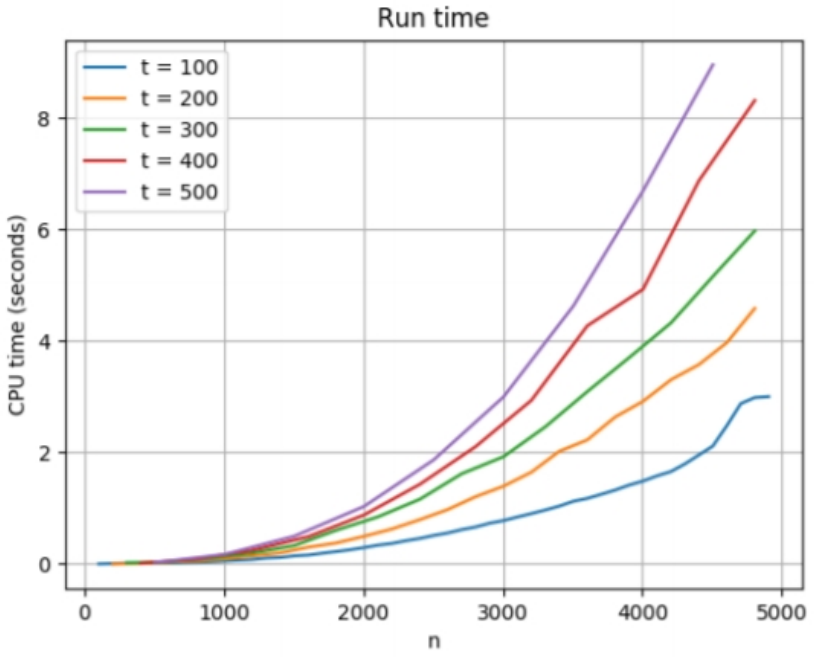
Define, for all $1 \leq j \leq n$, $1 \leq t' \leq t$,

$$F^*(j,t') = \text{max score obtained for dividing } \{j,\ldots,n\} \text{ into } t' \text{subsets}$$

Then

$$F^*(j,t') = \max_{k \in \{j+1,\ldots,(n+1-t')\}} \left(f(\sum_{i=j}^{k} x_i, \sum_{i=j}^{k} y_i) + F^*(k+1, t'-1) \right)$$





Simulation Results – Risk Partitioning

Setup: Simulate data with $\alpha = 2, 3, 10$ true partitions across a variety of signal strengths (separability).

q (relative risk):

$\alpha = 2$: $q = [1 - \epsilon, 1 + \epsilon]$

$\alpha = 3$: $q = [1 - \epsilon, 1, 1 + \epsilon]$

$\alpha = 10$: $q = [1 - \epsilon, 1 - 0.8\epsilon, \ldots, 1 + \epsilon]$

ϵ (signal strength):

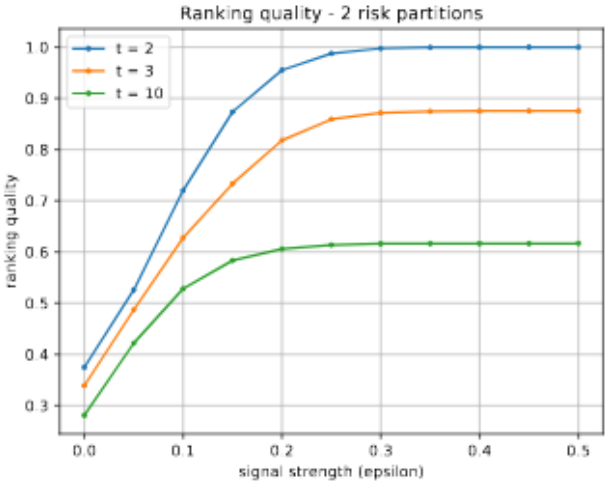
$\epsilon = [0.0, \ldots, 0.5]$

Compute $F^* = \max F(P)$, $P^* = \text{argmax}\, F(P)$.
Order $\{S_1^{true}, \ldots, S_m^{true}\}$, $\left\{S_1^{pred}, \ldots S_{m'}^{pred}\right\}$.

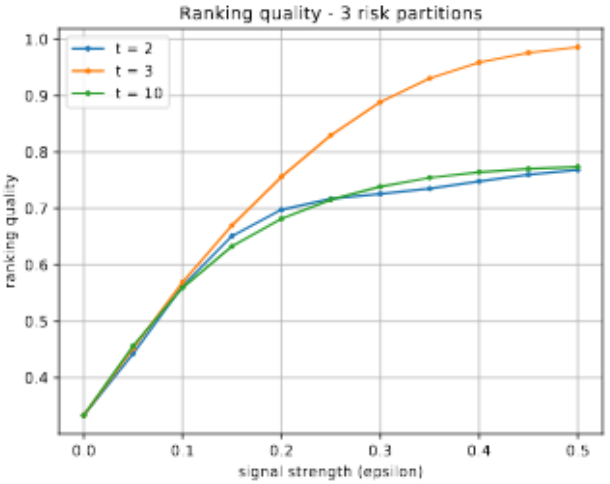
Ranking quality is the probability that clusters are preserved - randomly draw (a, b) with $1 \leq a, b \leq n$, and find $(a, b) \in (S_i^{true}, S_j^{true}), (a, b) \in (S_{i'}^{pred}, S_{j'}^{true})$, determine the probability that (i, j) and $(\imath', \jmath')$ are similarly ordered.

Simulation Results – Risk Partitioning

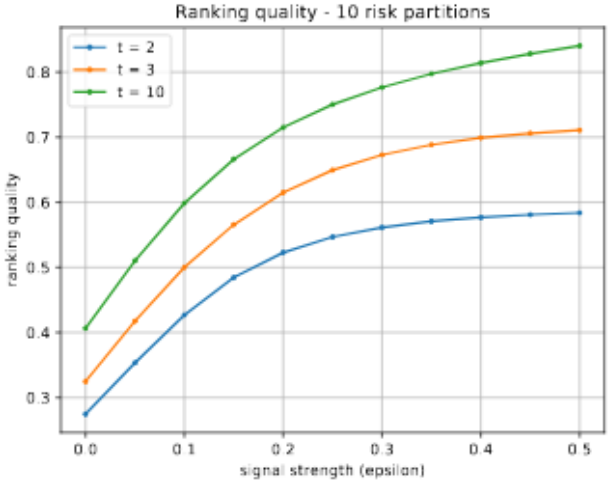
Risk Partitioning Simulation Results



(a) Two true partitions



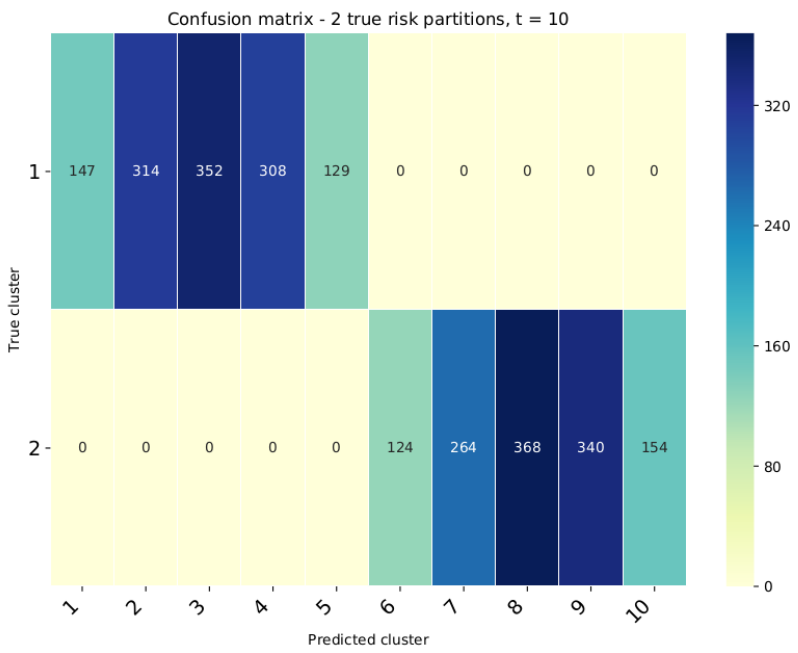
(b) Three true partitions

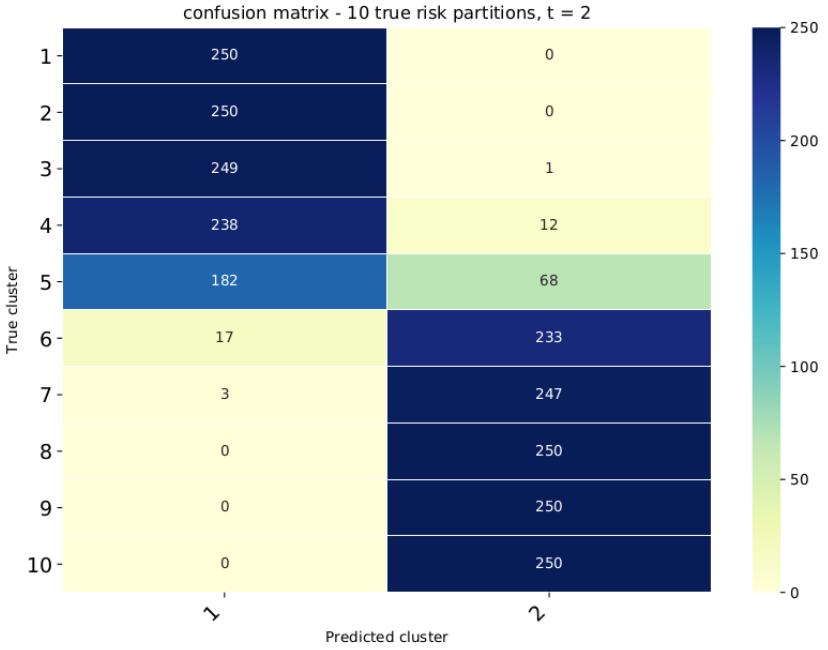


(c) Ten true partitions

Simulation Results – Risk Partitioning

"Smearing" and "folding" of true partitions when t is misspecified





Simulation Results – Multiple Clustering

Setup: Simulate data with varying relative risks $q_1 > q_2 > q = 1$ so that the q_1, q_2 clusters each account for 10 percent of the data.
q (scenarios):

low: $(q_1, q_2, q) = (1 + \epsilon, 1 + \frac{\epsilon}{4}, 1)$

medium: $(q_1, q_2, q) = (1 + \epsilon, 1 + \frac{\epsilon}{2}, 1)$

high: $(q_1, q_2, q) = (1 + \epsilon, 1 + \frac{3\epsilon}{4}, 1)$

Detection power is the probability that the generated optimal score falls above a 95% threshold for the null hypothesis
Detection accuracy measures how well clusters identified by the method correspond to the true clusters used to generate the data. 3 types:

Primary cluster detection accuracy - ability to distinguish q_1 cluster from $q = 1$ cluster, for (n_1, n_2) randomly sampled from $\mathcal{V} \times \mathcal{V}$ with corresponding datapoints in q_1 and $q = 1$ cluster, the probability that the predicted clusters $(S_{i'}^{pred}, S_{j'}^{pred})$ satisfy $i' > j'$.

Secondary cluster detection accuracy - ability to distinguish q_2 cluster from $q = 1$ cluster, defined similarly

Cluster differentiation accuracy - ability to distinguish q_1 cluster from q_2 cluster, defined similarly

Simulation Results – Multiple Clustering Low Scenario

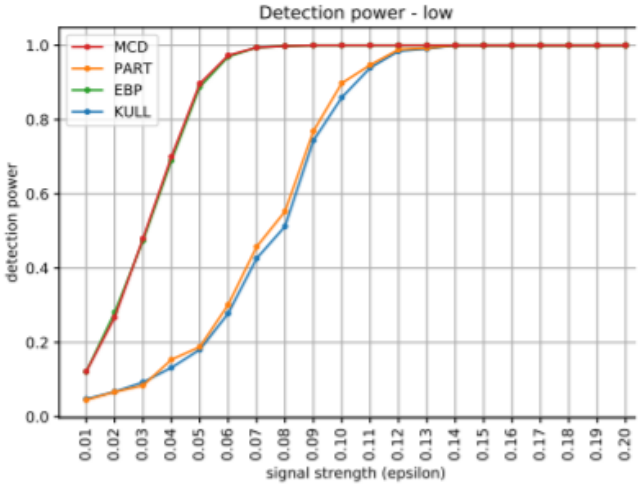
Classifier specification:

type	$t=2$	$t=3$
$RP:$	$KULL$	$PART$
$MCD:$	EBP	MCD

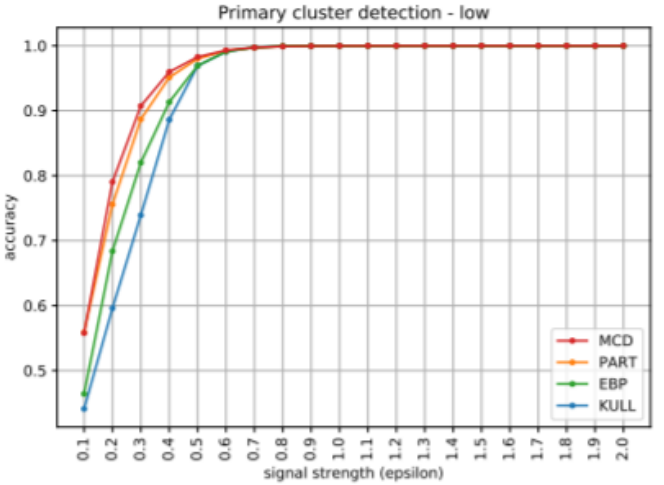
low: $(q_1, q_2, q) = (1+\epsilon, 1+\frac{\epsilon}{4}, 1)$

medium: $(q_1, q_2, q) = (1+\epsilon, 1+\frac{\epsilon}{2}, 1)$

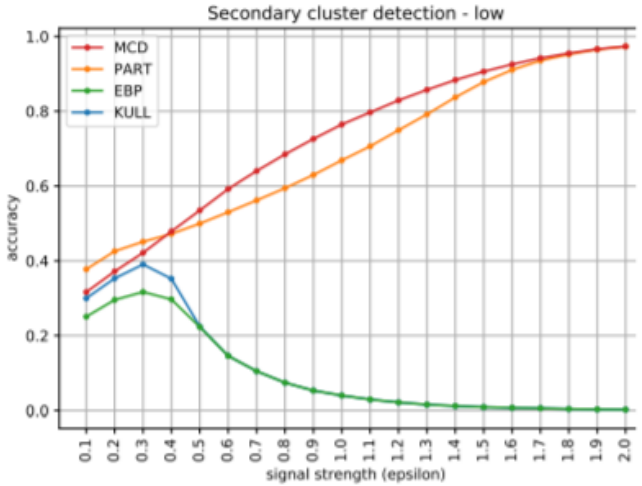
high: $(q_1, q_2, q) = (1+\epsilon, 1+\frac{3\epsilon}{4}, 1)$



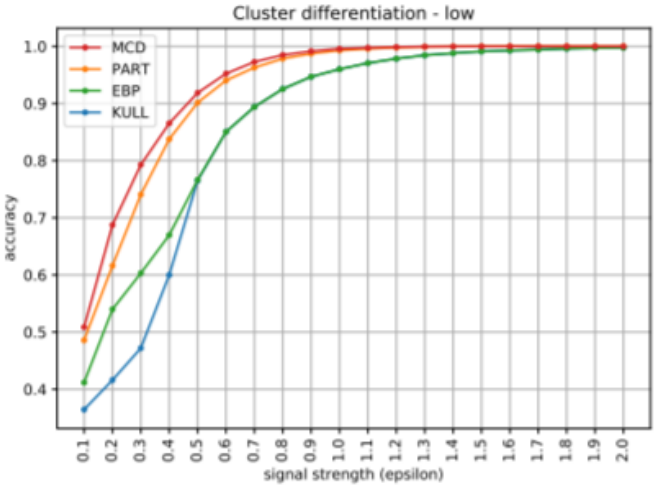
(a) Detection power



(b) Primary cluster detection accuracy



(c) Secondary cluster detection accuracy



(d) Cluster differentiation accuracy

Simulation Results – Multiple Clustering Medium Scenario

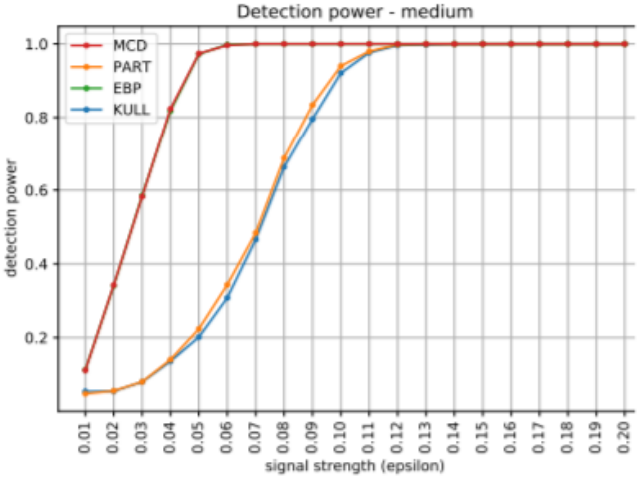
Classifier specification:

type	$t = 2$	$t = 3$
RP :	$KULL$	$PART$
MCD :	EBP	MCD

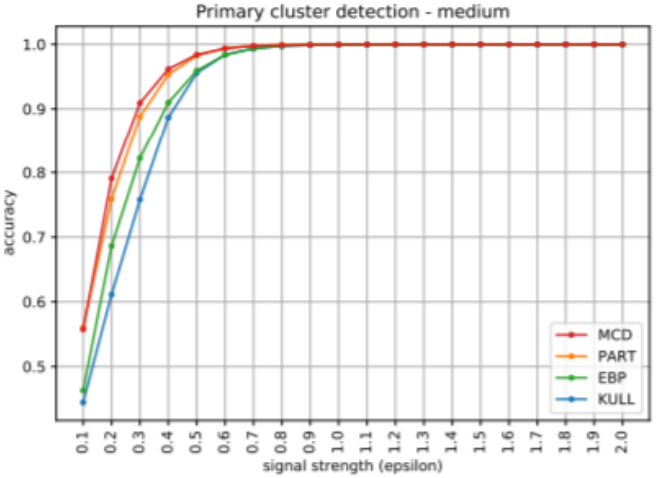
low: $(q_1, q_2, q) = (1 + \epsilon, 1 + \frac{\epsilon}{4}, 1)$

medium: $(q_1, q_2, q) = (1 + \epsilon, 1 + \frac{\epsilon}{2}, 1)$

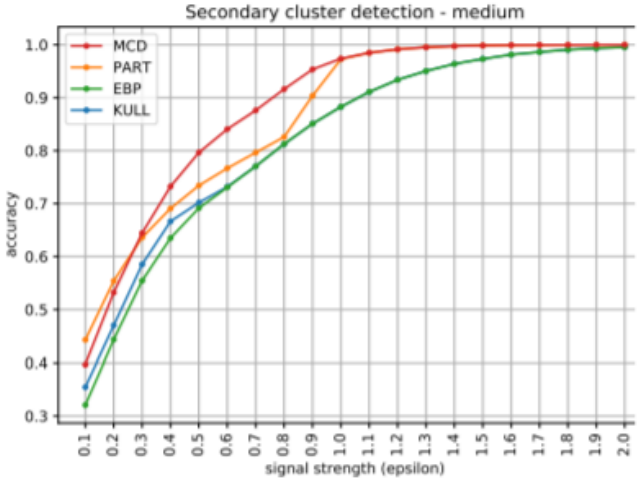
high: $(q_1, q_2, q) = (1 + \epsilon, 1 + \frac{3\epsilon}{4}, 1)$



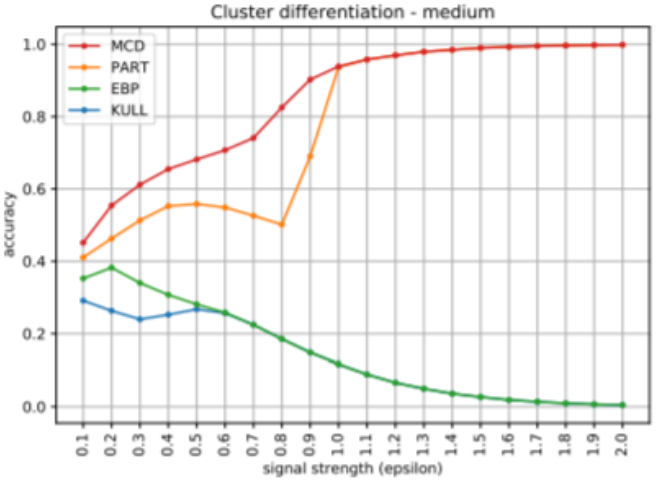
(a) Detection power



(b) Primary cluster detection accuracy



(c) Secondary cluster detection accuracy



(d) Cluster differentiation accuracy

Simulation Results – Multiple Clustering High Scenario

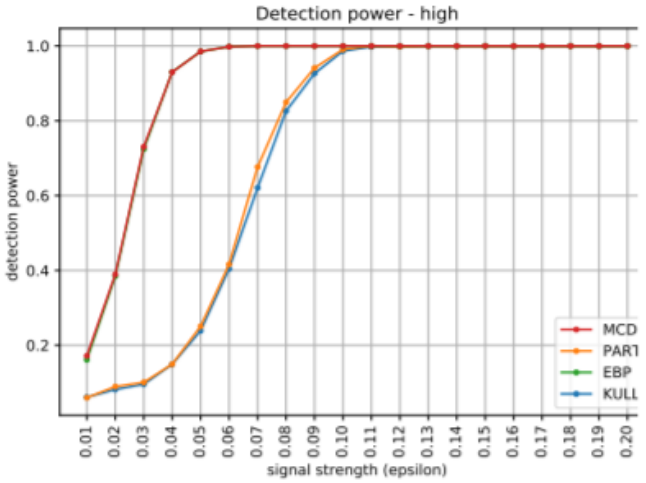
Classifier specification:

type	$t = 2$	$t = 3$
$RP:$	$KULL$	$PART$
$MCD:$	EBP	MCD

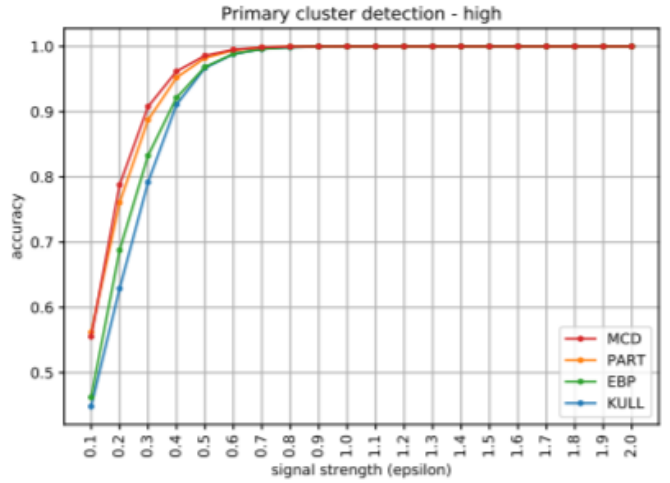
low: $(q_1, q_2, q) = (1 + \epsilon, 1 + \frac{\epsilon}{4}, 1)$

medium: $(q_1, q_2, q) = (1 + \epsilon, 1 + \frac{\epsilon}{2}, 1)$

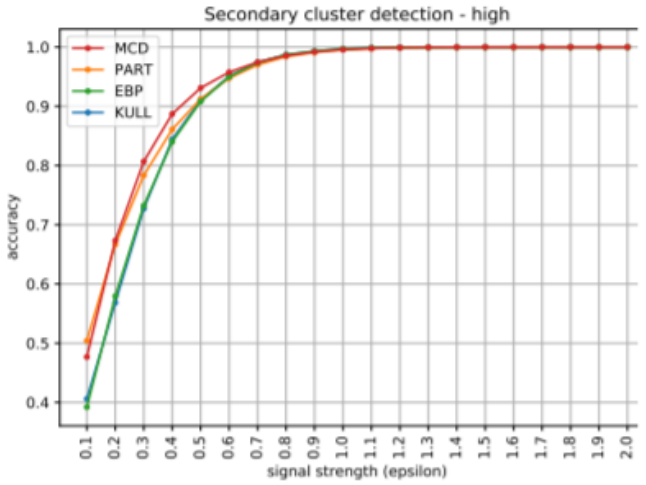
high: $(q_1, q_2, q) = (1 + \epsilon, 1 + \frac{3\epsilon}{4}, 1)$



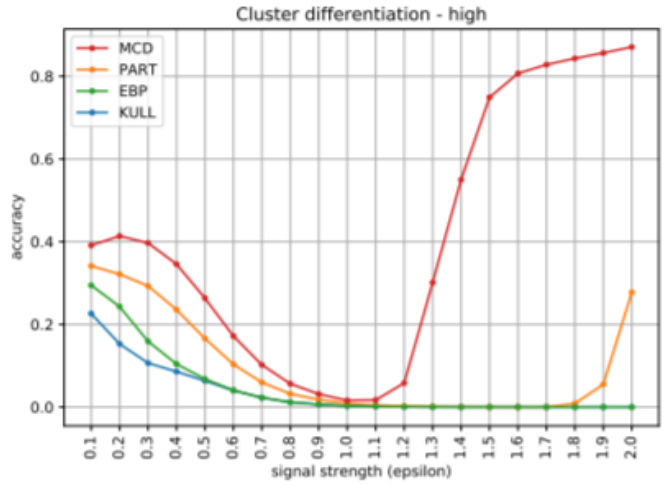
(a) Detection power



(b) Primary cluster detection accuracy



(c) Secondary cluster detection accuracy



(d) Cluster differentiation accuracy

Future Directions?

- Gradient Boosting

Gradient boosting iterative boosting step attempts to choose additive update f_t to minimize loss l:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{2}$$

where $l(y_i, \hat{y}_i)$ is an arbitrary convex loss function, and $\Omega(f_t)$ is an l^2-regularization term.
The algorithm approximates the loss by a quadratic expansion

$$\mathcal{L}^{(t)} = \sum_{j=1}^{t} [(\sum_{i \in S_j} \beta_i) w_j + \frac{1}{2} (\sum_{i \in S_j} \alpha_i + \lambda) w_j^2] + \gamma t \tag{3}$$

where $\beta_i = \frac{\partial l(y_i, \hat{y}^{(t-1)})}{\partial \hat{y}^{(t-1)}}$, $\alpha_i = \frac{\partial^2 l(y_i, \hat{y}^{(t-1)})}{\partial^2 \hat{y}^{(t-1)}}$.
Obtaining the optimal leaf sets and values is equivalent to solving

$$\mathcal{L}^{(t)} = \max_{\mathcal{P}=\{S_1 \ldots, S_t\}} \sum_{j=1}^{t} \frac{(\sum_{i \in S_j} \beta_i)^2}{\sum_{i \in S_j} \alpha_i + \lambda} + \gamma t \tag{4}$$

and setting $w_j = -\frac{\sum_{i \in S_j} \alpha_i}{\sum_{i \in S_j} \beta_i + \lambda}$.

Future Directions?

This suggests a bottom-up, "inductive" approach to the splitting choice - fix t at each step, choose optimal partition and w_j, $j = 1, \ldots t$, iterate, using an inductive tree classifier.

Future Directions?

- Multidimensional approaches?
- Incorporating spatial constraints into the partition scan
- Equivalence results; interplay between f, F.

References

Martin Kulldorff, A spatial scan statistic, *Communications in Statistics: Theory and Methods*, 26(6), 1997: 1481–1496.

Daniel B. Neill, Fast subset scan for spatial pattern detection, *Journal. Royal Statist. Soc.* B (2012) 74, Part 2, pp. 337–360.

Charles A. Pehlivanian, Daniel B. Neill, Efficient optimization of partition scan statistics via the consecutive partitions property, preprint, 2021.