Efficient Optimization of Partition Scan Statistics via the Consecutive Partitions Property

Charles A. Pehlivanian & Daniel B. Neill

To cite this article: Charles A. Pehlivanian & Daniel B. Neill (2022): Efficient Optimization of Partition Scan Statistics via the Consecutive Partitions Property, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2022.2077351

To link to this article: https://doi.org/10.1080/10618600.2022.2077351

Published online: 29 Jun 2022.
Efficient Optimization of Partition Scan Statistics via the Consecutive Partitions Property

Charles A. Pehlivanian\textsuperscript{a} and Daniel B. Neill\textsuperscript{b}

\textsuperscript{a} JP Morgan Chase & Co., New York, NY; \textsuperscript{b} New York University, New York, NY

1. Introduction

The spatial scan statistic is a widely used methodological approach for spatial and space-time cluster detection, first proposed by Kulldorff (Kulldorff 1997; Kulldorff and Nagarwalla 1995) and building on prior work in scan statistics by Nauss, Glaz, and others (Nauss 1965a, 1965b; Glaz, Nauss, and Wallenstein 2001). Spatial scanning has been used for detection of high-risk clusters for cancer and other chronic diseases, emerging infections in human and animal populations, suspicious network activity, areas of increased brain activity from imaging data, and many other applications (Kulldorff 1997; Neill 2012). In the usual spatial scan setting, data elements $s_i$, for $i \in \{1, \ldots, n\}$, represent spatial locations with associated values $x_i$ (representing counts or concurrent measurements) and $y_i$ (representing baselines, expectations, or populations). The goal is to identify a subset of locations with unusual (typically, increased) values of $x_i$ as compared to the expected $y_i$. This is done by maximizing a log-likelihood ratio statistic $F(S)$ over subsets $S \subseteq \{s_1, \ldots, s_n\}$, assuming a parametric model for the $x_i$. For example, for Kulldorff’s spatial scan statistic (Kulldorff 1997), we assume $x_i \sim \text{Poisson}(q_i y_i)$ for all $s_i$. Under the null hypothesis $H_0$, $q_i = q_{\text{all}}$ everywhere for some constant $q_{\text{all}}$, while under the alternative hypothesis $H_1(S)$, $q_i = q_{\text{in}}$ for $s_i \in S$ and $q_i = q_{\text{out}}$ for $s_i \notin S$, for constants $q_{\text{in}} > q_{\text{out}}$. The values of $q_{\text{in}}$, $q_{\text{out}}$, and $q_{\text{all}}$ are fit by maximum likelihood, leading to the generalized log-likelihood ratio statistic:

$$F(S) = \log \frac{\max_{q_{\text{in}}>q_{\text{out}}} \Pr(\text{Data} \mid H_1(S))}{\max_{q_{\text{all}}} \Pr(\text{Data} \mid H_0)} \left( \frac{\max_{q_{\text{in}}>q_{\text{out}}} \prod_{s_i \in S} \Pr(x_i \sim \text{Poisson}(q_{\text{in}} y_i)) \prod_{s_j \notin S} \Pr(x_j \sim \text{Poisson}(q_{\text{out}} y_j))}{\max_{q_{\text{all}}} \prod_{s_i \in S} \Pr(x_i \sim \text{Poisson}(q_{\text{all}} y_i))} \right) \times 1 \left\{ \frac{C_{\text{in}}}{B_{\text{in}}} > \frac{C_{\text{out}}}{B_{\text{out}}} \right\},$$

where $C_{\text{in}} = \sum_{s_i \in S} y_i$, $C_{\text{out}} = \sum_{s_i \notin S} y_i$, and similarly $B_{\text{in}} = \sum_{s_j \in S} y_j$, $B_{\text{out}} = \sum_{s_j \notin S} y_j$, and $B_{\text{all}} = \sum_{s_j} y_j$. Often, baselines are computed such that $B_{\text{all}} = C_{\text{all}}$, and thus $B_{\text{all}} \log \frac{C_{\text{all}}}{B_{\text{all}}} = 0$; in any case, this term is the same for all $S$ and can be neglected when computing $S^* = \text{arg max}_S F(S)$. Once the highest-scoring regions are found, a randomization test is used to determine whether they are statistically significant at a given level (Kulldorff 1997), or in some simple cases a limiting
distribution is known. Alternatively, baselines can be updated under a Bayesian framework, and significance can be estimated from the posterior (Neill, Moore, and Cooper 2006).

While Kulldorff’s original approach focused on maximizing $F(S)$ over the set of circular spatial regions, which can be done exhaustively in $O(n^2)$ time, Neill (2012) reframed the problem as a (constrained or unconstrained) search over all subsets $S$, thus, allowing detection of irregularly shaped clusters as well as extensions to nonspatial data. While exhaustive enumeration and evaluation of the $2^n$ subsets of data elements is computationally infeasible, many log-likelihood ratio scan statistics (including the Kulldorff statistic defined above) were shown to satisfy the linear-time subset scanning property, enabling exact optimization over the exponentially many subsets while requiring only a linear number of subsets to be evaluated (Neill 2012).

The spatial and subset scans can be thought of as partitioning the data elements $s_i$ into two disjoint subsets: a “high-risk” subset $S$ and a “low-risk” subset $\{s_1, ..., s_n\} \setminus S$, assuming constant relative risk $q$ within each subset. Real-world datasets, however, may not satisfy this strong assumption, and may be better modeled by identifying multiple disjoint subsets $S_1, S_2, ..., S_t$ of varying risk. This finer-grained partitioning of the data is useful for detection of multiple clusters (Li et al. 2011), for risk stratification in disease mapping (Lawson et al. 1999), and for discretization of high-arity attributes such as occupations (Kull- dorff, Fang, and Walsh 2003).

In this work, we demonstrate that a large class of log-likelihood ratio statistics, including the Poisson scan statistic defined above, can be exactly and efficiently optimized over partitionings of the data, requiring worst-case time $O(n^2 t)$ to divide a set of size $n$ into $t$ partitions. This efficient optimization follows from two sets of results. First, we define partition score functions and show that a large class of partition score functions satisfy the consecutive partitions property, a generalization of linear-time subset scanning from $t = 2$ to $t \geq 2$ partitions. Second, while naive optimization of the score function over consecutive partitionings would be computationally infeasible for large $t$, with time complexity $O(n^{t-1})$, we provide a dynamic programming algorithm which reduces the time cost to quadratic in $n$ and linear in $t$, while still guaranteeing that the optimal partitioning will be found.

## 2. Partition Scan Statistics

Let $n \in \mathbb{N}$ be positive and set $V = \{1, ..., n\}$. Let $X = \{x_1, ..., x_n\}$ and $Y = \{y_1, ..., y_n\}$ be finite real sequences, with $y_i > 0$ for all $i$. In many of the applications that follow, each tuple $(x_i, y_j)$ corresponds to realized and baseline counts or measures of an attribute for the item or spatial location $s_i$.

Denote by $D = D_{X,Y}$ the set of tuples $\{(x_i, y_j)\}$ associated with $X$ and $Y$.

Given the dataset $D$ and a target number of partitions $t$, our goal is to maximize the partition scan statistic $F(P) = F((S_1, ..., S_t))$ over all possible partitionings $P$ of size $t$, where each subset $S_j \subseteq \{1, ..., n\}$ is nonempty, $S_j \cap S_f = \emptyset$ for $j \neq f$, and $\bigcup_{j=1}^{t} S_j = V$. The partition score $F(P)$ is defined in terms of a score function $f(x, y)$ as

$$F(P) = \sum_{j=1}^{t} F(S_j) = \sum_{j=1}^{t} f \left( \sum_{i \in S_j} x_i, \sum_{i \in S_j} y_i \right).$$

As a concrete example, the Kulldorff scan statistic defined above is a partition scan statistic for size $t = 2$, with associated score function $f(x, y) = x \log \frac{x}{y}$, and the extension to Poisson partition scan statistics of size $t > 2$ follows naturally from the above:

$$F(P) = \sum_{j=1}^{t} C_j \log \frac{C_j}{B_j},$$

where $C_j = \sum_{i \in S_j} x_i$ and $B_j = \sum_{i \in S_j} y_i$ are the aggregate count and baseline for partition $S_j$.

We will also refer to the rational score functions $f(x, y) = x^\alpha y^{-\beta}$ for constants $\alpha, \beta > 0$, and their corresponding partition scan statistics $F(P)$. More formally, we define a score function as follows:

**Definition 1.** A score function is a continuous function $f(x, y) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ with continuous extension to the origin in any wedge $W = (\mu_1, \mu_2) = \{(x, y) : y > 0, \mu_1 \leq x \leq \mu_2\}$, for $-\infty < \mu_1 \leq \mu_2 < \infty$, with $\lim_{(x,y) \in W \rightarrow (0,0)} f(x, y) = 0$.

The regularity condition on $W$ in wedges simply guarantees a continuous extension to the origin on any positive cone in $\mathbb{R}^+$; for rational score functions it is equivalent to the constraint $\alpha > \beta$. Several additional properties have been associated with score functions, namely monotonicity in $x$ or $y$, quasi-convexity, convexity, or an isotone differences constraint $\frac{\partial f}{\partial x y} \leq 0$ related to submodularity. We do not assume smoothness beyond continuity, nor (quasi-)convexity, etc., unless explicitly stated.

A score function $f$ naturally gives rise to a set function $F : 2^V \rightarrow \mathbb{R}$, as above, defined by summation over subsets, $F(S) = f(\sum_{i \in S} x_i, \sum_{i \in S} y_i)$, for $S \subseteq V$. Associating $S \subseteq V$ with its $X$ $Y$ attributes as in $X_S = \sum_{i \in S} x_i$, $Y_S = \sum_{i \in S} y_i$ allows us to write $F(S) = f(\sum_{i \in X_S} x_i, \sum_{i \in Y_S} y_i) = f(X_S, Y_S)$, etc. Likewise, the partition scan statistic $F(P) = F(S_1, ..., S_t)$ can be written as $\sum_{j=1}^{t} F(S_j)$, as above, and we are interested in identifying the optimal partitioning

$$P^* = \arg \max_{P \in \{S_1, ..., S_t\}} \sum_{j=1}^{t} F(S_j)$$

$$= \arg \max_{P \in \{S_1, ..., S_t\}} \sum_{j=1}^{t} f \left( \sum_{i \in S_j} x_i, \sum_{i \in S_j} y_i \right).$$

Common approaches to solving Equation (1) rely on properties of the set function $F$. For $F$ submodular, a class of greedy algorithms can provide $(1 - \frac{1}{e})$-approximate solutions for the single subset problem in polynomial time (Vondrák 2008; Filmus and Ward 2012), based on Monte Carlo approximation of the multidimensional extension $f_m(x_1, ..., x_n) = \sum_{S \subseteq V} F(S) \prod_{i \in S} x_i \prod_{i \notin S} (1 - x_i)$, for $x \in \{0, 1\}^n$. Minimization of a submodular $F$ over subsets of $2^V$ admits an exact solution using the Lovasz extension (Grötschel, Lovasz,
and Schrijver 1981), which is guaranteed to be convex. The essence of these approaches is a “relaxation by expectation” to a continuous problem on the $n$-dimensional unit hypercube, associating $S \subseteq V$ with the point $(1_{1 \in S}(S), \ldots, 1_{n \in S}(S))$. In this way every boundary point of the hypercube is associated with a subset of $V$. In our setting, the ambient extension is already specified as $f$, and the natural extensions to $\mathbb{R}^+$ (multilinear, Lovasz) do not allow for a convenient restriction of the admissible set of partitionings. Nevertheless, we will demonstrate an alternative approach based on consecutive partitions which enables exact and efficient optimization of Equation (1), as described in Section 3.

### 2.1. Partition Scan Statistics for the Separable Exponential Family

Let us assume a parametric specification for the generating distribution $\Psi$ for the $x_i$ values, where the null and alternative hypotheses are defined as follows:

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

where $P = \{S_1, \ldots, S_t\}$ is a partition of $D$ of size $t$, and the constants $q_1 \ldots q_t$ (the “relative risk” for each partition $S_j \in P$) and $q_{\text{all}}$ are computed by maximum likelihood. The most common choices of distribution are Poisson and Gaussian. For many distributional assumptions, the score function is expressed only in terms of sufficient statistics of the underlying $\Psi$, and closed-form solutions for the $q_j$ can be solved for and resubstituted. Let us consider the (generalized) log-likelihood ratio statistic:

$$F(P) = F(S_1, \ldots, S_t) = \log \frac{\max_{\mu_1, \ldots, \mu_t} \prod_{j=1}^t \prod_{i \in S_j} \Pr(x_i | x_i \sim \Psi (\cdot; \mu = q_j \mu_i))}{\max_{\mu_{\text{all}}} \prod_{i} \Pr(x_i | x_i \sim \Psi (\cdot; \mu = q_{\text{all}} \mu_i))}. \quad (2)$$

Following Neill (2012), we now assume that $\Psi$ comes from an exponential family, in which case we can write $\Psi$ in terms of its mean $\mu$ as $\log \Pr(x | \mu) = T(x) \theta(\mu) - \psi(\theta(\mu))$, where $T(x)$ is the sufficient statistic, $\theta(\mu)$ is a function mapping the mean $\mu$ to the natural parameter $\theta$, $\psi$ is the log-partition function, and $\phi$ is the convex conjugate (in the Legendre-Fenchel sense) of $\psi$. Plugging in the log-likelihood ratio statistic for $F(P)$, we obtain

$$F(P) = \sum_{j=1}^t \sum_{x_i \in S_j} \left( T(x_i) \theta(q_j \mu_i) - q_j \mu_i \theta(q_j \mu_i) + \phi(q_j \mu_i) \right) - \sum_{x_i} \left( T(x_i) \theta(q_{\text{all}} \mu_i) - q_{\text{all}} \mu_i \theta(q_{\text{all}} \mu_i) + \phi(q_{\text{all}} \mu_i) \right).$$

For distributions $\Psi$ in a separable exponential family (Neill 2012), including the Poisson, Gaussian, and exponential distributions, we can further write $\theta(q_j \mu_i) = z_j \theta_0(q) + v_i$, where the function $\theta_0$ depends only on $q$, while $z_j$ and $v_i$ can depend on $\mu_i$ and $c_j$ but are independent of $q$. We also have $\phi(q_j \mu_i) = \mu_i z_j \phi_0(q) + \mu_i v_i q + K_i$, where $\phi_0(q) = \int \theta_0(q) \, dq$, and $K_i$ is independent of the expression $F(P)$ can then be simplified to

$$F(P) = \sum_{j=1}^t \sum_{x_i \in S_j} \left( C_j \theta_0(q_j) + B_j \left( \phi_0(q) - q_j \theta_0(q) \right) \right) - C_{\text{all}} \theta_0(q_{\text{all}}) - B_{\text{all}} \phi(q_{\text{all}}) - q_{\text{all}} \phi(q_{\text{all}}),$$

where $C_j$ and $B_j$ are the sufficient statistics for subset $S_j$, $C_j = \sum_{x_i \in S_j} T(x_i) z_j$, and $B_j = \sum_{x_i \in S_j} \mu_i z_j$, respectively. Similarly, we have $C_{\text{all}} = \sum_{x_i} T(x_i) z_j$ and $B_{\text{all}} = \sum_{x_i} \mu_i z_j$. To obtain the maximum likelihood estimate of $q_j$, we set $\frac{\partial F(P)}{\partial q_j} = 0$, obtaining $q_j = \frac{C_j}{B_j}$, and similarly $q_{\text{all}} = \frac{C_{\text{all}}}{B_{\text{all}}}$. Substituting these values of $q_j$ and $q_{\text{all}}$ into the equation for $F(P)$ and simplifying, we find that

$$F(P) = \sum_{j=1}^t B_j \left( \phi_0 \left( \frac{C_j}{B_j} \right) - \phi_0 \left( \frac{C_{\text{all}}}{B_{\text{all}}} \right) \right) = B_{\text{all}} D_{f} (\hat{C} \mid \hat{B}),$$

where $\hat{C}$ and $\hat{B}$ are the normalized count and baseline vectors $\frac{C_j}{B_j} (C_1, \ldots, C_t)$ and $\frac{C_{\text{all}}}{B_{\text{all}}} (B_1, \ldots, B_t)$, respectively, and $D_{f}$ is the $f$-divergence, $D_{f} (P \mid Q) = \sum_{j=1}^t Q_j f \left( \frac{p_j}{q_j} \right)$, where $f(q) = \phi_0 (\frac{C_j}{B_j}) - \phi_0 (\frac{C_{\text{all}}}{B_{\text{all}}}).$ \footnote{In the common case in which $B_{\text{all}}$ is set equal to $C_{\text{all}}$, the $f$-divergence $D_{f}$ simplifies to $B_{\text{all}} D_{\phi}(\hat{C} \mid \hat{B})$, which is similar to the expectation-based scan statistics for the separable exponential family (Neill 2012), $F(S) = B_{\text{all}} D_{\phi}(\hat{C} \mid \hat{B})$, but in the latter case $D$ is the Bregman divergence rather than $f$-divergence corresponding to $\phi_0$.}

### 3. The Consecutive Partitions Property

We now return to the general formulation of the partition scan statistic, $F(P) = F(S_1, \ldots, S_t) = \sum_{j=1}^t f(\sum_{i \in S_j} x_i, \sum_{i \in S_j} c_i)$, and identify conditions on the score function $f$ that will enable us to efficiently compute the optimal partitioning $P^* = \arg \max_P F(P)$.
We first note that exhaustively searching over all partitions $P$ is computationally infeasible, even for relatively small values of $n$ and $t$. More precisely, the number of such partitions $\binom{n}{t}$ is a Stirling number of the second kind. The Stirling numbers have an asymptotic growth rate which is exponential in $n$ for fixed $t$, $\binom{n}{t} \sim \frac{2^n}{\sqrt{\pi t}}$. For example, for $n=30$ and $t=10$, we already have $\binom{n}{t} \approx 1.73 \times 10^{22}$.

Moreover, it is well known that even for monotone submodular set functions $F$, the maximization $\max_{S \in \mathcal{V}} F(S)$ is NP-hard, while the submodular minimization problem can be solved in polynomial time (Calinescu et al. 2007; Bach 2013). For nonmonotone submodular objectives, any polynomial time algorithm for the maximization is only guaranteed to satisfy a lower bound of $(1 - \frac{1}{e})$ from optimality, that is, $F(S) \geq (1 - \frac{1}{e})F(S^*)$, where $S^*$ is the argmax solution (Freige, Mirrokni, and Vondrak 2011). For monotone submodular objectives, a greedy algorithm can do better, but only efficient approximations are known. Monotonicity and submodularity are generally not satisfied by even the rational score functions.

On the other hand, the number of candidate partitions would be greatly reduced by the requirement that each partition $S_i$ be consecutive, that is, of the form $\{s_i, s_{i+1}, \ldots, s_k\}$ for $1 \leq i \leq k \leq n$, assuming some appropriate ordering of the data elements. The set $\mathbb{T}_n$ of partitions of $\mathcal{V}$ of size $t$ containing only consecutive elements has size $(n-1)^t$, which grows polynomially in $n$, as $O(n^{t-1})$. We will show that, under certain sufficient conditions on $f$, the optimal partitioning is guaranteed to be consecutive when the set $D$ is ordered under an appropriate priority function $g$. Formally, we define:

**Definition 2.** A priority function is a function $g: \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}$ that induces an ordering on the dataset $D$. We refer to $g(x, y) = \frac{x}{y}$ as the standard priority function.

Unless otherwise stated, the sets $X$, $Y$ will be assumed to be already indexed in standard priority order, that is, $g(x_1, y_1) \leq \cdots \leq g(x_n, y_n)$, where $g$ is the standard priority function.

Given the dataset $D$ ordered by priority, we can now formally define the **consecutive partitions property**.

**Definition 3.** A nonempty subset $S \subseteq D$ is consecutive if it is of the form $\{s_i, s_{i+1}, \ldots, s_k\}$, for $1 \leq i \leq k \leq n$.

**Definition 4.** A partitioning $P = \{S_1, \ldots, S_j\}$ of $D$ of size $t$ is consecutive if each partition $S_i$ is a consecutive subset.

**Definition 5.** A partition scan statistic $F(P)$, with associated score function $f(x, y)$ and priority function $g(x, y)$, satisfies the **consecutive partitions property (CPP)** if, for all datasets $D = \{(x_i, y_i)\}$ ordered by priority, and all sizes $1 \leq t \leq |D|$, the highest scoring partitioning of $D$ of size $t$, $p^* = \arg \max_{P=\{S_1, \ldots, S_j\}} \sum_{j=1}^n f \left( \sum_{i \in S_j} x_i, \sum_{i \in S_j} y_i \right)$, is consecutive.

We also define a weaker version of CPP, as follows:

**Definition 6.** A partition scan statistic $F(P)$, with associated score function $f(x, y)$ and priority function $g(x, y)$, satisfies the **weak consecutive partitions property (WCPP)** if, for all datasets $D = \{(x_i, y_i)\}$ ordered by priority, and all sizes $1 \leq t \leq |D|$, the highest scoring partitioning of $D$ of size $t' \leq t$, $p^* = \arg \max_{P=\{S_1, \ldots, S_j\}, t' \leq t} \sum_{j=1}^{t'} f \left( \sum_{i \in S_j} x_i, \sum_{i \in S_j} y_i \right)$, is consecutive.

It is clear from these definitions that, if $F(P)$ satisfies CPP, then it also satisfies WCPP. However, the converse is not necessarily true. As a concrete example, Chakravarty, Orlin, and Rothblum (1982) show that, if $f(x, y)$ is convex and $y_i > 0$ for all $i$, then a consecutive partitioning of size $t' \leq t$ exists that is maximal in the sense of (1) for the standard priority function; in other words, $F(P)$ satisfies WCPP. Let us consider the rational score function $f(x, y) = \frac{x}{y}$, which is convex, and consider the dataset $D = (X, Y)$, where $X = \{8, 2, 9\}$ and $Y = \{8, 1, 3\}$, ordered by priority. We observe that the highest-scoring partitioning $F(P)$ of size $t = 2$ consists of $\{s_1, s_3\}$ and $\{s_2\}$, which is not consecutive, so CPP does not hold. The highest scoring partitioning of size $t' \leq 2$ is the trivial (size-1) partitioning $P = \{s_i\}$, where $s_1 = \{s_1, s_2, s_3\}$. In such cases, we are unable to say anything about the optimal partitioning for sizes $t > 1$.

Thus, we wish to identify sufficient conditions under which CPP and WCPP hold. These are provided by our main results, Theorems 1 and 2:

**Theorem 1.** Given a partition scan statistic $F(P)$ with associated score function $f(x, y)$ and priority function $g(x, y) = \frac{x}{y}$ if $f$ is convex and subadditive, then $F$ satisfies CPP.

**Theorem 2.** Given a partition scan statistic $F(P)$ with associated score function $f(x, y)$ and priority function $g(x, y) = \frac{x}{y}$ if $f$ is convex, then $F$ satisfies WCPP.

Note that convexity and subadditivity are both preserved under function addition. The proofs of Theorems 1 and 2 are based on a novel geometric construction, described in detail in the following section. It follows that partition scan statistics for the separable exponential family satisfy CPP:

**Corollary 1.** Let $F(P)$ be the partition scan statistic corresponding to the log-likelihood ratio statistic for a distribution $\Psi$ in a separable exponential family, with associated score function $f(x, y) = y \phi_0 \left( \frac{x}{y} \right)$, where $\phi_0$ is defined as above, and priority function $g(x, y) = \frac{x}{y}$. Then $F(P)$ satisfies CPP.

**Proof.** The proof of Corollary 1 follows from the convexity and subadditivity of $f(x, y)$, and from Theorem 1. Convexity of $f$ follows from convexity of $\phi_0$ and the fact that the perspective of a convex function is also convex. To show subadditivity, we have:

$$f(x_1 + x_2, y_1 + y_2) = (y_1 + y_2) \phi_0 \left( \frac{x_1 + x_2}{y_1 + y_2} \right) = (y_1 + y_2) \phi_0 \left( \frac{\lambda x_1}{y_1} + (1 - \lambda) \frac{x_2}{y_2} \right) \leq (y_1 + y_2) \left( \lambda \phi_0 \left( \frac{x_1}{y_1} \right) + (1 - \lambda) \phi_0 \left( \frac{x_2}{y_2} \right) \right) = y_1 \phi_0 \left( \frac{x_1}{y_1} \right) + y_2 \phi_0 \left( \frac{x_2}{y_2} \right) = f(x_1, y_1) + f(x_2, y_2).$$
where \( \lambda = \frac{n}{y_1 + y_2} \) and the inequality follows from convexity of \( \phi_0 \).

We can also easily characterize the partition scan statistics corresponding to the rational score functions:

**Corollary 2.** Let \( F(P) \) be the partition scan statistic with associated score function \( f(x, y) = x^\alpha y^{-\beta} \), for constants \( \alpha, \beta > 0 \), and priority function \( g(x, y) = \frac{\lambda}{y} \). Assume \( x > 0 \). If \( \alpha - \beta \geq 1 \), then \( F(P) \) satisfies WCP. If \( \alpha - \beta = 1 \), then \( F(P) \) satisfies CPP.

**Proof.** We note that the Hessian of \( f \) has principal minors
\[
M_1 = \alpha(\alpha - 1)x^{\alpha - 2}y^{-\beta},
\]
\[
M_2 = \alpha\beta(\alpha - 1)x^{2(\alpha - 1)}y^{-2(\beta + 1)},
\]
so that for \( x \in \mathbb{R}^+, f \) is convex iff \( \alpha - \beta \geq 1 \), and \( f \) is subadditive iff \( \alpha - \beta \leq 1 \). The result then follows from Theorems 1 and 2.

A similar result holds when \( x \) is not restricted to be positive, but in this case \( \alpha \) must be an even integer:

**Corollary 3.** Let \( F(P) \) be the partition scan statistic with associated score function \( f(x, y) = x^\alpha y^{-\beta} \), for constants \( \alpha, \beta > 0 \), and priority function \( g(x, y) = \frac{\lambda}{y} \). If \( \alpha \in \mathbb{N} \) is even and \( \alpha - \beta \geq 1 \), then \( F(P) \) satisfies WCP. If \( \alpha \in \mathbb{N} \) is even and \( \alpha - \beta = 1 \), then \( F(P) \) satisfies CPP.

**Proof.** For \( x \in \mathbb{R} \), \( f \) is convex iff \( \alpha - \beta \geq 1 \), and \( \alpha \in \mathbb{N} \) is even. \( f \) is subadditive iff \( \alpha - \beta \leq 1 \), and \( \alpha, \beta \in \mathbb{N} \). The result then follows from Theorems 1 and 2.

4. A Geometric Proof of the Consecutive Partitions Property

Let \( D = \{(x_i, y_i)\}_{i=1}^n \) be a dataset, where \( x_i \in \mathbb{R} \) and \( y_i \in \mathbb{R}^+ \), ordered by the standard priority function \( g(x, y) = \frac{\lambda}{y} \). Let us consider the subsets \( S \subseteq V \) and their corresponding sums \( s_k = \sum_{i \in S} x_i \) and \( s_y = \sum_{i \in S} y_i \), associating each subset \( S \) with a corresponding point in \( \mathbb{R}^2 \), \( p_S = (s_x, s_y) \), which we term the partition point of \( S \). By convention, \( p_S = (0, 0) \). Our proofs of Theorems 1 and 2 are based on a geometric characterization of the convex hull of the partition points \( p_S \) for \( S \subseteq V \). Denoting the convex hull of a set \( S \subseteq \mathbb{R}^m \) as \( \mathcal{S} \), we define:

**Definition 7.** For \( S \subseteq V \), let \( p_S = (\sum_{i \in S} x_i, \sum_{i \in S} y_i) \in \mathbb{R}^2 \). The partition polytope \( \mathcal{P} \) and the constrained partition polytope \( \mathcal{C} \) are defined by the convex hulls \( \mathcal{C} = \mathcal{P} \) and \( \mathcal{C} = \mathcal{P} \), respectively, where \( P = \{p_S: S \subseteq V\} \) and \( \mathcal{P} = \{p_S: S \subseteq V, S \neq \emptyset, S \neq V\} \).

We now characterize the extreme points of \( \mathcal{C} \) and \( \mathcal{C} \). It will then follow from the convexity of the score function \( f(x, y) \) that the optimal size-2 partitioning is \( P^* = \{S, V \setminus S\} \) for some extreme point \( p_S \), and the remainder of the proofs of Theorems 1 and 2 proceed by induction. Let us define:

**Definition 8.** A consecutive subset \( S \subseteq V \) is nonsplitting if both \( S \) and \( V \setminus S \) are consecutive; otherwise, it is splitting. A consecutive nonsplitting subset of the form \( \{1, \ldots, j\} \) is ascending, and a consecutive nonsplitting subset of the form \( \{j, \ldots, n\} \) is descending. By convention, \( \emptyset \) and \( V \) are consecutive nonsplitting, both ascending and descending. Define
\[
\mathcal{T} = \mathcal{T}_n = \{S \subseteq V: S \text{ is consecutive}\},
\]
\[
\mathcal{U} = \mathcal{U}_n = \{S \subseteq V: S \text{ is consecutive nonsplitting}\},
\]
\[
\mathcal{U} = \mathcal{U}_n = \{S \subseteq V \setminus \emptyset, V\}.
\]

A special case occurs when \( S \) is the singleton \( \{j\} \): the pair \( S, V \setminus S \) is called a singleton splitting or nonsplitting pair, depending on whether \( V \setminus S \) is consecutive. The set of singleton splitting subsets \( S \) and their complements in \( V \) is denoted by
\[
\mathcal{S} = \mathcal{S}_n = \{S \subseteq V: (S, V \setminus S) \text{ form a singleton splitting pair}\}.
\]

We note that \( S \subseteq \mathcal{T} \) and \( \mathcal{U} \subseteq \mathcal{U} \subseteq \mathcal{T} \), but \( \mathcal{S} \not\subseteq \mathcal{T}, \mathcal{S} \), since \( \mathcal{S} \) includes the nonconsecutive complement \( V \setminus S \) for each singleton splitting subset \( S \). It is easy to see that \( |\mathcal{U}_n| = 2n, |\mathcal{S}_n| = \frac{(n-1)(n-2)}{2} \), with \( |\mathcal{T}_1| = |\mathcal{U}_1| + |\mathcal{S}_1| = \frac{n(n+1)}{2} + 1 \), while \( |\mathcal{S}_n| = 2n - 4 \).

For any partitioning \( P = \{S_1, \ldots, S_l\} \), or more generally, a set of subsets of \( V \), denote by \( \Pi(P) \) the mapping taking \( P \) to the point set \( \{p_{S_1}, \ldots, p_{S_l}\} \subset \mathbb{R}^2 \), so \( \Pi(S) \subset \Pi(T), \Pi(U) \subseteq \Pi(T) \), etc.

**Proposition 1.** Let \( D = \{(x,y)\} \) be a dataset ordered by the standard priority function \( g(x, y) = \frac{\lambda}{y} \). Let \( \mathcal{E} \) and \( \mathcal{C} \) be the sets of extreme points of the partition polytope \( \mathcal{C} \) and the constrained partition polytope \( \mathcal{C} \), respectively. Then
\[
1. \mathcal{E} \subseteq \Pi(U),
\]
\[
2. \mathcal{E} \subseteq \Pi(U) \cup \Pi(S).
\]

In particular, \( \mathcal{E} \) contains only consecutive nonsplitting subsets, while \( \mathcal{C} \) may also contain consecutive splitting subsets and nonconsecutive subsets, but only singletons and their complements.

**Proof.** Let \( p^* = (x^*, y^*) \in \mathcal{E} \). Then there is an affine \( l(x, y) = Ax + By + C \) which separates \( p^* \) in \( \mathcal{C} \), or equivalently a vector \( v = (A, B) \in \mathbb{R}^2 \) for which
\[
(x^*, y^*) = \text{argmin } v \cdot p = \text{argmin } v \cdot p_S, \quad p_S \in \mathcal{C}, S \subseteq V.
\]

The point \( p_S^* \), where \( S^* = \{i \in V: \text{v} \cdot (x_i, y_i) \leq 0\} \) minimizes the last expression. If \( A = 0 \) then necessarily \( B \neq 0 \), in which case \( S^* = \{i \in V: \frac{y_i}{x_i} \leq 0\} = \emptyset \) or \( S^* = \{i \in V: \frac{y_i}{x_i} \geq 0\} = V \), so that \( S^* \) and \( V \) are both consecutive. Otherwise \( S^* = \{i \in V: \frac{x_i}{y_i} \leq -\frac{B}{A} \} \) or \( S^* = \{i \in V: \frac{x_i}{y_i} \geq -\frac{B}{A} \} \), depending on the sign of \( A \), and the same conclusion holds. So \( \mathcal{E} \subseteq \Pi(U) \).

We now prove that \( \mathcal{E} \subseteq \Pi(U) \cup \Pi(S) \). For \( p = (x, y) \in \mathcal{E} \), we can again find a separating affine function \( l(x, y) = Ax + By + C \) and corresponding \( v = (A, B) \) satisfying
\[
p = \text{argmin } v \cdot p = \text{argmin } v \cdot p_S, \quad p_S \in \mathcal{C}, S \subseteq V, \quad S \neq \emptyset, S \neq V.
\]
Let \( p = \min_{p \in C \subseteq \mathcal{V}} p \mathcal{S} \), which by the previous argument is of the form \( p \mathcal{S} \) for some consecutive nonsplitting subset \( S \). If \( 0 < |S| < n \), then it also minimizes the expression in (4), and \( p \in \Pi(U) \). Otherwise there are two cases.

**Case 1.** \( S = \emptyset \). Since the minimizing subset \( S \) can be written as \( \{ i \in \mathcal{V} \mid \mathcal{V} \cdot (x_i, y_i) \leq 0 \} \), we must have \( Ax_i + By_i > 0 \) for all \((x_i, y_i) \in \mathcal{C} \). So expression (4) is minimized by selecting \( p = p_S \), for \( S = \{ i \} \), where \( p^* = \max_{i \in \mathcal{V}} (Ax_i + By_i) \), a singleton. It follows that \( p \) lies in either \( \Pi(U) \) or \( \Pi(S) \).

Thus, \( C \subseteq \Pi(U) \cup \Pi(S) \).

For a given score function \( f \), we now define the symmetric score function \( \bar{f} : C \subseteq \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R} \) by

\[
\bar{f}(x, y) = \begin{cases} 
  f(x, y) + f(C^n_y - x, C^n_y - y), & (x, y) \in C \setminus \{(0, 0), (C^n_x, C^n_y)\} \\
  f(C^n_x, C^n_y), & (x, y) \in \{(0, 0), (C^n_x, C^n_y)\}
\end{cases}
\]

where \( C^n_x = \sum_{i=1}^{n} x_i \), \( C^n_y = \sum_{j=1}^{n} y_j \). We note that \( \bar{f} \) is continuous on \( C \) for any score function \( f \), and is convex on \( C \) for any convex score function \( f \).

By Proposition 1, a convex \( \bar{f} \) will take its maximum on \( C \) at either a splitting or nonsplitting consecutive partition point. We show that the former case represents a kind of degenerate behavior, associated with a collapse of the unconstrained case to the trivial partitioning:

**Proposition 2.** Let \( f(x, y) \) be a convex score function and define the symmetric score function \( \bar{f}(x, y) \) as above. If the set \( \text{argmax}_{(x,y)\in\mathcal{C}} \bar{f}(x,y) \) contains only partition points associated with consecutive splitting subsets, then \( p_S \in \Pi(S) \).

**Proof.** Let \( p = \argmax_{(x,y)\in\mathcal{C}} \bar{f}(x,y) \), with \( p \) consecutive splitting in \( \mathcal{V} \). Let \( p = \argmax_{(x,y)\in\mathcal{C}} \bar{f}(x,y) \). Convexity of \( \bar{f} \) on \( C \) follows from convexity of \( f \), so \( p \in C \). Since \( \bar{f}(p) \geq \bar{f}(p) \), and \( p \not\in \mathcal{C} \) by Proposition 1, we have \( \bar{f}(p) \geq \bar{f}(p) \). So \( p \) cannot lie in \( \mathcal{C} \), and thus \( p \in \mathcal{C} \cup \mathcal{I} = \{ p_{\mathcal{V}}, p_{\mathcal{V}} \} \).

Given Propositions 1 and 2, we can now prove the main theorems:

**Theorem 1.** Given a partition scan statistic \( F(p) \) with associated score function \( f(x, y) \) and priority function \( g(x, y) = \frac{x}{y} \). If \( f \) is convex and subadditive, then \( F \) satisfies SCCP.

**Proof.** The proof proceeds by induction on \( t \). Let \( t = 2 \), and note that \( \text{argmax}_{p \in \mathcal{P}} F(p) = \{ S^n, \mathcal{V} \setminus S^n \} \), where \( S^n = \text{argmax}_{p \in \mathcal{P}} \bar{f}(p) \). By convexity of \( \bar{f} \) and Proposition 1, the maximum of \( \bar{f} \) on \( C \) occurs at a point \( p_S \in \Pi(U) \cup \Pi(S) \). Since \( f \) is subadditive, the maximum of \( \bar{f} \) on \( C \) cannot occur at \( p_{\mathcal{V}} \), hence, \( p_S \in \Pi(U) \cup \Pi(S) \).

So the maximal partitioning \( \{ S, \mathcal{V} \setminus S \} \) is consecutive.

Now let \( t \geq 3 \). The arguments here are motivated by Chakravarty, Orlin, and Rothblum (1982). Define, for any subset \( S \subseteq \mathcal{V} \), \( M_S = \max i : i \in S \), \( m_S = \min i : i \in S \), and \( d(S) = M_S - m_S \). Let \( S = \{ S_1, \ldots, S_t \} \) be a size-\( t \) partitioning that maximizes \( F(p) \) and also minimizes \( \sum_{i=1}^{t} d(S_i) \) among all optimal size-\( t \) partitionings. If not all \( S_i \) are consecutive, then we can find two partitions \( S_i, S_j \in P \) and an element \( k \in S_j \) with \( m_{S_k} < k < M_{S_k} \). Note that \( \min(M_{S_k}, M_{S_j}) \geq k \), and \( \max(m_{S_k}, m_{S_j}) \leq k \), so that \( \min(M_{S_k}, M_{S_j}) - \max(m_{S_k}, m_{S_j}) \geq 0 \).

By the induction hypothesis, we can find an optimal partitioning of \( S_i \cup S_j \) into nonempty consecutive sets \( S_i, S_j \). Then \( d(S_i) + d(S_j) \leq \max(M_{S_k}, M_{S_j}) - \min(m_{S_k}, m_{S_j}) - 1 = d(S_i) + d(S_j) \).

Thus, \( d(S) + d(S) \leq d(S_i) + d(S_j) \), and we have a (possibly trivial) maximal partitioning of \( \mathcal{V} \) with size \( t' \leq 2 \).

The inductive step proceeds as before, and assuming \( S_i, S_j \) nonconsecutive, we produce the consecutive partitions \( S_i' \) with \( S_i' \cup S_j = S_i \cup S_j \), with possibly one of \( S_i', S_j' \) empty.

If, without loss of generality, \( S_j' = \emptyset \), then \( d(S_i') + d(S_j') \leq \max(M_{S_k}, M_{S_j}) - \min(m_{S_k}, m_{S_j}) - 1 = d(S_i) + d(S_j) \).

Thus, \( d(S) + d(S) \leq d(S_i) + d(S_j) \), and we arrive at a contradiction. Thus, we obtain, for any \( t \), an optimal consecutive partitioning \( P \) of \( \mathcal{V} \), possibly containing empty sets and thus having size \( t' \leq t \). Therefore, WCPP holds for \( F \).
### 4.1. Further Characterization of the Partition Polytopes

Proposition 1 demonstrates that the extreme points of the unconstrained partition polytope $C$ consist only of consecutive nonsplitting subsets, while the extreme points of the constrained partition polytope $C$ may include consecutive nonsplitting subsets (excluding $\emptyset$ and $\mathcal{V}$), singleton splitting subsets, and their (nonconsecutive) complements. We can further characterize these polytopes by showing that (i) all consecutive nonsplitting subsets $p_S \in \Pi(\mathcal{U})$ are on the boundary of the unconstrained partition polytope, and (ii) all consecutive nonsplitting subsets $p_S \in \Pi(\mathcal{U})$ are on the boundary of the constrained partition polytope.

Given the dataset $\mathcal{D} = \{(x_i, y_i)\}$, we define the summations $C^j_x = \sum_{i=1}^{n} x_i$, $C^j_y = \sum_{i=1}^{n} y_i$, $C_x^{-1} = \sum_{i=1, n} x_i$, and $y_1, \ldots, y_n$ for $1 \leq j \leq n$, with the conventions that $C_x^0 = C_y^0 = C_x^{-1} = C_y^{-1} = 0$ and $x_0 = y_0 = 0$.

Thus, $p_C^j = (x_j y, C^j_x, C^j_y)$ is the partition point associated with the ascending nonsplitting subset $C_j = \{1, \ldots, j\}$, while $p_{C^{-j}} = (x_j y, C_x^{-j}, C_y^{-j})$ is the partition point associated with the descending nonsplitting subset $C_j = \{j, \ldots, n\}$.

We can now construct the convex hull of the points $p_C$ for $j = 0, \ldots, n$ and $p_{C^{-j}}$ for $j = 2, \ldots, n$ using a Graham scan (Graham 1972). By repeated application of the inequality $\frac{x_j}{y_j} \leq \frac{x_{j+1}}{y_{j+1}} \leq \frac{x_k}{y_k}$ for $j < k$, it follows that $C^j_x \leq \ldots \leq C^{n}_x \leq C^j_y \leq \ldots \leq C^{n}_y$.

Thus, $p_C^j$ is on the boundary of the unconstrained partition polytope, and is an extreme point if $\frac{x_j-1}{y_j-1}$ is strictly less than $\frac{x_j}{y_j}$.

From this argument, we can see that all of the consecutive nonsplitting subsets $S \in \mathcal{U}$ correspond to points $p_S$ on the boundary of the unconstrained partition polytope $C$, and to extreme points if the priority function $g(x_j, y_j) = \frac{x_j}{y_j}$ is strictly increasing with $j$. Again, from Proposition 1, no other extreme points exist.

For the constrained partition polytope $C$, we remove $p_0 = p_{C^0}$ and $p_{\mathcal{V}} = p_{C^{n}}$, and consider the convex hull of the remaining points. All of the points $p_S$ corresponding to the consecutive nonsplitting subsets $S \in \mathcal{U}$ remain on the boundary of the convex hull when these two points are removed. Additionally, from Proposition 1, the singleton splitting pairs must be considered, as some or all of these may be vertices. See Figure 1 for an example of unconstrained and constrained partition polytopes, where removal of $p_0$ and $p_{\mathcal{V}}$ adds a singleton splitting pair to the boundary of the convex hull.

If any singleton is a vertex of $C$, it will lie between $p_{C^{-n}}$ and $p_C$ when the vertices of $C$ are traversed in a clockwise direction along the partition polytope. Since the singleton splitting subsets $p_{\{i\}}$ are indexed by $1 < i < n$, they can be enumerated and tested successively. Thus, we can perform a Graham scan to identify the lower convex hull of the points $p_{C \ldots n}$, $p_{(n-1)}$, $p_{(n-2)}$, $\ldots$, $p_{\{1\}}$, $p_C$ in clockwise order. For any points $p = (x,y)$, $p' = (x', y')$, and $p'' = (x'', y'')$ in the given order, if $(x'-x)(y''-y)-(y'-y)(x''-x) > 0$, then $p'$ is to the left of $C$ and can be removed, and this process can be repeated until all remaining points are on the boundary of $C$.

![Figure 1. Example of partition polytope C and constrained partition polytope C](image)

Finally, the extreme point classification allows us to specify sufficient conditions on the dataset $\mathcal{D} = \{(x, y)\}$ under which the solution to expression (1) occurs for a consecutive partitioning of size $t$, for $f$ convex but not necessarily subadditive:
Theorem 3. Given a partition scan statistic $F(P)$ with associated score function $f(x, y)$ and priority function $g(x, y) = \frac{x}{y}$, and a dataset $D = \{(x_i, y_i)\}$ (for $i = 1 \ldots n$) ordered by priority. If $f$ is convex, and $(x_i - x_n)(y_1 - y_n) - (y_i - y_n)(x_1 - x_n) \geq 0$ for $i = 2 \ldots n - 1$, then the solution to

$$\arg\max_{P=\{S_1, \ldots, S_t\}} \sum_{j=1 \ldots t} f \left( \sum_{i \in S_j} x_i \right) \sum_{i \in S_j} y_i$$

is a consecutive partitioning of size $t$.

Proof. By the right turn rule, if $(x_i - x_n)(y_1 - y_n) - (y_i - y_n)(x_1 - x_n) \geq 0$, then the singleton splitting subset $P_{\{i\}}$ lies above the segment joining the vertices $p_{C_i}$ and $p_{C_1}$, and is thus in the interior of $C_i$, not an extreme point. If all singleton splitting subsets are in the interior, then by Proposition 1, the only extreme points of $C$ correspond to consecutive nonsplitting subsets, and the symmetric score function $f_n(x, y)$ is maximized at one of these subsets. Finally, following Theorems 1 and 2, the partition scan statistic is maximized for a consecutive partitioning.

For example, for the sets $X = \{7.0, 9.4, 8.9, 7.5, 6.9, 7.2, 7.5, 6.3\}$ and $Y = \{8.4, 9.3, 6.7, 4.7, 1.8, 1.4, 0.6, 0.1\}$, we know that any partition scan statistic with a convex score function $f$ satisfies CPP for this particular choice of $X$ and $Y$.

5. Efficient Optimization over Consecutive Partitionings

As noted in Section 3, an exhaustive search over all size-$t$ partitionings $P$, in order to find the optimal partitioning $P^* = \arg\max_{P=\{S_1, \ldots, S_t\}} F(P)$, is computationally infeasible even for relatively small datasets, requiring time exponential in the number of data records $n$. However, if a partition scan statistic is known to satisfy CPP or WCPP, then the corresponding optimization can be accelerated substantially by evaluating only the much smaller set of consecutive partitionings.

A constrained optimization over the set of all size-$t$ consecutive partitionings has cost that grows as $(n-t-1)$, that is, as a polynomial of degree $t-1$ in $n$. For very small $t$, exhaustive enumeration of the consecutive partitionings is a low-degree polynomial in $n$, and thus computationally feasible, but this approach does not scale well for larger $t$.

Thus, we propose an alternative approach based on dynamic programming (Algorithm 1), which runs in quadratic $O(n^2t)$ time and requires linear $O(nt)$ space. Algorithm 1 finds the highest-scoring consecutive partitioning, $\arg\max_{P=\{S_1, \ldots, S_t\}} F(P)$, for each size $t' \in \{1, \ldots, t\}$. For a partition scan statistic $F(P)$ satisfying CPP, we are guaranteed that the highest-scoring consecutive partitioning of size $t$ is the optimal size-$t$ partitioning. For a partition scan statistic satisfying WCPP but not CPP, the highest-scoring consecutive partitioning of size $t$ is only guaranteed to be the optimal size-$t$ partitioning if no other consecutive partitioning of size $t' < t$ has higher score.

For each size $t' \in \{1, \ldots, t\}$ and each element $j \in \{1, \ldots, n+1-t'\}$, Algorithm 1 computes two quantities: (a) the maximum score $F^*(j, t')$ for dividing elements $\{j, \ldots, n\}$ into $t'$ consecutive partitions, and (b) the starting element $\rho(j, t')$ of the second partition of the highest-scoring consecutive size-$t'$ partitioning of $\{j, \ldots, n\}$. In the final stage of the algorithm (lines 25–36), the highest-scoring consecutive partitioning of each size $t' \in \{1, \ldots, t\}$ is recovered by repeatedly applying $\rho$, starting at $j = 1$, and its corresponding score is $F^*(1, t')$.

The key step of the algorithm (lines 16–20) is that $F^*(j, t') = \max_{k \in \{1, \ldots, n+1-t'\}} F^*(j, t', k) + F^*(k+1, t'-1)$, where $k \in \{j, \ldots, n+1-t'\}$, $F^*(j, t', k)$ is the score of the partition formed by elements $\{j, \ldots, k\}$, and $F^*(k+1, t'-1)$ is the maximum score for dividing elements $\{k+1, \ldots, n\}$ into $t' - 1$ consecutive partitions. We note that the sufficient statistics $(C_x, C_y)$ are aggregated iteratively (lines 4 and 15), giving constant-time computation for each combination of $t \in \{1, \ldots, t'\}$, $j \in \{1, \ldots, n+1-t'\}$, and $k \in \{j, \ldots, n+1-t'\}$, for a total run time of $O(t^2n)$.

A display of run times of Algorithm 1 as a function of $n$, for various values of $t$, is shown in Figure 2. We observe the quadratic dependence of run time on $n$ and linear dependence on $t$. Average run time was under 2.7 sec for all $n \leq 5000$ and $t \leq 100$.

6. Partition Scan Statistics for Multiple Cluster Detection

The partition scan statistics presented in Section 2.1 generalize Kulldorff’s spatial scan statistic (Kulldorff 1997) from $t = 2$ to $t > 2$ partitions and from the Poisson distribution to other distributions in a separable exponential family. As such, they are best applied to risk partitioning, where the goal is to optimally divide the dataset into $t$ partitions of differing risks $q_1, \ldots, q_t$. We now consider the application of partition scan statistics to the related problem of multiple cluster detection, where we place additional constraints on the values of $q_1, \ldots, q_t$ (and $q_{\text{null}}$) in order to distinguish clusters of elevated risk from the remaining background data. More precisely, consider the null and alternative hypotheses in Section 2.1,

$$H_0: x_i \sim \Psi(\cdot; \mu = q_{\text{null}}, \Sigma) \forall S_I,$$

$$H_1(P): x_i \sim \Psi(\cdot; \mu = q_i, \Sigma) \forall S_i \in S_I,$$
with constraints \(q_j > 1\) for \(j \in \{2, \ldots, t\}\), and \(q_1 = q_{\text{all}} = 1\). Then Equation (2) simplifies to:

\[
F_{\text{clust}}(P) = F_{\text{clust}}((S_1, \ldots, S_t)) = \log \max_{q_2, \ldots, q_t > 1} \prod_{j=2}^{t} \prod_{s_j \in S_j} \frac{\Pr(x_j | x_i \sim \psi(\mu = q_j \mu_j))}{\Pr(x_i | x_i \sim \psi(\mu = \mu_j))}.
\]

This generalizes the expectation-based scan statistic (Neill 2012) from a single cluster \((t = 2)\) to multiple clusters \((t > 2)\). For consistency with Neill (2012), we allow partitions to be empty, so that the detected clusters (nonempty partitions with \(q_j > 1\)) could include some, all, or none of the data elements.

For expectation-based scan statistics in a separable exponential family, following Neill (2012):

\[
F_{\text{clust}}(P) = F_{\text{clust}}((S_1, \ldots, S_t)) = \sum_{j=2}^{t} f(C_j, B_j),
\]

where \(C_j\) and \(B_j\) are the sufficient statistics for partition \(S_j\), \(C_j = \sum_{i \in S_j} T(x_i) x_i\) and \(B_j = \sum_{i \in S_j} \mu_i z_i\), respectively, and the score function \(f(x, y) = yD_{\phi_0}(\frac{x}{y}, 1)\) for \(x > y\) or 0 otherwise, where \(D_{\phi_0}\) is a Bregman divergence. We can now show:

**Theorem 4.** Let \(F_{\text{clust}}(P) = F_{\text{clust}}((S_1, \ldots, S_t)) = \sum_{j=2}^{t} f(\sum_{i \in S_j} x_i, \sum_{i \in S_j} y_i)\) be the multiple cluster scan statistic corresponding to the expectation-based scan statistic for a distribution \(\Psi\) in a separable exponential family, with associated score function \(f(x, y) = yD_{\phi_0}(\frac{x}{y}, 1)\), where \(D_{\phi_0}\) is a Bregman divergence, and priority function \(g(x, y) = \frac{x}{y}\). Let dataset \(D = \{(x_i, y_i)\}, i = 1 \ldots n\) be ordered by priority. Then the solution to \(P^* = \arg \max_{P \in \left\{(S_1, \ldots, S_t)\right\}} F_{\text{clust}}(P)\), allowing empty partitions, is a consecutive partitioning of \(D\), of size \(t' \leq t\), with \(S_1\) ascending consecutive.

**Proof.** We first note that the score function \(f(x, y)\) is convex, since Bregman divergences are convex in their first argument and the perspective of a convex function is convex. We now proceed by induction. For the case \(t = 2\), the convex score function \(f(x, y)\) is maximized at an extreme point of the partition polytope \(C\), which is consecutive by Proposition 1. By the reasoning in Section 4.1, the order of clockwise traversal of the vertices of \(C\), starting from the origin \(p_{C0} = p_{C_{t-1}}(\mu)\), is \(p_{C1}, \ldots, p_{Cn}, p_{C_{t-1}}, \ldots, p_{Cn}\). Since the \(y\)-coordinates are increasing on \(p_{C1}, \ldots, p_{Cn}\), it follows that for each point \(p_{Cj}\) for \(j \in \{1, \ldots, n - 1\}\), there is a \(\delta > 0\) such that \((C_k + \delta, C_j)\) lies in \(C\). Since \(f\) is nondecreasing in \(x\), the maximizing point must be descending consecutive of the form \(p_{Cj-k+1}\), for \(1 \leq k \leq (n + 1)\). Thus, we have \(S_2\) descending consecutive and \(S_1\) ascending consecutive, allowing for the possibility that either partition could be empty.

For \(t > 2\), consider any pair of subsets \(S_i\) and \(S_j\), and assume without loss of generality that \(j \neq 1\). There are two cases. First, if \(i = 1\), then by the above reasoning, the optimal subset \(S_j'\) of \(S_i \cup S_j\) is descending consecutive, and the corresponding \(S_j\) is ascending consecutive. Second, if \(i \neq 1\), we proceed as in Theorems 1 and 2. Since \(f\) is convex, the symmetric score function \(\tilde{f}\) is also convex, and is maximized at an extreme point of \(C\). By identical reasoning to Theorems 1 and 2, there exist optimal consecutive partitions \(S_j'\) and \(S_j''\) whose union is \(S_i \cup S_j\). \(\square\)

This generalizes the linear-time subset scanning property for expectation-based scan statistics in the separable exponential family (Neill 2012) from a single detected cluster \((t = 2)\) to multiple detected clusters \((t > 2)\). Randomization testing can be used to assess the statistical significance of the detected clusters, as in Kulldorff (1997) and Neill (2012). For \(t = 2\), the fast subset scan approach of Neill (2012) can be used to identify the highest-scoring subset in \(O(n \log n)\) time while evaluating only \(O(n)\) of the \(2^n\) subsets. For \(t > 2\), a slight modification

---

**Algorithm 1** Dynamic Programming Algorithm for Optimizing over Consecutive Partitionings

1: \# Base case \((t' = 1)\)
2: \((C_x, C_y) = (0, 0)\)
3: for \(j \in \{n, n-1, \ldots, 1\}\) do
4: \((C_x, C_y) = (C_x, C_y) + (x_j, y_j)\)
5: \(F^*(j, 1) = f(C_x, C_y)\)
6: \(\rho(j, 1) = n + 1\)
7: end for
8: \# Iterative step
9: for \(t' \in \{2, \ldots, t\}\) do
10: for \(j \in \{1, \ldots, (n + 1 - t')\}\) do
11: \(F^*(j, t') = -\infty\)
12: \((C_x, C_y) = (0, 0)\)
13: for \(k \in \{j, \ldots, (n + 1 - t')\}\) do
14: \((C_x, C_y) = (C_x, C_y) + (x_k, y_k)\)
15: \(F^*(j, k) = f(C_x, C_y) + F^*(k + 1, t' - 1)\)
16: if \(F^*(j, k) > F^*(j, t')\) then
17: \(F^*(j, t') = F^*(j, k)\)
18: \(\rho(j, t') = k + 1\)
19: end if
20: end for
21: end for
22: end for
23: end for
24: \# Recover the highest-scoring consecutive partitioning for each \(t'\)
25: for \(t' \in \{1, \ldots, t\}\) do
26: \(\text{scores}[t'] = F^*(1, t')\).
27: \(\text{partitions}[t'] = []\)
28: \(j = 1;\)
29: for \(t'' \in \{t', (t' - 1), \ldots, 1\}\) do
30: \(j_{\text{next}} = \rho(j, t'')\)
31: Append \([j, j_{\text{next}} - 1]\) to \(\text{partitions}[t']\)
32: \(j = j_{\text{next}}\)
33: end for
34: end for
35: return \(\text{scores, partitions}\)

---

2We note that quasi-convexity of \(f\) is sufficient for the \(t = 2\) case (as proved in Neill 2012) but not for \(t > 2\), since the symmetric score function \(\tilde{f}\) corresponding to a quasi-convex score function \(f\) is not necessarily quasi-convex. Subadditivity is not necessary here, since we allow empty partitions.
We now consider how to choose the number of partitions of the dynamic programming approach described in Section 5 can identify the highest scoring partitioning, and thus detect multiple clusters, in $O(n^2)$ time. To do so, we compute $F^w_t(j, t')$ as above, and $F^w_{\text{clust}} = \max_{t'} \max_j F^w(j, t')$, for $t' \in \{1, \ldots, t - 1\}$ and $j \in \{1 \ldots (n + 1 - t')\}$.

Several alternative approaches have been proposed for extending the spatial scan statistic to multiple cluster detection (Kulldorff 1997; Zhang, Assuncao, and Kulldorff 2010; Li et al. 2011; Takahashi and Shimadzu 2020). Most of these approaches work sequentially, first detecting the primary cluster and removing it from the data before searching for significant secondary clusters. Kulldorff’s original spatial scan (Kulldorff 1997) compares the scores of the secondary clusters to the same detection threshold, obtained by randomization testing, as the primary cluster. Zhang, Assuncao, and Kulldorff (2010) note that this approach is conservative and adjust the threshold for more sensitive detection of secondary clusters, while Takahashi and Shimadzu (2020) propose a generalized linear model and information-theoretic criterion to select the number of clusters, starting with candidate clusters identified by the sequential method. Repeated single cluster detection is computationally efficient, scaling linearly with the number of detected clusters, but does not directly optimize the log-likelihood ratio statistic for multiple clusters. In contrast, we exactly and efficiently find the highest scoring partitioning in the unconstrained sense, which can be thought of as maximizing the joint LLR score corresponding to multiple detected clusters. This is most similar to Li et al. (2011), who explicitly scan over tuples of clusters and compute joint scores, but their approach requires an exhaustive search that scales exponentially with the number of clusters to be detected (and thus is typically limited to two or three clusters), while our approach can easily scale to very large $n$ and $t$. Note that the goal of these previously proposed methods is different from ours: they focus entirely on spatial cluster detection (typically scanning over circular regions in space), while our approach identifies an optimal partitioning of the data without an explicit spatial constraint.

### 7. Choosing the Number of Partitions $t$

We now consider how to choose the number of partitions $t$, in cases when the desired value of $t$ is not known a priori. We propose a novel, computationally efficient heuristic for finding $t$, and show in our experiments below that this approach can obtain the correct $t$ value in practice. For a given dataset $D$, let $F^w_t$ denote the maximum score $\max_{t'=\{n, \ldots, n\} \cup F(P)$ for separating $D$ into $t$ partitions, and assume that we wish to choose $t \in \{1, 2, \ldots, t_{\text{max}}\}$ for some constant $t_{\text{max}}$. Our heuristic is based on several observations. First, as shown in Figure 3(a), we identify a power law relationship between the average score difference $\Delta_t = F^w_t - F^w_{t-1}$ and the value of $t$, for data generated under the null hypothesis $H_0$, such that $\log \Delta_t$ is linearly related to $\log t$. Second, as shown in Figure 3(b), for data generated under the alternative hypothesis $H_t(P)$ with $t_{\text{true}}$ distinct partitions, the relationship between $\log \Delta_t$ and $\log t$ deviates from linearity. In this case, we observe that the point $(\log t$, $\log \Delta_t)$ corresponding to $t = t_{\text{true}} + 1$ partitions has a large negative residual, indicating that the increase in score from adding partition $t_{\text{true}} + 1$ is less than expected given the overall trend. Intuitively, when partitions are sufficiently distinct, the $\Delta_t$ from separating two true partitions, when $t \leq t_{\text{true}}$, will be large compared to the $\Delta_t$ from splitting apart a single true partition, when $t > t_{\text{true}}$. Finally, we note that all scores $F^w_1, \ldots, F^w_{t_{\text{max}} + 1}$ can be obtained efficiently from a single run of Algorithm 1, as opposed to requiring a separate run for each $t$.

This suggests the following algorithmic approach:

1. Compute $F^w_{t_{\text{max}} + 1}$ as in Algorithm 1, storing the intermediate results $F^w_1, \ldots, F^w_{t_{\text{max}}}$.  
2. Compute $\log \Delta_t = \log(F^w_t - F^w_{t-1})$ for each $t \in \{2, 3, \ldots, t_{\text{max}} + 1\}$.  
3. Fit a linear model $\log \Delta_t = w_1(\log t) + w_0$, and compute residuals for each $t$.  
4. Set $t_{\text{opt}} = \arg\min_t(\log \Delta_t - w_1(\log t) - w_0) - 1$.  

The time complexity of this algorithm is $O(n^2 t_{\text{max}}) + O(t_{\text{max}}) = O(n^2 t_{\text{max}})$, where the first term for the dynamic programming dominates the second term for the linear regression.

### 8. Simulation Experiments

We now perform two sets of simulation experiments to compare the partition scan to the previously proposed Kulldorff and expectation-based Poisson scan statistics, in the risk partitioning and multiple cluster detection settings, respectively.

---

**Figure 3.** Plot of $\log(F^w_t - F^w_{t-1})$ versus $\log t$, for data generated under the null hypothesis $H_0$ and under the alternative hypothesis $H_1$ (with five true partitions and $\epsilon = 0.75$), respectively. Values of $t \in \{2, 3, \ldots, 11\}$ are shown. Note the large negative residual at $t = 6$ in plot (b).
### 8.1. Risk Partitioning Simulations

For the first set of simulations, we evaluate the partition scan statistic with Poisson score function, 
\[ F(P) = \sum_{j=1}^{t} f(C_j, B_j) - f(C_{all}, B_{all}), \]
where 
\[ f(x, y) = x \log(\frac{x}{y}). \]
We compare \( t = 2, t = 3, \) and \( t = 10 \) partitions, where the \( t = 2 \) case corresponds to the original Poisson scan (Kulldorff 1997).

We consider three different simulated scenarios (2 true partitions, 3 true partitions, and 10 true partitions). In each case, we generate simulated datasets each consisting of \( n = 2500 \) data records, where each data record has baseline \( y_i \sim \text{Poisson(100)} \) and count \( x_i \sim \text{Poisson}(q_i y_i) \), where \( q_i \) is the relative risk for that data record.

Each of the three scenarios is indexed by the "signal strength" \( \epsilon \), where we consider values of \( \epsilon \) ranging from 0 (no signal) to 0.5 (strongest signal). For "two true partitions," the relative risks \( q_i \) are \( 1 - \epsilon \) and \( 1 + \epsilon \) for data records in the first and second partition, respectively. For "three true partitions," we use relative risks of \( 1 - \epsilon, 1, \) and \( 1 + \epsilon \) for the three partitions. For "ten true partitions," the relative risks are \( 1 - \epsilon, 1 - 0.8\epsilon, 1 - 0.6\epsilon, 1 - 0.4\epsilon, 1 - 0.2\epsilon, 1 + 0.2\epsilon, 1 + 0.4\epsilon, 1 + 0.6\epsilon, 1 + 0.8\epsilon, \) and \( 1 + \epsilon \).

For all three scenarios, the partitions are equal in size. For each scenario and each signal strength \( \epsilon \), we generated 500 simulated datasets. For each dataset and for each parameter setting (\( t = 2, t = 3, \) and \( t = 10 \)), we compute the optimal partitioning \( P^* = \arg \max_P F(P) \) and its score \( F^* = F(P^*) \).

The methods were evaluated based on ranking quality, which measures how well the partitions created by the method correspond to the true partitions used to generate the data. To compute ranking quality, we consider the true partitions \( S_1^{\text{true}}, \ldots, S_m^{\text{true}} \), ordered from smallest to largest relative risk, and the detected partitions \( S_1, \ldots, S_t \), likewise ordered from smallest to largest relative risk. We then compute the probability that a pair of data records \( (i, j) \) drawn uniformly at random with replacement, with corresponding true partitions \( S_i^{\text{true}} \) and \( S_j^{\text{true}} \), and corresponding detected partitions \( S_i \) and \( S_j \), are ordered correctly by the method (i.e., \( S_i > S_j \) if \( S_i^{\text{true}} > S_j^{\text{true}} \), \( S_i < S_j \) if \( S_i^{\text{true}} < S_j^{\text{true}} \), and \( S_i = S_j \) if \( S_i^{\text{true}} = S_j^{\text{true}} \)). Ranking quality can be calculated as

\[
RQ = \frac{\sum_{i=1}^{m} \sum_{j'=1}^{t} \sum_{i'=1}^{m} \sum_{j=1}^{t} N_{ij} N_{ij'} (\mathbb{1}[i < i'] \mathbb{1}[j < j'] + \mathbb{1}[i > i'] \mathbb{1}[j > j'] + \mathbb{1}[i = i'] \mathbb{1}[j = j'])}{\sum_{i=1}^{m} \sum_{j'=1}^{t} \sum_{i'=1}^{m} \sum_{j=1}^{t} N_{ij} N_{ij'}},
\]

**Figure 4.** Ranking quality for 2, 3, and 10 true partitions, comparing partition scans with \( t = 2 \) (solid line), \( t = 3 \) (dotted line), and \( t = 10 \) (dashed line) partitions.
where \( N_{ij} \) is the number of data records assigned to \( S_i \cap S_j \).

### 8.1.1. Risk Partitioning Simulation Results

We compare the ranking quality for partition scans with \( t = 2 \), \( t = 3 \), and \( t = 10 \), for varying numbers of true partitions, in Figure 4. For two true partitions (Figure 4(a)), we observe that the partition scan with \( t = 2 \) converges to the two correct partitions, while the ranking quality of the \( t = 3 \) and \( t = 10 \) partition scans is reduced because a single true partition is split between multiple detected partitions. For three true partitions (Figure 4(b)), we observe that the ranking quality of the three methods is very similar for low \( \epsilon \) values. For higher \( \epsilon \), the correct number of partitions \( t = 3 \) outperforms \( t = 2 \) and \( t = 10 \), with near-perfect partitions for the highest \( \epsilon \) values considered. For 10 true partitions (Figure 4(c)), we again observe substantial improvement in ranking quality for the correct number of partitions \( t = 10 \) as compared to \( t = 3 \) and \( t = 2 \).

Across these three scenarios, we observe that, when the number of partitions \( t \) is correctly specified, the partition scan con-
verges to the correct partitioning of the data as signal strength \( \epsilon \) increases. However, when the number of partitions \( t \) is mis-specified, ranking quality is reduced either because a single true partition is split across multiple partitions, or because multiple true partitions are combined into a single partition. This is illustrated in Figure 5, which shows sample confusion matrices with \( \epsilon = 0.5 \) for the \( t = 10 \) case, when two true partitions exist, and the \( t = 2 \) case, when 10 true partitions exist.

These examples demonstrate the importance of choosing the correct value of the number of partitions \( t \). Thus, we evaluate our efficient heuristic for choosing \( t \), described in Section 7. We compute the mean and 95% confidence interval of the chosen \( t \) (across the 500 simulated datasets) for the cases of two, three, five, and seven true partitions, as shown in Figure 6. We observe that, as the signal strength \( \epsilon \) increases, our heuristic is able to reliably choose the correct number of partitions \( t \). However, the signal strength required for convergence increases with the number of true partitions, from \( \epsilon \approx 0.2 \) for two true partitions, to \( \epsilon \approx 0.9 \) for seven true partitions.

### 8.2. Multiple Cluster Detection Simulations

For the second set of simulations, we evaluate the multiple cluster detection (MCD) scan statistic with Poisson score function, \( F(P) = \sum_{j=2}^{t} f(C_j, B_j) \), where \( f(x, y) = x \log(x/y) + y - x \) if \( x > y \) and 0 otherwise. We compare \( t = 2 \) and \( t = 3 \) partitions, where the \( t = 2 \) case corresponds to the expectation-based Poisson (EBP) scan statistic (Neill 2012). We denote the \( t = 2 \) and \( t = 3 \) cases as EBP and MCD, respectively. We also compare performance to Kulldorff’s Poisson scan (i.e., partition scan with \( t = 2 \) partitions) and the partition scan with \( t = 3 \) partitions, which we denote as KULL and PART, respectively.

As in the risk partitioning experiments, we generate datasets consisting of \( n = 2500 \) data records, where each data record has baseline \( y_i \sim \text{Poisson}(100) \) and count \( x_i \sim \text{Poisson}(q_i y_i) \). We focus on the case where there are two true clusters to be detected, a primary cluster with relative risk \( q_1 \) and a secondary cluster with relative risk \( q_2 \), where \( q_1 > q_2 > 1 \). Each cluster consists of 10% of the data records, while the remaining 80% of records are generated with \( q_i = 1 \). As in the risk partitioning experiments, we consider a range of signal strengths \( \epsilon \), where the primary cluster has relative risk \( q_1 = 1 + \epsilon \). We consider three different scenarios: in the “low” scenario, the secondary cluster has relative risk \( 1 + \frac{q_2 q_1}{q_1} \), in the “medium” scenario, the secondary cluster has relative risk \( 1 + \frac{q_2}{2} \), and in the “high” scenario, the secondary cluster has relative risk \( 1 + \frac{3 q_2}{4} \). For each scenario and each signal strength \( \epsilon \), we generated 500 simulated datasets.

![Figure 7](image-url). Multiple cluster detection with medium-strength secondary cluster. MCD (solid line) is multiple cluster detection with \( t = 3 \) partitions, PART (dotted line) is risk partitioning with \( t = 3 \), EBP (dashed line) is cluster detection with \( t = 2 \), and KULL (dash-dotted line) is risk partitioning with \( t = 2 \).
datasets. An additional 10,000 datasets were generated under the null hypothesis $H_0$ ($q_i = 1$ for all data records), to be used for computing detection power. For each dataset and for each method (MCD, PART, EBP, and KULL), we compute the optimal partitioning $P^* = \arg\max_P F(P)$ and its score $F^* = F(P^*)$.

The methods were evaluated based on two criteria, detection power, which measures a method's ability to distinguish datasets generated under the alternative hypothesis from those generated under the null hypothesis of no clusters, and detection accuracy, which measures how well the clusters identified by the method correspond to the true clusters used to generate the data. To compute detection power, we first compute the threshold score needed for a method to detect at a fixed false positive rate of $\alpha = 0.05$. To do so, we compute the maximum scores $P^n$ for 10,000 datasets under the null hypothesis and use the 95th percentile of these scores as the detection threshold. We then compute the proportion of the 500 datasets (for a given scenario and a given signal strength $\epsilon$) for which $F^n$ exceeds the threshold. Thus, detection power is the statistical power ($1 - \text{Type II error}$) at a fixed false positive rate (Type I error) of 5%.

We compute three different pairwise measures of detection accuracy which together give a more complete picture of each method’s ability to identify the primary and secondary clusters. Consider the detected partitions $S_1, \ldots, S_t$, ordered from smallest to largest relative risk. Then primary cluster detection accuracy is defined as the probability that, for a data record $i$ from the primary cluster and an unaffected data record $j$, each selected uniformly at random, with corresponding partitions $S_i$ and $S_j$, that $S_i > S_j$. Similarly, secondary cluster detection accuracy is defined as the probability that $S_i > S_j$ for a data record $i$ from the secondary cluster and an unaffected data record $j$. Finally, cluster differentiation accuracy is defined as the probability that $S_i > S_j$ for data records $i$ from the primary cluster and $j$ from the secondary cluster. For $i = 1 \ldots t$, let $N_i^{(p)}$, $N_i^{(s)}$, and $N_i^{(u)}$ equal the number of records in the primary cluster, secondary cluster, and unaffected records, respectively, that are assigned to partition $i$. Then primary cluster detection accuracy is defined as

$$\text{PCDA} = \frac{\sum_{i=2}^{t-1} \sum_{i'=1}^{t-1} N_i^{(p)} N_{i'}^{(u)} - \sum_{i=1}^{t-1} \sum_{i'=1}^{t-1} N_i^{(p)} N_{i'}^{(u)}}{\sum_{i=1}^{t-1} \sum_{i'=1}^{t-1} N_i^{(p)} N_{i'}^{(u)}}.$$  

Secondary cluster detection accuracy and cluster differentiation accuracy are defined analogously. For $t = 3$ partitions, perfect detection—where all unaffected records are assigned to partition $S_1$, all records in the secondary cluster are assigned to partition $S_2$, and all records in the primary cluster are assigned to partition $S_3$—would score 1 for each of these three accuracy measures.
Figure 9. Multiple cluster detection with high-strength secondary cluster. MCD (solid line) is multiple cluster detection with \( t = 3 \) partitions, PART (dotted line) is risk partitioning with \( t = 3 \), EBP (dashed line) is cluster detection with \( t = 2 \), and KULL (dash-dotted line) is risk partitioning with \( t = 2 \).

8.2.1. Multiple Cluster Detection Simulation Results

We first consider the "medium" scenario, for which the primary and secondary clusters have relative risks \( 1 + \epsilon \) and \( 1 + \frac{\epsilon}{2} \), respectively. Detection power, primary cluster detection accuracy, secondary cluster detection accuracy, and cluster differentiation accuracy for the MCD, PART, EBP, and KULL methods are shown in Figure 7.\(^4\) For detection power, we observe that the cluster detection methods (MCD and EBP) have substantially higher detection power than the risk partitioning methods (PART and KULL). PART slightly outperforms KULL, while MCD and EBP have very similar detection power. For primary cluster detection accuracy, the methods with \( t = 3 \) partitions (MCD and PART) outperform the methods with \( t = 2 \) (EBP and KULL), and the cluster detection methods slightly outperform the risk partitioning methods. For secondary cluster detection accuracy, MCD and PART again outperform EBP and KULL. MCD outperforms PART across most of the range of \( \epsilon \) values, while KULL slightly outperforms EBP. Finally, for cluster differentiation accuracy, we see a clear ordering of methods with MCD > PART > EBP > KULL. As the signal strength \( \epsilon \) increases, MCD and PART correctly distinguish the primary cluster, the secondary cluster, and the remaining records, while EBP and KULL detect the primary and secondary cluster together as a single cluster, and thus, fail to distinguish primary from secondary.

Next we consider the "low" scenario, for which the primary and secondary clusters have relative risks \( 1 + \epsilon \) and \( 1 + \frac{\epsilon}{4} \), respectively. Detection power, primary cluster detection accuracy, secondary cluster detection accuracy, and cluster differentiation accuracy for the MCD, PART, EBP, and KULL methods are shown in Figure 8. We see the same general trends in detection power, primary cluster detection accuracy, and cluster differentiation accuracy as in the medium case. One notable difference is in secondary cluster detection accuracy: as signal strength increases, the methods with \( t = 2 \) partitions (EBP and KULL) only detect the primary cluster and fail to detect the secondary cluster, while the methods with \( t = 3 \) (MCD and PART) correctly distinguish the primary cluster, the secondary cluster, and the remaining records.

Finally, we consider the "high" scenario, for which the primary and secondary clusters have relative risks \( 1 + \epsilon \) and \( 1 + \frac{3\epsilon}{4} \), respectively. Detection power, primary cluster detection accuracy, secondary cluster detection accuracy, and cluster differentiation accuracy for the MCD, PART, EBP, and KULL methods are shown in Figure 9. We see the same trends in detection power and cluster detection accuracy as in the medium case. One notable difference is in cluster differentiation accuracy. As in the medium case, the methods with \( t = 2 \) partitions (KULL

\(^4\)Note the smaller x-axis ranges for Figures 7(a), 8(a), and 9(a) as compared to the rest of Figures 7–9. All methods had perfect detection power for \( \epsilon > 0.2 \), so we focus on the lower signal strengths for the detection power graphs.
Figure 10. Multiple cluster detection with medium-strength secondary cluster, fixed signal strength of $\epsilon = 0.5$, and varying cluster size. MCD (solid line) is multiple cluster detection with $t = 3$ partitions, PART (dotted line) is risk partitioning with $t = 3$, EBP (dashed line) is cluster detection with $t = 2$, and KULL (dash-dotted line) is risk partitioning with $t = 2$.

and EBP) detect the primary and secondary cluster together as a single cluster, and thus, fail to distinguish primary from secondary. However, the methods with $t = 3$ (MCD and PART) also have difficulty distinguishing primary from secondary: for risk partitioning with $t = 3$, even for high signal strengths, the primary and secondary clusters are detected as a single cluster in the third (highest risk) partition, while a number of unaffected records are incorrectly detected as a “false positive” cluster in the second (medium-risk) partition. For multiple cluster detection with $t = 3$, this problem occurs for medium signal strengths ($\epsilon \approx 1$), while for high signal strengths ($\epsilon \approx 2$), MCD is able to correctly distinguish the primary cluster, secondary cluster, and unaffected records.

Thus, across these three scenarios, we observe that multiple cluster detection with $t = 3$ (MCD) outperforms risk partitioning with $t = 3$ (PART) as well as the methods with $t = 2$ (EBP and KULL). Moving from single to multiple cluster detection (i.e., moving from $t = 2$ to $t = 3$) consistently improves detection accuracy and ability to differentiate between the primary and secondary clusters, while the single cluster detection methods will either fail to detect the secondary cluster (if its signal is much weaker than the primary cluster) or combine the primary and secondary clusters into a single detected cluster. Additionally, while the partition scan performs adequately for the multiple cluster detection task, the multiple cluster detection scan consistently outperforms PART in terms of both detection power and detection accuracy.

To confirm these results, we perform an additional set of simulation runs, based on the “medium-strength secondary cluster” scenario. Instead of fixing each cluster size at 10% of the data records, and allowing the signal strength $\epsilon$ to vary, we fix $\epsilon = 0.5$ and allow cluster sizes to vary between 1% and 20% of the data records, as shown in Figure 10. As expected, smaller cluster size, like weaker signal strength, reduces the performance of all methods, while keeping the relative performance of methods unchanged. We observe that MCD outperforms PART, EBP, and KULL in terms of both detection power and accuracy. PART has higher primary cluster detection accuracy, secondary cluster detection accuracy, and cluster differentiation accuracy than EBP and KULL, while EBP has higher detection power than PART for small cluster sizes.

9. Empirical Study—Cancer Incidence

We apply the multiple cluster detection scan approach to identify high-risk cancer clusters in New York City, using cancer incidence data from 2013 to 2017, grouped by census tract. Data were obtained from the New York State Cancer Reg-
istry (New York State Dept. of Health 2021), including the observed count (number of cases) and the expected count for the three most common cancer types (prostate, lung, and breast) for each tract during the five-year time period. Expected counts were population- and age-adjusted, using the statewide cancer rate as a baseline. In this way we identify regions of high cancer risk and possibly correlate them with a common feature: demographic, socio-economic, geographic, etc. We use the approach described in Section 7 to choose the number of partitions \( t \) between \( t = 1 \) and \( t = 20 \), obtaining \( t = 3 \), \( t = 4 \), and \( t = 5 \) (i.e., 2, 3, and 4 detected clusters) for lung, breast, and prostate cancers, respectively. The results are shown in Figure 11(a)–(c), respectively, where darker shades correspond to partitions with higher relative risks \( q \). We observe that all three cancer types cluster spatially, and that much of this clustering can be explained by neighborhood demographic characteristics. For example, spatial clustering is most evident for prostate cancer, which is known to afflict African–American males at a much higher rate than white males (New York State Dept. of Health 2021). We also compare the results to single cluster detection \( (t = 2) \) for each cancer type; we show prostate cancer as an example in Figure 11(d). We observe that the higher numbers of partitions \( t \) provide a finer-grained (and thus, more informative) gradation of risk as compared to the single cluster detection \( (t = 2) \) case, distinguishing between clusters with a higher and lower degree of elevated risk. For example, for prostate cancer, cluster relative...
risks varied from $q = 1.21$ to $q = 2.43$, as compared to $q = 1.70$ for the single cluster detection approach. Additionally, as expected, the chosen $t$ values more closely fit the observed data than $t = 2$, as measured by total log-likelihood: $LL = -5095$ versus $-5205$ for lung cancer, $-5322$ versus $-5441$ for breast cancer, and $-5580$ versus $-5786$ for prostate cancer.

### 10. Conclusions

The empirical results shown above demonstrate several advantages of generalizing the spatial and subset scan statistics to multiple partitions of the data: increased detection power and accuracy for multiple clusters, as well as the ability to accurately capture finer-grained gradations in risk. The practical utility of our risk partitioning and multiple cluster detection scan statistics was enabled by three novel methodological contributions: (i) formulating these partition scan statistics as generalizations of the population-based and expectation-based scan statistics, respectively; (ii) identifying sufficient conditions under which the optimal partitioning is guaranteed to be consecutive; and (iii) developing a new dynamic programming algorithm for identification of the optimal consecutive partitioning in quadratic time. Together, these contributions enable exact and efficient solutions to large-scale risk partitioning and multiple cluster detection problems for any partition scan statistic satisfying the Consecutive Partitions Property. Further, since CPP generalizes the linear-time subset scanning property from $t = 2$ to $t \geq 2$ partitions of the data, our novel, geometric proof of CPP also provides a new approach to proving LTSS.

One important limitation of the present approach, which we will believe will provide a fertile ground for future work, is that it only solves the "best unconstrained partitioning" problem, while for real-world problems it is desirable to include constraints, such as spatial proximity or connectivity, on the detected subsets or partitions of the data. We believe that, just as the LTSS property has proved to be a useful building block for solving constrained pattern detection problems (Neill 2012; Speakman, McFowland III, and Neill 2015), CPP will enable scalable solutions to a variety of constrained partitioning problems. However, significant work remains in defining constraints or penalties which encode desirable features of a partitioning while still preserving the efficient and exact optimization guaranteed by CPP.

### Supplementary Materials

C++ and Python code: The library contains exact solvers for partition scan statistics (both risk partitioning and multiple clustering scans) for score functions satisfying the Consecutive Partitions Property. Multiple objective functions from the generalized exponential family are supported, as is determination of the optimal partition size. The C++ routines are standalone or exposed through Python bindings. Scripts are provided to generate synthetic data and reproduce figures from the paper.

### Data Availability Statement

Code for this article is available at [https://github.com/pehlivanian/PartitionSolvers](https://github.com/pehlivanian/PartitionSolvers).

### Funding

DBN gratefully acknowledges funding support from the National Science Foundation Program on Fairness in Artificial Intelligence in Collaboration with Amazon, grant IIS-2040898.

### References


