

# Fixed and adaptive landmark sets for finite metric spaces

Jason Cory Brunson

Yara Skaf

# Introduction

Topological data analysis (TDA) is a maturing field in data science, at the interface of statistics, computer science, and mathematics. Topology is the discipline at the interface of geometry (the study of shape) and analysis (the study of continuity) that focuses on geometric properties that are preserved under continuous transformations. TDA consists in the use of computational theories of continuity to investigate the shape or structure of data. While TDA is most commonly associated with persistent homology (PH) and mapper, it can be understood to encompass and generalize many conventional and even classical techniques, including cluster analysis, network analysis, and nearest neighbors prediction.

Two basic maneuvers in TDA are locality preservation and cardinality reduction. *Locality preservation* is the property of some functions, such as projections or hashes, that nearby cases or points in the domain have nearby images in the codomain. (Continuity, as defined in analysis and topology, is a type of locality preservation.) This is the defining property of many non-linear dimensionality reduction techniques, including t-SNE and UMAP, but also an asymptotic property of nearest neighbors prediction and the locally-sensitive hashing used in its implementations. We use the term *cardinality reduction*<sup>1</sup>, in contrast to dimension reduction, to describe techniques that produce more tractable or comprehensible representations of complex data by reducing the number of cases or points rather than the number of variables or dimensions used to represent them. Cardinality reduction describes cluster analysis and association (co-occurrence or correlation) network analysis, for example.

More elaborate TDA techniques combine both maneuvers, often through the use of covering methods, as with the approximation of PH through witness complexes [deSilva2004] or in the mapper construction [Singh2007]. Covering methods, in turn, can be enhanced by strategic sampling from large data sets [deSilva2004], as can other proximity-based techniques like nearest neighbors. The maxmin procedure is often used for this purpose, as it is deterministic, is computationally efficient, and generates more evenly distributed samples than random selection. However, maxmin comes with its own potential limitations in the analysis of data that vary greatly in density or have multiplicities. This is a frequent concern when sparse, heterogeneous, and incomplete data are modeled as finite pseudometric spaces.

In this paper, we develop a landmark sampling procedure complementary to maxmin, based on the rankings of points by a distance or similarity measure rather than on the raw values of such a measure. In the remainder of this section, we motivate the procedure, which we call lastfirst, as a counterpart to maxmin obtained by adapting this procedure from the use of fixed-radius balls to the use of fixed-cardinality neighborhoods. We then formally describe the procedure and prove some of its basic properties in Section . In Section we report the results of benchmark tests and robustness checks of lastfirst on simulated and empirical data sets. We describe some basic and novel applications to real-world data in Section .

## Conventions

$(X, d_X)$  will refer to a finite pseudometric space with point set  $X$  and pseudometric  $d_X : X \times X \rightarrow \mathbb{R}_{\geq 0}$ , which by definition satisfies all of the properties of a metric except that  $d_X(x, y) = 0$  implies  $x = y$ .  $(X, d_X)$  may be shortened to  $X$ , and  $d_X$  to  $d$ , when clear from context. If  $x \neq y$  but  $d(x, y) = 0$  then  $x$  and  $y$  are said to be indistinguishable or co-located. The cardinality of  $Y \subseteq X$  (counting multiplicities) is denoted  $|Y|$ , and the set of distinguishable points of  $Y$ —or of equivalence classes under co-location—is denoted  $[Y]$ . When  $Y, Z \subseteq X$ , let  $Y \setminus Z$  denote the set of points in  $Y$  (with multiplicities) that are distinguishable from all points in  $Z$ . This means that, when defined,  $\min_{y \in Y \setminus Z, z \in Z} d(y, z) > 0$ .

---

<sup>1</sup>The term “cardinality reduction” takes different meanings in the data analysis literature, including the combining of different values of a categorical variable [MicciBarreca2001; Refaat2010] or of time series [Hu2011] (also “numerosity reduction” [Lin2007]). Our meaning is that of ByczkowskaLipinska2017: an  $n \times p$  data table of  $n$  cases (rows) and  $p$  variables (columns) can be dimension-reduced to an  $n \times q$  table, where  $q < p$ , or cardinality-reduced to an  $m \times p$  table, where  $m < n$ . The most common cardinality reduction method is data reduction.

We denote the diameter of  $Y$  as  $D(Y) = \max_{x,y \in Y} d(x,y)$ , and we write:

$$\begin{aligned} d(Y, Z) &= \min_{y \in Y, z \in Z} d(y, z) & D(Y, Z) &= \max_{y \in Y, z \in Z} (y, z) \\ d(x, Y) &= d(\{x\}, Y) & D(x, Y) &= D(\{x\}, Y) \end{aligned}$$

If  $d(x, y) = d(x, z) \implies y = z$ , then  $X$  is considered to be in *locally general position*, even if  $d(x, y) = d(z, w) \not\implies \{x, y\} = \{z, w\}$ . This condition implies that  $X$  is Hausdorff:  $d(x, y) = 0 \implies x = y$ .  $f : X \rightarrow Y$  will denote a morphism of pseudometric spaces, meaning a function (or morphism of sets) that preserves locality in the sense that  $d_X(x, y) \geq d_Y(f(x), f(y))$ .

We use the ball notation  $B_\varepsilon(x)$  for the set of points less than distance  $\varepsilon$  from a point  $x$ ; that is,  $B_\varepsilon(x) = \{y \mid d(x, y) < \varepsilon\}$ . We use an overline to also include points exactly distance  $\varepsilon$  from  $x$ :  $\overline{B}_\varepsilon(x) = \{y \mid d(x, y) \leq \varepsilon\}$ . (While these connote openness and closedness in the discrete topology, every such ball is both open and closed.) If  $|\overline{B}_\varepsilon(x)| \geq k$  and  $\varepsilon' < \varepsilon \implies |\overline{B}_{\varepsilon'}(x)| < k$ , then we say that  $N_k^+(x) = \overline{B}_\varepsilon(x)$  is the  $k$ -nearest neighborhood of  $x$ . When  $X$  is in general position, therefore,  $|N_k(x)| = k$ .

Throughout, let  $N = |X|$ . For convenience, we assume  $0 \in \mathbb{N}$ .

## Background

We designed the lastfirst procedure to address an issue with the maxmin procedure that arises when, due to the use of binary or categorical variables or to limits on measurement resolution, a data set includes many duplicate or otherwise indistinguishable cases. These render the finite metric space representation of the data non-Hausdorff. While these issues may be negligible when such points are rare, they raise computational and interpretative concerns when they are common. Because our procedure is motivated by the same practical needs as the maxmin procedure, we begin with a discussion of those needs.

The earliest appearance of the maxmin<sup>2</sup> procedure of which we are aware is by @deSilva2004. The authors propose witness complexes, later generalized to alpha complexes [ @], for the rapid approximation of persistent homology: Given a point cloud, a set of landmark points and their overlapping neighborhoods define a nerve, which stands in for the Vietoris–Rips complex at each scale. They use the maxmin procedure, which we define in Section , as an alternative to selecting landmark points uniformly at random. The procedure ensures that the landmarks are locally separated and roughly evenly distributed. While the procedure improved little upon uniform random selection in most use cases, on some tasks it far outperformed.

Subsequent uses of maxmin include the selection of a sample of points from a computationally intractable point cloud for the purpose of downstream topological analysis, as when performing the mapper construction [ @Singh2007]; and the optimization of a fixed-radius ball cover of a point cloud, in the sense of minimizing both the number of balls and their shared radius [ @Dlotko2019]. In addition to approximating persistent homology [ @deSilva2004; @Dlotko2019], maxmin has been used to reduce the sizes of simplicial complex models of point cloud data for the sake of visualization and exploration [ @Singh2007; @Dlotko2019].

## Motivation

The ball covers mentioned above have been proposed as an alternative to mapper [ @Dlotko2019], where they exchange complexity for computational cost, and the mapper construction itself relies on a crucial covering step that has received limited theoretical attention. Conventionally, mappers rely on covers consisting either of overlapping intervals (when the lens is one-dimensional) or of their cartesian products (higher-dimensional). For this purpose, we propose that ball covers, heuristically optimized using the maxmin procedure, have a potential advantage over conventional covers, alongside a potential disadvantage.

Conventionally, mappers use low-dimensional lens spaces  $\mathbb{R}^p$  and one of two types of cover, based on overlapping intervals either of fixed length or at fixed quantiles [ @Piekenbrock2020]. We think of this length or quantile as the *resolution* of the cover. When  $m > 1$ , covers for  $Y \subset \mathbb{R}^m$  can be obtained as the cartesian

---

<sup>2</sup>This procedure is distinct from many other uses of “maxmin” and related terms.

products of those of the coordinate projections of  $Y$ —so, if  $\pi_1, \pi_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$  are the coordinate projections, then interval covers  $\{I_\alpha\}$  and  $\{J_\beta\}$  of  $\pi_1(Y)$  and  $\pi_2(Y)$  give rise to the rectangle cover  $\{I_\alpha \times J_\beta\}$  for  $Y$ . While these cover types are manageable in very few dimensions, the number of sets scales geometrically with  $p$ , holding the resolution fixed. Moreover, eventually most of the resulting cover sets will contain no points of  $X$ , and additional calculations will be needed to restrict to the non-empty sets.

In contrast, a cover obtained by centering balls at a subset of landmark points in  $Y$  will have greater up-front computational cost but will be guaranteed to contain no empty sets, and the number of sets required to capture the topology of  $Y$  will increase only with the geometric and topological complexity of  $Y$ , not with  $p$ . (We test this hypothesis in Section .)

Nevertheless, the maxmin cover requires a meaningful distance metric: the dissimilarity of cases  $x$  and  $y$  is captured by their distance  $d(x, y)$ , regardless of where  $x$  and  $y$  are located in  $X$ , and the neighborhoods  $B_r(x)$  and  $B_r(y)$  about landmarks  $x$  and  $y$  play an equal role in the cover.<sup>3</sup> This means that cover sets centered at landmarks in sparse regions of  $X$  will be more numerous and of lower cardinality than those centered in dense regions. The assumption is violated in many real-world settings, including much of biomedicine. For example, in psychometric terms, this would mean that inter-case distance is an *interval*, not only an *ordinal*, variable, so that the distances between cases in a point cloud representation has a definite meaning independent of which cases are considered. This assumption is often made for convenience, but it generally does not follow from theory.

This motivates us to produce a counterpart to the ball cover that we call the *neighborhood cover*, each set of which may have a different radius but (nearly) the same cardinality. Especially in analyses of medical and healthcare data, underlying variables can often only be understood as ordinal. Other representations of high-dimensional data sets are commonly defined by similarity (or dissimilarity) measures such as cosine similarity rather than by coordinates and associated metrics. Furthermore, because measurements are coarse and often missing, such data often contain indistinguishable entries—cases all of whose measurements are equal and that are therefore represented as multiple instances of the same point. All of these attributes violate the assumptions of the ball cover approach and suggest the need for an ordinal counterpart.

## Procedures

In this section we review the maxmin procedure and introduce the lastfirst procedure as a complement to it.

### Maxmin procedure

As defined for the purpose of landmark selection, maxmin takes as input a proper subset  $L \subset X$  and returns as output a point  $x \in X \setminus L$ . A related procedure, which we will call minmax, takes as input only  $X$  and returns its *Chebyshev center*, the point  $x \in X$  for which  $D(x, X \setminus \{x\})$  is minimized. Whereas maxmin naturally increments a landmark set, minmax naturally initializes one: The Chebyshev center is the landmark that produces the single-set cover using the smallest radius. To help build intuition around these concepts, consider them together with two other procedures: Each procedure calculates either minmax or maxmin, and each does so either with reference to a proper subset of  $X$  or with respect to  $X$  itself.

Given  $L \subset X$  and  $x \in X \setminus L$ , define the *minmax sets*

$$\begin{aligned} \text{minmax}(L) &= \{x \in X \setminus L \mid D(x, L) = \min_{y \in X \setminus L} D(y, L)\} \\ \text{minmax}(X) &= \{x \in X \mid D(x, X \setminus \{x\}) = \min_{y \in X} D(y, X \setminus \{y\})\} \end{aligned}$$

consisting of *minmax points* and the *maxmin sets*

$$\begin{aligned} \text{maxmin}(L) &= \{x \in X \setminus L \mid d(x, L) = \max_{y \in X \setminus L} d(y, L)\} \\ \text{maxmin}(X) &= \{x \in X \mid d(x, X \setminus \{x\}) = \max_{y \in X} d(y, X \setminus \{y\})\} \end{aligned}$$

---

<sup>3</sup>Is there a term for this property, e.g. something being a “universal constant”?

consisting of *maxmin points*. Note that both  $\text{minmax}(\cdot)$  and  $\text{maxmin}(\cdot)$  are nonempty and that, when  $X$  is in locally general position, they both have cardinality 1.

---

**Algorithm 1** Select a maxmin landmark set.

---

**Require:** finite metric space  $(X, d)$

**Require:** at least one parameter  $\varepsilon \geq 0$  or  $n \in \mathbb{N}$

**Require:** seed point  $\ell_0 \in X$

**if** only  $\varepsilon$  is given **then**

$n \leftarrow 1$

**end if**

**if** only  $n$  is given **then**

$\varepsilon \leftarrow \infty$

**end if**

$L \leftarrow \emptyset$

$i \leftarrow 0$

**repeat**

$L \leftarrow L \cup \{\ell_i\}$

$i \leftarrow i + 1$

$\ell_i \in \text{maxmin}(L)$

$d_{\max} \leftarrow d(\ell_i, L)$

**until**  $d_{\max} < \varepsilon$  and  $|L| \geq n$

**return** maxmin landmark set  $L$

---

The *maxmin procedure* for generating a landmark set  $L \subseteq X$  proceeds as follows (see Algorithm 1): First, choose a number  $k \leq |[X]|$  of landmark points to generate or a radius  $\varepsilon \geq 0$  for which to require that the balls  $\overline{B}_\varepsilon(\ell)$  minimally cover  $X$ . Choose a first landmark point  $\ell_0 \in X$ . This choice may be arbitrary; we specifically consider three selection rules: the first point index in the object representing  $X$ , selection at random, and (random selection from)  $\text{minmax}(X)$ . Inductively over  $i \in \mathbb{N}$ , if ever  $i \geq k$  or  $d(L, X \setminus L) \leq \varepsilon$ , then stop. Otherwise, when  $L = \{\ell_0, \dots, \ell_{i-1}\}$ , choose  $\ell_i \in \text{maxmin}(L)$ , again according to a preferred rule. If a fixed number  $k$  of landmarks was prescribed, then set  $\varepsilon = d(L, X \setminus L)$ ; if  $\varepsilon$  was prescribed, then set  $k = |L|$ .

We will write the elements of landmark sets  $L = \{\ell_0, \dots, \ell_{k-1}\}$  in the order in which they were generated. Note that, if  $k = |[X]|$  or  $\varepsilon = 0$ , then  $L = [X]$ . When the procedure stops,  $X = \bigcup_{i=0}^{k-1} \overline{B}_\varepsilon(\ell_i)$ , and this cover is minimal in two senses: The removal of any  $\overline{B}_\varepsilon(\ell_i)$  or any decrease in  $\varepsilon$  will obtain a collection of sets that fail to cover  $X$ . More generally, a non-minimal cover can be obtained by specifying both  $k$  and  $\varepsilon$  in a compatible way. In Section , we describe two adaptive parameters implemented in our software package that make these choices easier.

## Lastfirst procedure

The lastfirst procedure is defined analogously to the maxmin procedure, substituting “rank-distance” for the pseudometric  $d_X$ .

## Rank-distances

Rank-distance is an adaptive notion of distance with respect to nearest neighborhoods.<sup>4</sup> It relies on the underlying pseudometric  $d_X$  but takes values in  $\mathbb{N}$  given by the ordinal of one point’s distance from another.

**Definition 1.** (Rank-distance) For  $x, y \in X$ , define the *rank-distance*  $q_X : X \times X \rightarrow \mathbb{N}$  as follows:

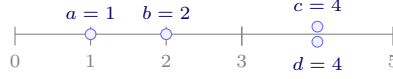
$$q_X(x, y) = |\{z \in X \mid d_X(x, z) < d_X(x, y)\}| + 1$$

---

<sup>4</sup>As used here, rank-distance is distinct from the *rank-distance* between permutations, which is used to define rank correlation coefficients, and from the *ordinal distance* proposed by Pattanaik and Xu (2008), a loosening of the concept of pseudometric that dispenses with the triangle inequality.

As with  $d$ , we allow ourselves to write  $q = q_X$  when clear from context. Note that  $q(x, x) = 1$  and  $q(x, y) \leq N$ , and that  $q$  is not, in general, symmetric. (It is therefore not a pseudometric.)<sup>5</sup>

**Example 2.** Consider the simple case  $X = \{a, b, c, d\}$ , visualized below, equipped with the standard Euclidean metric:



Lack of symmetry of  $q$  is shown by points  $b$  and  $c$ :

$$\begin{aligned} q(a, c) &= |\{x \in X \mid |x - a| < |c - a| = 2\}| + 1 & q(c, a) &= |\{x \in X \mid |x - c| < |a - c| = 2\}| + 1 \\ &= |\{a, b\}| + 1 & &= |\{b, c, d\}| + 1 \\ &= 3 & &= 4 \end{aligned}$$

Observe in particular that  $\max_{x \in X} q(a, x) = 3 < |X|$ ;  $q(x, \cdot)$  will not max out at  $N$  when the most distant points from  $x$  have multiplicity.

However,  $q(x, x) = 1$  whether  $x$  has multiplicity or not:

$$\begin{aligned} q(a, a) &= |\{x \in X \mid |x - a| < |a - a| = 0\}| + 1 & q(c, c) &= |\{x \in X \mid |x - c| < |c - c| = 0\}| + 1 \\ &= |\emptyset| + 1 & &= |\emptyset| + 1 \\ &= 1 & &= 1 \end{aligned}$$

We term the unary rankings  $q(x, \cdot)$  and  $q(\cdot, x)$  the *out-* (from  $x$ ) and *in-* (to  $x$ ) *rankings* of  $X$ , respectively. These can then be used to define *out-* and *in-neighborhoods* of  $x$ .<sup>6</sup>

**Definition 3.** ( $k$ -neighborhoods) For  $x \in X$ , define the  $k$ -out-neighborhoods  $N_k^+$  and  $k$ -in-neighborhoods  $N_k^-$  of  $x$  as the sets

$$\begin{aligned} N_k^+(x) &= \{y \in X \mid q(x, y) \leq k\} \\ N_k^-(x) &= \{y \in X \mid q(y, x) \leq k\} \end{aligned}$$

Note that  $\emptyset \subseteq N_1^\pm(x) \subseteq \dots \subseteq N_N^\pm(x) = X$ . The  $k$ -out-neighborhoods of  $x$  are the sets of points in  $X$  that have rank-distance at most  $k$  from  $x$ . This is equivalent to the  $k$ -nearest neighbors of  $x$ . The  $k$ -in-neighborhoods of  $x$  are the sets of points in  $X$  from which  $x$  has rank-distance at most  $k$ . These definitions can be adapted as follows to be relative to a subset  $Y \subseteq X$ , using the notation  $q_X$  to emphasize that the points in  $X \setminus (Y \cup \{x\})$  are still involved in the calculation:

$$\begin{aligned} N_k^+(x, Y) &= \{y \in Y \mid q_X(x, y) \leq k\} \\ N_k^-(x, Y) &= \{y \in Y \mid q_X(y, x) \leq k\} \end{aligned}$$

**Example 4.** Consider the same  $X$  as in Example 2. Compute  $N_k^+$  and  $N_k^-$  for  $a$  and  $c$ , using  $k = 3$ :

$$\begin{aligned} N_3^+(a) &= \{x \in X \mid q(a, x) \leq 3\} & N_3^+(c) &= \{x \in X \mid q(c, x) \leq 3\} \\ &= \{a, b, c, d\} & &= \{b, c, d\} \\ N_3^-(a) &= \{x \in X \mid q(x, a) \leq 3\} & N_3^-(c) &= \{x \in X \mid q(x, c) \leq 3\} \\ &= \{a, b\} & &= \{a, b, c, d\} \end{aligned}$$

<sup>5</sup>These images might be made larger, with less or no transparency.

<sup>6</sup>The terminology and notation are adapted from the theory of directed graphs. These definitions are the same as those for a complete directed graph on  $X$  with directed arcs  $x \rightarrow y$  weighted by rank-distance  $q(x, y)$ .

Rank-distances are not as straightforward to compare among subsets of points. For example, for  $y \neq x$ ,  $q(x, y)$  takes integer values between  $\overline{B}_0(x) + 1$  and  $N$ . To explain and motivate their use, we now intuitively adapt the minmax and maxmin procedures to this setting, then state and prove a formal definition for their analogs.

### Rank sequences and landmark selection

Consider the case of choosing the next landmark point given a subset  $L = \{\ell_0, \dots, \ell_{i-1}\} \subset X$  of collected landmark points. For a ball cover, we choose  $\ell_i \in X \setminus L$  so that the *minimum radius*  $\varepsilon$  required for some  $B_\varepsilon(\ell_j)$  to contain  $\ell_i$  is *maximized*. If  $X$  is in general position, the choice of  $\ell_i$  will be unique; otherwise,  $\ell_i$  can be chosen at random from the subset of  $X$  that satisfies this criterion. Analogously, for a neighborhood cover, we want that the *minimum cardinality*  $k$  required for some  $N_k^+(\ell_j)$  to contain  $\ell_i$  is *maximized*. Switching perspective from out- to in-, reversing the roles of  $L$  and  $\ell_i$ , we want  $N_k^-(\ell_i, L) = 1$  for the latest (largest)  $k$  possible, say  $k_1^-$ . When indistinguishable points abound, this may still not uniquely determine  $\ell_i$ , so we may extend the principle: Among those  $\ell \in X \setminus L$  for which  $N_{k_1^-}^-(\ell_i, L) = 1$ , choose  $\ell_i$  for which  $N_k^-(\ell_i, L) = 2$  for the latest  $k$  possible, say  $k_2^- \geq k_1^-$ . Continue this process until only one candidate  $\ell$  remains (up to multiplicity), or until  $N_k^-(\ell, L) = |L|$ , in which case we consider all remaining candidates equivalent.

Now consider the case of choosing an initial landmark point with respect to a subset  $L \subset X$ . For a ball cover, we above considered the Chebyshev center  $\ell_0 \in X$  so that the *maximum radius*  $\varepsilon$  required for  $\overline{B}_\varepsilon(\ell_0)$  to contain any  $\ell \in L$  is *minimized*. This means that a one-set ball cover of  $L$  centered at  $\ell_0$  would have minimum radius. Analogously, for a neighborhood cover, we would want the *maximum cardinality*  $k$  required for  $N_k^+(\ell_0)$  to contain any  $\ell \in L$  to be *minimized*. That is, we would want  $N_k^+(\ell_0, L) = |L|$  for the earliest (smallest)  $k$  possible, say  $k_{|L|}^+$ . To decide among several points  $\ell$  with this property, we would then choose one for which  $N_k^+(\ell, X) = |L| - 1$  for the earliest  $k$  possible, say  $k_{|L|-1}^+ \leq k_{|L|}^+$ . This process would likewise continue until only one candidate  $\ell$  remains, or until  $N_k^+(\ell, L) = 0$  and all remaining candidates are considered equivalent.

We formalize these two procedures, and companion procedures without reference to  $L$ , by defining two sequences of neighborhood sizes that encode the optimized cardinalities  $k_i$  and imposing on them two total orders that encode the preferences between them.

**Definition 5.** (Rank sequences) For  $x \in X$  and  $L \subset X$ , define the *out-rank* ( $Q^+$ ) and *in-rank* ( $Q^-$ ) sequences of  $k$ -neighborhood cardinalities:

$$Q^\pm(x) = (|N_k^\pm(x)|)_{k=1}^N \quad Q^\pm(x, L) = (|N_k^\pm(x, L)|)_{k=1}^{|L|}$$

**Proposition 6.** Taking the  $k_i^\pm$  as defined above,

$$Q^+(x, L) = (k_1^+ - 1, k_2^+ - 1, \dots, k_{|L|}^+ - 1, |L|) \\ Q^-(x, L) = (1^{k_1^-}, 2^{k_2^- - k_1^-}, \dots, (|L| - 1)^{k_{|L|-1}^- - k_{|L|-2}^-}, |L|^{L - k_{|L|-1}^-})$$

**Lemma 7.** If  $Q = Q^+(x, L)$ , then  $Q_i \geq i$ .

*Proof.* For convenience, assume that  $d(x, \ell_0) \leq d(x, \ell_1) \leq \dots \leq d(x, \ell_{k-1})$ , and consider  $i > 1$ . If  $d(x, \ell) \leq d(x, \ell_{i-1})$ , then  $q(x, \ell) = |\{\ell' \in L \mid d(x, \ell') < d(x, \ell)\}| + 1 \leq |\{\ell' \in L \mid d(x, \ell') < d(x, \ell_{i-1})\}| + 1 \leq |\{\ell_0, \dots, \ell_{i-2}\}| + 1 = i$ ; whereas, if  $d(x, \ell) > d(x, \ell_{i-1})$ , then  $q(x, \ell) = |\{\ell' \in L \mid d(x, \ell') < d(x, \ell)\}| + 1 > |\{\ell' \in L \mid d(x, \ell') < d(x, \ell)\} \setminus \{\ell_{i-1}\}| + 1 \geq |\{\ell' \in L \mid d(x, \ell') < d(x, \ell_{i-1})\}| + 1 = |\{\ell_0, \dots, \ell_{i-2}\}| + 1 = i$ . (The reader may check that  $q(x, \ell) \leq 1$  if and only if  $d(x, \ell) \leq d(x, \ell_0)$ .) In the first instance, then,  $Q_i = |N_i^+(x, L)| = |\{\ell \in L \mid q(x, \ell) \leq i\}| = |\{\ell \in L \mid d(x, \ell) \leq d(x, \ell_{i-1})\}| \geq |\{\ell_0, \dots, \ell_{i-1}\}| = i$ .  $\square$

**Corollary 8.** Given  $L = \{\ell_0, \dots, \ell_{k-1}\}$ , the maximum possible out-rank sequence  $Q^+(x, L)$  is  $Q = (1, 2, \dots, k)$ .

An example of this computation is shown in Example 9.

**Example 9.** Continuing Example 4, we can compute the other  $N_k^+$  and  $N_k^-$  to obtain  $Q^+$  and  $Q^-$  for  $a$  and  $c$ :

$$\begin{aligned}
Q^+(a) &= (|N_1^+(a)|, |N_2^+(a)|, |N_3^+(a)|, |N_4^+(a)|) & Q^+(c) &= (|N_1^+(c)|, |N_2^+(c)|, |N_3^+(c)|, |N_4^+(c)|) \\
&= (|\{a\}|, |\{a, b\}|, |\{a, b, c, d\}|, |\{a, b, c, d\}|) & &= (|\{c, d\}|, |\{c, d\}|, |\{b, c, d\}|, |\{a, b, c, d\}|) \\
&= (1, 2, 4, 4) & &= (2, 2, 3, 4)
\end{aligned}$$

$$\begin{aligned}
Q^-(a) &= (|N_1^-(a)|, |N_2^-(a)|, |N_3^-(a)|, |N_4^-(a)|) & Q^-(c) &= (|N_1^-(c)|, |N_2^-(c)|, |N_3^-(c)|, |N_4^-(c)|) \\
&= (|\{a\}|, |\{a, b\}|, |\{a, b\}|, |\{a, b, c, d\}|) & &= (|\{c, d\}|, |\{c, d\}|, |\{a, b, c, d\}|, |\{a, b, c, d\}|) \\
&= (1, 2, 2, 4) & &= (2, 2, 4, 4)
\end{aligned}$$

**Definition 10.** (Total orders on rank sequences) Let  $a_n = (a_1, \dots, a_M), b_n = (b_1, \dots, b_M) \in \mathbb{N}^M$ . Then

- $a_n < b_n$  in the reverse colexicographic (revcolex) order if  $\exists i, a_i > b_i \wedge (\forall j > i, a_j = b_j)$ .
- $a_n < b_n$  in the reverse lexicographic (revlex) order if  $\exists i, a_i > b_i \wedge (\forall j < i, a_j = b_j)$ .

Impose the revcolex order on the  $Q^+$  to emphasize later out-distances. Impose the revlex order on the  $Q^-$  to emphasize earlier in-distances. In both cases, sequences with more large values indicate points with lower rank-distances to or from more other points, which is why both orders are reversed.

We now define counterparts to the minmax and maxmin procedures using these totally ordered sequences.

**Definition 11.** (Firstlast and lastfirst) Given  $L \subset X$ , define the *firstlast sets*

$$\begin{aligned}
\text{fl}(L) &= \{x \in X \setminus L \mid Q^+(x, L) = \min_{y \in X \setminus L} Q^+(y, L)\} \\
\text{fl}(X) &= \{x \in X \mid Q^+(x, X \setminus \{x\}) = \min_{y \in X} Q^+(y, X \setminus \{y\})\}
\end{aligned}$$

consisting of *firstlast points* and the *lastfirst sets*

$$\begin{aligned}
\text{lf}(L) &= \{x \in X \setminus L \mid Q^-(x, L) = \max_{y \in X \setminus L} Q^-(y, L)\} \\
\text{lf}(X) &= \{x \in X \mid Q^-(x, X \setminus \{x\}) = \max_{y \in X} Q^-(y, X \setminus \{y\})\}
\end{aligned}$$

consisting of *lastfirst points*.

**Remark 12.** Replacing each  $Q^\pm(x, X \setminus \{x\})$  with  $Q^\pm(x)$  would simply increase the integer in each position by 1, so the latter could be used equivalently to the former.

**Example 13.** Return again to  $X = \{a, b, c, d\}$  from Example 2. We calculate an exhaustive lastfirst landmark set, seeded with a firstlast point:

1. We have

$$\begin{aligned}
Q^+(a, \{b, c, d\}) &= (0, 1, 3, 3) & Q^+(b, \{a, c, d\}) &= (0, 1, 3, 3) \\
Q^+(c, \{a, b, d\}) &= (1, 1, 2, 3) & Q^+(d, \{a, b, c\}) &= (1, 1, 2, 3)
\end{aligned}$$

Under the revcolex order,  $\min_{x \in X} Q^+(x, X \setminus \{x\}) = (0, 1, 3, 3)$ , and we randomly select  $\ell_0 = a$ .

2. We now have

$$Q^-(b, \{a\}) = (0, 1, 1, 1) \quad Q^-(c, \{a\}) = (0, 0, 1, 1) \quad Q^-(d, \{a\}) = (0, 0, 1, 1)$$

Under the revlex order,  $\max_{x \in X \setminus \{a\}} Q^-(x, \{a\}) = (0, 0, 1, 1)$ , and without loss of generality, we select  $\ell_1 = c$ , so that now  $L = \{a, c\}$ .

3. Only one point remains in  $X \setminus L = \{b\}$ , so the exhaustive landmark set is  $\{a, c, b\}$ .



## Algorithms

For illustration, we provide a complete algorithm to identify the firstlast set of  $X$  and a simplified algorithm to obtain a lastfirst landmark set using specified parameters. The former requires first calculating  $Q^+(x, L)$  from  $x$  and  $L$ , which is done in Algorithm 2. This algorithm uses the notation  $[a, b]$  as shorthand for the arithmetic sequence  $(a, a + 1, \dots, b)$  of indices. Making use of these out-rank sequences, the recovery of a firstlast set proceeds as in Algorithm 3.

---

**Algorithm 2** Compute the out-rank sequence of a point with respect to a proper subset.

---

**Require:** finite pseudometric space  $(X, d)$   
**Require:** landmark set  $L = \{\ell_0, \dots, \ell_{k-1}\} \subset X$   
**Require:** point  $x \in X$   
 $D \leftarrow \text{sort}(d(x, \ell_0), \dots, d(x, \ell_{k-1})) = (d_1 \leq \dots \leq d_k)$   
 $\tilde{D} \leftarrow (\tilde{d}_0 = -\infty, \tilde{d}_1 = d_1, \dots, \tilde{d}_k = d_k, \tilde{d}_{k+1} = +\infty)$   
 $\tilde{Q} \leftarrow (0, \dots, 0) \in \mathbb{N}^{[0, k+1]}$   
 $j \leftarrow 0$   
**for**  $i = 1$  **to**  $k + 1$  **do**  
  **if**  $\tilde{d}_i > \tilde{d}_{i-1}$  **then**  
    **while**  $j < i$  **do**  
       $j \leftarrow j + 1$   
       $\tilde{Q}_j \leftarrow \tilde{Q}_{j-1}$   
    **end while**  
  **end if**  
   $\tilde{Q}_j \leftarrow \tilde{Q}_j + 1$   
**end for**  
 $Q \leftarrow \tilde{Q}_{[1, k]}$   
**return** out-rank sequence  $Q$

---

**Lemma 14.** *Algorithm 2 returns the out-rank sequence.*

*Proof.* The algorithm takes  $k + 1$  steps, which we consider in three groups. At any step  $i$ ,  $j$  never decreases and increases at most to  $j = i$ , so the value of  $\tilde{Q}_{j'}$  is fixed once  $j > j'$ .

$i = 1$ . In this step,  $\tilde{d}_i = \tilde{d}_1 \geq 0 > -\infty = \tilde{d}_0 = \tilde{d}_{i-1}$ , so the **while** loop is run. The loop begins with  $j = 0$  and ends with  $j = 1$ , so it has only the vacuous effect  $\tilde{Q}_1 \leftarrow \tilde{Q}_0$  on  $\tilde{Q}$ . The only effect in this case is the final increment  $\tilde{Q}_1 \leftarrow \tilde{Q}_1 + 1$ , so that the step ends with  $\tilde{Q} = (0, 1, 0, \dots, 0)$ .

$1 < i \leq k$ . The intermediate steps proceed in two ways, depending on whether  $d_i = \tilde{d}_i > \tilde{d}_{i-1} = d_{i-1}$ —that is, whether the next-closest point in  $L$  is farther from  $x$  than the previous. For convenience, let  $i' \leq i - 1$  denote the previous value of  $i$  at which the distance from  $x$  increased.

$d_i = d_{i-1}$ . In this case, the only change to  $\tilde{Q}$  is to increase  $\tilde{Q}_{i'}$  by 1. Thus, at the end of the step,  $\tilde{Q}$  equals the number of members of  $L$  up to index  $i$  at distance at most  $d_{i'} = d_i$  from  $x$ .

$d_i > d_{i-1}$ . In this case,  $j$  starts out at  $i'$ . If  $i' < i - 1$ , then the **while** loop copies the value  $\tilde{Q}_{i'}$  to the values of  $\tilde{Q}_{[i'+1, i-1]}$ . By the same reasoning as in the previous case plus the fact that  $d_i > d_{i-1}$ ,  $\tilde{Q}_{i'}$  at this step equals  $|N_{i'}^+(x, L)| = |N_{i-1}^+(x, L)|$ . Therefore, at the end of this step,  $\tilde{Q}_{[i', i-1]} = (Q^+(x, L))_{[i', i-1]}$  and  $j > i - 1$ , so that the returned values  $Q_{[i', i-1]}$  will be correct. Regardless, this value is also copied to  $\tilde{Q}_i$ , and  $\tilde{Q}_i$  is incremented by 1, so that  $\tilde{Q}_i = \tilde{Q}_{i-1} + 1$ .

Regardless of the case, at each step  $\max \tilde{Q}$  increases by 1, so that after step  $i = k$  we have  $\max \tilde{Q} = k$ . The last step wraps up the process in the case that  $d_k = d_{k-1}$ .

$i = k + 1$ . In this step,  $\tilde{d}_i = \tilde{d}_{k+1} = +\infty > d_k = \tilde{d}_{i-1}$ , so the **while** loop is run. If  $d_k > d_{k-1}$ , then this has no effect on  $Q = \tilde{Q}_{[1, k]}$ . If, instead,  $d_k = d_{k-1}$ , then the effect is to copy  $\tilde{Q}_{i'}$  to  $\tilde{Q}_{[i'+1, k]}$ . As in the previous step, this makes  $\tilde{Q}_{[i'+1, k]} = (Q^+(x, L))_{[i'+1, k]}$ , which ensures that the returned  $Q_{[i', k]}$  are correct.

□

---

**Algorithm 3** Identify a firstlast set with respect to a proper subset.

---

**Require:** finite metric space  $(X, d)$ **Require:** subset  $L = \{\ell_0, \dots, \ell_{k-1}\} \subset X$  $F \leftarrow \emptyset$  $Q \leftarrow (k, \dots, k) \in \mathbb{N}^k$ **for**  $x \in X$  **do** $Q' \leftarrow Q^+(x, L) \in \mathbb{N}^k$  (Algorithm 2) $j \leftarrow k$ **while**  $Q'_j = Q_j$  **do** $j \leftarrow j - 1$ **end while****if**  $j = 0$  **then** $F \leftarrow F \cup \{x\}$ **else****if**  $Q'_j < Q_j$  **then** $F \leftarrow \{x\}$  $Q \leftarrow Q'$ **end if****end if****end for****return** firstlast set  $F$ 


---

**Proposition 15.** *Algorithm 3 returns the firstlast set.*

*Proof.* In the algorithm,  $Q$  begins at what we know from Corollary 8 to be the maximum possible out-rank sequence. The algorithm then simply computes the out-rank sequence of every point  $x \in X$  and either stores it in a list, if it is equal to the current minimum, or resets and replaces the stored list, if it is smaller. Once all points have been considered, those in the stored list will constitute the firstlast set. □

The construction of a lastfirst landmark set proceeds as follows, and as outlined in Algorithm 4. Let  $L$  denote the set of landmark points. At the start of the algorithm,  $L = \emptyset$ . An initial point  $\ell_0$  is selected and added to the landmark set so that  $L = \{\ell_0\}$ . This point seeds the procedure. At each iteration,  $\text{lf}(L)$  is computed and the next landmark point  $\ell_i$  is selected from this set. The procedure terminates when  $X = \bigcup_{j \leq i} N_1^+(\ell_j)$ , hence  $[L] = [X]$ .

The implementation of  $\text{lf}$  is more involved: Because an in-rank sequence  $Q^-(x)$  cannot be calculated from the distances  $\{d(x, y) \mid y \in X\}$  alone, much more computation is required to generate  $Q^-(x, L)$  at each step. Our (partially vectorized) R implementation uses some combinatorial identities to expedite this calculation.

**Proposition 16.** *The landmark set  $L$  returned by Algorithm 4 gives the following properties:*

1. If  $k$  is given, the  $k$ -th term in the  $Q^-(x, L)$  sequence, denoted  $Q_k^-(x, L)$ , is at most  $k$ . Equivalently,

$$\max_{x \in X \setminus L} Q_k^-(x, L) := \max_{x \in X \setminus L} |N_k^-(x, L)| \leq k$$

2. If  $n$  is given and  $k$  is not,  $|L| = n$ . If  $n$  and  $k$  are both given,  $|L| \geq n$ . Otherwise  $L$  is minimal, meaning it has the smallest cardinality required for  $k$ -neighborhoods of the points in  $L$  to cover the space  $X$ .

*Proof.* Let  $(X, d)$  be a finite metric space and  $\ell_0 \in X$  be a seed point as required by Algorithm 4.

Recall that for the algorithm to terminate its loop and subsequently return the resulting landmark set  $L$ , both of the following exit conditions must hold: (1)  $q_{\max} \leq k$  and (2)  $|L| \geq n$ .

---

**Algorithm 4** Select a lastfirst landmark set.

---

**Require:** finite metric space  $(X, d)$

**Require:** at least one parameter  $k > 0$  or  $n \in \mathbb{N}$

**Require:** seed point  $\ell_0 \in X$

**Require:** selection procedure **pick**

**if** only  $k$  is given **then**

$n \leftarrow 0$

**end if**

**if** only  $n$  is given **then**

$k \leftarrow |X|$

**end if**

$L \leftarrow \emptyset$

$i \leftarrow 0$

**repeat**

$L \leftarrow L \cup \{\ell_i\}$

$i \leftarrow i + 1$

$F \leftarrow \text{lf}(L)$

$\ell_i \leftarrow \text{pick}(F)$

$q_{\max} \leftarrow q(\ell_i, L)$

**until**  $q_{\max} < k$  and  $|L| \geq n$

**return** lastfirst landmark set  $L$

---

1. Suppose  $k$  is given. By (1),  $q_{\max} := \min_{\ell_j \in L} q(\ell_i, \ell_j) \leq k$ , where  $\ell_i \in \text{lf}(L)$ .

**No longer sure this part is true.**

2. Suppose  $n$  is given. Regardless of whether  $k$  is provided, (2) must always hold for the algorithm to terminate, so  $|L| \geq n$  for any  $k$ . If  $k$  is not given, it is set to  $|X|$  before the loop, which means (1) holds from the first iteration onward since  $|X|$  is by definition the maximum value  $q$  can attain. Then the algorithm terminates as soon as condition (2) is first met, which is when  $|L| = n$ .<sup>7</sup>

Now suppose  $n$  is not given. Then  $k$  must be given since the algorithm requires at least one of the two parameters  $n, k$  to be provided. Therefore,  $n$  is set to 0 before the loop, meaning (2)  $|L| \geq n = 0$  always holds, so the algorithm returns  $L$  as soon as (1) is satisfied, i.e. when  $q_{\max} := q(\ell_i, L) \leq k$  for any  $\ell_i \in \text{lf}(L)$ . This means the point  $\ell_i \in \text{lf}(L)$  that is farthest from any  $\ell_j \in L$  by  $q$  is still within a  $k$ -neighborhood of some landmark  $\ell_j$ . Since this occurs and causes the loop to exit as soon as the space  $X$  can be covered by  $k$ -neighborhoods of points in  $L$ ,  $|L|$  is as small as possible.

□

## Tie handling

We might have defined two *rank-distances*  $\check{q}, \hat{q} : X \times X \rightarrow \mathbb{N}$  (“ $q$ -check” and “ $q$ -hat”) as follows:

$$\check{q}(x, y) = |\{z \in X \mid d(x, z) < d(x, y)\}| + 1$$

$$\hat{q}(x, y) = |\{z \in X \mid d(x, z) \leq d(x, y)\}|$$

In this notation,  $\check{q} = q$ , while  $\hat{q}(x, y)$  is the cardinality of the smallest ball centered at  $x$  that contains  $y$ . Letting  $\dot{q}$  stand in for either  $\check{q}$  or  $\hat{q}$ , we can define  $\dot{N}_k^\pm(x)$  and  $\dot{Q}^\pm(x)$  as before and arrive at corresponding notions of firstlast and lastfirst sets.

Note that then  $\check{N}_1^\pm(x) \subseteq \{x\} \subseteq \hat{N}_1^\pm(x)$ , and that  $\hat{q}(x, x) > 1$  when  $x$  has multiplicity. These two rank-distances derive from two tie-handling schemes for calculating rankings of lists with duplicates: For example, if

---

<sup>7</sup>Note that  $|L| > n$  if and only if  $q_{\max} \leq k$  is not satisfied when  $|L| = n$ , meaning  $X$  would not be covered by  $k$ -neighborhoods around  $n$  landmark points, so more landmarks must be chosen to guarantee the algorithm produces a valid cover by neighborhoods of size  $k$ .

$a < b = c < d$  are the distances from  $x$  to  $y_1, y_2, y_3, y_4$ , respectively, then  $(\check{q}(x, y_1), \check{q}(x, y_2), \check{q}(x, y_3), \check{q}(x, y_4)) = (1, 2, 2, 4)$  and  $(\hat{q}(x, y_1), \hat{q}(x, y_2), \hat{q}(x, y_3), \hat{q}(x, y_4)) = (1, 3, 3, 4)$ .

Conceptually, these tools would produce landmark sets that yield neighborhood covers with smaller, rather than larger, neighborhoods in regions of high multiplicity. While we do not use these ideas in this study, they may be suitable in some settings or for some purposes, for example when high multiplicity indicates a failure to distinguish similar but distinct cases that the analyst wishes to maintain greater separation between. They may also play a role in stability analyses, e.g. by producing an interleaving sequence of nerves of covers.

**Example 17.** Recall  $X = \{a, b, c, d\}$  from Example 2. The rank-distance  $\hat{q}$  is also asymmetric:

$$\begin{aligned} \hat{q}(b, c) &= |\{x \in X \mid |x - b| \leq |c - b| = 2\}| & \hat{q}(c, b) &= |\{x \in X \mid |x - c| \leq |b - c| = 2\}| \\ &= |\{a, b, c, d\}| & &= |\{b, c, d\}| \\ &= 4 & &= 3 \end{aligned}$$

Observe that  $\hat{q}(x, x) = 1$  only for distinguishable points  $x \in X$ :

$$\begin{aligned} \hat{q}(a, a) &= |\{x \in X \mid |x - a| \leq |a - a| = 0\}| & \hat{q}(c, c) &= |\{x \in X \mid |x - c| \leq |c - c| = 0\}| \\ &= |\{a\}| & &= |\{c, d\}| \\ &= 1 & &= 2 \end{aligned}$$

Finally, observe that  $\hat{q}(x, \cdot)$  always maxes out at  $|X|$ :  $\hat{q}(a, c) = \hat{q}(b, c) = \hat{q}(c, a) = \hat{q}(d, a) = |X|$ .

Continuing on as in Example 4, we can compute  $\hat{N}_2^+$  and  $\hat{N}_2^-$  for  $b$  and  $c$ :

$$\begin{aligned} \hat{N}_2^+(b) &= \{x \in X \mid \hat{q}(b, x) \leq 2\} & \hat{N}_2^+(c) &= \{x \in X \mid \hat{q}(c, x) \leq 2\} \\ &= \{a, b\} & &= \{c, d\} \\ \hat{N}_2^-(b) &= \{x \in X \mid \hat{q}(x, b) \leq 2\} & \hat{N}_2^-(c) &= \{x \in X \mid \hat{q}(x, c) \leq 2\} \\ &= \{a, b\} & &= \{c, d\} \end{aligned}$$

Similarly, we can compute the other  $\hat{N}_k^+$  and  $\hat{N}_k^-$  to obtain  $\hat{Q}^+$  and  $\hat{Q}^-$  for  $b$  and  $c$ :

$$\begin{aligned} \hat{Q}^+(b) &= (|\hat{N}_1^+(b)|, |\hat{N}_2^+(b)|, |\hat{N}_3^+(b)|, |\hat{N}_4^+(b)|) & \hat{Q}^+(c) &= (|\hat{N}_1^+(c)|, |\hat{N}_2^+(c)|, |\hat{N}_3^+(c)|, |\hat{N}_4^+(c)|) \\ &= (|\{b\}|, |\{a, b\}|, |\{a, b\}|, |\{a, b, c, d\}|) & &= (|\emptyset|, |\{c, d\}|, |\{b, c, d\}|, |\{a, b, c, d\}|) \\ &= (1, 2, 2, 4) & &= (0, 2, 3, 4) \\ \hat{Q}^-(b) &= (|\hat{N}_1^-(b)|, |\hat{N}_2^-(b)|, |\hat{N}_3^-(b)|, |\hat{N}_4^-(b)|) & \hat{Q}^-(c) &= (|\hat{N}_1^-(c)|, |\hat{N}_2^-(c)|, |\hat{N}_3^-(c)|, |\hat{N}_4^-(c)|) \\ &= (|\{b\}|, |\{a, b\}|, |\{a, b, c, d\}|, |\{a, b, c, d\}|) & &= (|\emptyset|, |\{c, d\}|, |\{c, d\}|, |\{a, b, c, d\}|) \\ &= (1, 2, 4, 4) & &= (0, 2, 2, 4) \end{aligned}$$

## Implementation

We have implemented the firstlast and lastfirst procedures, together with minmax and maxmin, in the R package landmark [Brunson2021a]. Each procedure is implemented in C++ using Rcpp [Eddelbuettel2011] and separately in R. For rank-distance-based procedures, the user can choose between  $\check{q}$  and  $\hat{q}$ . The landmark-generating procedures return the indices of the selected landmarks, optionally together with the sets of indices of the points in the cover set (ball or neighborhood) centered at each landmark. In addition to the number of landmarks  $m$  and either the radius  $\varepsilon$  of the balls or the cardinality  $k$  of the neighborhoods, the user may also specify additive and multiplicative extension factors for  $m$  and for  $\varepsilon$  or  $k$ . These will produce additional landmarks ( $m$ ) and larger cover sets ( $\varepsilon$  or  $k$ ) with increased overlaps, in order (for example) to construct more robust nerve complexes.

## Validation

We validated the firstlast and lastfirst procedures against several small example data sets, including that of Example 2. We also implemented each of the maxmin and lastfirst procedures, using C++ (for Euclidean distances only) and R (which uses the proxy package [Meyer2021] to calculate distances other than Euclidean), and validated these against each other on several larger data sets, including as part of the benchmark tests reported in the next section. We invite readers to install the package and experiment with new cases, as well as to request or write any desired additional features.

## Benchmark tests

We benchmarked the C++ and R implementations on three data sets: uniform samples from the unit circle  $S^1 \subset \mathbb{R}^2$  convoluted with Gaussian noise, samples with duplication from the integer lattice  $[0, 23] \times [0, 11]$  using the probability mass function  $p(a, b) \propto 2^{-ab}$ , and patients recorded at each critical care unit in MIMIC-III using the RT-similarity measure (Section ). We conducted benchmarks using the bench package [Hester2020] on the University of Florida high-performance cluster HiPerGator.

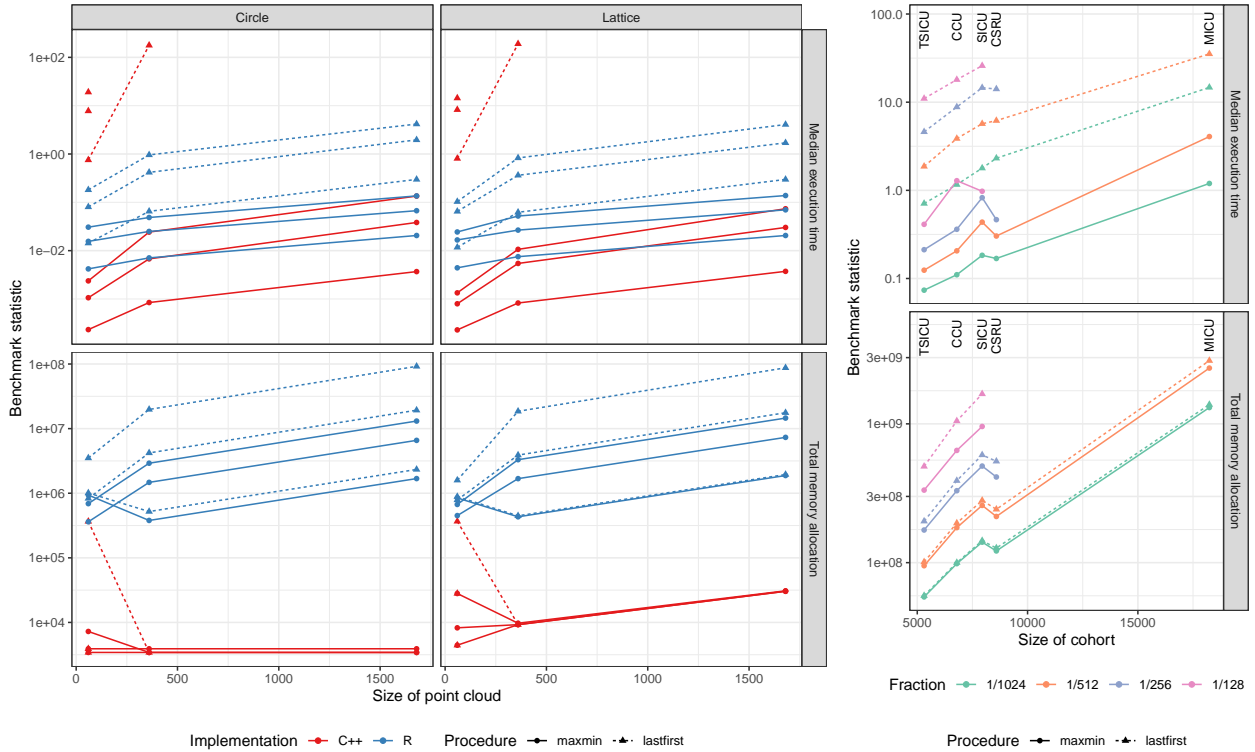


Figure 1. Benchmark results for computing landmarks on two families of artificial data (circle and lattice) and one collection of empirical data (RT-similarity space of critical care units in MIMIC-III). Some points are missing because benchmark tests did not complete within 1 hour.

Benchmark results are reported in Figure 1. R implementations used orders of magnitude more memory and took slightly longer. They appeared to scale slightly better in terms of time and slightly worse in terms of memory. The additional calculations required for the lastfirst procedure increase runtimes by a median factor of 2.5 in our R implementations. The C++ implementation of lastfirst is based on combinatorial definitions and not optimized for speed, and as a result takes much longer—a median factor of almost 2000 relative to maxmin in C++—and failed to complete in many of our tests.

# Experiments

## Empirical data

### MIMIC-III

The open-access critical care database MIMIC-III (“Medical Information Mart for Intensive Care”), derived from the administrative and clinical records for 58,976 admissions of 46,520 patients over 12 years and maintained by the MIT Laboratory for Computational Physiology and collaborating groups, has been widely used for education and research [Goldberger2000; Johnson2016]. For our analyses we included data for patients admitted to five care units: coronary care (CCU), cardiac surgery recovery (CSRU), medical intensive care (MICU), surgical intensive care (SICU), and trauma/surgical intensive care (TSICU).<sup>8</sup> For each patient admission, we extracted the set of ICD-9/10 codes from the patient’s record and several categorical demographic variables: age group (18–29, decades 30–39 through 70–79, and 80+), recorded gender (M or F), stated ethnicity (41 values),<sup>9</sup> stated religion (Catholic, unspecified/unobtainable/missing, Protestant Quaker, Jewish, other, Episcopalian, Greek Orthodox, Christian Scientist, Buddhist, Muslim, Jehovah’s Witness, Unitarian-Universalist, 7th Day Adventist, Hindu, Romanian Eastern Orthodox, Baptist, Hebrew, Methodist, or Lutheran), marital status (married, single, widowed, divorced, unknown/missing, separated, or life partner), and type of medical insurance (Medicare, private, Medicaid, government, or self pay). Following Zhong2020, we transformed these *relational-transaction* (RT) data into a binary case-by-variable matrix  $X \in \mathbb{B}^{n \times p}$  suitable for the cosine similarity measure, which was converted to a distance measure by subtraction from 1. Because cosine similarity is monotonically related to the angle metric, our topological results will be the same up to this rescaling, so for simplicity we use cosine similarity in our experiments.

### Mexican Department of Health

The Mexican Department of Health (MXDH) has released official open-access data containing an assortment of patient-level clinical variables related to COVID-19 infection and outcomes. These data have been compiled into a database and made freely available on Kaggle<sup>10</sup>, a collaborative data science platform, where they are maintained and updated regularly. The database includes information regarding over 724,000 patients confirmed to be COVID-positive via diagnostic laboratory testing. Two main types of information are present for each patient: (1) temporal data, and (2) categorical or binary variables. The temporal data consist of key dates associated with the clinical course of infection such as date of symptom onset, date of admission to a healthcare institution, and date of death (if applicable). The categorical or binary fields encode clinical factors likely to be associated with COVID-19 infection, severity, or outcome. These variables include information such as sex, state of patient residence, and intubation status, as well as binary fields encoding the presence or absence of a wide variety of comorbidities such as asthma, hypertension, cardiovascular disease. (For a full description of each field included in the data set, see Kaggle.\*) Though these variables are categorical rather than continuous/numeric, there are sufficiently many of them (~50) to distinguish between many phenotypic subtypes of COVID-19 patients. Further, this data set is very complete in that every patient is required to contain a valid value for every field, which minimizes concerns around missing data.

### Homology recovery

We compared the suitability of three landmarking procedures (uniformly random, maxmin, lastfirst) on datasets with varying density and duplication patterns by extending an example of deSilva2004. Each experiment proceeded as follows: We sampled  $n = 540$  points from the sphere  $\mathbb{S}^2 \subset \mathbb{R}^3$  and selected  $k = 12$

<sup>8</sup><https://mimic.physionet.org/mimictables/transfers/>

<sup>9</sup>White, Black/African American, Unknown/Not Specified, Hispanic or Latino, Other, Unable to Obtain, Asian, Patient Declined to Answer, Asian – Chinese, Hispanic Latino – Puerto Rican, Black/Cape Verdean, White – Russian, Multi Race Ethnicity, Black/Haitian, Hispanic/Latino – Dominican, White – Other European, Asian – Asian Indian, Portuguese, White – Brazilian, Asian – Vietnamese, Black/African, Middle Eastern, Hispanic/Latino – Guatemalan, Hispanic/Latino – Cuban, Asian – Filipino, White – Eastern European, American Indian/Alaska Native, Hispanic/Latino – Salvadoran, Asian – Cambodian, Native Hawaiian or Other Pacific Islander, Asian – Korean, Asian – Other, Hispanic/Latino – Mexican, Hispanic/Latino – Central American (Other), Hispanic/Latino – Colombian, Caribbean Island, South American, Asian – Japanese, Hispanic/Latino – Honduran, Asian – Thai, American Indian/Alaska Native Federally Recognized Tribe

<sup>10</sup><https://www.kaggle.com/lalish99/covid19-mx>

landmark points. We then used the landmarks to compute PH and recorded the statistics  $R_0, R_1, K_0, K_1$  as defined by @deSilva2004. From these statistics we compute the *relative dominance*  $(R_1 - R_0)/K_0$  and *absolute dominance*  $(R_1 - R_0)/K_1$  of the last interval over which all Betti numbers are correctly calculated.

The points  $x = (r, \theta, \phi)$  were sampled using four procedures: uniform sampling, skewed sampling, uniform sampling with skewed boosting, and skewed sampling with skewed boosting. The first procedure was used by @deSilva2004 and here serves as a baseline case. For a sample  $S$  (with multiplicities) generated from each of the other three procedures, the expected density  $\lim_{\varepsilon \rightarrow 0} \lim_{n \rightarrow \infty} \frac{1}{n} |\{x = (r, \theta, \phi) \in S \mid \alpha - \varepsilon < \phi < \alpha + \varepsilon\}|$  of points near a given latitude  $\alpha \in [0, \pi]$  is proportional to the quartic function  $p : [0, 1] \rightarrow [0, 1]$  defined by  $p(x) = (\frac{\phi}{\pi})^4$ . Skewed sampling is performed via rejection sampling: Points  $x_i = (r_i, \theta_i, \phi_i)$  are sampled uniformly and rejected at random if a uniform random variable  $t_i \in [0, 1]$  satisfies  $(\frac{\phi_i}{\pi})^\alpha < t_i$  until  $n$  points have been kept [Diaconis2013]. Skewed boosting is performed by first obtaining a (uniform or skewed) sample  $T$  of size a fraction  $\frac{n}{6}$  of the total, then sampling  $n$  points (with replacement) from  $T$  using the probability mass function satisfying  $P(x_i) \propto (\frac{\phi_i}{\pi})^\beta$ . When performed separately, skewed sampling and skewed boosting use  $\alpha = \beta = 4$ ; when performed in sequence, they use  $\alpha = \beta = 2$ .

The landmark points were selected in three ways: uniform random selection (without replacement), the maxmin procedure, and the lastfirst procedure. We computed PH in Python GUDHI, using three implementations: Vietoris–Rips (VR) filtrations on the landmarks, alpha complexes on the landmarks, and witness complexes on the landmarks with the full sample as witnesses [Maria2021; Rouvreau2021; Kachanovich2021].

The skewed data sets are dense (or multiple) at the south pole and sparse (or singular) at the north pole. We expect lastfirst to be more sensitive to this variation and place more landmarks toward the south. As measured by dominance, therefore, we hypothesized that lastfirst would be competitive with maxmin when samples are uniform and inferior to maxmin when samples are skewed. Put differently, we expected lastfirst to better reject the homology of  $S^2$ , i.e. to detect the statistical void at the north pole.

Figure 2 compares the relative dominance of the spherical homology groups in each case. When PH is computed using VR or alpha complexes, maxmin better recovers the homology of the sphere except on uniform samples, while lastfirst and random selection better detect the void. Random selection is usually better than lastfirst selection at detecting this void when samples are non-uniform, which indicates that lastfirst selection still oversamples from less dense regions. Lastfirst and maxmin perform similarly when PH is computed using witness complexes.

## Covers and nerves

Cardinality reduction techniques can be used to model a large number of cases represented by a large number of variables as a smaller number of clusters with similarity or overlap relations among them. The deterministic sampling procedures maxmin and lastfirst provide clusters (cover sets) defined by proximity to the landmark cases and relations defined by their overlap. The clusters obtained by these procedures occupy a middle ground between the regular intervals or quantiles commonly used to cover samples from Euclidean space and the emergent clusters obtained heuristically by penalizing between-cluster similarity and rewarding within-cluster similarity. The maxmin procedure produces cover sets of (roughly) fixed radius, analogous to overlapping intervals of fixed length, while the lastfirst procedure produces cover sets of fixed size, analogous to the quantiles of an adaptive cover. This makes them natural solutions to the task of covering an arbitrary finite metric space that may or may not contain important geometric or topological structure [Singh2007].

As a practical test of this potential, we loosely followed the approach of Dlotko2019 to construct covers and their nerves for each care unit of MIMIC-III, using maxmin and lastfirst. We varied the number of landmarks (6, 12, 24, 36, 48, 60, 120) and the multiplicative extension of the cover sets’ sizes (0, .1, .2). We evaluated the procedures in three ways:

- **Clustering quality:** Both procedures yield *fuzzy* clusters—that is to say, clusters that allow for some overlap. While clustering quality measures might be useful, most, including almost all that have been proposed for fuzzy clusterings, rely on coordinate-wise calculations, specifically data and cluster centroids [Bouguessa2006; Wang2007; Falasconi2010]. To our knowledge, the sole exception to have appeared in a comprehensive comparison of such measures is the *modified partition coefficient*

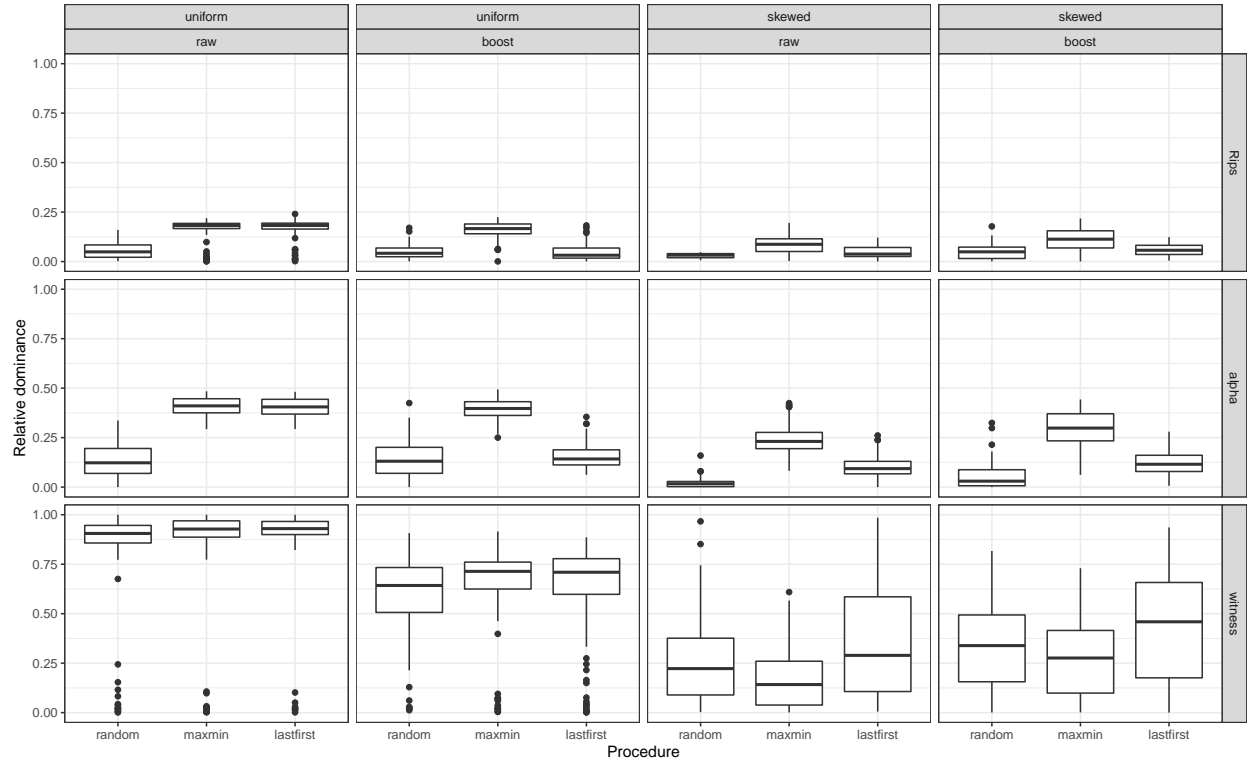


Figure 2. Relative dominance of the spherical homology groups in the persistent homology of four samples from the sphere, using each of three landmark procedures and three persistence computations. Similar plots of absolute dominance (not shown) tell a consistent story, but the distributions are more skewed so the comparisons are less clear.



[@Dave1996], defined as

$$\text{MPC} = 1 - \frac{k}{k-1} \left(1 - \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k u_{ij}^2\right)$$

where  $U = (u_{ij})$  is the  $n \times k$  fuzzy partition matrix:  $u_{ij}$  encodes the extent of membership of point  $x_i$  in cluster  $c_j$ , and  $\sum_{j=1}^k u_{ij} = 1$  for all  $i$ . When a point  $x_i$  is contained in  $m$  cover sets  $c_j$ , we equally distribute its membership so that  $u_{ij} = \frac{1}{m}$  when  $x_i \in c_j$  and  $u_{ij} = 0$  otherwise. Thus, the MPC quantifies the extent of overlap between all pairs of clusters. Like the partition coefficient from which it is adapted, the MPC takes the value 1 on crisp partitions and is penalized by membership sharing, but it is standardized so that its range does not depend on  $k$ .

- **Discrimination of risk:** For purposes of clinical phenotyping, patient clusters are more useful that better discriminate between low- and high-risk subgroups. We calculate a cover-based risk estimate from individual outcomes  $y_i$  as follows: For each cover set  $c_j \subset X$ , let  $p_j = \frac{1}{|c_j|} \sum_{x_i \in c_j} y_i$  be the incidence of the outcome in that set. Then compute the weighted sum  $q_i = \sum_{x_i \in c_j} u_{ij} p_j$  of these incidence rates for each case. We measure how well the cover discriminates risk as the area under the receiver operating characteristic curve (AUROC).

We hypothesized that lastfirst covers would exhibit less overlap than maxmin covers by virtue of their greater sensitivity to local density, and that they would outperform maxmin covers at risk prediction by reducing the sizes of cover sets in denser regions of the data (taking advantage of more homogeneous patient cohorts).

Figure 3 presents the sizes of the nerves of the covers and the two evaluation statistics as functions of the number of landmarks. The numbers of 1- and of 2-simplices grew at most roughly quadratically and roughly cubically, respectively. This suggests that the densities of the simplicial complex models were at most roughly constant, regardless of the number of landmarks. Landmark covers grew fuzzier and generated more accurate predictions until the number of landmarks reached around 60, beyond which point most covers grew crisper while performance increased more slowly (and in one case decreased). This pattern held for covers with any fixed multiplicative extension. Naturally, these extensions produced fuzzier clusters, but they also reduced the overall accuracy of the predictive model. In addition to (i.e. independently of) these patterns, models fitted to smaller care units tended to outperform those fitted to larger care units. Contrary to expectations, unextended maxmin covers were usually crisper than their lastfirst counterparts and yielded more accurate predictions, though extensions reduced the crispness of maxmin covers more dramatically than of lastfirst covers. The same patterns were observed in the risk discrimination of maxmin versus lastfirst covers, with maxmin covers yielding the most accurate predictions when unextended but lastfirst covers retaining more accuracy after extension.

Figure 4 presents the same evaluations for covers of the MXDH data. In contrast to the MIMIC experiments, lastfirst-based nerves of the MXDH data grow sub-polynomially and are significantly sparser than maxmin-based nerves. Lastfirst covers tend to be crisper, especially as the number of landmarks and the extension factors increase. This indicates that the nearest neighborhoods form a more parsimonious cover of the data than the centered balls. The predictive accuracies of the maxmin- and lastfirst-selected cover-based models converge with increasing landmarks<sup>11</sup>, though for smaller numbers different selection procedures perform best for different outcomes.

## Interpolative nearest neighbors prediction

Landmark points may also be used to trade accuracy for memory in neighborhood-based prediction modeling. Consider the following approach: A modeling process involves predictor data  $X \in \mathbb{R}^{n \times p}$  and response data  $y \in \mathbb{R}^{n \times 1}$ , partitioned into training and testing sets  $X_0, X_1$  and  $y_0, y_1$  according to a partition  $I_0 \sqcup I_1 = \{1, \dots, n\}$  of the index set. Given  $x \in X_1$ , a nearest neighbors model computes the prediction  $p(x) = \frac{1}{k} \sum_{q(x, x_i) \leq k} y_i$  by averaging the responses for the  $k^{\text{th}}$  nearest neighbors of  $x$  in  $X_0$ . By selecting a landmark set  $L \subset X_0$ , a researcher can reduce the computational cost of the model as follows: For each  $\ell \in L$ , calculate  $p(\ell)$  as above. Then, for each  $x \in X_1$ , calculate  $p_L(x) = \sum_{\ell \in L} w(d(x, \ell)) p(\ell) / \sum_{\ell \in L} w(d(x, \ell))$ , where  $w : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  is a

<sup>11</sup>We should increase the maximum number of landmarks to be sure.

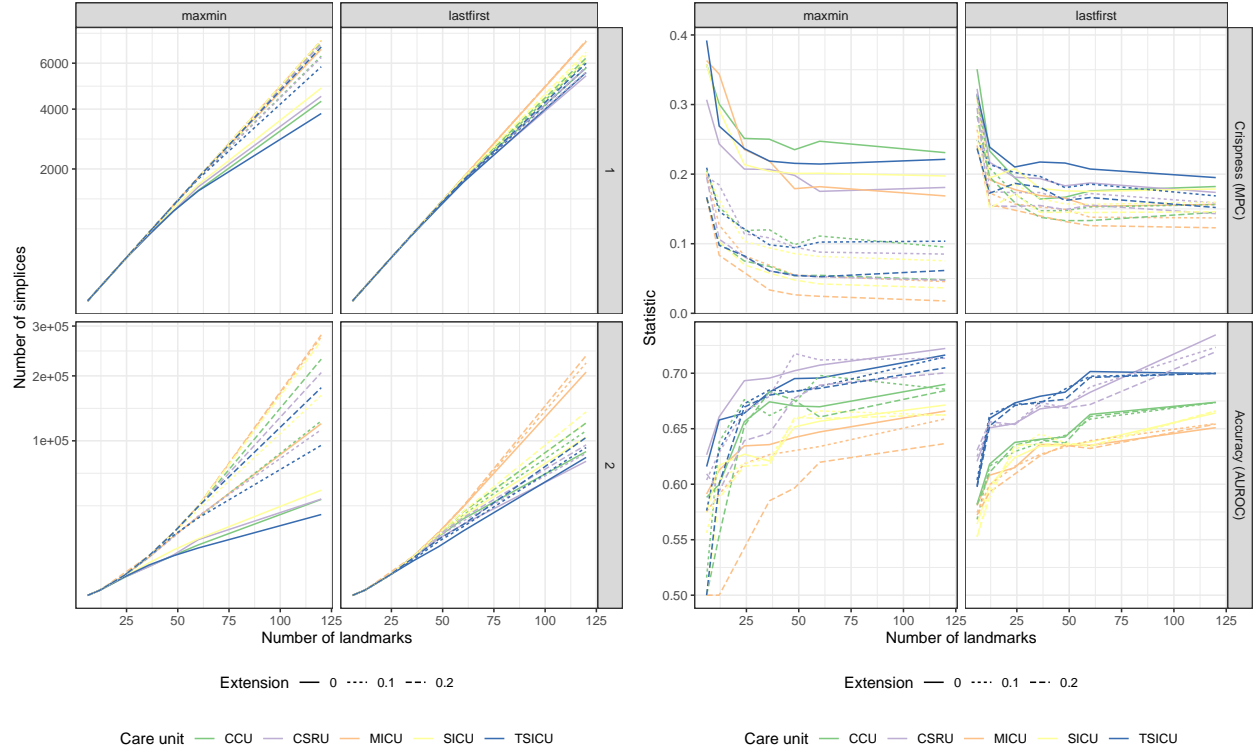


Figure 3. Summary and evaluation statistics versus number of 0-simplices (landmarks) for the covers generated using the maxmin and lastfirst procedures, with three multiplicative extensions in their size. Left: the sizes of their nerves, as numbers of 1- and 2-simplices, using a square root-transformed vertical scale. Right: the modified partition coefficient (MPC) and the c-statistic of the risk prediction model based on the cover sets (AUROC).

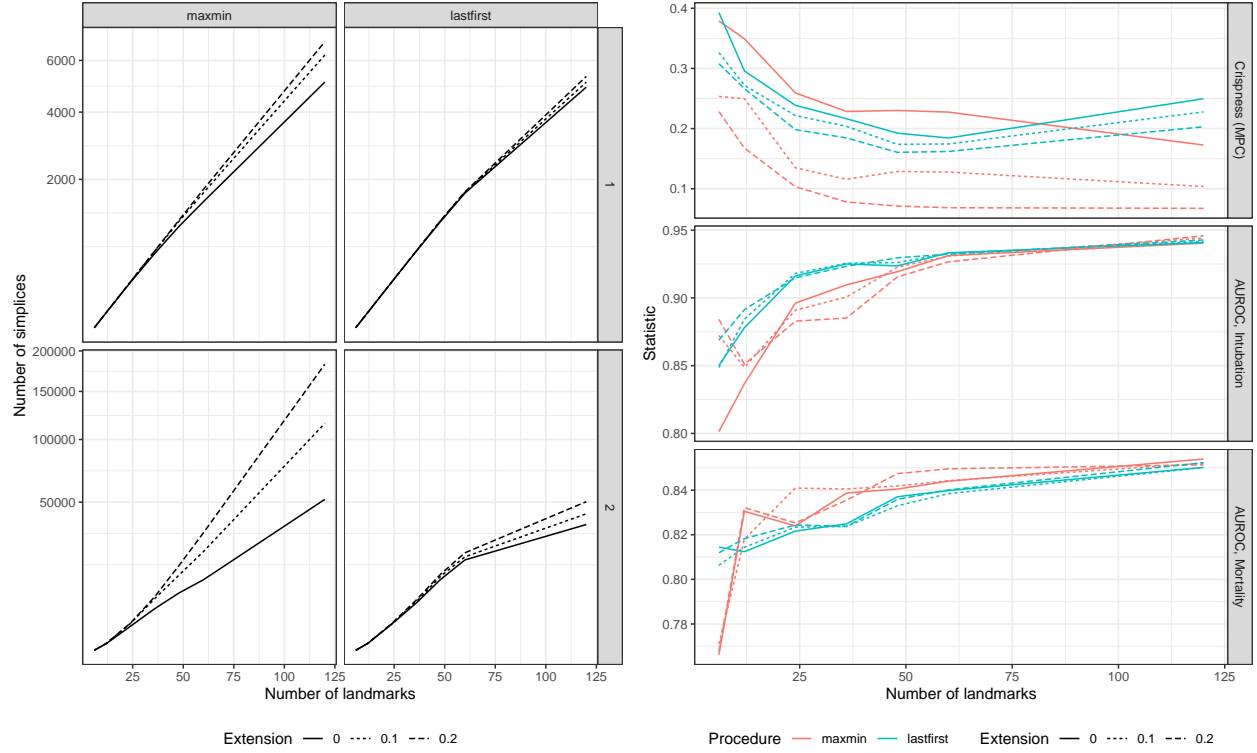


Figure 4. Summary and evaluation statistics versus number of 0-simplices (landmarks) for the covers generated using the maxmin and lastfirst procedures, with three multiplicative extensions in their size. Left: the sizes of their nerves, as numbers of 1- and 2-simplices, using a square root-transformed vertical scale. Right: the modified partition coefficient (MPC) and the c-statistics of the risk prediction models based on the cover sets (AUROC).

weighting function (for example,  $w(d) = d^{-1}$ ). The nearest neighbor predictions for  $L$  thus serve as proxies for the responses associated with  $X_0$ .

We took this approach to the prediction of in-hospital mortality for patients with records in each critical care unit of MIMIC-III. We then implemented the following procedure:

1. Determine a nested  $6 \times 6$ -fold split for train-fit-test cross-validation. That is, partition  $[n] = \bigsqcup_{i=1}^6 I_i$  into roughly equal parts, and partition each  $[n] \setminus I_i = \bigsqcup_{j=1}^6 J_{ij}$  into roughly equal parts.
2. Iterate the following steps over each  $i, j$ :
  - a) Generate a sequence  $L$  of landmarks from the points  $X_{([n] \setminus I_i) \setminus J_{ij}}$ .
  - b) Identify the 180 nearest neighbors  $N_{180}^+(\ell)$  of each landmark  $\ell$ . This was a fixed parameter, chosen for being slightly larger than the optimal neighborhood size in a previous study of individualized models [Lee2015].
  - c) Find the value of  $k \in [180]$  and the weighting function  $w$  (among those available) for which the predictions  $p_L : X_{J_{ij}} \rightarrow [0, 1]$  maximize the AUROC.
  - d) Use the AUROC to evaluate the performance of the predictions  $p_L : X_{I_i} \rightarrow [0, 1]$  using these  $k$  and  $w$ .

We replicated the experiment for each combination of procedure (random, maxmin, lastfirst) and number of landmarks ( $|L| = 36, 60, 180, 360$ ). We hypothesized that, as measured by overall accuracy of the resulting predictive model, the maxmin and lastfirst procedures would outperform random selection, and that lastfirst would outperform maxmin, for similar reasons to those in the previous section.

Boxplots of the AUROCs for each cross-validation step are presented in Figure 5. While both landmark procedures yielded stronger results than random selection, lastfirst performed on average slightly worse than maxmin on each data set. Importantly, both landmark procedures also yielded more accurate predictions than a basic unweighted nearest-neighbors model, lending support to the modeling approach itself. Interestingly, only on the largest data set (the MICU) did increasing the number of landmarks from 36 to 360 appreciably improve predictive accuracy (using all three selection procedures).

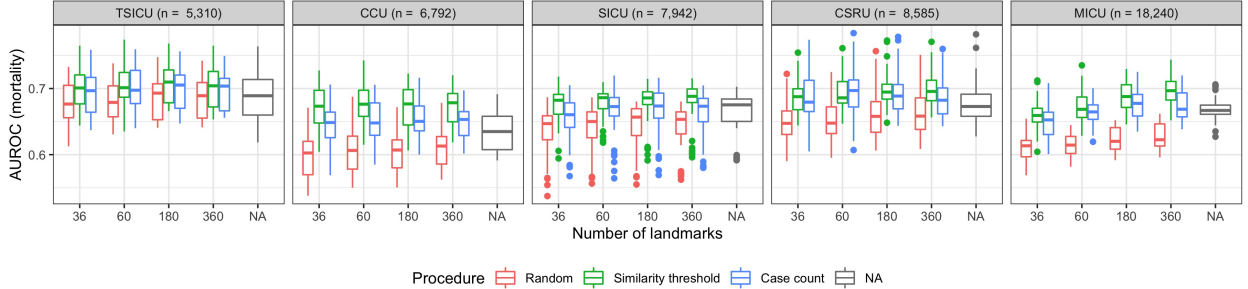


Figure 5. AUROCs of the interpolative predictive models of mortality in five MIMIC-III care units based on covers constructed using random, maxmin, and lastfirst procedures to generate landmarks. Each boxplot summarizes AUROCs from  $6 \times 6 = 36$  models, one for each combination of outer and inner fold. AUROCs of simple nearest-neighbor predictive models are included for comparison.

Over the course of the COVID-19 pandemic, hospitals and other facilities experienced periods of overburden and resource depletion, and best practices were continually learned and disseminated. As a result, outcomes in the MXDH data reflect institutional- as well as population-level factors. We took advantage of the rapid learning process in particular by adapting the nested CV approach above to a nested-longitudinal CV approach: We partitioned the data by week, beginning with Week 11 (March 11–17) and ending with Week 19 (May 6–9, the last dates for which data were available). For each week  $i$ ,  $11 < i \leq 19$ , we trained prediction models on the data from Week  $i - 1$ . We then randomly partitioned Week  $i$  into six roughly equal parts and optimized and evaluated the models as above. (For this analysis, we only considered Gaussian weighting.)

Line plots of model performance are presented in Figure 6, with one curve (across numbers of landmarks) per

selection procedure, outcome, and week. Again both landmark selection procedures yielded stronger results than random selection. Interestingly, this was more pronounced in later weeks, as the pandemic progressed, even as overall predictive accuracy declined.<sup>12</sup> Overall, performance improved slightly as the number of landmarks increased from 50 to 150 but either plateaued or declined from 150 to 250.

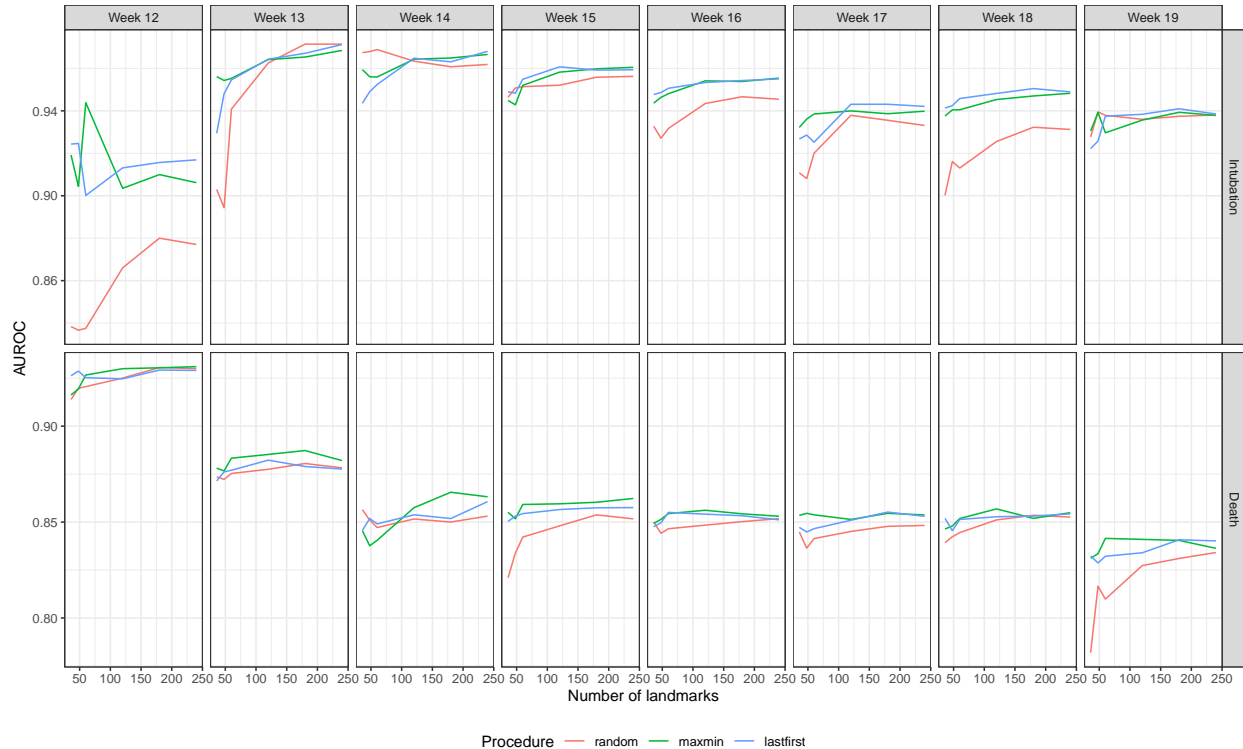


Figure 6. AUROC of the sliding-window interpolative predictive models of intubation and mortality in the MXDH data based on covers constructed using random, maxmin, and lastfirst procedures to generate landmarks.

## Discussion

The definitions of our lastfirst and firstlast procedures are analogous to those of maxmin and minmax, substituting ranks in the role of distances. In this way, lastfirst is an alternative to maxmin that is adaptive to the local density of the data, similar to the use of fixed quantiles in place of fixed-length intervals. The maxmin and lastfirst procedures implicitly construct a minimal cover whose sets are centered at the selected landmarks, and the fixed-radius balls of maxmin correspond to the fixed-cardinality neighborhoods of lastfirst. The rank-based procedures are more combinatorially complex and computationally expensive, primarily because rank-distances are asymmetric, which doubles (in the best case) or squares (in the worst case) the number of distances that must be calculated. Nevertheless, the procedure can be performed in a reasonable time for many real-world uses.

We ran an experiment to compare maxmin and lastfirst landmarks at expediting the computation of persistent homology, extending an experiment of @deSilva2004. In addition to a uniform sample from a sphere, we drew skewed and bootstrapped samples in order to simulate data sets with variable density and multiplicity, in this case exhibiting a statistical void at one pole of the sphere, opposite a concentration at the other

<sup>12</sup>Do we know what major events in Brazil might have influenced these outcomes? We should at least also plot the number of cases, stratified by outcomes, per day over this period.

pole. Whereas the maxmin procedure would sample more uniformly across the sphere despite this skew, the lastfirst procedure would concentrate landmarks toward the south. Classical persistent homology is notoriously sensitive to outliers, and maxmin better recovered spherical homology than lastfirst, as expected. This is likely due in part to the filtration itself being based on distances rather than rank-distances between landmarks (and other points as witnesses). Yet, compared to random sampling, lastfirst still oversampled from less dense regions. This is desirable for settings, such as healthcare, in which regions of missingness often indicate limitations of the data collection rather than rarity of case types in the population.

We also ran several experiments that used landmarks to obtain well-separated clusters of patients with common risk profiles and to more efficiently generate nearest neighbor predictions. Because we designed lastfirst to produce cover sets of equal size despite variation in the density or multiplicity of the data, we expected it to outperform maxmin with respect to the crispness of clusters and to the accuracy of predictions. In particular, we expected that the optimal neighborhood size for outcome prediction would be roughly equal across our data; as a result, by assigning each landmark case an equally-sized cohort of similar cases, we expected predictions based on these cohorts to outperform those based on cohorts using a fixed similarity threshold.

Contrary to expectations, maxmin produced crisper clusterings on average, and in the case of MIMIC-III more accurate predictions. However, when the sets of these covers had their radii or cardinalities extended by a fixed proportion, those of lastfirst better preserved these qualities. Additionally, in the case of MXDH, neither landmark selection procedure produced consistently more accurate predictions.

A possible explanation for the stronger performance of maxmin on MIMIC is that the data did not exhibit very strongly the patterns for which the lastfirst procedure is designed to account, namely variation in density and multiplicity. As a result, the RT-similarity measure is in fact meaningful across the population: Whatever the baseline presentation of a patient, rather than a cohort of similar patients of some fixed size, their prognosis would be better guided by a cohort cut off at a fixed minimum similarity (or maximum distance). This suggests that the use of personalized cohorts to improve predictive modeling, as employed by @Lee2015, may be strengthened by optimizing a fixed similarity threshold rather than a fixed cohort size. It is worth noting that @Park2006 reached a similar conclusion. In contrast, this stronger performance was not evident on MXDH, which contained fewer variables and as a result exhibited many more occurrences of duplication.

Another way to think about these results is in terms of a balance between relevance and pwoer, with fixed-radius balls (respectively, fixed-cardinality neighborhoods) providing training cohorts of roughly equal relevance (statistical power) to all test cases. With sufficiently rich data, relevance can be more precisely measured and becomes more important to cohort definition, as with MIMIC. When variables are fewer, as with MXDH, relevance is more difficult to measure, so that larger samples can improve performance even at the expense of such a measure.

## References