Progressive Scattered Data Filtering

Armin Iske

Abstract. Given a finite point set $Z \subset \mathbb{R}^d$, the *covering radius* of a non-empty subset $X \subset Z$ is the minimum distance $r_{X,Z}$ such that every point in Z is at a distance of at most $r_{X,Z}$ from some point in X. This paper concerns the construction of a sequence of subsets of decreasing sizes, such that their covering radii are small. To this end, a method for progressive data reduction, referred to as *scattered data filtering*, is proposed. The resulting scheme is a composition of greedy *Thinning*, a recursive point removal strategy, and *Exchange*, a postprocessing local optimization procedure. The paper proves adaptive a priori lower bounds on the minimal covering radii, which allows us to control for any current subset the deviation of its covering radius from the optimal value at run time. Important computational aspects of greedy Thinning and Exchange are discussed. The good performance of the proposed filtering scheme is finally shown by numerical examples.

Key words: Thinning algorithms, progressive scattered data reduction, scattered data modelling, k-center-problem, data clustering.

1 Introduction

Let $Z \subset \mathbb{R}^d$, $d \geq 1$, be a finite scattered point set of size $M = |Z|$, and let $\mathcal{Z} = \{X \subset Z : X \neq Z, X \neq \emptyset\}$ denote the power set of its $2^M - 2$ non-empty (strict) subsets. Moreover, let $\|\cdot\|$ be any norm on $\mathbb{R}^d$, and for any point $z \in Z$ let

$$d_X(z) = \min_{x \in X} \|x - z\|$$

denote the distance between z and a subset $X \in \mathcal{Z}$.

This paper concerns the construction of a sequence $\{X_n\}_{n=1}^{M-1} \subset \mathcal{Z}$ of subsets, with decreasing sizes $|X_n| = M - n$, such that for each $X \equiv X_n \subset Z$ its *covering radius*

$$r_{X,Z} = \max_{z \in Z} d_X(z)$$

on Z is small. The progressive construction of such a sequence is accomplished by using *filter operators*, one at a time, each of whose action on Z returns a *locally optimal* subset $X_n \subset Z$ (a precise definition for the term "locally optimal" is given in Definition 1, Section 2). The resulting data reduction scheme is termed *progressive scattered data filtering*.

This work is mainly driven by applications in scattered data modelling. This is subject of the discussion in the previous papers [9, 10], where the utility of scattered data filtering for least squares approximation and multilevel interpolation by radial basis functions is shown. In order to briefly explain this particular application, we remark that scattered data modelling requires reconstructing an unknown function $f : \mathbb{R}^d \to \mathbb{R}$ from its function values sampled at the points in Z. In radial basis function schemes this is done by using, for a fixed radial function $\Phi \equiv \phi(\|\cdot\|)$, approximations of the form

$$s = \sum_{x \in X} c_x \phi(\|\cdot - x\|), \tag{1}$$

where $X \subset Z$. Hence, the approximation space is spanned by X-translates of the basis function Φ. The coefficients c_x, $x \in X$, in (1) are computed by the underlying approximation scheme (for more details see the recent tutorial [11]). According to the discussion in [9, 10], for the purpose of combining good approximation quality with low computational costs, it is desirable to select a *small* set $X \in \mathcal{Z}$ such that its covering radius $r_{X,Z}$ on Z is *small*. But this requires carefully balancing the size of X and the value $r_{X,Z}$. To this end, one *good* subset $X \subset Z$ from the sequence $\{X_n\}_n$, generated by the proposed filtering scheme, is selected.

Before we proceed with explaining details on this filtering scheme, which is the subject of most of this paper, let us first make a few general remarks. Observe that for the above purpose one ideally wants to pick one subset $X \equiv X_n^* \subset Z$, with (small) size $|X_n^*| = M - n$, which is *optimal* by minimizing the covering radius $r_{X,Z}$ among all subsets $X \subset Z$ of equal size, so that

$$r_n^* = r_{X_n^*,Z} = \min_{\substack{X \subset Z \\ |X| = M-n}} r_{X,Z}. \tag{2}$$

The problem of finding an algorithm which outputs for any possible input pair (Z, n), $1 \leq n < |Z|$, such an optimal subset X_n^* satisfying (2) is one particular instance of the *k-center problem* (in a more general setting, the norm $\|\cdot\|$ may be replaced by any arbitrary metric).

But the k-center problem is, due to Kariv & Hakimi [12], NP-hard. Moreover, the problem of finding an *α-approximation algorithm*, $\alpha \geq 1$, for the k-center problem which outputs for any input pair (Z, n), $1 \leq n < |Z|$, a subset $X_n \subset Z$ of size $|X_n| = M - n$ satisfying

$$r_{X_n,Z} \leq \alpha \cdot r_n^* \tag{3}$$

is for any $\alpha < 2$ NP-complete. Hochbaum & Shmoys [8] were the first to provide a 2-approximation algorithm (i.e. $\alpha = 2$) for the k-center problem,

which is best possible unless P=NP. For a comprehensive discussion on the k-center problem we refer to the textbook [7, Section 9.4.1] and the survey paper [14], where the Hochbaum-Shmoys algorithm is explained.

In contrast to the situation in the k-center problem, our filtering scheme does not work with a beforehand selection for $M-n$, the size of the output $X_n \in \mathcal{Z}$. Instead of this, our algorithms picks one *good* subset X_n at run time. This selection relies on adaptive bounds of the form

$$r_{X_n,Z} \leq \alpha_{X_n,Z} \cdot r_n^*, \tag{4}$$

where $\alpha_{X_n,Z} = r_{X_n,Z}/\sigma_n$ denotes the *quality index* of X_n, and the numbers σ_n solely depend on the distribution of the points in Z. Note that the upper bound on $r_{X_n,Z}$ in (4) looks similar to the one in (3). However, while $\alpha_{X_n,Z}$ in (4) depends on both Z and $X_n \in \mathcal{Z}$, the universal constant α in (3) does not even depend on Z. In fact, the sequence of numbers $\alpha_{X_n,Z}$, recorded at run time, helps us to control the deviation between any current covering radius $r_{X_n,Z}$ and the optimal value r_n^*. Details on this are explained in Section 3.

The filtering scheme itself, which is subject of the following Section 2, is a composition of greedy *Thinning*, a recursive point removal scheme, and *Exchange*, a postprocessing local optimization strategy. Greedy Thinning is discussed in Section 4, and important computational aspects concerning Exchange are addressed in Section 5. Numerical examples in Section 6 finally show how the proposed filtering scheme performs in comparison with α-approximation algorithms for the k-center problem.

2 Scattered Data Filtering

In this section, details on the construction of the abovementioned sequence $\{X_n\}_n \subset \mathcal{Z}$ by progressive scattered data filtering are discussed. This filtering scheme is associated with a sequence $\mathcal{F} = \{F_n\}_n$ of filter operators $F_n : Z \to \mathcal{Z}$, satisfying $|F_n(Z)| = M - n$, so we let $X_n = F_n(Z)$. Moreover, it is required that every subset $X_n \in \mathcal{Z}$ output by the operator F_n is locally optimal in Z.

Definition 1 *Let $X \in \mathcal{Z}$ and $Y = Z \setminus X \in \mathcal{Z}$. The set X is said to be* `locally optimal` *in Z, iff there is no pair $(x, y) \in X \times Y$ of points satisfying*

$$r_{X,Z} > r_{(X\setminus x)\cup y,Z}. \tag{5}$$

A point pair $(x, y) \in X \times Y$ satisfying (5) is said to be `exchangeable`.

Hence, if $X \in \mathcal{Z}$ is locally optimal in Z, then the covering radius $r_{X,Z}$ of X on Z cannot be reduced by one single exchange between a point $x \in X$ and a point y in the difference set $Y = Z \setminus X$. Note that every (globally) optimal subset X_n^* satisfying $r_{X_n^*,Z} = r_n^*$ is also locally optimal.

Now the idea of progressive scattered data filtering is to combine a recursive point removal scheme, termed *Thinning*, with a postprocessing local optimization procedure, termed *Exchange*. Exchange outputs, on any given $X \in \mathcal{Z}$, a locally optimal subset of equal size $|X|$. This is accomplished, according to the following algorithm, by iteratively swapping exchangeable point pairs between X and $Z \setminus X$.

Algorithm 1 (Exchange).

INPUT: $X \in \mathcal{Z}$;

(1) *Let* $Y = Z \setminus X$;
(2) **WHILE** (X *not locally optimal in* Z)
  (2a) *Locate an exchangeable pair* $(x, y) \in X \times Y$;
  (2b) *Let* $X = (X \setminus x) \cup y$ *and* $Y = (Y \setminus y) \cup x$;

OUTPUT: $X \in \mathcal{Z}$, *locally optimal in* Z.

Note that the Exchange Algorithm terminates after finitely many steps. Indeed, this is because the set Z is assumed to be finite, and each exchange in step **(2b)** strictly reduces the current (non-negative) covering radius $r_{X,Z}$. By construction, the output set $X \in \mathcal{Z}$ is then locally optimal. A characterization of exchangeable point pairs is provided in Section 5. This yields useful criteria for the efficient localization of such point pairs.

Now let us turn to Thinning. This class of recursive point removal schemes is used for multilevel scattered data interpolation in [5], and moreover analyzed in [6]. A generic formulation of Thinning is given by the following algorithm.

Algorithm 2 (Thinning).

INPUT: Z *with* $|Z| = M$, *and* $n \in \{1, \ldots, M-1\}$;

(1) *Let* $X_0 = Z$;
(2) **FOR** $k = 1, \ldots, n$
  (2a) *Locate a* `removable` *point* $x \in X_{k-1}$;
  (2b) *Let* $X_k = X_{k-1} \setminus x$;

OUTPUT: $X_n \in \mathcal{Z}$, *of size* $|X_n| = M - n$.

In order to select a specific Thinning strategy, it remains to give a definition for a removable point in step **(2a)** of the Algorithm 2. Details on our preferred Thinning strategy are discussed in Section 4.

For the subsequent discussion in this paper, it is convenient to associate with any Thinning algorithm a *Thinning operator* T. The operation of T on any non-empty subset $X \subset Z$ is defined by $T(X) = X \setminus x$ for one unique $x \in X$, so by the action of T on X the point x is removed from X. Therefore, any subset X_n output by Algorithm 2 can be written as $X_n = T^n(Z)$, where $T^n = T \circ \cdots \circ T$ denotes the n-fold composition of T. Likewise, the Exchange Algorithm 1 is viewed as an operator $E : \mathcal{Z} \to \mathcal{Z}$, which returns on any given argument $X \in \mathcal{Z}$ a locally optimal subset $E(X) \in \mathcal{Z}$ of equal size $|E(X)| = |X|$. Hence, E is a projector onto the locally optimal sets in $\mathcal{Z}$.

Having specified such operators T and E, this already yields by the composition $F_n = E \circ T^n$ a sequence $\mathcal{F} = \{F_n\}_n$ of filter operators with the desired properties. Indeed, any subset $X_n = F_n(Z)$ output by the operator $F_n = E \circ T^n$ is locally optimal in Z and it moreover satisfies $|X_n| = M - n$ by construction.

3 Adaptive Bounds on the Covering Radii

In this section, adaptive bounds on the covering radii $r_{X,Z}$, $X \in \mathcal{Z}$, are proven. To this end, assume without loss of generality that the points in $Z = \{z_1, \ldots, z_M\}$ are ordered such that their *significances*

$$\sigma(z) = d_{Z \setminus z}(z), \quad \text{for } z \in Z, \tag{6}$$

are increasing, i.e.

$$\sigma(z_1) \leq \sigma(z_2) \leq \cdots \leq \sigma(z_M). \tag{7}$$

Note that for any $z \in Z$ its significance $\sigma(z)$ in (6) is the distance to its nearest neighbour in Z. Hence, according to the above assumption (7) on the ordering of the points in Z, the value $\sigma(z_1)$ yields the minimal distance between two points in Z. In fact, since this minimum is attained by at least two points in Z, we have $\sigma(z_1) = \sigma(z_2)$.

For notational simplicity, we let $\sigma_n = \sigma(z_n)$, $1 \leq n \leq M$. Moreover, for any $X \in \mathcal{Z}$ of size $|X| = M - n$, we let $Y = Z \setminus X$ denote the difference set, whose size is then $|Y| = n$. Starting point of the subsequent discussion is the following lower bound on the covering radius $r_{X,Z}$ for $X \in \mathcal{Z}$.

Theorem 1 *For any $X \in \mathcal{Z}$ of size $|X| = M - n$ the inequality*

$$\sigma_n \leq r_{X,Z} \tag{8}$$

holds.

Proof: Since for any $y \in Y = Z \setminus X$ the inequality

$$d_{Z\setminus Y}(y) \geq d_{Z\setminus y}(y) = \sigma(y)$$

holds, we conclude

$$r_{X,Z} = r_{Z\setminus Y,Z} = \max_{z\in Z} d_{Z\setminus Y}(z) = \max_{y\in Y} d_{Z\setminus Y}(y) \geq \max_{y\in Y} \sigma(y). \tag{9}$$

By our assumption (7) on the ordering of the points in Z and by $|Y| = n$, it follows that

$$\max_{y\in Y} \sigma(y) \geq \sigma(z_n) = \sigma_n$$

which completes, by using (9), our proof. □

Note that the above inequality (8) holds in particular for any optimal set $X_n^* \in \mathcal{Z}$ of size $|X_n^*| = M - n$ satisfying $r_{X_n^*,Z} = r_n^*$, which yields $\sigma_n \leq r_n^*$ for $n = 1, \ldots, M-1$. This immediately implies

$$r_{X,Z} = \alpha_{X,Z} \cdot \sigma_n \leq \alpha_{X,Z} \cdot r_n^*,$$

where we let $\alpha_{X,Z} = r_{X,Z}/\sigma_n$. This is the adaptive upper bound (4) stated in the introduction. In summary, we draw the following conclusion from Theorem 1.

Corollary 1 *For any $X \in \mathcal{Z}$ of size $|X| = M - n$ the inequalities*

$$\sigma_n \leq r_n^* \leq r_{X,Z} \leq \alpha_{X,Z} \cdot r_n^* \tag{10}$$

hold, where $\alpha_{X,Z} = r_{X,Z}/\sigma_n \geq 1$. □

The above upper bound on $r_{X,Z}$ in (10) is particularly useful for our purposes. In fact, (10) implies

$$\left| \frac{r_{X,Z} - r_n^*}{r_n^*} \right| \leq \alpha_{X,Z} - 1,$$

which allows us to control, for any current $X \equiv X_n \in \mathcal{Z}$ in the sequence $\{X_n\}_n$, the relative deviation between the current covering radius $r_{X_n,Z}$ and

the optimal value r_n^*. The quality indices $\alpha_{X_n,Z}$ are recorded at run time during the filtering. Whenever $\alpha_{X_n,Z}$ is close to one, this then indicates that the set X_n is close to one optimal set of equal size $M-n$. In our applications in [9, 10], this turns out to be a useful criterion for the subset selection.

In situations where the optimal value r_n^* is known, the following observation may help to construct an optimal subset by using the initial significances of the points in Z.

Theorem 2 *Suppose $X \in \mathcal{Z}$ is an optimal subset of size $|X| = M - n$. Then, $\sigma(y) \leq r_n^*$ for all $y \in Y = Z \setminus X$.*

Proof: Note that every point $y \in Y$ satisfies

$$\sigma(y) = d_{Z\setminus y}(y) \leq d_{Z\setminus Y}(y) = d_X(y) \leq r_{X,Z}. \tag{11}$$

Moreover, since X is optimal, we have $r_{X,Z} = r_n^*$. This in combination with (11) implies $\sigma(y) \leq r_n^*$ for every $y \in Y$, as stated. □

Note that the above characterization implies that any optimal $X^* \in \mathcal{Z}$ of size $M - n$ is necessarily a superset of $Z \setminus \{z \in Z : \sigma(z) \leq r_n^*\}$. We come back to this point in the following section.

4 Greedy Thinning

Greedy algorithms are known as efficient and effective methods of dynamic programming for solving optimization problems. Greedy algorithms typically go through a sequence of steps, where for each step a choice is made that looks best at the moment. For a general introduction to greedy algorithms we recommend the textbook [1, Chapter 16].

4.1 Characterization of Removable Points

In our particular situation, a greedy Thinning algorithm is one where at each step one point is removed, such that the resulting covering radius is minimal among all other possible point removals. This leads us to the following definition for a removable point in step **(2a)** of Algorithm 2.

Definition 2 *For any $X \in \mathcal{Z}$ with $|X| \geq 2$, a point $x^* \in X$ is said to be* `removable` *from X, iff x^* minimizes the covering radius $r_{X\setminus x,Z}$ among all points in X, i.e.*

$$r_{X\setminus x^*,Z} = \min_{x\in X} r_{X\setminus x,Z}.$$

We remark that this definition for a removable point is different from those used in [5, 6, 10], where a removable point is one which minimizes the distance to its nearest neighbour in the *current* subset X. In contrast to this, the removal criterion of Definition 2 depends also on the points in $Y = Z \setminus X$ which have already been removed in previous steps. This idea is also favourably used in the recent paper [3].

At first sight, the task of locating a removable point may look costly. The computation can, however, be facilitated by using the following characterization for removable points, which works with Voronoi diagrams [13]. To this end, recall that for any finite point set X and $x \in X$ the convex polyhedron

$$V_X(x) = \left\{ y \in \mathbb{R}^d : d_X(y) = \|y - x\| \right\}$$

denotes the *Voronoi tile* of x w.r.t. X, comprising all points in space whose nearest neighbour in X is x.

Theorem 3 *Let $X \in \mathcal{Z}$ with $|X| \geq 2$. Every point $x \in X$ which minimizes the* `local covering radius`

$$r(x) = r_{X \setminus x, Z \cap V_X(x)} \tag{12}$$

among all points in X is removable from X.

Proof: Let $Y = Z \setminus X$. Note that

$$\begin{aligned} r_{X\setminus x,Z} &= \max_{y \in Y \cup x} d_{X\setminus x}(y) \\ &= \max\left(\max_{y \in Y \setminus V_X(x)} d_{X\setminus x}(y), \max_{y \in Y \cap V_X(x)} d_{X\setminus x}(y), d_{X\setminus x}(x) \right) \\ &= \max\left(\max_{y \in Y \setminus V_X(x)} d_X(y), \max_{y \in Y \cap V_X(x)} d_{X\setminus x}(y), d_{X\setminus x}(x) \right). \end{aligned}$$

Since $d_{X\setminus x}(y) \geq d_X(y)$ for all $y \in Y \cap V_X(x)$, this implies

$$r_{X\setminus x,Z} = \max\left(\max_{y \in Y} d_X(y), \max_{y \in Y \cap V_X(x)} d_{X\setminus x}(y), d_{X\setminus x}(x) \right). \tag{13}$$

Moreover, since

$$\max\left(\max_{y \in Y \cap V_X(x)} d_{X\setminus x}(y), d_{X\setminus x}(x) \right) = \max_{y \in (Y \cap V_X(x)) \cup x} d_{X\setminus x}(y) = r_{X\setminus x, Z \cap V_X(x)}$$

and $r_{X,Z} = \max_{y \in Y} d_X(y)$, we obtain, by using (12), the equality

$$r_{X\setminus x,Z} = \max\left(r_{X,Z}, r(x)\right)$$

directly from (13). Therefore, $r_{X\setminus x,Z} \leq r_{X\setminus \tilde{x},Z}$, whenever $r(x) \leq r(\tilde{x})$ for any $x, \tilde{x} \in X$, which completes our proof. $\square$

4.2 Computational Costs of Greedy Thinning

In this subsection, the *efficient* implementation of greedy Thinning is explained and the resulting computational costs are analyzed. To this end, recall the Definition 2 for a removable point, and the characterization in Theorem 3.

Following along the lines of the discussion in the previous papers [3, 6], during the performance of the greedy Thinning algorithm the points of the current set X are stored in a heap, here and in the following called `X-heap`. Recall that a heap is a binary tree which can be used for the maintenance of a priority queue. Each node $x \in X$ in the `X-heap` bears its local covering radius $r(x)$ in (12) as its significance value. Recall that building the initial `X-heap` costs $\mathcal{O}(M \log M)$ operations [1], where $M = |Z|$ is the size of the input point set Z. Likewise, building the initial Voronoi diagram costs $\mathcal{O}(M \log M)$ operations [13].

Now due to the heap condition, the significance of a node in the `X-heap` is *smaller* than the significances of its two children. Hence, the root of the `X-heap` contains a removable point. Therefore, the point removal in steps **(2a)** and **(2b)** of the Thinning Algorithm 2 can be performed by *popping* the root of the `X-heap`. But the employed data structures, the heap and the Voronoi diagram, need to be updated accordingly. To be more precise, the steps **(2a)** and **(2b)** in each iteration of the Thinning Algorithm 2 are performed as follows.

(T1) Pop the root x^* from the heap and update the heap.

(T2) Remove x^* from the Voronoi diagram. Update the Voronoi diagram in order to obtain the Voronoi diagram of the point set $X \setminus x^*$.

(T3) Let $X = X \setminus x^*$ and so $Y = Y \cup x^*$.

(T4) Update the local covering radii of the Voronoi neighbours of x^* in X, whose Voronoi tiles were changed by the update in step **(T2)**. Update the positions of these points in the heap.

In addition, during the performance of greedy Thinning, each $y \in Y$ is attached to a Voronoi tile containing y. Thus in step **(T2)**, by the removal of x^*, the points in $Y \cap V_X(x^*)$ and x^* itself need to be *reattached* to new Voronoi tiles $V_{X \setminus x^*}(\,\cdot\,)$ of Voronoi neighbours of x^*. These (re)attachments facilitate the required updates of the local covering radii in step **(T4)**.

We remark that the updates in the above steps **(T2)** and **(T4)** require merely local operations on the Voronoi diagram. Moreover, each update in

the heap costs $\mathcal{O}(\log n)$ operations [1], where $n \leq M$ is the number of (current) nodes in the heap. Using similar arguments as in [3], this shows that each removal step of greedy Thinning costs at most $\mathcal{O}(\log M)$ operations. Since the number n of iterations in the Thinning Algorithm 2 is bounded above by M, we obtain the following result concerning the computational costs of greedy Thinning.

Theorem 4 *The performance of the Thinning Algorithm 2, by using the removal criterion of Definition 2, and according to the steps* **(T1)**-**(T4)** *requires at most* $\mathcal{O}(M \log M)$ *operations.* □

4.3 Localization of Optimal Subsets

In the remainder of this section, one useful (theoretical) property of greedy Thinning is discussed. This is concerning the selection of optimal subsets during the removal. To this end, we use the notation $T_*^n(Z) \in \mathcal{Z}$ for a subset output by greedy Thinning after n point removals. In particular, we have $|T_*^n(Z)| = M - n$ for the size of $T_*^n(Z)$, and so by Corollary 1 in Section 3 we obtain for any $n \in \{1, \ldots, M-1\}$ the adaptive bounds

$$\sigma_n \leq r_n^* \leq r_{T_*^n(Z),Z} \leq \alpha_{T_*^n(Z),Z} \cdot r_n^*$$

on the covering radius of $T_*^n(Z)$ on Z, where $\alpha_{T_*^n(Z),Z} = r_{T_*^n(Z),Z}/\sigma_n \geq 1$.

Now, if $\alpha_{T_*^n(Z),Z} = 1$, this then would directly imply that the subset $T_*^n(Z)$ is optimal with satisfying $r_{T_*^n(Z),Z} = \sigma_n = r_n^*$. For instance, at the first point removal (i.e. when $n = 1$) greedy Thinning returns the optimal subset $T_*(Z) = Z \setminus x^*$, with x^* some removable point, satisfying

$$r_{T_*(Z),Z} = r_{Z\setminus x^*,Z} = d_{Z\setminus x^*}(x^*) = \sigma(x^*) = \sigma_1.$$

But for general n, it is not true that σ_n coincides with $r_{T_*^n(Z),Z}$. In fact, this depends also on Z, which leads us to the following definition.

Definition 3 *An index* n, $1 \leq n < M$, *is said to be an* `optimal breakpoint` *for* Z, *iff there is one* $X \in \mathcal{Z}$ *of size* $|X| = M - n$ *satisfying* $r_{X,Z} = \sigma_n$.

Hence, for any Z, $n = 1$ is always an optimal breakpoint. Indeed, in this case $X = T_*(Z)$ satisfies $r_{X,Z} = \sigma_1$. But in general, i.e. for $n > 1$, it is not necessarily true that n is an optimal breakpoint for Z. Nevertheless, whenever any n is an optimal breakpoint for Z, we can show that the subset $T_*^n(Z) \subset Z$ generated by greedy Thinning is the *unique* optimum satisfying $T_*^n(Z) = \sigma_n$, provided that $\sigma_n < \sigma_{n+1}$.

Theorem 5 *Suppose n is an optimal breakpoint for Z. If $\sigma_n < \sigma_{n+1}$, then the set*

$$X_n^* = Z \setminus \{z_1, \ldots, z_n\} \in \mathcal{Z}$$

is optimal by satisfying $r_{X_n^*,Z} = \sigma_n$. Moreover, X_n^* is the unique minimizer of the covering radius $r_{X,Z}$ among all sets $X \in \mathcal{Z}$ of equal size $|X| = M - n$.

Proof: Since n is an optimal breakpoint in Z, there is at least one optimal subset $X \subset Z$ of size $|X| = M - n$ satisfying $r_{X,Z} = \sigma_n$. Let $Y = Z \setminus X$. Then, due to Theorem 2, this implies

$$\sigma(y) \leq \sigma_n \tag{14}$$

for every $y \in Y$. But since $\sigma_n < \sigma_{n+1}$ and by (7), the condition (14) is only satisfied by the points in the set $Z_n = \{z_1, \ldots, z_n\}$, and so $Y \subset Z_n$. But $|Y| = |Z_n| = n$, and therefore $Y = Z_n$, which implies $X = Z \setminus Z_n = X_n^*$. □

Corollary 2 *Suppose n is an optimal breakpoint for Z, and $\sigma_n < \sigma_{n+1}$. Then the set $T_*^n(Z)$ output by greedy Thinning is optimal by satisfying $r_{T_*^n(Z)} = \sigma_n$. Moreover, $T_*^n(Z)$ is the unique minimizer of the covering radius $r_{X,Z}$ among all sets $X \in \mathcal{Z}$ of equal size $|X| = M - n$.*

Proof: Due to Theorem 5, it is sufficient to show that $T_*^n(Z) = Z \setminus Z_n$ holds, where $Z_n = \{z_1, \ldots, z_n\}$. We prove this by induction. Since $r_{Z\setminus z,Z} = \sigma(z)$, and due to the assumption $\sigma_n < \sigma_{n+1}$, greedy Thinning removes one point from Z_n in its first step, i.e. $T_*(Z) = Z \setminus z^*$ for some $z^* \in Z_n$.

Now suppose, for any $1 \leq k < n$, that $T_*^k(Z) = Z \setminus Y$ holds with $Y \subset Z_n$. Then, on the one hand, for every $y \in Z_n \setminus Y \subset Z_n$ we have

$$r_{T_*^k(Z)\setminus y,Z} = r_{Z\setminus(Y\cup y),Z} \leq r_{Z\setminus Z_n,Z} = \sigma_n. \tag{15}$$

Indeed, this is due to the monotonicity of the covering radius, i.e.

$$r_{X,Z} \leq r_{\tilde{X},Z}, \quad \text{for all } X, \tilde{X} \in \mathcal{Z} \text{ with } \tilde{X} \subset X.$$

On the other hand, for every $z \in Z \setminus Z_n = \{z_{n+1}, \ldots, z_M\}$ we have

$$d_{T_*^k(Z)\setminus z}(z) = d_{(Z\setminus Y)\setminus z}(z) \geq d_{Z\setminus z}(z) = \sigma(z) > \sigma_n,$$

and so $r_{T_*^k(Z)\setminus z,Z} > \sigma_n$. This in combination with (15) shows that

$$r_{T_*^k(Z)\setminus y,Z} < r_{T_*^k(Z)\setminus z,Z}, \quad \text{for all } y \in Z_n \setminus Y, z \in Z \setminus Z_n.$$

Therefore, greedy Thinning removes one point from $Z_n \setminus Y \subset Z_n$ in its next step. After n removals, we have $T_*^n(Z) = Z \setminus Z_n$ as desired. □

5 Exchange

This section is devoted to the characterization of exchangeable point pairs. Moreover, the following discussion addresses important computational aspects concerning the efficient implementation of the Exchange Algorithm 1. In fact, this section provides useful criteria for an efficient localization of exchangeable point pairs, as required in step **(2a)** of Algorithm 1.

5.1 Characterization of Exchangeable Point Pairs

For the moment of the discussion in this section, $X \in \mathcal{Z}$ denotes a fixed subset of Z and we let $Y = Z \setminus X$. Moreover,

$$Y^* = \{y \in Y : d_X(y) = r_{X,Z}\}$$

stands for the set of all points $y \in Y$ where the maximum $r_{X,Z}$ is attained. The following theorem yields a necessary and sufficient condition for exchangeable point pairs. The subsequent two corollaries provide sufficient conditions, which are useful for the purpose of quickly locating exchangeable points.

Theorem 6 *A point pair $(\hat{x}, \hat{y}) \in X \times Y$ is exchangeable, if and only if all of the following three statements are true.*

(a) $r_{X,Z} > d_{X \cup \hat{y}}(y)$ *for all* $y \in Y^*$*;*

(b) $r_{X,Z} > d_{(X \setminus \hat{x}) \cup \hat{y}}(\hat{x})$*;*

(c) $r_{X,Z} > d_{(X \setminus \hat{x}) \cup \hat{y}}(y)$ *for all* $y \in Y \cap V_X(\hat{x})$*.*

Proof: Suppose all of the three statements (a),(b), and (c) are true. Note that condition (a), together with the definition for Y^*, implies

$$r_{X,Z} > d_{X \cup \hat{y}}(y) \text{ for all } y \in Y. \tag{16}$$

Moreover, for any $y \in Y \setminus V_X(\hat{x})$ we have $d_X(y) = d_{X \setminus \hat{x}}(y)$, and therefore

$$d_{X \cup \hat{y}}(y) = d_{(X \setminus \hat{x}) \cup \hat{y}}(y) \text{ for all } y \in Y \setminus V_X(\hat{x}).$$

This, in combination with statement (c) and (16), implies

$$r_{X,Z} > d_{(X \setminus \hat{x}) \cup \hat{y}}(y) \text{ for all } y \in Y. \tag{17}$$

By combining (17) with condition (b), we find

$$\begin{aligned} r_{X,Z} &> \max\left(\max_{y\in Y\setminus\hat{y}} d_{(X\setminus\hat{x})\cup\hat{y}}(y), d_{(X\setminus\hat{x})\cup\hat{y}}(\hat{x})\right) \\ &= \max_{y\in(Y\setminus\hat{y})\cup\hat{x}} d_{(X\setminus\hat{x})\cup\hat{y}}(y) \\ &= \max_{z\in Z} d_{(X\setminus\hat{x})\cup\hat{y}}(z) \\ &= r_{(X\setminus\hat{x})\cup\hat{y},Z}, \end{aligned}$$

in which case the pair $(\hat{x},\hat{y})$ is, according to Definition 1, exchangeable.

As to the converse, suppose the pair $(\hat{x},\hat{y}) \in X \times Y$ is exchangeable, i.e. $r_{X,Z} > r_{(X\setminus\hat{x})\cup\hat{y},Z}$. This implies

$$r_{X,Z} > \max_{y\in(Y\setminus\hat{y})\cup\hat{x}} d_{(X\setminus\hat{x})\cup\hat{y}}(y) \geq \max_{y\in Y\setminus\hat{y}} d_{(X\setminus\hat{x})\cup\hat{y}}(y) = \max_{y\in Y} d_{(X\setminus\hat{x})\cup\hat{y}}(y), \quad (18)$$

and therefore

$$r_{X,Z} > \max_{y\in Y^*} d_{(X\setminus\hat{x})\cup\hat{y}}(y) \geq \max_{y\in Y^*} d_{X\cup\hat{y}}(y),$$

which shows that in this case statement (a) holds. Finally, note that (18) immediately implies the statements (b) and (c), which completes our proof. □

Corollary 3 *Let $\hat{y} \in Y$ satisfy condition (a) of Theorem 6. Moreover, let $\hat{x} \in X$ satisfy $r(\hat{x}) < r_{X,Z}$. Then, the pair $(\hat{x},\hat{y}) \in X \times Y$ is exchangeable.*

Proof: Recall the expression $r(\hat{x}) = r_{X\setminus\hat{x},Z\cap V_X(\hat{x})}$ in (12) for the local covering radius of $\hat{x}$, which yields

$$\begin{aligned} r(\hat{x}) &= \max\left(\max_{y\in Y\cap V_X(\hat{x})} d_{X\setminus\hat{x}}(y), d_{X\setminus\hat{x}}(\hat{x})\right) \\ &\geq \max\left(\max_{y\in Y\cap V_X(\hat{x})} d_{(X\setminus\hat{x})\cup\hat{y}}(y), d_{(X\setminus\hat{x})\cup\hat{y}}(\hat{x})\right). \end{aligned}$$

Therefore, the assumption $r_{X,Z} > r(\hat{x})$ directly implies that the pair $(\hat{x},\hat{y})$ satisfies the conditions (b) and (c) in Theorem 6. In combination with the other assumption on $\hat{y}$, all of the three conditions (a), (b), and (c) in Theorem 6 are satisfied by $(\hat{x},\hat{y})$. Therefore, the point pair $(\hat{x},\hat{y})$ is exchangeable. □

In many situations, the set Y^* contains merely one point y^*. In this case, the point $y^* \in Y^* \subset Y$ is potentially a good candidate for an exchange,

since it satisfies the condition (a) in Theorem 6. This observation yields, by using the criterion in Corollary 3, the following sufficient condition for an exchangeable pair.

Corollary 4 *Let $y^* \in Y$ satisfy $d_X(y^*) > d_X(y)$ for all $y \in Y \setminus \{y^*\}$. Then, for $\hat{x} \in X$, the pair $(\hat{x}, y^*) \in X \times Y$ is exchangeable, if they satisfy*

$$d_X(y^*) > r(\hat{x}). \tag{19}$$

Proof: Note that the first assumption on y^* implies

$$r_{X,Z} = d_X(y^*) > d_X(y) \geq d_{X \cup y^*}(y), \quad \text{for all } y \in Y.$$

Hence, the point y^* satisfies the condition (a) of Theorem 6. Moreover, by the other assumption in (19), we obtain $r_{X,Z} = d_X(y^*) > r(\hat{x})$. But in this case, the point pair $(\hat{x}, y^*)$ is, due to Corollary 3, exchangeable. □

5.2 Computational Costs of Exchange

In this subsection, the implementation of the Exchange Algorithm 1 and the resulting computational costs are discussed. In particular, we explain how the exchange in steps **(2a)** and **(2b)** of the Exchange Algorithm 1 can be done efficiently. To this end, we merely work with the sufficient criterion of Corollary 4 for locating exchangeable point pairs.

Recall from the discussion in Section 4 that greedy Thinning works with a heap, called `X-heap`, for maintaining removable points. In the `X-heap`, the significance of a node $x \in X$ is given by the value $r(x)$ of its current local covering radius. We use this `X-heap` also for the performance of the Exchange algorithm. Note that the `X-heap` is already available when the greedy Thinning algorithm terminates, so that no additional computational costs are required for building the `X-heap`.

Moreover, during the performance of the Exchange algorithm we use another heap, called `Y-heap`, where the points of the current set $Y = Z \setminus X$ are stored. The priority of a node $y \in Y$ in the `Y-heap` is given by its distance $d_X(y)$ to the set X. The nodes in the `Y-heap` are ordered such that the significance of a node is *greater* than the significances of its two children. Hence, the root of the `Y-heap` contains a point y^* from the set Y^*, so that $d_X(y^*) = r_{X,Z}$.

We remark that the `Y-heap` may either be built immediately before the performance of the Exchange algorithm, or it may be maintained during the performance of the greedy Thinning algorithm. In either case, building

the `Y-heap` costs at most $\mathcal{O}(M \log M)$ operations. We can explain this as follows. First note that the abovementioned attachments of the points in $Y = Z \setminus X$ to corresponding Voronoi tiles (see Subsection 4.2) can be used in order to facilitate this. Indeed, by these attachments the significance $d_X(y)$ of any $y \in Y \cap V_X(x)$ is already given by the Euclidean distance between y and $x \in X$. Now since the number $|Y|$ of points in Y is at most M, and each insertion into the `Y-heap` costs at most $\mathcal{O}(\log M)$ operations, this altogether shows that we require at most $\mathcal{O}(M \log M)$ operations for building the initial `Y-heap`.

Now let us return to the performance of the steps **(2a)** and **(2b)** of the Exchange Algorithm 1. In order to locate an exchangeable pair in **(2a)**, we compare the significance $r(x^*)$ of the point x^* (the point in the root of the `X-heap`) with the significance $d_X(y^*)$ of y^* (the point in the root of the `Y-heap`). If $r(x^*) < d_X(y^*)$ and $Y^* = \{y^*\}$, then the pair $(x^*, y^*) \in X \times Y$ is, due to Corollary 4, exchangeable. Step **(2b)** of the Exchange Algorithm 1 is then accomplished as follows.

(E1) Remove x^* from X by applying greedy Thinning on X. To this end, perform the steps **(T1)**-**(T4)**, described in the previous section.

(E2) Pop the root y^* from the `Y-heap` and update the `Y-heap`.

(E3) Add the point y^* to the Voronoi diagram of the set X [1] in order to obtain the Voronoi diagram of the set $X \cup y^*$.

(E4) Update the local covering radii of those points in X, whose Voronoi tiles were modified by the insertion of y^* in step **(E3)**. Update the positions of these points in the `X-heap`.

(E5) Update the significances $d_X(y)$ of those points in Y, whose surrounding Voronoi tile was deleted by the removal of x^* in step **(T2)** or by the insertion of y^* in step **(E3)**. Reattach each of these points to a new Voronoi tile, and update their positions in the `Y-heap`.

(E6) Let $X = X \cup y^*$ and so $Y = Y \setminus y^*$.

(E7) Compute the local covering radius $r(y^*)$ of y^*, and insert y^* into the `X-heap`.

(E8) Compute the significance $d_X(x^*)$ of x^*, and insert x^* into the `Y-heap`.

[1] Note that at this stage x^* has already been removed from X by step **(T3)**.

Now let us turn to the computational costs required for *one* exchange step of the Exchange Algorithm 1. As explained above, step **(2a)** requires only $\mathcal{O}(1)$ operations, when working with the two heaps, `X-heap` and `Y-heap`. The performance of one step **(2b)**, as described by the above instructions **(E1)**-**(E8)**, can be done in at most $\mathcal{O}(\log M)$ operations, provided that each Voronoi tile contains $\mathcal{O}(1)$ points from Y. We tacitly act on this reasonable assumption from now. In this case, the required updates of the local covering radii in steps **(E1)**, **(E4)**, and **(E7)** cost only $\mathcal{O}(1)$ time. Likewise, the updates of the significances in steps **(E5)** and **(E8)** cost $\mathcal{O}(1)$ time. Finally, each update in either of the two heaps in steps **(E1)**,**(E2)**,**(E4)**,**(E5)**,**(E7)**, and **(E8)** costs at most $\mathcal{O}(\log M)$ time.

Theorem 7 *One exchange step of the Exchange Algorithm 1, by performing the instructions* **(E1)**-**(E8)**, *requires at most* $\mathcal{O}(\log M)$ *operations.* □

We finally remark that we have no (non-trivial) upper bound on the number n_E of exchange steps (required in the Exchange Algorithm 1). But in all of our numerical experiments we observed that n_E is always much smaller than the size of the input point set Z, i.e., $n_E \ll M = |Z|$. We summarize the above results concerning the computational costs of scattered data filtering by combining the Theorems 4 and 7.

Theorem 8 *For any finite point set* Z *of size* $M = |Z|$, *and* $1 \leq n < M$, *the construction of the subset* $X_n = E \circ T_*^n(Z)$ *by* n *steps of the greedy Thinning Algorithm 2 followed by* n_E *steps of the Exchange Algorithm 1 requires at most* $\mathcal{O}(M \log M) + \mathcal{O}(n_E \log M)$ *operations.* □

6 Numerical Results

We have implemented the proposed scattered data filtering scheme in two dimensions, $d = 2$, by using the Euclidean norm $\|\cdot\| = \|\cdot\|_2$. For the purpose of locating exchangeable point pairs, in step **(2a)** of Algorithm 1, we decided to merely work with the sufficient criterion in Corollary 4, as explained in the previous section. Moreover, our implementation only removes interior points, though the algorithm could easily be extended so as to remove also boundary points.

Initially, on given input set Z, the significance $\sigma(z)$ in (6) is computed for every point $z \in Z$. Then, the occurring significances (but not the points!) are sorted in increasing order, so that we obtain the sequence $\sigma_1 \leq \sigma_2 \leq \ldots \leq \sigma_M$, which is required for recording the quality indices

$\alpha_{X_n,Z} = r_{X_n,Z}/\sigma_n$, where $X_n = E \circ T_*^n(Z)$ or $X_n = T_*^n(Z)$, at run time. Note that this preprocess costs only at most $\mathcal{O}(M \log M)$ operations [1], where $M = |Z|$.

The filtering scheme was applied on two different types of scattered data,

- *clustered data* from terrain modelling (Figure 1 **(a)**);
- *track data* from marine seismic data analysis (Figure 4 **(a)**).

The numerical results on these two examples are discussed, one after the other, in the following Subsections 6.1 and 6.2. We remark that the numerical experiments were prepared on a `Sun-Fire-480R` workstation (900 MHz processor, 16384 MB physical memory).

6.1 Terrain Data

Figure 1 **(a)** shows a scattered data sample of a terrain around `Gjøvik`, Norway, comprising $M = 7928$ data points. Note that the sampling density is subject to strong variation. In fact, the data is rather sparse in flat regions of the terrain, whereas a higher sampling rate around steep gradients of the terrain's surface leads to clusters.

For the purpose of graphical illustration, Figure 1 shows also the three different subsets **(b)** $F_{2000}(Z)$, **(c)** $F_{4000}(Z)$, and **(d)** $F_{6000}(Z)$, which were generated by using the proposed filtering scheme. The resulting covering radii and the quality indices of $T_*^n(Z)$ and $F_n(Z)$, $n = 2000, 4000, 6000$, are shown in Table 1. Moreover, Table 1 shows the CPU seconds $u(T)$ which were required for computing the subsets $T_*^n(Z)$ from Z by greedy Thinning, and the CPU seconds $u(E)$ for the postprocessing exchange of point pairs. Therefore, the sum $u(F) = u(T) + u(E)$ of these values are the total costs, in terms of CPU seconds, for computing the subsets $X_n = F_n(Z)$ from Z. The numbers n_E of exchange steps are also shown in Table 1.

n	$r_{T_*^n(Z),Z}$	$r_{F_n(Z),Z}$	$\alpha_{T_*^n(Z),Z}$	$\alpha_{F_n(Z),Z}$	$u(T)$	$u(E)$	n_E
2000	3.0321	2.9744	1.3972	1.3706	1.86	0.21	71
4000	5.2241	4.6643	1.6462	1.4698	2.84	1.18	409
6000	23.9569	7.9306	5.4281	1.7969	3.74	0.81	381

Table 1: Scattered data filtering on `Gjøvik`.

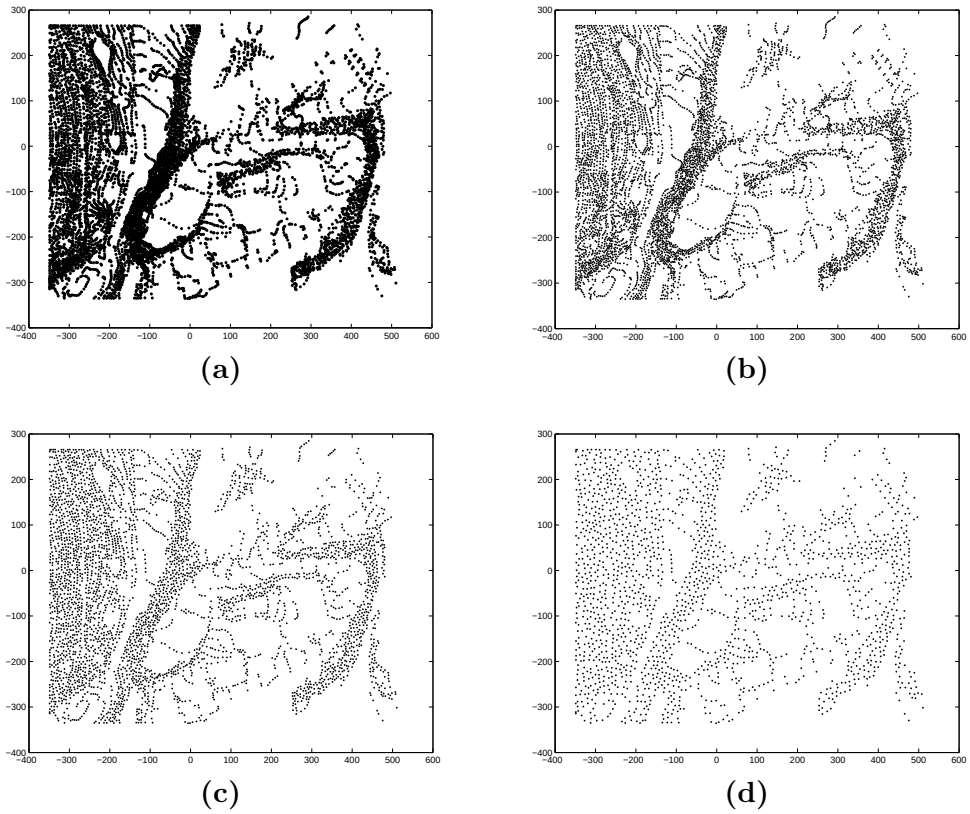


Figure 1: `Gjøvik`. **(a)** The input data set Z comprising 7928 points, and the subsets **(b)** X_{2000} of size 5928, **(c)** X_{4000} of size 3928, and **(d)** X_{6000} of size 1928, generated by scattered data filtering.

For further illustration, we have recordered the results in Table 1 for *all* possible n. The following Figures 2 and 3 reflect the results of the entire numerical experiment. The graphs of the resulting covering radii $r_{T_*^n(Z),Z}, r_{F_n(Z),Z}$ and the quality indices $\alpha_{T_*^n(Z),Z}, \alpha_{F_n(Z),Z}$, $100 \leq n \leq 7391$, are displayed in Figure 2. Figure 2 **(a)** shows also the graph of the initial significances σ_n. Recall that $\sigma_n \leq r_n^*$ by Theorem 1, i.e., the value σ_n is a lower bound for the optimal value r_n^*.

We remark that for large values of n the deviation between σ_n and the optimal value r_n^* is typically very large. For $n = M - 1$, for instance, we find $r_{M-1}^* = 516.264$ for the optimal covering radius, but $\sigma_{M-1} = 22.581$ for the penultimate significance value. This observation partly explains why

the quality indices of $\alpha_{T_*^n(Z),Z}$ and $\alpha_{F_n(Z),Z}$ in Figure 2 **(b)** are so rapidly growing for large n.

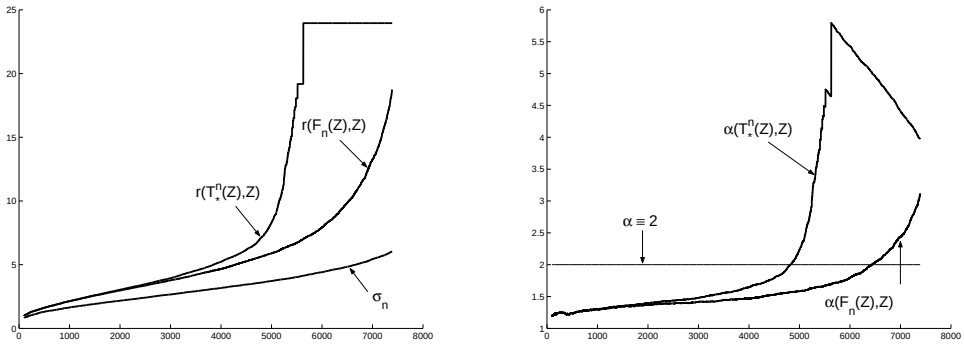


Figure 2: `Gjøvik`. **(a)** The covering radii $r_{T_*^n(Z),Z}, r_{F_n(Z),Z}$, and the significances σ_n. **(b)** The quality indices $\alpha_{T_*^n(Z),Z}$ and $\alpha_{F_n(Z),Z}$.

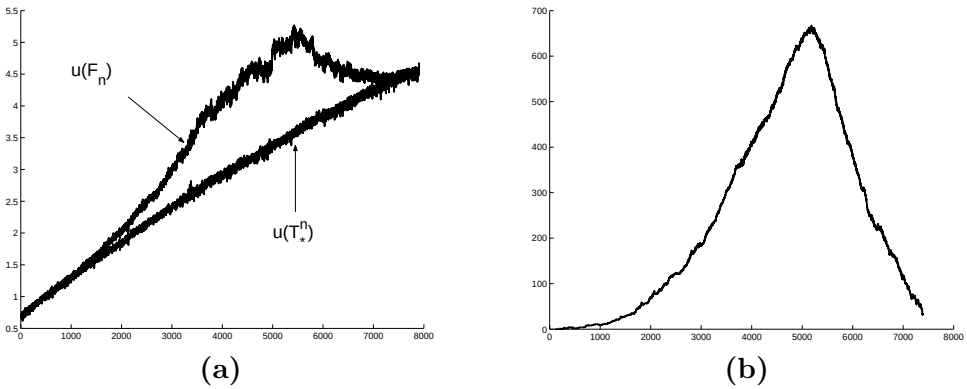


Figure 3: `Gjøvik`. **(a)** CPU seconds $u(F_n)$ required for computing $F_n(Z)$, and $u(T_*^n)$ for computing $T_*^n(Z)$; **(b)** number of exchange steps.

Nevertheless, for $n \leq 6435$, we found $\alpha(F_n(Z), Z) < 2$ and moreover, $\alpha(F_n(Z), Z) < \alpha_2 = \sqrt{2+\sqrt{3}} \approx 1.9319$ for $n \leq 6327$. We mention the latter because for the special case of the Euclidean norm, the best possible constant in (3) is $\alpha = \alpha_2$. In other words, there is for $\alpha < \alpha_2$ no α-approximation algorithm for the k-center problem, when using the Eu-

clidean norm, unless P=NP. This result is due to Feder & Greene [4] (see also [14, Section 4]).

In conclusion, the numerical results reflected by Figure 2 illustrate the good performance of the proposed filtering scheme, especially in comparison with possible α-approximation algorithms for the k-center-problem. The required seconds of CPU time and the number of exchange steps for computing the sets $X_n = F_n(Z)$ from $T_*^n(Z)$ are displayed Figures 3 **(a)** and **(b)**. Not surprisingly, we found that the CPU seconds $u(E)$ for the exchange are roughly proportional to the number n_E of exchange steps.

6.2 Track Data

In our second numerical experiment, we considered using one example from marine seismic data analysis. In this case, the spatial distribution of the sampled data is organized along *tracks*, since these data are acquired from ships. Figure 4 **(a)** shows such a seismic data set which was taken in a region of the North Sea. This data set, here referred to as `NorthSea`, comprises $M = 9758$ data points.

We have recorded the covering radii, $r_{T_*^n(Z),Z}$ and $r_{F_n(Z),Z}$, and the quality indices, $\alpha_{T_*^n(Z),Z}$ and $\alpha_{F_n(Z),Z}$, for all possible n. Figure 5 **(a)** displays the graphs of $r_{T_*^n(Z),Z}$ and $r_{F_n(Z),Z}$ along with that of the significances σ_n, whereas the graphs of $\alpha_{T_*^n(Z),Z}$ and $\alpha_{F_n(Z),Z}$, $1 \leq n \leq 8300$, are shown in Figure 5 **(b)**. Moreover, we have also recorded the elapsed CPU time required for computing $F_n(Z)$ and $T_*^n(Z)$, see Figure 6 **(a)**, as well as the number n_E of exchange steps, which are required for computing $F_n(Z)$ from $T_*^n(Z)$, see Figure 6 **(b)**.

We remark that both greedy thinning and the proposed scattered data filtering scheme perform very well on this data set. This is confirmed by the numerical results concerning the behaviour of the quality indices $\alpha_{T_*^n(Z),Z}$ and $\alpha_{F_n(Z),Z}$, see Figure 5 **(b)**. Indeed, the values $\alpha_{T_*^n(Z),Z}$ and $\alpha_{F_n(Z),Z}$ are very close to the best possible value $\alpha \equiv 1$ in the range $1083 \leq n \leq 5472$, where we find

$$1.00155 \leq \alpha_{T_*^n(Z),Z} \leq 1.00279, \qquad \text{for all } 1083 \leq n \leq 5472.$$

The quality index $\alpha_{F_n(Z),Z}$ continues to be very close to $\alpha \equiv 1$ beyond $n = 5472$, where we find

$$1.00135 \leq \alpha_{F_n(Z),Z} \leq 1.00272, \qquad \text{for all } 1083 \leq n \leq 5765.$$

Moreover, we have $\alpha_{F_n(Z),Z} < \alpha_2 = \sqrt{2+\sqrt{3}}$ for every $n \leq 6032$, and $\alpha_{F_n(Z),Z} < 2$ for every $n \leq 6908$.

The subsets $F_{5765}(Z)$ and $F_{6908}(Z)$ are shown in the Figures 4 **(b)** and **(c)**, along with the subset $F_{8112}(Z)$, which is displayed in Figure 4 **(d)**.

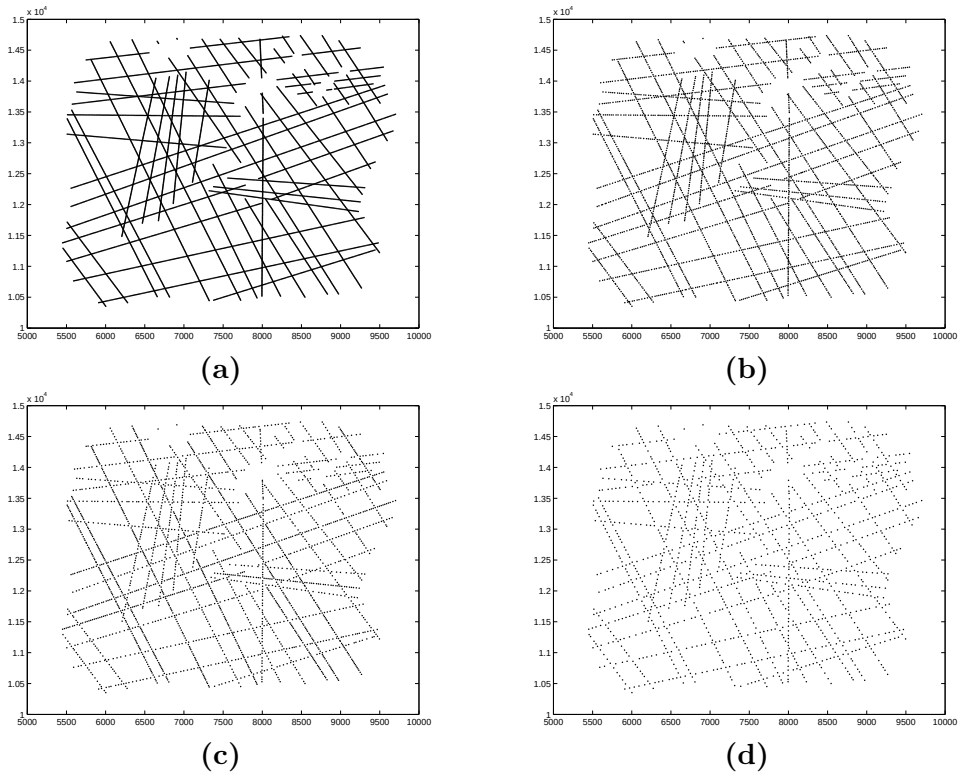


Figure 4: `NorthSea`. **(a)** The input data set comprising 9758 points, and the subsets **(b)** X_{5765} of size 3993, **(c)** X_{6908} of size 2850, and **(d)** X_{8112} of size 1646, generated by scattered data filtering.

Finally, let us spend a few remarks concerning the results in Figure 5.

Firstly, note from Figure 5 **(a)** that the significance values σ_n are almost constant for $n \geq 1083$, where we find $12.3987 = \sigma_{1083} \leq \sigma_n \leq \sigma_M = 12.5005$ for all $1083 \leq n \leq M$. This is due to the (almost) constant sampling rate of the data acquisition along the track lines. In fact, the smaller significances σ_n, for $n \leq 1082$, are attained at sample points near intersections between different track lines.

Secondly, observe from Figure 5 **(a)** the step-like behaviour of the covering radii $r_{T_*^n(Z),Z}$ and $r_{F_n(Z),Z}$. For the purpose of explaining the jumps in the graph of $r_{T_*^n(Z),Z}$, let us for the moment assume that the data contains

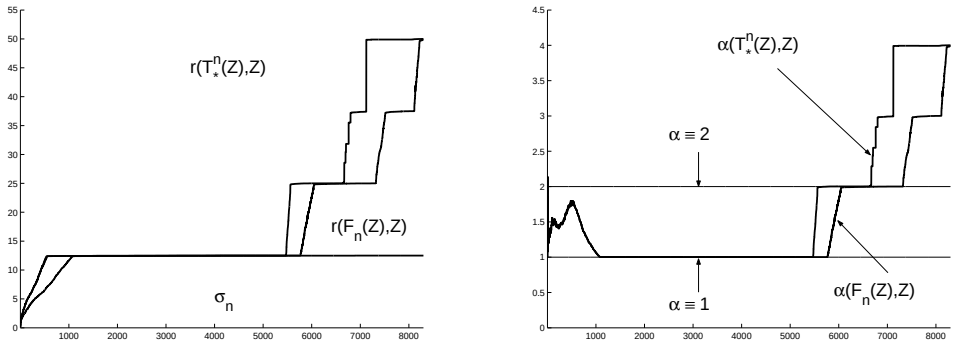
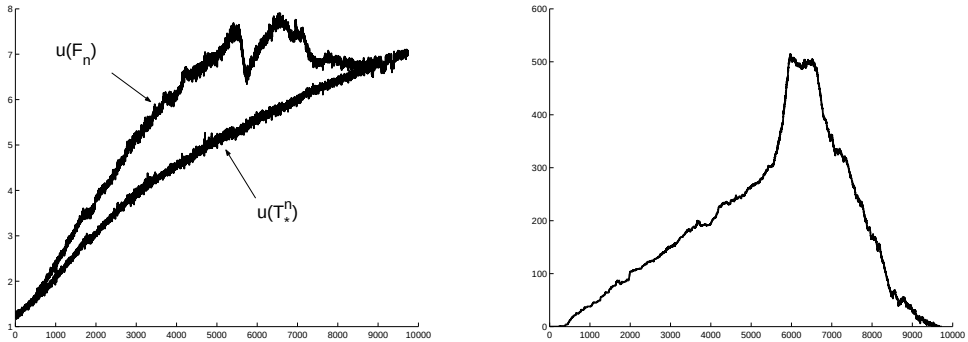


Figure 5: `NorthSea`. The graphs of **(a)** the covering radii $r_{T_*^n(Z),Z}, r_{F_n(Z),Z}$ and the significances σ_n; **(b)** the quality indices $\alpha_{T_*^n(Z),Z}$ and $\alpha_{F_n(Z),Z}$.

only *one* track line, with a constant sampling rate. In this case, the data points are *uniformly distributed* along one straight line, so that our discussion boils down to greedy Thinning on univariate data. But greedy Thinning on (uniformly distributed) univariate data is already well-understood [2]. In this case, greedy Thinning generates equidistributed subsets of points. To this end, in the beginning the algorithm prefers to remove *intermediate* points, each of whose left and right neighbour have not been removed by the algorithm, yet. Note that the covering radius is then constant. But the point removal leads, after sufficiently many steps, to a situation where the algorithm must remove a point, say x^*, in its next step, whose left and right neighbour have already been removed in previous steps. Now by the removal of x^*, the resulting covering radius will be doubled, which leads to the first jump in the graph of the covering radii. By recursion, the covering radius is kept constant for a while, before the next jump occurs at one later removal, and so on.

Now let us return to the situation of the data set `NorthSea`, which incorporates several track lines. Note that the interferences between the different track lines are rather small. In this case, the recursive point removal by greedy Thinning on the separate track lines can widely be done simultaneously. This in turn explains the jumps in the graph of the covering radii $r_{T_*^n(Z),Z}$ by following along the lines of the above arguments for the univariate case. Note that the postprocessing exchange algorithm can only delay, but not avoid, the jumps of the resulting covering radii of $r_{F_n(Z),Z}$. This also explains the step-like behaviour of the graph $r_{F_n(Z),Z}$ in Figure 5 **(a)**.

Thirdly, given the almost constant significances σ_n and the jumps in the graphs of $r_{T_*^n(Z),Z}$ and $r_{F_n(Z),Z}$, the resulting quality indices $\alpha_{T_*^n(Z),Z}$ and $\alpha_{F_n(Z),Z}$ are clearly also subject to jumps by definition, see Figure 5 **(b)**. Moreover, we remark that for large n, the differences between the significances σ_n and the optimal covering radii $r_n^*(Z)$ are very large, see Figure 5 **(a)**. For $n = M - 1$, for instance, we find $r_{M-1}^* = 2652.46$ for the optimal covering radius, but $\sigma_{M-1} = 12.5004$. In this case, albeit the adaptive bound in (10) is no longer a useful criterion for the subset selection (see the corresponding discussion immediately after Corollary 1), the proposed filtering scheme continues to generate subsets, whose sample points are uniformly distributed along the track lines. One example is given by the subset $F_{8112}(Z)$ in Figure 4 **(d)**, whose quality index is $\alpha_{F_{8112}(Z),Z} = 3.0012$.



(a) **(b)**

Figure 6: `NorthSea`. **(a)** CPU seconds $u(F_n)$ required for computing $F_n(Z)$, and $u(T_*^n)$ for computing $T_*^n(Z)$; **(b)** number of exchange steps.

Acknowledgment

The author was partly supported by the European Union within the project MINGLE (Multiresolution in Geometric Modelling), contract no. HPRN-CT-1999-00117.

References

[1] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 2nd edition, MIT Press, Cambridge, Massachusetts, 2001.

[2] N. Dyn, M.S. Floater, A. Iske, Univariate adaptive thinning, *Mathematical Methods for Curves and Surfaces: Oslo 2000*, T. Lyche and L.L. Schumaker (eds.), Vanderbilt University Press, Nashville (2001) 123–134.

[3] N. Dyn, M.S. Floater, A. Iske, Adaptive thinning for bivariate scattered data, J. Comp. Appl. Math. 145 (2002) 505–517.

[4] T. Feder, D.H. Greene, Optimal algorithms for approximate clustering, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing* (1988) 434–444.

[5] M.S. Floater, A. Iske, Multistep scattered data interpolation using compactly supported radial basis functions, J. Comp. Appl. Math. 73 (1996) 65–78.

[6] M.S. Floater, A. Iske, Thinning algorithms for scattered data interpolation, BIT 38 (1998) 705–720.

[7] D.S. Hochbaum (ed.), Approximation Algorithms for NP-hard Problems, PWS Publishing Company, Boston, 1997.

[8] D.S. Hochbaum, D.B. Shmoys, A best possible heuristic for the k-center problem, Mathematics of Operations Research 10 (1985) 180–184.

[9] A. Iske, Reconstruction of smooth signals from irregular samples by using radial basis function approximation, *Proceedings of the 1999 International Workshop on Sampling Theory and Applications*, Y. Lyubarskii (ed.), The Norwegian University of Science and Technology, Trondheim (1999) 82–87.

[10] A. Iske, Hierarchical scattered data filtering for multilevel interpolation schemes, *Mathematical Methods for Curves and Surfaces: Oslo 2000*, T. Lyche and L.L. Schumaker (eds.), Vanderbilt University Press, Nashville (2001) 211–221.

[11] A. Iske, Scattered data modelling using radial basis functions, *Tutorials on Multiresolution in Geometric Modelling*, A. Iske, E. Quak, and M.S. Floater (eds.), Springer-Verlag, Heidelberg (2002) 205–242.

[12] O. Kariv, S.L. Hakimi, An algorithmic approach to network location problems, part I: the p-centers, SIAM J. Appl. Math 37:3 (1979) 513–538.

[13] F.P. Preparata, M.I. Shamos, Computational Geometry, 2nd edition, Springer, New York, 1988.

[14] D.B. Shmoys, Computing near-optimal solutions to combinatorial optimization problems, DIMACS, Ser. Discrete Math. Theor. Comput. Sci. 20 (1995), 355–397.

Armin Iske
Zentrum Mathematik
Technische Universität München
D-85747 Garching, GERMANY
iske@ma.tum.de