

Efficient Algorithms for Least Square Piecewise Polynomial Regression

Daniel Lokshtanov ✉

Department of Computer Science, University of California, Santa Barbara, CA, USA

Subhash Suri ✉

Department of Computer Science, University of California, Santa Barbara, CA, USA

Jie Xue ✉

Department of Computer Science, University of California, Santa Barbara, CA, USA

Abstract

We present approximation and exact algorithms for piecewise regression of univariate and bivariate data using fixed-degree polynomials. Specifically, given a set S of n data points $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathbb{R}$ where $d \in \{1, 2\}$, the goal is to segment $\mathbf{x}_i$'s into some (arbitrary) number of disjoint pieces $P_1, \dots, P_k$, where each piece P_j is associated with a fixed-degree polynomial $f_j : \mathbb{R}^d \rightarrow \mathbb{R}$, to minimize the total loss function $\lambda k + \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2$, where $\lambda \geq 0$ is a regularization term that penalizes model complexity (number of pieces) and $f : \bigsqcup_{j=1}^k P_j \rightarrow \mathbb{R}$ is the *piecewise polynomial* function defined as $f|_{P_j} = f_j$. The pieces $P_1, \dots, P_k$ are disjoint intervals of $\mathbb{R}$ in the case of univariate data and disjoint axis-aligned rectangles in the case of bivariate data. Our error approximation allows use of any fixed-degree polynomial, not just linear functions.

Our main results are the following. For univariate data, we present a $(1 + \varepsilon)$ -approximation algorithm with time complexity $O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$, assuming that data is presented in sorted order of x_i 's. For bivariate data, we present three results: a sub-exponential exact algorithm with running time $n^{O(\sqrt{n})}$; a polynomial-time constant-approximation algorithm; and a quasi-polynomial time approximation scheme (QPTAS). The bivariate case is believed to be NP-hard in the folklore but we could not find a published record in the literature, so in this paper we also present a hardness proof for completeness.

2012 ACM Subject Classification Theory of computation $\rightarrow$ Computational geometry

Keywords and phrases regression analysis, piecewise polynomial, least square error

Digital Object Identifier 10.4230/LIPIcs.ESA.2021.63

1 Introduction

Line, or curve, fitting is a classical problem in statistical regression and data analysis, where the goal is to find a simple predictive model that best fits an observed data set. For instance, given a set of two-dimensional points (x_i, y_i) , $i = 1, \dots, n$, the least-square line fitting problem is to find a linear function $f : y = ax + b$ minimizing the cumulative error $\sum_{i=1}^n (y_i - (ax_i + b))^2$. This problem is easily solved in $O(n)$ time because the coefficients of the optimal line have a simple closed form solution in terms of input data. In most cases, however, a single line is a poor fit for the data, and instead the goal is to segment the data into multiple piece, with each piece represented by its own linear function. This problem of poly-line (or piecewise linear) fitting has been studied widely in computational geometry, where the goal is either to minimize the total error *for a given number of pieces* [8, 10], or to minimize the number of pieces for a given upper bound on the error [8], under a variety of error measures. In a related but technically different vein of work on “curve simplification”, the approximation must also form a polygonal chain – that is, the pieces representing neighboring segments must form a continuous curve, and it is conjectured that finding a polygonal chain of k pieces with minimum L_2 error is NP-hard [8]. In our regression setting, such continuity is not required.



© Daniel Lokshtanov, Subhash Suri, and Jie Xue;
licensed under Creative Commons License CC-BY 4.0

29th Annual European Symposium on Algorithms (ESA 2021).

Editors: Petra Mutzel, Rasmus Pagh, and Grzegorz Herman; Article No. 63; pp. 63:1–63:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

These best-fit formulations with a “hard-coded” value for the number of pieces k , however, suffer from the problem of having to specify k , rather than letting the structure in the data dictate the choice. This can be circumvented by running the algorithm for multiple values of k , and then stopping with the smallest number of pieces with an *acceptable* error. A significant issue, however, is the inherent tradeoff between the number of pieces and the error – the larger number of pieces, the smaller the error – which is recognized as the problem of “overfitting” in statistics and machine learning. In order to avoid this overfitting problem, regression typically uses “regularization” and includes a penalty term for the *size* of the representation (model) in the objective, often called the “loss” function. By optimizing the loss function, the algorithm automatically balances the two competing criteria: number of pieces k and approximation error.

In particular, suppose we have a set of data points $(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, for $i = 1, \dots, n$. We call $(\mathbf{x}_i, y_i)$ univariate data if $d = 1$ and bivariate if $d = 2$. We will consider piecewise approximation of these data points using polynomial functions of any fixed degree g , where linear functions are the special case when the degree is one. Our goal is to segment $\mathbf{x}_i$ ’s into some (arbitrary) number of disjoint pieces $P_1, \dots, P_k$, each associated with a constant-degree polynomial function f_j , to minimize the total *loss function*

$$\lambda k + \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2,$$

where $\lambda > 0$ is a pre-specified penalty term for regularizing the model complexity (number of pieces) and $f : \bigsqcup_{j=1}^k P_j \rightarrow \mathbb{R}$ is the *piecewise polynomial* function defined as $f|_{P_j} = f_j$. The pieces $P_1, \dots, P_k$ are disjoint intervals in $\mathbb{R}$ in the case of univariate data and are disjoint axis-aligned rectangles in $\mathbb{R}^2$ in the case of bivariate data.

Even for piecewise linear approximation of univariate data, the best bound currently known is $\Omega(kn^2)$ [2, 9, 15], and it is an important open problem to either find a sub-quadratic algorithm or prove a $\Omega(n^2)$ lower bound. We make progress on this problem by presenting a linear-time approximation scheme for this problem.

► **Theorem 1.** *There exists a $(1 + \varepsilon)$ -approximation algorithm for univariate piecewise polynomial regression which runs in $O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ time (excluding the time for pre-sorting).*

For bivariate data, we obtain the following results, including a sub-exponential exact algorithm, a constant-factor approximation in polynomial time, and a quasi-polynomial approximation scheme (QPTAS).

► **Theorem 2.** *There exists an exact algorithm for bivariate piecewise polynomial regression which runs in $n^{O(\sqrt{n})}$ time.*

► **Theorem 3.** *There exists a constant-factor approximation algorithm for bivariate piecewise polynomial regression which runs in polynomial time.*

► **Theorem 4.** *There exists a QPTAS for bivariate piecewise polynomial regression.*

Finally, while the bivariate case (and hence the case of more than two variables) is believed to be NP-hard in the folklore, we could not find a published record in the literature, so we also present a hardness proof for completeness.

► **Theorem 5.** *Bivariate piecewise regression is NP-hard for all fixed degree polynomials, including piecewise constant or piecewise linear functions.*

Related work. Curve fitting and piecewise regression related problems have been studied in computational geometry [6, 8], statistics [16] and machine learning [1, 9] as well as in database theory under the name *histogram approximation* [11, 14]. The main focus of research in computational geometry has been to approximate a curve, or a set of points sampled from a curve, by a fixed-size polygonal chain to minimize some measure of error, such as L_1, L_2, L_∞ error or Hausdorff error. For instance, Goodrich [10] presented an $O(n \log n)$ -time algorithm to compute a polyline (or a connected piecewise linear function) in the plane that minimizes the maximum vertical distance from a set of n points to the polyline, which improves upon the algorithms of [12, 17]. Aronov et al. present an FPTAS for the polyline fitting problem with the min-sum and least-square error measure, and conjecture that finding a polygonal chain of k pieces with minimum L_2 error is NP-hard [8]. Agarwal et al. [6] consider approximation under Hausdorff and Frechet distances.

Unlike these computational geometric models, in statistics, machine learning and database theory, the piecewise approximation is typically *not* required to be “connected”; instead, the goal is to partition the data into a given number k of pieces, each represented by a simple function. Such an optimal histogram (piecewise approximation) can be constructed in $O(kn^2)$ time using dynamic programming, where k is the number of pieces [11, 14]. A similar dynamic programming algorithm can also compute an optimal “regularized” piecewise approximation, where k is the number of pieces in the optimal solution [15]. It is an important open problem to either find a sub-quadratic algorithm or prove a $\Omega(n^2)$ lower bound.

In machine learning, Acharya et al. [2] study a “segmented regression” problem where the goal is to recover a function f , which is promised to be “nice” (say, piecewise linear with k pieces), and the sampled data from f has a small random noise. The quality of recovery is measured by the mean squared error. In this model, they present an algorithm for computing a function with $O(k)$ linear pieces in $O(n \log n)$ time [2]. An extension to multi-dimensional data with similar results is presented in [9]. Our focus is a little different from these results because (1) we do not assume a fixed value of k , and (2) we judge the error of our regression against worst-case input that is not necessarily drawn from a hypothetical k -piece input with small random noise. Thus, these two lines of research are complementary.

Finally, for bivariate data, Agarwal and Suri [7] considered the problem of computing a piecewise linear surface with smallest number of pieces whose vertical distance from data points is at most ε . They showed that the problem is NP-hard and gave a polynomial-time $O(\log n)$ -approximation algorithm.

Organization. Section 2 introduces some basic notations and concepts used throughout the paper. Our linear-time approximation scheme for univariate data (Theorem 1) is presented in Section 3, while our algorithms for bivariate data are presented in Section 4. Finally, in Section 5, we conclude the paper and pose some open questions. Due to limited space, the algorithm of Theorem 2 and the hardness result of Theorem 5 (as well as some proofs and details) are omitted in this version, which will appear in the full paper.

2 Basic notations and concepts

We begin with basic notation and concepts that are used throughout the paper. For an integer $g \geq 0$, we use $\mathbb{R}[x]_g$ and $\mathbb{R}[x, x']_g$ to denote the family of all univariate and bivariate polynomial functions with degree at most g . A univariate (resp., bivariate) *piecewise polynomial function* of degree at most g is a function $f : \bigsqcup_{j=1}^k P_j \rightarrow \mathbb{R}$, where $P_1, \dots, P_k$ are disjoint intervals in $\mathbb{R}^1$ (disjoint axis-parallel rectangles in $\mathbb{R}^2$) and $f|_{P_j} = f_j|_{P_j}$ for some

$f_j \in \mathbb{R}[x]_g$ (resp., $f_j \in \mathbb{R}[x, x']_g$), for all $j \in \{1, \dots, k\}$. The intervals (resp., rectangles) $P_1, \dots, P_k$ are the *pieces* of f , and the number k is the *complexity* of f , denoted by $|f|$. The notion of piecewise polynomial functions generalizes to higher dimensions (multi-variables), where the pieces becomes axis-parallel boxes but in this paper we only study univariate and bivariate piecewise polynomial functions.

Let Γ_g^d denote the family of piecewise polynomial functions with d variables and of degree at most g . For a set of n points $S = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}\}_{i=1}^n$, we define the *error* of a function $f \in \Gamma_g^d$ for S as

$$\sigma_S(f) = \lambda \cdot |f| + \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2,$$

where $\lambda > 0$ is a pre-specified (regularizer) parameter. We assume $\sigma_S(f) = \infty$ if the domain of f does not cover all $\mathbf{x}_i$'s. For a fixed constant g , the *piecewise polynomial regression* problem takes S and λ as the input, and aims to find the function $f^* \in \Gamma_g^d$ that minimizes the error $\sigma_S(f^*)$. By appropriate scaling of the y -values in the input, we can assume without loss of generality that $\lambda = 1$. Therefore, for convenience, we make this assumption throughout the paper.

3 Algorithm for univariate data

The input to the univariate regression problem is a dataset $S = \{(x_i, y_i) \in \mathbb{R} \times \mathbb{R}\}_{i=1}^n$, where $x_1 \leq \dots \leq x_n$, and the goal is to find the function $f^* \in \Gamma_g^1$ minimizing $\sigma_S(f^*)$, for some fixed constant $g \geq 0$, where we assume $\lambda = 1$, as mentioned earlier. This problem can be solved in $O(n^2)$ time with a straightforward dynamic program, and no subquadratic-time (even approximation) algorithm is known. Our main result in this section is a linear-time approximation scheme, which for any $\varepsilon > 0$ computes in $O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ time a piecewise function $f \in \Gamma_g^1$ whose error is at most $(1 + \varepsilon) \cdot \text{opt}$, assuming that the points in S are pre-sorted by their x -coordinates.

In order to explain the main ideas behind our algorithm, it is helpful to first briefly review the quadratic-time dynamic programming algorithm. That algorithm performs n iterations, where the i th iteration computes an optimal piecewise regression for the subset of points $(x_1, y_1), \dots, (x_i, y_i)$. If the rightmost piece in the optimal solution for this subproblem covers the points $(x_j, y_j), \dots, (x_i, y_i)$, then the solution combines the optimal regression for $(x_1, y_1), \dots, (x_{j-1}, y_{j-1})$ with the best fitting degree g polynomial for $(x_j, y_j), \dots, (x_i, y_i)$. By dynamic programming, the former is already computed in the $(j-1)$ th iteration, and the latter can be computed for all subproblems with an $O(n^2)$ -time preprocessing step. There are $O(i)$ candidates for the rightmost piece, and so the i th iteration takes $O(i)$ time, resulting in an $O(n^2)$ time algorithm.

A natural idea for improving the dynamic program's time complexity is to reduce the number of guesses needed for the rightmost piece in each iteration: ideally, we would like to find the “best” rightmost piece without trying all possibilities. This, however, seems quite difficult if we want the exact optimal solution. Our main idea is to show that this is possible if we only need a $(1 + \varepsilon)$ approximation of the minimum error. Our algorithm builds on three key steps. First, we prove a structural lemma (Lemma 7) showing that there exists an approximate solution f in which the squared error of each piece (essentially) is bounded by $O(1/\varepsilon)$, and therefore contributes between 1 and $1 + O(1/\varepsilon)$ to the final objective $\sigma_S(f)$. The second key idea is to show that, for each $i \in [n]$, there exist a set of $O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon})$ “candidate” pieces with right endpoint x_i such that a $(1 + \varepsilon)$ -approximate solution can be found using

only these pieces (Lemma 8). Thus, assuming that these candidate pieces and their best fit degree g polynomials are known, we only have to make $O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon})$ guesses in each iteration, which leads to an $O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ -time algorithm. The final, and third, step is to compute all the candidate pieces efficiently, which we show can be done using prefix sum and the standard formula for least-square polynomial regression—the details of this part will appear in the full paper.

With this preamble, we are ready to describe our algorithm in detail. For $a, b \in [n]$ satisfying $a \leq b$, we define

$$f[a, b] = \arg \min_{f \in \mathbb{R}[x]_g} \sum_{i=a}^b (y_i - f(x_i))^2 \quad \text{and} \quad \delta[a, b] = \min_{f \in \mathbb{R}[x]_g} \sum_{i=a}^b (y_i - f(x_i))^2.$$

That is, $f[a, b]$ is the best-fit polynomial in $\mathbb{R}[x]_g$ for the set of points $(x_a, y_a), \dots, (x_b, y_b)$ (in terms of square error) and $\delta[a, b]$ is the square error of $f[a, b]$. We have the following simple observation.

► **Lemma 6.** *If $a' \leq a$ and $b' \geq b$, then $\delta[a', b'] \geq \delta[a, b]$. Furthermore, for a sequence of numbers $a_0, a_1, \dots, a_r$ where $a-1 \leq a_0 < \dots < a_r \leq b$, we have $\delta[a, b] \geq \sum_{j=1}^r \delta[a_{j-1}+1, a_j]$.*

Let ε be the approximation factor, which we assume is sufficiently small, say $0 < \varepsilon \leq 1$. Let $\tilde{\varepsilon} > 0$ be such that it satisfies $(1 + \tilde{\varepsilon})^2 = 1 + \varepsilon$. Then, we have $\varepsilon/3 \leq \tilde{\varepsilon} \leq \varepsilon$ since $\varepsilon \leq 1$. For an index $i \in [n]$, we say i is a *left* (resp., *right*) *break point* if $x_{i-1} < x_i$ (resp., $x_{i+1} > x_i$). For a function $f \in \Gamma_g^1$ and a piece P of f , the *cost* of P is defined as $\sum_{x_i \in P} (y_i - f(x_i))^2$. Thus, the total error $\sigma_S(f)$ is simply $|f|$ plus the cost of all the pieces of f .

► **Lemma 7.** *There exists a function $f \in \Gamma_g^1$ such that $\sigma_S(f) \leq (1 + \tilde{\varepsilon}) \cdot \text{opt}$ and each piece of f either has cost at most $2/\tilde{\varepsilon}$ or is a singleton point.*

Proof. Let $f^* \in \Gamma_g^1$ be an optimal solution, and so $\sigma_S(f^*) = \text{opt}$. Consider a piece of f^* , say, $P^* = [x_a, x_b]$ where a is a left break point and b is a right break point and $a, b \in [n]$. Since f^* is optimal, the cost of P^* is $\delta[a, b]$. We replace P^* with $r < \tilde{\varepsilon} \cdot \delta[a, b] + 1$ pieces $P_1, \dots, P_r$ as follows. We say a pair (a', a'') of indices with $a' \leq a''$ is *legal* if $x_{a'} = x_{a''}$ or $\delta[a', a''] \leq 2/\tilde{\varepsilon}$. Starting with $a_0 = a - 1$, we create a sequence $a_0, a_1, a_2, \dots$ of indices, where a_{i+1} is the largest right break point in $\{a_i + 1, \dots, b\}$ such that $(a_i + 1, a_{i+1})$ is legal. The sequence ends at some $a_r = b$, and we claim that $r < \tilde{\varepsilon} \cdot \delta[a, b] + 1$. We first observe that since a_{i+1} is the largest right break point for which $(a_i + 1, a_{i+1})$ is legal, we have $\delta[a_i + 1, a_{i+2}] > 2/\tilde{\varepsilon}$ for all $i \in \{0, 1, \dots, r-2\}$. Now consider the sum $\sum_{i=0}^{\lfloor r/2 \rfloor - 1} \delta[a_{2i} + 1, a_{2(i+1)}]$. Each summand of this sum is greater than $2/\tilde{\varepsilon}$. On the other hand, we have $\delta[a, b] \geq \sum_{i=0}^{\lfloor r/2 \rfloor - 1} \delta[a_{2i} + 1, a_{2(i+1)}]$ by Lemma 6. It directly follows that $\lfloor r/2 \rfloor < \tilde{\varepsilon} \cdot \delta[a, b]/2$ and hence $r < \tilde{\varepsilon} \cdot \delta[a, b] + 1$. We define $P_i = [x_{a_{i-1}+1}, x_{a_i}]$ for $i \in [r]$. We replace P^* of f^* with $P_1, \dots, P_r$, and call them the *sub-pieces* of P^* . We do this for all pieces of f^* , which gives us our function $f \in \Gamma_g^1$, as follows. First, clearly, the domain of f is contained in the domain of f^* . Next, for each piece $P = [x_a, x_b]$ of f , the function $f|_P$ is simply the polynomial $f[a, b]$ restricted to P , whose cost is $\delta[a, b]$. All that remains is to bound the total error $\sigma_S(f)$. Consider a piece $P^* = [x_a, x_b]$ of f^* and its sub-pieces $P_1, \dots, P_r$. Let $c(P^*)$ be the total cost of all the sub-pieces $P_1, \dots, P_r$ plus r . By Lemma 6, the total cost of all the sub-pieces $P_1, \dots, P_r$ is at most $\delta[a, b]$, and since $r < \tilde{\varepsilon} \cdot \delta[a, b] + 1$ and $c^*(P^*) = \delta[a, b] + 1$, we get $c(P^*) \leq (1 + \tilde{\varepsilon}) \cdot c^*(P^*)$. This inequality holds for each piece of f^* , and so we get our result that $\sigma_S(f) \leq (1 + \tilde{\varepsilon}) \cdot \sigma_S(f^*)$. ◀

For convenience, we say a function $f \in \Gamma_g^1$ is *S-light* if each piece of f is either a singleton point or of cost at most $2/\tilde{\varepsilon}$. Similarly, for a subset $S' \subseteq S$, we say a function $f \in \Gamma_g^1$ is *S'-light* if each piece of f is either a singleton point or of cost *with respect to* S' (i.e., the sum of only the square error of the points in S') at most $2/\tilde{\varepsilon}$.

For a right break point $b \in [n]$ and an integer $i \geq 0$, let $a_i(b) \in [b]$ be the smallest left break point such that $\delta[a_i(b), b] \leq (1 + \varepsilon)^i - 1$; if such a left break point does not exist, we set $a_i(b)$ to be the largest left break point that is smaller than or equal to b . We define an index set $A(b) = \{a_i(b) : i \geq 0 \text{ and } (1 + \varepsilon)^{i-1} - 1 \leq 2/\varepsilon\}$. We say an interval I is *canonical* if $I = [x_a, x_b]$ for some $a, b \in [n]$ such that b is a right break point and $a \in A(b)$. A function $f \in \Gamma_g^1$ is *canonical* if all pieces of f are canonical intervals. The following lemma shows that we can limit our search to canonical functions.

► **Lemma 8.** *There exists a canonical function $f \in \Gamma_g^1$ such that $\sigma_S(f) \leq (1 + \varepsilon) \cdot \text{opt}$.*

Proof. We claim that for any S -light function $f_0 \in \Gamma_g^1$, there exists a canonical function $f \in \Gamma_g^1$ with $\sigma_S(f) \leq (1 + \varepsilon) \cdot \sigma_S(f_0)$. This claim in combination with Lemma 7 proves the lemma. We prove the claim using induction on the number r of distinct x -coordinates of the points in S , i.e., distinct elements in $\{x_1, \dots, x_n\}$. If $r = 1$, then $x_1 = \dots = x_n$ and the interval $I = [x_1, x_n]$ is a singleton point. Furthermore, in this case, 1 is the unique left break point, hence $1 \in A(n)$ and I is canonical. Therefore, the claim clearly holds. Assume that the claim holds if the number of distinct x -coordinates of the points in S is less than r , and consider the case where the number is r . Let $f_0 \in \Gamma_g^1$ be a S -light function, and we want to show that there exists a canonical function $f \in \Gamma_g^1$ such that $\sigma_S(f) \leq (1 + \varepsilon) \cdot \sigma_S(f_0)$. Consider the rightmost piece P of f_0 . Without loss of generality, we may assume that $P = [x_a, x_n]$ for some left break point $a \in [n]$. Let $c(P)$ be the cost of P . We consider two cases, $c(P) \leq 2/\varepsilon$ and $c(P) > 2/\varepsilon$. If $c(P) \leq 2/\varepsilon$, we define i as the smallest integer such that $(1 + \varepsilon)^i \geq c(P) + 1$. Therefore, $(1 + \varepsilon)^{i-1} \leq c(P) + 1 \leq (1 + \varepsilon)^i$. Since $c(P) \leq 2/\varepsilon$, we have $(1 + \varepsilon)^{i-1} - 1 \leq 2/\varepsilon$ and hence $a_i(n) \in A(n)$. By the definition of $a_i(n)$, we have $a_i(n) \leq a$ and $\delta[a_i(n), n] \leq (1 + \varepsilon)^i - 1$, i.e., $\delta[a_i(n), n] + 1 \leq (1 + \varepsilon)^i$. Since $(1 + \varepsilon)^{i-1} \leq c(P) + 1$, we further deduce that $\delta[a_i(n), n] + 1 \leq (1 + \varepsilon) \cdot (c(P) + 1)$. Now we define $S' = \{(x_1, y_1), \dots, (x_{a-1}, y_{a-1})\} \subseteq S$ and $S'' = \{(x_1, y_1), \dots, (x_{a_i(n)-1}, y_{a_i(n)-1})\} \subseteq S$. Let $f'_0 \in \Gamma_g^1$ be the function obtained by restricting f_0 to the union of the pieces other than P . Then f'_0 is both S' -light and S'' -light. Note that the number of distinct x -coordinates of the points in S'' is strictly less than r , as $a_i(n)$ is a left break point. Therefore, by our induction hypothesis, there exists some canonical function $f'' \in \Gamma_g^1$ with $\sigma_{S''}(f'') \leq (1 + \varepsilon) \cdot \sigma_{S''}(f'_0) \leq (1 + \varepsilon) \cdot \sigma_{S'}(f'_0)$, and we can assume without loss of generality that all pieces of f'' are contained in the range $(-\infty, x_{a_i(n)-1}]$. We define our function f as the “combination” of f'' and $f[a_i(n), n]$. Specifically, the pieces of f consists of all pieces of f'' and the interval $[x_{a_i(n)}, x_n]$. On the piece $[x_{a_i(n)}, x_n]$, f is the same as $f[a_i(n), n]$. On the other pieces, f is the same as f'' . Clearly, $f \in \Gamma_g^1$, and it is canonical because f'' is canonical and $[x_{a_i(n)}, x_n]$ is a canonical interval. Finally, we have

$$\begin{aligned} \sigma_S(f) &= \sigma_{S''}(f'') + \delta[a_i(n), n] + 1 \\ &\leq (1 + \varepsilon) \cdot \sigma_{S'}(f'_0) + (1 + \varepsilon) \cdot (c(P) + 1) \\ &= (1 + \varepsilon) \cdot \sigma_S(f_0). \end{aligned}$$

In the case $c(P) > 2/\varepsilon$, P must be a singleton point as f_0 is S -light. Thus, $x_a = x_n$ and a is the largest left break point smaller than or equal to n , which implies $a_0(n) = a$ and hence P is canonical. By our induction hypothesis, there exists some canonical function $f'' \in \Gamma_g^1$ with $\sigma_{S'}(f'') \leq (1 + \varepsilon) \cdot \sigma_{S'}(f'_0)$, where $S' = \{(x_1, y_1), \dots, (x_{a-1}, y_{a-1})\}$. Without loss of generality, we may assume all pieces of f'' are contained in the range $(-\infty, x_{a-1}]$. Similarly to the above, We define f as the combination of f'' and $f[a, n]$. Since $\sigma_{S'}(f'') \leq (1 + \varepsilon) \cdot \sigma_{S'}(f'_0)$ and the cost of P is at least $\delta[a, n]$, we have $\sigma_S(f) \leq (1 + \varepsilon) \cdot \sigma_S(f_0)$. ◀

We can find a canonical function $f \in \Gamma_g^1$ minimizing $\sigma_S(f)$ using dynamic programming, as shown in Algorithm 1. By Lemma 8, the result is a $(1 + \varepsilon)$ -approximation of the univariate regression problem.

■ **Algorithm 1** APPROXIMATE-REGRESSION-1D(S).

```

1:  $t \leftarrow 0$  and  $\text{opt}_0 \leftarrow 0$ 
2: for  $t$  from 1 to  $n$  do
3:   if  $t$  is a right break point then
4:      $\tilde{a} \leftarrow \arg \min_{a \in A(t)} \{\text{opt}_{a-1} + (\delta[a, t] + 1)\}$ 
5:      $\text{opt}_t \leftarrow \text{opt}_{\tilde{a}-1} + (\delta[\tilde{a}, t] + 1)$ 
6: return  $\text{opt}_n$ 

```

The correctness of Algorithm 1 is clear. To analyze its time complexity, we observe that $|A(b)| = O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon})$ for all right break points $b \in [n]$. Therefore, assuming that we know all the index sets $A(b)$ and all the $f[a, b]$ and $\delta[a, b]$, where $a \in A(b)$, Algorithm 1 can be directly implemented in $O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ time. The details of how to compute all $A(b)$ and all $f[a, b], \delta[a, b]$, where $a \in A(b)$, in $O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ time will appear in the full paper. The following theorem states the main result of this section.

► **Theorem 1.** *There exists a $(1 + \varepsilon)$ -approximation algorithm for univariate piecewise polynomial regression which runs in $O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ time (excluding the time for pre-sorting).*

4 Algorithms for bivariate data

In this section, we present our algorithms for piecewise polynomial regression for bivariate data. The input of the problem is a dataset $S = \{(x_i, x'_i), y_i\} \in \mathbb{R}^2 \times \mathbb{R}\}_{i=1}^n$, and our goal is to find a function $f^* \in \Gamma_g^2$ that minimizes $\sigma_S(f^*)$ (recall that $\lambda = 1$ by assumption).

We present three algorithms for this problem. The first is a polynomial-time constant-factor approximation. This is the simplest of the three results. The second algorithm computes the exact solution in sub-exponential time $n^{O(\sqrt{n})}$, which makes use of the planar separator theorem (this one will appear in the full paper). The third result is a quasi-polynomial time approximation scheme, and is technically the most sophisticated of the three algorithms.

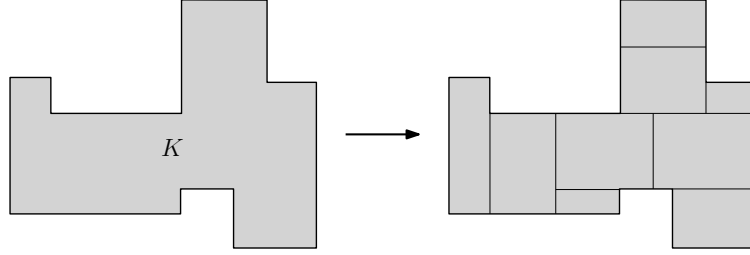
We begin with a brief overview of the high-level ideas underlying our algorithms. We first observe that a piecewise function corresponds to an orthogonal partition of the plane (induced by the pieces of the function). Therefore, the problem of finding the optimal function $f^* \in \Gamma_g^2$ is (essentially) equivalent to computing an optimal orthogonal partition of the plane (Lemma 9). Our constant-approximation algorithm (Section 4.1) follows easily from the observation that there always exists a *binary* orthogonal partition whose “cost” is a constant factor of the optimal solution (Lemma 11), and we can compute such a partition in polynomial-time using dynamic programming. To obtain our subexponential-time algorithm, we observe that an orthogonal partition of the plane forms a planar graph, and so we can use a divide-and-conquer approach by utilizing balanced separators of this graph. Finally, our QPTAS (Section 4.2) is more complicated. It is also based on a planar separator theorem, together with a cutting lemma (Lemma 13) of [3]. The basic idea is to guess a balanced separator of the planar graph of the cutting and do divide-and-conquer. We then carefully analyze the quality of the solution computed by this divide-and-conquer process (Lemma 15 and Corollary 16), and show it is indeed a $(1 + \varepsilon)$ -approximation.

We begin with introducing some notations and concepts. Let $\Delta > 0$ be a sufficiently small number such that $3\Delta \leq |x_i - x_j|$ for all $i, j \in [n]$ with $x_i \neq x_j$ and $3\Delta \leq |x'_i - x'_j|$ for all $i, j \in [n]$ with $x'_i \neq x'_j$. Define $X = \{x_i - \Delta : i \in [n]\} \cup \{x_i + \Delta : i \in [n]\}$ and $X' = \{x'_i - \Delta : i \in [n]\} \cup \{x'_i + \Delta : i \in [n]\}$. We say a rectangle $[x_-, x_+] \times [x'_-, x'_+]$ is *regular* if $x_-, x_+ \in X \cup \{-\infty, \infty\}$ and $x'_-, x'_+ \in X' \cup \{-\infty, \infty\}$. Let $\mathcal{R}_{\text{reg}}$ denote the set of all regular rectangles. The total number of different regular rectangles is $O(n^4)$, i.e., $|\mathcal{R}_{\text{reg}}| = O(n^4)$, because $|X| = O(n)$ and $|X'| = O(n)$. Note that if R is a regular rectangle, then for any $i \in [n]$, the point (x_i, x'_i) is either contained in the interior of R or outside R . We say a regular rectangle R is *nonempty* if $(x_i, x'_i) \in R$ for some $i \in [n]$, and *empty* otherwise. For a nonempty rectangle R , we define

$$\delta_R = 1 + \min_{f \in \mathbb{R}[x, x']_g} \sum_{(x_i, x'_i) \in R} (y_i - f(x_i, x'_i))^2.$$

Note that δ_R can be computed in $n^{O(1)}$ time using the standard approach for least-square polynomial regression. For a set $\mathcal{R}$ of regular rectangles, denote by $\mathcal{R}_\bullet \subseteq \mathcal{R}$ the subset of nonempty rectangles, and define $\sigma_S(\mathcal{R}) = \sum_{R \in \mathcal{R}_\bullet} \delta_R$. A *regular region* refers to a subset of $\mathbb{R}^2$ that is the union of regular rectangles.

An *orthogonal partition* (OP) Π of a region $K \subseteq \mathbb{R}^2$ is a set of interior-disjoint (axis-parallel) rectangles whose union is K (see Figure 1 for an illustration). An OP Π is *regular* if all rectangles in Π are regular. The following lemma shows that our problem can be reduced to computing a regular OP Π of the plane which minimizes $\sigma_S(\Pi)$.



■ **Figure 1** An orthogonal partition (OP) of the region K .

► **Lemma 9.** *For any $f \in \Gamma_g^2$, there exists a regular OP Π of $\mathbb{R}^2$ such that $|\Pi| \leq 5|f| + 1$ and $\sigma_S(\Pi) \leq \sigma_S(f)$. Conversely, given a regular OP Π of $\mathbb{R}^2$, one can compute in $n^{O(1)}$ time a function $f \in \Gamma_g^2$ such that $\sigma_S(f) = \sigma_S(\Pi)$.*

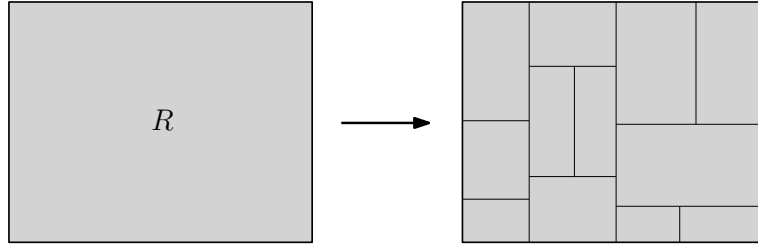
Using the reduction of Lemma 9, we establish our algorithms for piecewise polynomial regression for bivariate data. Section 4.1 presents a polynomial-time constant-approximation algorithm (Theorem 3), and Section 4.2 presents a QPTAS (Theorem 4).

4.1 A polynomial-time constant-approximation algorithm

In this section, we present a polynomial-time constant-approximation algorithm for the problem. Let Π^* be a regular OP of $\mathbb{R}^2$ that minimizes $\sigma_S(\Pi^*)$. In order to describe our algorithm, we need to introduce the notion of *binary* OP (and regular binary OP).

► **Definition 10** (binary OP). *Let R be an axis-parallel rectangle. A **binary OP** of R is an OP defined using the following recursive rule:*

- *The trivial partition $\{R\}$ is a binary OP of R .*



■ **Figure 2** A binary OP of the rectangle R .

- If ℓ is a horizontal or vertical line that partitions R into two smaller rectangles R_1 and R_2 , and Π_1 (resp., Π_2) are binary OPs of R_1 (resp., R_2), then $\Pi_1 \cup \Pi_2$ is a binary OP of R .

A binary OP is **regular** if it only consists of regular rectangles.

See Figure 2 for an illustration of binary OP. The basic idea of our approximation algorithm is to, instead of computing an optimal regular OP, compute an optimal *binary* regular OP, i.e., a regular binary OP Π of $\mathbb{R}^2$ that minimizes $\sigma_S(\Pi)$. This task can be solved in polynomial time by a simple dynamic programming algorithm as follows. Suppose we want to compute an optimal binary regular OP Π of a regular rectangle R . Then Π is either the trivial partition $\{R\}$ of R , or there exists a horizontal or vertical line ℓ separating R into two rectangles R_1 and R_2 , and $\Pi = \Pi_1 \cup \Pi_2$ where Π_1 (resp., Π_2) is a regular binary OPs of R_1 (resp., R_2). In the latter case, the equation of the line ℓ must be $x = \tilde{x}$ for some $\tilde{x} \in X$ or $x' = \tilde{x}'$ for some $\tilde{x}' \in X'$, because Π has to be a regular OP. This implies that R_1 and R_2 are regular rectangles. Furthermore, Π_1 and Π_2 must be optimal regular binary OPs of R_1 and R_2 , respectively, in order to minimize $\sigma_S(\Pi)$. Therefore, if we already know the optimal regular binary OPs of all regular rectangles R' such that $\text{area}(R') < \text{area}(R)$, then an optimal regular binary OPs of R can be computed in $O(n)$ time. The details of our algorithm is shown in Algorithm 2, which computes an optimal regular binary OP of $\mathbb{R}^2$. Since $|\mathcal{R}_{\text{reg}}| = O(n^4)$, it is clear that Algorithm 2 runs in polynomial time.

Let Π_{bin} be the optimal regular binary OP of $\mathbb{R}^2$ computed by Algorithm 2 and Π^* be the regular OP of $\mathbb{R}^2$ that minimizes $\sigma_S(\Pi^*)$. We shall show that $\sigma_S(\Pi_{\text{bin}}) = O(\sigma_S(\Pi^*))$. To this end, we need the following two lemmas.

► **Lemma 11.** *For any regular OP Π of $\mathbb{R}^2$, there exists a regular binary OP Π' of $\mathbb{R}^2$ such that $|\Pi'| = O(|\Pi_\bullet|)$ and for any $R' \in \Pi'_\bullet$ there exists $R \in \Pi_\bullet$ such that $R' \subseteq R$.*

► **Lemma 12.** *Let Π and Π' be two regular OP of $\mathbb{R}^2$. If for any $R' \in \Pi'_\bullet$ there exists $R \in \Pi_\bullet$ such that $R' \subseteq R$, then we have $\sigma_S(\Pi') - \sigma_S(\Pi) \leq |\Pi'_\bullet| - |\Pi_\bullet|$.*

By Lemma 11, there exists a regular binary OP Π' of $\mathbb{R}^2$ such that $|\Pi'_\bullet| \leq O(|\Pi_\bullet^*|)$ and for any $R' \in \Pi'_\bullet$ there exists $R \in \Pi_\bullet^*$ such that $R' \subseteq R$. Then by Lemma 12, we have $\sigma_S(\Pi')/\sigma_S(\Pi^*) = 1 + (\sigma_S(\Pi') - \sigma_S(\Pi^*))/\sigma_S(\Pi^*) \leq 1 + (|\Pi'_\bullet| - |\Pi_\bullet^*|)/|\Pi_\bullet^*| = |\Pi'_\bullet|/|\Pi_\bullet^*| = O(1)$. Because Π_{bin} is an optimal regular binary OP of $\mathbb{R}^2$, we further have $\sigma_S(\Pi_{\text{bin}}) \leq \sigma_S(\Pi') \leq O(\sigma_S(\Pi^*))$. We have $\sigma_S(\Pi^*) \leq \text{opt}$ by the first statement of Lemma 9, and hence $\sigma_S(\Pi_{\text{bin}}) \leq O(\text{opt})$. Using the second statement of Lemma 9, we then compute a function $f \in \Gamma_g^2$ in $O(n \cdot |\Pi_{\text{bin}}|) = O(n^5)$ time such that $\sigma_S(f) = \sigma_S(\Pi_{\text{bin}}) \leq O(\text{opt})$.

► **Theorem 3.** *There exists a constant-factor approximation algorithm for bivariate piecewise polynomial regression which runs in polynomial time.*

Algorithm 2 OPTBINPARTITION(S).

```

1:  $N \leftarrow |\mathcal{R}_{\text{reg}}|$ 
2: sort the rectangles in  $\mathcal{R}_{\text{reg}}$  as  $R_1, \dots, R_N$  such that  $\text{area}(R_1) \leq \dots \leq \text{area}(R_N)$ 
3: for  $i$  from 1 to  $N$  do
4:    $\Pi[R_i] \leftarrow \{R_i\}$  and  $\text{opt}[R_i] \leftarrow \sigma_S(\Pi[R_i])$ 
5:   suppose  $R_i = [x_-, x_+] \times [x'_-, x'_+]$ 
6:   for all  $z \in X$  such that  $x_- < z < x_+$  do
7:      $R'_i \leftarrow [x_-, z] \times [x'_-, x'_+]$  and  $R''_i \leftarrow [z, x_+] \times [x'_-, x'_+]$ 
8:     if  $\text{opt}[R_i] > \text{opt}[R'_i] + \text{opt}[R''_i]$  then
9:        $\Pi[R_i] \leftarrow \Pi[R'_i] \cup \Pi[R''_i]$  and  $\text{opt}[R_i] \leftarrow \sigma_S(\Pi[R_i])$ 
10:  for all  $z' \in X'$  such that  $x'_- < z' < x'_+$  do
11:     $R'_i \leftarrow [x_-, x_+] \times [x'_-, z']$  and  $R''_i \leftarrow [x_-, x_+] \times [z', x'_+]$ 
12:    if  $\text{opt}[R_i] > \text{opt}[R'_i] + \text{opt}[R''_i]$  then
13:       $\Pi[R_i] \leftarrow \Pi[R'_i] \cup \Pi[R''_i]$  and  $\text{opt}[R_i] \leftarrow \sigma_S(\Pi[R_i])$ 
14: return  $\Pi[\mathbb{R}^2]$ 

```

4.2 A quasi-polynomial-time approximation scheme

In this section, we design a quasi-polynomial-time approximation scheme (QPTAS) for the problem, that is, a $(1 + \varepsilon)$ -approximation algorithm which runs in $n^{\log^{O(1)} n}$ time for any fixed $\varepsilon > 0$. To this end, we borrow an idea from the geometric independent set literature [3, 4, 5, 13], which combines the cutting lemma and the planar separator theorem. We need the following cutting lemma.

► **Lemma 13.** *Given a set $\mathcal{R}$ of interior-disjoint regular rectangles and a number $1 \leq r \leq |\mathcal{R}|$, there exists a regular OP Π of $\mathbb{R}^2$ with $|\Pi| = O(r)$ such that each rectangle in Π intersects at most $|\mathcal{R}|/r$ rectangles in $\mathcal{R}$.*

Proof. This lemma follows directly from a result of [3] (Lemma 3.12). The original statement in Lemma 3.12 of [3] only claims the existence of a partition Π of $\mathbb{R}^2$ satisfying the desired properties. However, by the construction in [3], if $\mathcal{R}$ consists of regular rectangles, then the partition Π is a regular OP. ◀

Using the above cutting lemma and the (weighted) planar separator theorem, we obtain the following corollary.

► **Corollary 14.** *Given a set $\mathcal{R}$ of interior-disjoint regular rectangles in $\mathbb{R}^2$ and a number $1 \leq r \leq |\mathcal{R}|$, there exists a set Σ of $O(\sqrt{r})$ interior-disjoint regular rectangles such that each rectangle in Σ intersects at most $|\mathcal{R}|/r$ rectangles in $\mathcal{R}$ and for any regular region $K \subseteq \mathbb{R}^2$, the closure of each connected component U of $K \setminus (\bigcup_{R \in \Sigma} R)$ entirely contains at most $\frac{2}{3}|\mathcal{R}|$ rectangles in $\mathcal{R}$.*

Proof. We shall use the following weighted version of the planar separator theorem. Let $G = (V, E)$ be a planar graph with m vertices where each vertex has a non-negative weight, and W be the total weight of the vertices. The weighted planar separator theorem states that one can partition the vertex set V into three parts V_1, V_2, Σ such that (i) there is no edge between V_1 and V_2 , (ii) $|\Sigma| \leq O(\sqrt{m})$, and (iii) the total weight of the vertices in V_i is at most $\frac{2}{3}W$ for $i \in \{1, 2\}$.

Let Π be the regular partition of $\mathbb{R}^2$ described in Lemma 13 satisfying that $|\Pi| = O(r)$ and each rectangle in Π intersects at most $|\mathcal{R}|/r$ rectangles in $\mathcal{R}$. Consider the planar graph G_Π induced by Π . We assign each vertex of G_Π (i.e., each rectangle in Π) a non-negative

weight as follows. For each rectangle $R \in \mathcal{R}$, let $m(R)$ be the number of rectangles in Π that intersects R . The weight of each rectangle $R' \in \Pi$ is the sum of $1/m(R)$ for all $R \in \mathcal{R}$ that intersects R' . Note that the total weight W is equal to $|\mathcal{R}|$ because each rectangle in $\mathcal{R}$ contributes exactly 1 to the total weight. Applying the weighted planar separator theorem to the vertex-weighted graph G_Π , we now partition Π into three parts V_1, V_2, Σ such that (i) there is no edge between V_1 and V_2 in G_Π , (ii) $|\Sigma| \leq O(\sqrt{r})$, and (iii) the total weight of the vertices in V_i is at most $\frac{2}{3}|\mathcal{R}|$ for $i \in \{1, 2\}$. The separator Σ is just the desired set of interior-disjoint regular rectangles described in the corollary. The fact that each rectangle in Σ intersects at most $|\mathcal{R}|/r$ rectangles in $\mathcal{R}$ follows directly from the property of Π . So it suffices to show that for any regular region $K \subseteq \mathbb{R}^2$, (the closure of) each connected component of $K \setminus (\bigcup_{R \in \Sigma} R)$ intersects at most $\frac{2}{3}|\mathcal{R}|$ rectangles in $\mathcal{R}$. Let U be a connected component of $K \setminus (\bigcup_{R \in \Sigma} R)$. The rectangles in Π that are contained in the closure of U induces a connected subgraph of G_Π , and hence they either all belong to V_1 or all belong to V_2 (because there is no edge between V_1 and V_2 in G_Π). It follows that the total weight of these rectangles is at most $\frac{2}{3}|\mathcal{R}|$, which further implies that the number of rectangles in $\mathcal{R}$ that are (entirely) contained in the closure of U is at most $\frac{2}{3}|\mathcal{R}|$. ◀

With the above corollary in hand, we are ready to describe our QPTAS. Roughly speaking, our algorithm “guesses” the set Σ in Corollary 14 for the optimal regular OP $\mathcal{R}$ (and some parameter r polynomial in $\log n$ and $1/\varepsilon$) and then recursively solve the sub-problem in each rectangle in Σ and in each connected component of the complement of $\bigcup_{R \in \Sigma} R$. The nice properties of Σ described in Corollary 14 can be used to show (with a careful analysis) that the final solution we compute is a $(1 + \varepsilon)$ -approximation of the optimal solution.

Formally, let $r = \omega(1)$ be an integer parameter to be determined later and c be a sufficiently large constant. For a regular region $K \subseteq \mathbb{R}^2$ and an integer m , we denote by $\text{opt}_{K,m}$ as the minimum $\sigma_S(\Pi)$ for a regular OP Π of K with $|\Pi_\bullet| \leq m$. We shall design a procedure $\text{APPXPARTITION}(S, K, m)$, which computes a regular OP Π of the regular region K such that $\sigma_S(\Pi)$ is “not much larger” than $\text{opt}_{K,m}$ (note that we do *not* require $|\Pi_\bullet| \leq m$); what we mean by “not much larger” will be clear shortly.

■ **Algorithm 3** $\text{APPXPARTITION}(S, K, m)$.

```

1:  $\Pi_{\text{opt}} \leftarrow \emptyset$  and  $\text{opt} \leftarrow \infty$ 
2: for all  $\Pi \subseteq \mathcal{R}_{\text{reg}}$  with  $|\Pi| \leq r$  do
3:   if the rectangles in  $\Pi$  are interior-disjoint and contained in  $K$  then
4:     construct an arbitrary regular OP  $\Pi'$  of  $K$  such that  $\Pi \subseteq \Pi'$ 
5:     if  $\sigma_S(\Pi') < \text{opt}$  then  $\Pi_{\text{opt}} \leftarrow \Pi'$  and  $\text{opt} \leftarrow \sigma_S(\Pi')$ 
6: if  $m \leq r$  then return  $\Pi_{\text{opt}}$ 
7: for all  $\Sigma \subseteq \mathcal{R}_{\text{reg}}$  with  $|\Sigma| \leq c\sqrt{r}$  do
8:   if the rectangles in  $\Sigma$  are interior-disjoint then
9:      $\mathcal{U} \leftarrow \text{Components}(K \setminus (\bigcup_{R \in \Sigma} R))$ 
10:     $\Pi_R \leftarrow \text{APPXPARTITION}(S, K \cap R, m/r)$  for all  $R \in \Sigma$ 
11:     $\Pi_U \leftarrow \text{APPXPARTITION}(S, \text{Closure}(U), \frac{3}{4}m)$  for all  $U \in \mathcal{U}$ 
12:     $\Pi \leftarrow (\bigcup_{R \in \Sigma} \Pi_R) \cup (\bigcup_{U \in \mathcal{U}} \Pi_U)$ 
13:    if  $\sigma_S(\Pi) < \text{opt}$  then  $\Pi_{\text{opt}} \leftarrow \Pi$  and  $\text{opt} \leftarrow \sigma_S(\Pi)$ 
14: return  $\Pi_{\text{opt}}$ 

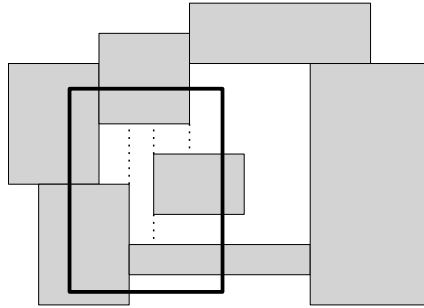
```

Algorithm 3 shows how $\text{APPXPARTITION}(S, K, m)$ works step-by-step, and here we provide an intuitive explanation of the algorithm. Let Π^* be a (unknown) regular OP of K such that $|\Pi^*| \leq m$ and $\sigma_S(\Pi^*) = \text{opt}_{K,m}$. We consider two cases separately: $|\Pi^*| \leq r$ and

$|\Pi_\bullet^*| > r$. The for-loop of Line 2-6 handles the case $|\Pi_\bullet^*| \leq r$. We simply guess the (at most) r rectangles in $\Pi_\bullet^*$. Note that when we correctly guess $\Pi_\bullet^*$, i.e., $\Pi = \Pi_\bullet^*$ in Line 2, any regular OP Π' of K such that $\Pi \subseteq \Pi'$ satisfies $\sigma_S(\Pi') = \sigma_S(\Pi) = \sigma_S(\Pi_\bullet^*) = \sigma_S(\Pi^*)$, because $(x_i, x'_i) \notin K \setminus (\bigcup_{R \in \Pi} R)$ for all $i \in [n]$. Therefore, in the case $|\Pi_\bullet^*| \leq r$, we already have $|\Pi_{\text{opt}}| \leq \text{opt}_{K,m}$ after the for-loop of Line 2-6. The remaining case is $|\Pi_\bullet^*| > r$, which implies $m > r$. This case is handled in the for-loop of Line 8-15. We guess the set Σ described in Corollary 14 with $\mathcal{R} = \Pi_\bullet^*$ (Line 8 of Algorithm 3), which consists of at most $c\sqrt{r}$ interior-disjoint regular rectangles (recall that c is sufficiently large). Let $\mathcal{U}$ be the set of connected components of $K \setminus (\bigcup_{R \in \Sigma} R)$. By Corollary 14, for each $R \in \Sigma$, the regular region $K \cap R$ intersects at most $|\Pi_\bullet^*|/r$ (and hence at most m/r) rectangles in $\mathcal{R}$, and for each $U \in \mathcal{U}$, the closure of U contains at most $\frac{2}{3}|\Pi_\bullet^*|$ rectangles (and hence at most $\frac{2}{3}m$) in $\mathcal{R}$. We then recursively call $\text{APPXPARTITION}(S, K \cap R, m/r)$ for all $R \in \Sigma$ and $\text{APPXPARTITION}(S, \text{Closure}(U), \frac{3}{4}m)$ for all $U \in \mathcal{U}$; see Line 11-12 of Algorithm 3. Each recursive call returns us a regular OP of the corresponding sub-region of K ; we set Π to be the union of all the returned regular OPs, which is clearly a regular OP of K (Line 13 of Algorithm 3). Intuitively, $\sigma_S(\Pi)$ should be “not much larger” than $\sigma_S(\Pi^*)$ if our guess for Σ is correct. More precisely, we have the following observation.

► **Lemma 15.** $\sum_{R \in \Sigma} \text{opt}_{K \cap R, m/r} + \sum_{U \in \mathcal{U}} \text{opt}_{\text{Closure}(U), \frac{3}{4}m} \leq (1 + O(1/\sqrt{r})) \cdot \sigma_S(\Pi^*)$.

Proof. We first show that there exists a regular OP Π of K satisfying (i) $|\Pi_\bullet| - |\Pi_\bullet^*| = O(|\Pi_\bullet^*|/\sqrt{r})$, (ii) each rectangle in Π is either contained in some $R \in \Sigma$ or interior-disjoint with all $R \in \Sigma$, (iii) each $R \in \Sigma$ contains at most m/r nonempty rectangles in Π and $\text{Closure}(U)$ contains at most $\frac{3}{4}m$ nonempty rectangles in Π for each $U \in \mathcal{U}$. Consider the regular OP Π^* of K . We further partition each rectangle $R^* \in \Pi^*$ into smaller (regular) rectangles as follows. Let $m(R^*)$ denote the number of rectangles in Σ that intersect (the interior of) R^* . Since the rectangles in Σ are interior-disjoint, the boundaries of these $m(R^*)$ rectangles cut R^* into $m(R^*) + 1$ regions (which are not necessarily rectangles). Now we construct the vertical decomposition the boundaries of these $m(R^*)$ rectangles inside R^* as follows (similarly to what we did in the proof of Lemma 9). For each top (resp., bottom) vertex of the $m(R^*)$ rectangles, if the vertex is contained in the interior of R^* , we shoot an upward (resp., downward) vertical ray from the vertex, which goes upwards (resp., downwards) until hitting the boundary of R^* or the boundary of some other $R \in \Sigma$. See Figure 3 for an illustration. Including one ray cuts R^* into one more piece, and the total number of the rays we shoot is at most $4m(R^*)$. Therefore, the vertical decomposition induces a regular OP of R^* into at most $5m(R^*) + 1$ rectangles. We do this for every rectangle $R^* \in \Pi^*$. After that, we obtain our desired regular OP Π . Next, we verify that Π satisfies



■ **Figure 3** The vertical decomposition inside R^* . The grey rectangles are those in Σ . The rectangle with bolder boundary is R^* .

the three conditions. We have $|\Pi_\bullet| \leq \sum_{R^* \in \Pi_\bullet^*} (5m(R^*) + 1) = \sum_{R^* \in \Pi_\bullet^*} 5m(R^*) + |\Pi_\bullet^*|$ since each rectangle $R^* \in \Pi_\bullet^*$ is partitioned into at most $5m(R^*) + 1$ smaller rectangles in Π (note that the rectangles in $\Pi^* \setminus \Pi_\bullet^*$ do not contribute any nonempty rectangle to Π). Because $|\Sigma| = O(\sqrt{r})$ and each rectangle in Σ intersects at most $|\Pi_\bullet^*|/r = |\Pi_\bullet^*|/r$ rectangles in $\Pi_\bullet^*$, we have $\sum_{R^* \in \Pi_\bullet^*} m(R^*) = O(|\Pi_\bullet^*|/\sqrt{r})$. It follows that $|\Pi_\bullet| - |\Pi_\bullet^*| = O(|\Pi_\bullet^*|/\sqrt{r})$, i.e., Π satisfies condition (i). Condition (ii) follows directly from our construction of Π . It suffices to show condition (iii). Let $R \in \Sigma$ be a rectangle. By our construction of Π , inside each $R^* \in \Pi^*$ that intersects (the interior of) R , there is exactly one rectangle in Π that is contained in R . Since R only intersects at most $|\Pi_\bullet^*|/r$ nonempty rectangles in Π^* and $|\Pi_\bullet^*| \leq m$, R contains at most m/r nonempty rectangles in Π . Let $U \in \mathcal{U}$ be a connected component of $K \setminus (\bigcup_{R \in \Sigma} R)$. Denote by $\Pi_\bullet^*(U) \subseteq \Pi_\bullet^*$ be the subset of rectangles that intersect U . Clearly, the number of nonempty rectangles in Π that are contained in $\text{Closure}(U)$ is at most $\sum_{R^* \in \Pi_\bullet^*(U)} (5m(R^*) + 1) = |\Pi_\bullet^*(U)| + O(|\Pi_\bullet^*(U)|/\sqrt{r})$. By Corollary 14, $\text{Closure}(U)$ entirely contains at most $\frac{2}{3}|\Pi_\bullet^*(U)|$ rectangles in $\Pi_\bullet^*(U)$. All the other rectangles in $\Pi_\bullet^*(U)$ are partially contained in $\text{Closure}(U)$. Note that if a rectangle is partially contained in $\text{Closure}(U)$, then it intersects some $R \in \Sigma$. Therefore, the number of rectangles in $\Pi_\bullet^*(U)$ that are partially contained in $\text{Closure}(U)$ is bounded by $O(|\Pi_\bullet^*(U)|/\sqrt{r})$, because $|\Sigma| = O(\sqrt{r})$ and each rectangle in Σ intersects at most $|\Pi_\bullet^*|/r$ rectangles in $\Pi_\bullet^*$. It follows that $|\Pi_\bullet^*(U)| = \frac{2}{3}|\Pi_\bullet^*(U)| + O(|\Pi_\bullet^*(U)|/\sqrt{r})$ and the number of rectangles in Π that are contained in $\text{Closure}(U)$ is bounded by $\frac{2}{3}|\Pi_\bullet^*(U)| + O(|\Pi_\bullet^*(U)|/\sqrt{r})$, which is no more than $\frac{3}{4}m$ because $|\Pi_\bullet^*| \leq m$ and we require $r = \omega(1)$.

Now we are ready to prove the lemma. Let Π be the regular OP of K we constructed above. Condition (ii) above guarantees that each rectangle in Π is either contained in some $R \in \Sigma$ or contained in $\text{Closure}(U)$ for some $U \in \mathcal{U}$. For each $R \in \Sigma$, let $\Pi(R) \subseteq \Pi$ denote the subset of rectangles contained in R . Similarly, for each $U \in \mathcal{U}$, let $\Pi(U) \subseteq \Pi$ denote the subset of rectangles contained in $\text{Closure}(U)$. Condition (iii) above guarantees that $|\Pi(R)_\bullet| \leq m/r$ for all $R \in \Sigma$ and $|\Pi(U)_\bullet| \leq \frac{3}{4}m$ for all $U \in \mathcal{U}$. So we have

$$\sigma_S(\Pi) = \sum_{R \in \Sigma} \sigma_S(\Pi(R)) + \sum_{U \in \mathcal{U}} \sigma_S(\Pi(U)) \geq \sum_{R \in \Sigma} \text{opt}_{K \cap R, m/r} + \sum_{U \in \mathcal{U}} \text{opt}_{\text{Closure}(U), \frac{3}{4}m}.$$

On the other hand, we have $\sigma_S(\Pi) - \sigma_S(\Pi^*) \leq |\Pi_\bullet| - |\Pi_\bullet^*| = O(|\Pi_\bullet^*|/\sqrt{r})$ by Lemma 12 and condition (i) above. Because $|\Pi_\bullet^*| \leq \sigma_S(\Pi^*)$, we further have $\sigma_S(\Pi) \leq (1 + O(1/\sqrt{r})) \cdot \sigma_S(\Pi^*)$. Combining the two inequalities above gives us the inequality in the lemma. $\blacktriangleleft$

► **Corollary 16.** *Let Π_{opt} be the regular OP of K returned by $\text{APPXPARTITION}(S, K, m)$. Then we have $\sigma_S(\Pi_{\text{opt}}) \leq (1 + O(1/\sqrt{r}))^{O(\log m)} \cdot \text{opt}_{K, m}$.*

Proof. As before, let Π^* be a (unknown) regular OP of K such that $|\Pi_\bullet^*| \leq m$ and $\sigma_S(\Pi^*) = \text{opt}_{K, m}$. We prove that $\sigma_S(\Pi_{\text{opt}}) \leq (1 + O(1/\sqrt{r}))^{\log_{3/4} m} \cdot \text{opt}_{K, m}$ by induction on m . In the base case where $m \leq r$, we have $\sigma_S(\Pi_{\text{opt}}) \leq \sigma_S(\Pi^*) = \text{opt}_{K, m}$ after the for-loop of Line 2-6 (as argued before). Now suppose $m > r$. If $|\Pi_\bullet^*| \leq r$, then we still have $\sigma_S(\Pi_{\text{opt}}) \leq \text{opt}_{K, m}$ after the for-loop of Line 2-6 (as argued before). So it suffices to consider the case $|\Pi_\bullet^*| > r$. We show that when we correctly guess the set Σ in Line 8, the regular OP Π of K we construct in Line 13 satisfies $\sigma_S(\Pi) \leq (1 + O(1/\sqrt{r}))^{\log_{3/4} m} \cdot \text{opt}_{K, m}$. Let $\mathcal{U}$ be the set of connected components of $K \setminus (\bigcup_{R \in \Sigma} R)$, as in Line 10. We have $\Pi = (\bigcup_{R \in \Sigma} \Pi_R) \cup (\bigcup_{U \in \mathcal{U}} \Pi_U)$ where $\Pi_R = \text{APPXPARTITION}(S, K \cap R, m/r)$ and $\Pi_U = \text{APPXPARTITION}(S, \text{Closure}(U), \frac{3}{4}m)$.

Recall that $r = \omega(1)$, and hence $m/r \leq \frac{3}{4}m$. By our induction hypothesis and Lemma 15,

$$\begin{aligned}
\sigma_S(\Pi) &= \sum_{R \in \Sigma} \sigma_S(\Pi_R) + \sum_{U \in \mathcal{U}} \sigma_S(\Pi_U) \\
&\leq (1 + O(1/\sqrt{r}))^{\log_{3/4} m - 1} \cdot \left(\sum_{R \in \Sigma} \text{opt}_{K \cap R, m/r} + \sum_{U \in \mathcal{U}} \text{opt}_{\text{Closure}(U), \frac{3}{4}m} \right) \\
&\leq (1 + O(1/\sqrt{r}))^{\log_{3/4} m - 1} \cdot (1 + O(1/\sqrt{r})) \cdot \sigma_S(\Pi^*) \\
&= (1 + O(1/\sqrt{r}))^{\log_{3/4} m} \cdot \sigma_S(\Pi^*),
\end{aligned}$$

which completes the proof. $\blacktriangleleft$

By Corollary 16, if we set $r = c' \cdot (\log^2 n / \varepsilon^2)$ for a sufficiently large constant c' , then for any regular region K and any $m = O(n)$, the procedure $\text{APPXPARTITION}(S, K, m)$ will return a regular partition Π_{opt} of K such that $\sigma_S(\Pi_{\text{opt}}) \leq (1 + \varepsilon) \cdot \text{opt}_{K, m}$. To solve our problem, we only need to call $\text{APPXPARTITION}(S, \mathbb{R}^2, 5n + 1)$, which will return a regular partition Π_{opt} of $\mathbb{R}^2$ such that $\sigma_S(\Pi_{\text{opt}}) \leq (1 + \varepsilon) \cdot \text{opt}_{\mathbb{R}^2, 5n+1}$. By the first statement of Lemma 9, we have $\text{opt}_{\mathbb{R}^2, 5n+1} \leq \text{opt}$. Therefore, it suffices to use the second statement of Lemma 9 to compute a function $f \in \Gamma_g^2$ such that $\sigma_S(f) = \sigma_S(\Pi_{\text{opt}}) \leq (1 + \varepsilon) \cdot \text{opt}$.

Time complexity. If $m \leq r$, the procedure $\text{APPXPARTITION}(S, K, m)$ takes $n^{O(r)} = n^{O(\log^2 n / \varepsilon^2)}$ time. In the case $m > r$, there are $n^{O(\sqrt{r})}$ sets Σ to be considered in Line 8. For each Σ , we have $c\sqrt{r}$ recursive calls in Line 11 and $n^{O(1)}$ recursive calls in Line 12, and all the other work in the for-loop of Line 8-15 can be done in $n^{O(1)}$ time. In addition, Line 1-6 takes $n^{O(r)}$ time. Therefore, if we use $T(m)$ to denote the running time of $\text{APPXPARTITION}(S, K, m)$, we have the recurrence

$$T(m) = \begin{cases} n^{O(\sqrt{r})} \cdot T(m/r) + n^{O(\sqrt{r})} \cdot T(\frac{3}{4}m) + n^{O(r)} & \text{if } m > r, \\ n^{O(r)} & \text{if } m \leq r, \end{cases}$$

which solves to $T(m) = n^{O(\sqrt{r} \log m + r)}$. Since our initial call is $\text{APPXPARTITION}(S, \mathbb{R}^2, 5n + 1)$, the total running time of our algorithm is $n^{O(\sqrt{r} \log n + r)} = n^{O(\log^2 n / \varepsilon^2)}$.

► **Theorem 4.** *There exists a QPTAS for bivariate piecewise polynomial regression.*

5 Conclusion and future work

In this paper, we studied the regression problem for univariate and bivariate data using piecewise polynomial functions. The loss of a k -piece polynomial function is measured as the sum of λk and its square error, where $\lambda \geq 0$ is a pre-specified parameter. For univariate data, we gave a $(1 + \varepsilon)$ -approximation algorithm that runs in $O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ time, assuming the data points are pre-sorted. For bivariate data, we presented three results, a subexponential-time exact algorithm, a polynomial-time constant-approximation algorithm, and a QPTAS. Finally, for completeness, we also proved the problem for bivariate data is NP-hard.

Our work suggests several open problems and future research directions. The complexity of solving the problem exactly for the univariate data remains a challenging open problem. Is there a subquadratic time algorithm, or is there a (conditional or unconditional) near-quadratic lower bound? For bivariate data, does there exist a PTAS, namely, a polynomial-time $(1 + \varepsilon)$ -approximation algorithm for any fixed $\varepsilon > 0$? Finally, designing efficient approximation algorithms for regression problems with more than two variables is an interesting problem.

References

- 1 Jayadev Acharya, Ilias Diakonikolas, Chinmay Hegde, Jerry Zheng Li, and Ludwig Schmidt. Fast and near-optimal algorithms for approximating distributions by histograms. In *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 249–263, 2015.
- 2 Jayadev Acharya, Ilias Diakonikolas, Jerry Li, and Ludwig Schmidt. Fast algorithms for segmented regression. In *International Conference on Machine Learning*, pages 2878–2886, 2016.
- 3 Anna Adamaszek, Sarel Har-Peled, and Andreas Wiese. Approximation schemes for independent set and sparse subsets of polygons. *Journal of the ACM (JACM)*, 66(4):1–40, 2019.
- 4 Anna Adamaszek and Andreas Wiese. Approximation schemes for maximum weight independent set of rectangles. In *2013 IEEE 54th annual symposium on foundations of computer science*, pages 400–409. IEEE, 2013.
- 5 Anna Adamaszek and Andreas Wiese. A qptas for maximum weight independent set of polygons with polylogarithmically many vertices. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 645–656. SIAM, 2014.
- 6 Pankaj K Agarwal, Sarel Har-Peled, Nabil H Mustafa, and Yusu Wang. Near-linear time approximation algorithms for curve simplification. *Algorithmica*, 42(3-4):203–219, 2005.
- 7 Pankaj K Agarwal and Subhash Suri. Surface approximation and geometric partitions. *SIAM Journal on Computing*, 27(4):1016–1035, 1998.
- 8 Boris Aronov, Tetsuo Asano, Naoki Katoh, Kurt Mehlhorn, and Takeshi Tokuyama. Polyline fitting of planar points under min-sum criteria. *International journal of computational geometry & applications*, 16(02n03):97–116, 2006.
- 9 Ilias Diakonikolas, Jerry Li, and Anastasia Voloshinov. Efficient algorithms for multidimensional segmented regression. *arXiv preprint arXiv:2003.11086*, 2020.
- 10 Michael T Goodrich. Efficient piecewise-linear function approximation using the uniform metric: (preliminary version). In *Proceedings of the tenth annual Symposium on Computational geometry*, pages 322–331, 1994.
- 11 Sudipto Guha. On the space–time of optimal, approximate and streaming algorithms for synopsis construction problems. *The VLDB Journal*, 17(6):1509–1535, 2008.
- 12 S Louis Hakimi and Edward F Schmeichel. Fitting polygonal functions to a set of points in the plane. *CVGIP: Graphical Models and Image Processing*, 53(2):132–136, 1991.
- 13 Sarel Har-Peled. Quasi-polynomial time approximation scheme for sparse subsets of polygons. In *Proceedings of the thirtieth annual symposium on Computational geometry*, pages 120–129, 2014.
- 14 Hosagrahar Visvesvaraya Jagadish, Nick Koudas, S Muthukrishnan, Viswanath Poosala, Kenneth C Sevcik, and Torsten Suel. Optimal histograms with quality guarantees. In *VLDB*, volume 98, pages 24–27, 1998.
- 15 Jon Kleinberg and Eva Tardos. *Algorithm design*. Pearson Education India, 2006.
- 16 Frederick Mosteller, John Wilder Tukey, et al. *Data analysis and regression: a second course in statistics*. Pearson, 1977.
- 17 DP Wang, NF Huang, HS Chao, and Richard CT Lee. Plane sweep algorithms for the polygonal approximation problems with applications. In *International Symposium on Algorithms and Computation*, pages 515–522. Springer, 1993.