Efficient Algorithms for Least Square Piecewise Polynomial Regression

Daniel Lokshtanov
Department of Computer Science, University of California, Santa Barbara, USA
daniello@ucsb.edu

Subhash Suri
Department of Computer Science, University of California, Santa Barbara, USA
suri@cs.ucsb.edu

Jie Xue
Department of Computer Science, University of California, Santa Barbara, USA
jiexue@ucsb.edu

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 $f_j : \mathbb{R}^d \to \mathbb{R}$, 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 $f : \bigsqcup_{j=1}^{k} P_j \to \mathbb{R}$ 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 are disjoint axis-aligned rectangles in the case of bivariate data. Our error approximation allows use of any fixed-degree polynomial, and 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 → Design and analysis of algorithms

Keywords and phrases Regression analysis, Piecewise polynomial, Least square error

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

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

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:26
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

form a continuous curve, and it is conjectured that finding a polygonal chain of k pieces with minimum L_2 error is NP-hard. In our regression setting, such continuity is not required.

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 $i = 1, \ldots, n$. 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 $P_1, \ldots, P_k$, 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 $f : \bigsqcup_{j=1}^{k} P_j \to \mathbb{R}$ is the *piecewise polynomial* function defined as $f|_{P_j} = f_j$. The pieces $P_1, \ldots, P_k$ 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 three results, including a sub-exponential time exact algorithm, a polynomial-time constant-approximation algorithm, and a quasi-polynomial time 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-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 are well-studied in computational geometry [6, 8] and statistics [16], 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 from the algorithms of [12, 18]. Aronov et al. [8] gave an FPTAS for the polyline fitting problem with the min-sum and least-square error measure. Specifically, they considered two problems: minimizing the total error for a given number of pieces of the polyline, and minimizing the number of pieces of the polyline for a given upper bound on the error. Agarwal et al. [6] consider approximation under Hausdorff and Frechet distances. Unlike these computational geometric models, in regression and in database theory, the piecewise approximation is *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, where k is the number of pieces [11, 14]. A similar dynamic programming algorithm can also compute an optimal "regularized" piecewise approximation, where the number of pieces k is not fixed but included in the objective function, in $O(kn^2)$ time, where k is the number of pieces in the optimal solution [15]. In machine learning, "segmented" piecewise regression aims to recover a function f, which is promised to be piecewise linear with an unknown number k pieces. A common assumption in that line of work is that data samples are drawn from a "tame" distribution, such as Gaussian, with i.i.d. noise [1, 9]. In that model also, the best known algorithm for computing an optimal piecewise function has complexity $O(kn^2)$ [1].

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. Our algorithms for bivariate data are presented in Section 4, with the exception that the sub-exponential time exact algorithm (Theorem 2) is presented in Appendix C. The hardness result for bivariate data (Theorem 5) is presented in Appendix D. Also, due to limited space, some proofs and details are deferred to the appendix.

2 Basic notations and concepts

In this section, we introduce some basic notations and concepts which will be use 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 $f : \bigsqcup_{j=1}^{k} P_j \to \mathbb{R}$, where $P_1, \ldots, P_k$ 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 $j \in \{1, \ldots, k\}$. The intervals (resp., rectangles) $P_1, \ldots, P_k$ are the *pieces* of f, and the number k is the *complexity* of f, denoted by $|f|$. Clearly, the notion of piecewise polynomial functions can be generalized to higher dimensions (i.e., more variables), where the pieces becomes axis-parallel boxes.

But in most part of this paper, we only study univariate and bivariate piecewise polynomial functions. Let $\mathit{\Gamma}_g^d$ denote the family of piecewise polynomial functions with d variables and of degree at most g. For a dataset $S = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}\}_{i=1}^n$ of points, we define the *error* of a function $f \in \mathit{\Gamma}_g^d$ 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 parameter; we set $\sigma_S(f) = \infty$ 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 $f^* \in \mathit{\Gamma}_g^d$ that minimizes $\sigma_S(f^*)$. As mentioned before, we usually study the case $d = 1$ or $d = 2$. Note that without loss of generality, we can assume $\lambda = 1$ by scaling the y-values of the points in S. Therefore, for convenience, we make this assumption throughout the paper.

3 A linear-time approximation scheme for univariate data

We consider the piecewise polynomial regression problem for univariate data. Let $g \geq 0$ is a fixed constant. The input of the problem is a dataset $S = \{(x_i, y_i) \in \mathbb{R} \times \mathbb{R}\}_{i=1}^n$ where $x_1 \leq \cdots \leq x_n$. Note that we do *not* assume that $x_1, \dots, x_n$ are distinct. Our goal is to find the function $f^* \in \mathit{\Gamma}_g^1$ that minimizes $\sigma_S(f^*)$ (recall that $\lambda = 1$ by assumption). Using dynamic programming, this problem can be straightforwardly solved in $O(n^2)$ time. However, no subquadratic-time algorithm was known.

In this section, we present the first linear-time approximation scheme for the problem. Specifically, we show that, for any $\varepsilon > 0$, one can find a function $f \in \mathit{\Gamma}_g^1$ in $O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ time such that $\sigma_S(f) \leq (1+\varepsilon) \cdot \mathsf{opt}$, where $\mathsf{opt} = \min_{f^* \in \mathit{\Gamma}_g^1} \sigma_S(f^*)$, provided that the points in S are pre-sorted by their x-coordinates. For $a, b \in [n]$ satisfying $a \leq b$, we define

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

▶ **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 $a - 1 \leq a_0 < \cdots < a_r \leq b$, we have $\delta[a,b] \geq \sum_{j=1}^{r} \delta[a_{j-1}+1, a_j]$.*

Let $\varepsilon > 0$ be a given approximation factor. Since we are interested in the asymptotical running time, we may assume that ε is sufficiently small, say $\varepsilon \leq 1$. Let $\tilde{\varepsilon} > 0$ be the number satisfying $(1+\tilde{\varepsilon})^2 = 1 + \varepsilon$. 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., $x_{i+1} > x_i$).

Before introducing our algorithm, we first establish a structural lemma of an approximation solution. For a function $f \in \mathit{\Gamma}_g^1$ 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, $\sigma_S(f)$ is equal to the sum of $|f|$ and the costs of the pieces of f.

▶ **Lemma 7.** *There exists a function $f \in \mathit{\Gamma}_g^1$ such that $\sigma_S(f) \leq (1+\tilde{\varepsilon}) \cdot \mathsf{opt}$ and each piece of f is either a single point or of cost at most $2/\tilde{\varepsilon}$.*

Proof. Let $f^* \in \mathit{\Gamma}_g^1$ be an optimal solution, i.e., $\sigma_S(f^*) = \mathsf{opt}$. Consider a piece P^* of f^*. Without loss of generality, we may assume that $P^* = [x_a, x_b]$ for some $a, b \in [n]$ where a is a left break point and b is a right break point. Since f^* is optimal, the cost of P^* is equal to $\delta[a,b]$. 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 $a' \leq a''$ *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 $a_r = b$. We first claim that $r < \tilde{\varepsilon}\cdot\delta[a,b]+1$. We observe that $\delta[a_i+1,a_{i+2}] > 2/\tilde{\varepsilon}$ for all $i \in \{0,1,\dots,r-2\}$. To see this, note that all a_i's are right break points. If $\delta[a_i+1,a_{i+2}] \leq 2/\tilde{\varepsilon}$, then (a_i+1,a_{i+2}) is legal, which contradicts with the fact that 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. 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 $i \in [r]$. As mentioned above, we replace the piece P^* of f^* with the pieces $P_1,\dots,P_r$. We call $P_1,\dots,P_r$ the *sub-pieces* of P^*. We do this for all pieces of f^*, and collect all the sub-pieces. Our function $f \in \Gamma_g^1$ is constructed as follows. The pieces of f are just the sub-pieces, therefore the domain of f is contained in the domain of f^*. On each piece $P = [x_a, x_b]$ of f, we define $f_{|P}$ as the polynomial $f[a,b]$ restricted to P, and thus the cost of the piece P is $\delta[a,b]$. Thus, $f \in \Gamma_g^1$. Furthermore, by our construction, each piece of f is either a single point or of cost at most $2/\tilde{\varepsilon}$. It now suffices to show that $\sigma_S(f) \leq (1+\tilde{\varepsilon})\cdot\sigma_S(f^*)$. Consider a specific piece $P^* = [x_a,x_b]$ of f^*, and suppose $P_1,\dots,P_r$ are the sub-pieces of P^*. As argued before, the cost of P^* is $\delta[a,b]$. Let $c^*(P^*) = \delta[a,b]+1$ and $c(P^*)$ be the sum of the costs of $P_1,\dots,P_r$ (regarded as pieces of f) plus r. We have showed that $r < \tilde{\varepsilon}\cdot\delta[a,b]+1$. By Lemma 6, the sum of the costs of $P_1,\dots,P_r$ is at most $\delta[a,b]$. Therefore, $c(P^*) \leq (1+\tilde{\varepsilon})\cdot c^*(P^*)$. Note that $\sigma_S(f^*) = \sum_{P^*\in\mathcal{P}^*} c^*(P^*)$ and $\sigma_S(f) = \sum_{P^*\in\mathcal{P}^*} c(P^*)$, where $\mathcal{P}^*$ denote the set of all pieces of f^*. It immediately follows 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 single 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 single 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 $\delta[a_i(b),b] \leq (1+\tilde{\varepsilon})^i - 1$; 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 $A(b) = \{a_i(b) : i \geq 0 \text{ and } (1+\tilde{\varepsilon})^{i-1}-1 \leq 2/\tilde{\varepsilon}\}$. 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. Based on Lemma 7, we have the following observation.

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

Proof. We claim that for any S-light function $f_0 \in \Gamma_g^1$, there exists a canonical function $f \in \Gamma_g^1$ such that $\sigma_S(f) \leq (1+\tilde{\varepsilon})\cdot\sigma_S(f_0)$. By Lemma 7, this claim directly implies 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 $x_1 = \cdots = x_n$ and the interval $I = [x_1,x_n]$ is a single 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 $\sigma_S(f) \leq (1+\tilde{\varepsilon})\cdot\sigma_S(f_0)$. 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, $c(P) \leq 2/\tilde{\varepsilon}$ and $c(P) > 2/\tilde{\varepsilon}$. If $c(P) \leq 2/\tilde{\varepsilon}$, we define i as the smallest integer such that $(1+\tilde{\varepsilon})^i \geq c(P)+1$. Therefore, $(1+\tilde{\varepsilon})^{i-1} \leq c(P)+1 \leq (1+\tilde{\varepsilon})^i$. Since $c(P) \leq 2/\tilde{\varepsilon}$, we have $(1+\tilde{\varepsilon})^{i-1}-1 \leq 2/\tilde{\varepsilon}$ and hence

$a_i(n) \in A(n)$. By the definition of $a_i(n)$, we have $a_i(n) \leq a$ and $\delta[a_i(n), n] \leq (1+\tilde{\varepsilon})^i - 1$, i.e., $\delta[a_i(n), n] + 1 \leq (1+\tilde{\varepsilon})^i$. Since $(1+\tilde{\varepsilon})^{i-1} \leq c(P) + 1$, we further deduce that $\delta[a_i(n), n] + 1 \leq (1+\tilde{\varepsilon}) \cdot (c(P)+1)$. Now we define $S' = \{(x_1, y_1), \ldots, (x_{a-1}, y_{a-1})\} \subseteq S$ and $S'' = \{(x_1, y_1), \ldots, (x_{a_i(n)-1}, y_{a_i(n)-1})\} \subseteq S$. 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 $f'' \in \Gamma_g^1$ such that $\sigma_{S''}(f'') \leq (1+\tilde{\varepsilon}) \cdot \sigma_{S''}(f_0) \leq (1+\tilde{\varepsilon}) \cdot \sigma_{S'}(f_0)$, 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, $f \in \Gamma_g^1$. Also, f is canonical because f'' is canonical and $[x_{a_i(n)}, x_n]$ is a canonical interval. Finally, we have

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

In the case $c(P) > 2/\tilde{\varepsilon}$, P must be a single 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 $f'' \in \Gamma_g^1$ such that $\sigma_{S'}(f'') \leq (1+\tilde{\varepsilon}) \cdot \sigma_{S'}(f_0)$, where $S' = \{(x_1, y_1), \ldots, (x_{a-1}, y_{a-1})\}$. 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 $\sigma_{S'}(f'') \leq (1+\tilde{\varepsilon}) \cdot \sigma_{S'}(f_0)$ and the cost of P is at least $\delta[a, n]$, we have $\sigma_S(f) \leq (1+\tilde{\varepsilon}) \cdot \sigma_S(f_0)$. ◀

According to the above lemma, to compute a $(1+\varepsilon)$-approximation solution for the problem, it suffices to find the canonical function $f \in \Gamma_g^1$ that minimizes $\sigma_S(f)$. This can be simply solved using the dynamic programming algorithm shown in Algorithm 1.

Algorithm 1 APPROXIMATE-REGRESSION-1D(S)

```
t ← 0 and opt_0 ← 0
for t from 1 to n do
    if t is a right break point then
        ã ← arg min_{a∈A(t)} {opt_{a−1} + (δ[a, t] + 1)}
        opt_t ← opt_{ã−1} + (δ[ã, t] + 1)
return opt_n
```

The correctness of Algorithm 1 is clear. Next, we show that how to implement Algorithm 1 in $O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ time. We first observe that $|A(b)| = O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon})$ for all right break points $b \in [n]$. Therefore, if we already have all index sets $A(b)$ and all $f[a, b], \delta[a, b]$ where $a \in A(b)$ in hand, Algorithm 1 can be directly implemented in $O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ time. In other words, it suffices 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. We show how to achieve this in Appendix B.

▶ **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).

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 $\sigma_S(\Pi)$.

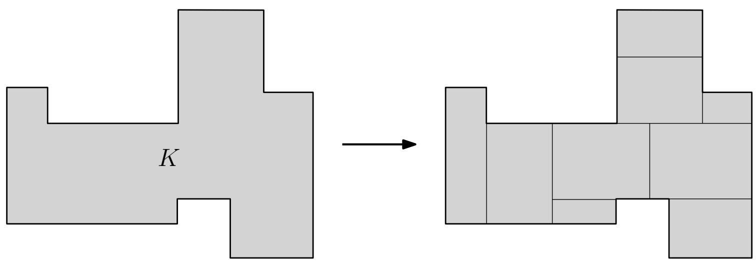

Figure 1 An orthogonal partition (OP) of the region K

▶ **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). Due to limited space, our sub-exponential exact algorithm (Theorem 2) is deferred to Appendix C, as it follows easily from Lemma 9 and the planar separator theorem.

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).

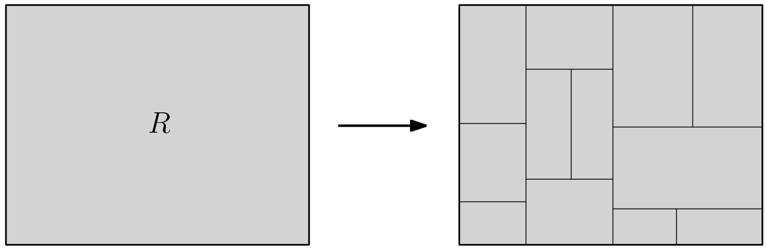

Figure 2 A binary OP of the rectangle R

▶ **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.*
- *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 $\mathsf{area}(R') < \mathsf{area}(R)$, 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 $\sigma_S(\Pi^*) \leq \mathsf{opt}$ by the first statement of Lemma 9, and hence $\sigma_S(\Pi_{\text{bin}}) \leq O(\mathsf{opt})$. 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 $\sigma_S(f) = \sigma_S(\Pi_{\text{bin}}) \leq O(\mathsf{opt})$.

Algorithm 2 $\textsc{OptBinPartition}(S)$

$N \leftarrow |\mathcal{R}_{\text{reg}}|$
sort the rectangles in $\mathcal{R}_{\text{reg}}$ as $R_1, \dots, R_N$ such that $\mathsf{area}(R_1) \leq \cdots \leq \mathsf{area}(R_N)$
for i from 1 to N **do**
  $\Pi[R_i] \leftarrow \{R_i\}$ and $\text{opt}[R_i] \leftarrow \sigma_S(\Pi[R_i])$
  suppose $R_i = [x_-, x_+] \times [x'_-, x'_+]$
  for all $z \in X$ such that $x_- < z < x_+$ **do**
    $R'_i \leftarrow [x_-, z] \times [x'_-, x'_+]$ and $R''_i \leftarrow [z, x_+] \times [x'_-, x'_+]$
    if $\text{opt}[R_i] > \text{opt}[R'_i] + \text{opt}[R''_i]$ **then**
      $\Pi[R_i] \leftarrow \Pi[R'_i] \cup \Pi[R''_i]$ and $\text{opt}[R_i] \leftarrow \sigma_S(\Pi[R_i])$
  for all $z' \in X'$ such that $x'_- < z' < x'_+$ **do**
    $R'_i \leftarrow [x_-, x_+] \times [x'_-, z']$ and $R''_i \leftarrow [x_-, x_+] \times [z', x'_+]$
    if $\text{opt}[R_i] > \text{opt}[R'_i] + \text{opt}[R''_i]$ **then**
      $\Pi[R_i] \leftarrow \Pi[R'_i] \cup \Pi[R''_i]$ and $\text{opt}[R_i] \leftarrow \sigma_S(\Pi[R_i])$
return $\Pi[\mathbb{R}^2]$

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

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 [4, 3, 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 can 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 each connected component U of $K \backslash (\bigcup_{R \in \Sigma} R)$, there are at most $\frac{2}{3}|\mathcal{R}|$ rectangles in $\mathcal{R}$ that are entirely contained in U.*

Now we are ready to describe our QPTAS. 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 $\mathsf{opt}_{K,m}$ as the minimum $\sigma_S(\Pi)$ for a regular OP Π of K with $|\Pi_\bullet| \leq m$. We shall design a procedure $\textsc{AppxPartition}(S, K, m)$, which computes a regular OP Π of the regular region K such that $\sigma_S(\Pi)$ is "not much larger" than $\mathsf{opt}_{K,m}$ (note that we do *not* require $|\Pi_\bullet| \leq m$); what we mean by "not much larger" will be clear shortly.

Algorithm 3 shows how $\textsc{AppxPartition}(S, K, m)$ 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 $\sigma_S(\Pi^*) = \mathsf{opt}_{K,m}$. We consider two cases separately: $|\Pi^*_\bullet| \leq r$ 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 $(x_i, x_i') \notin K \backslash (\bigcup_{R \in \Pi} R)$ for all $i \in [n]$. Therefore, in the case $|\Pi^*_\bullet| \leq r$, we already have $|\Pi_{\text{opt}}| \leq \mathsf{opt}_{K,m}$ 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 $K \backslash (\bigcup_{R \in \Sigma} R)$. 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 $\textsc{AppxPartition}(S, K \cap R, m/r)$ for all $R \in \Sigma$ and $\textsc{AppxPartition}(S, \mathsf{Closure}(U), \frac{3}{4}m)$ 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.** $\sum_{R \in \Sigma} \mathsf{opt}_{K \cap R, m/r} + \sum_{U \in \mathcal{U}} \mathsf{opt}_{\mathsf{Closure}(U), \frac{3}{4}m} \leq (1 + O(1/\sqrt{r})) \cdot \sigma_S(\Pi^*)$.

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 $\mathsf{Closure}(U)$ 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 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 $\Pi^* \backslash \Pi^*_\bullet$ 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)**. Conditions **(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 $U \in \mathcal{U}$

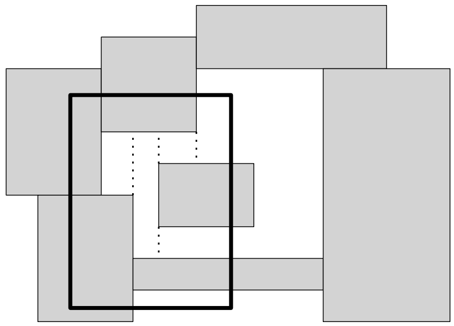

Figure 3 The vertical decomposition inside R^*. The grey rectangles are those in Σ. The rectangle with bolder boundary is R^*.

be a connected component of $K \backslash (\bigcup_{R \in \Sigma} R)$. 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 $\mathsf{Closure}(U)$ is at most $\sum_{R^* \in \Pi_\bullet^*(U)} (5m(R^*)+1) = |\Pi_\bullet^*(U)| + O(|\Pi_\bullet^*|/\sqrt{r})$. By Corollary 14, $\mathsf{Closure}(U)$ entirely contains at most $\frac{2}{3}|\Pi_\bullet^*|$ rectangles in $\Pi_\bullet^*(U)$. All the other rectangles in $\Pi_\bullet^*(U)$ are partially contained in $\mathsf{Closure}(U)$. Note that if a rectangle is partially contained in $\mathsf{Closure}(U)$, then it intersects some $R \in \Sigma$. Therefore, the number of rectangles in $\Pi_\bullet^*(U)$ that are partially contained in $\mathsf{Closure}(U)$ is bounded by $O(|\Pi_\bullet^*|/\sqrt{r})$, because $|\Sigma| = O(\sqrt{r})$ and each rectangle in Σ intersects at most $|\Pi_\bullet^*|/r$ rectangles in $\Pi_\bullet^*$. It follows that $|\Pi_\bullet^*(U)| = \frac{2}{3}|\Pi_\bullet^*| + O(|\Pi_\bullet^*|/\sqrt{r})$ and the number of rectangles in Π that are contained in $\mathsf{Closure}(U)$ is bounded by $\frac{2}{3}|\Pi_\bullet^*| + O(|\Pi_\bullet^*|/\sqrt{r})$, 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 $\mathsf{Closure}(U)$ 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 $\mathsf{Closure}(U)$. 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

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

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. ◀

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

Proof. As before, let Π^* be a (unknown) regular OP of K such that $|\Pi_\bullet^*| \leq m$ and $\sigma_S(\Pi^*) = \mathsf{opt}_{K,m}$. We prove that $\sigma_S(\Pi_{\text{opt}}) \leq (1 + O(1/\sqrt{r}))^{\log_{3/4} m} \cdot \mathsf{opt}_{K,m}$ by induction on m. In the base case where $m \leq r$, we have $\sigma_S(\Pi_{\text{opt}}) \leq \sigma_S(\Pi^*) = \mathsf{opt}_{K,m}$ after the for-loop of Line 2-6 (as argued before). Now suppose $m > r$. If $|\Pi_\bullet^*| \leq r$, then we still have $\sigma_S(\Pi_{\text{opt}}) \leq \mathsf{opt}_{K,m}$ 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 $\sigma_S(\Pi) \leq (1 + O(1/\sqrt{r}))^{\log_{3/4} m} \cdot \mathsf{opt}_{K,m}$. Let $\mathcal{U}$ be the set of connected components of $K \backslash (\bigcup_{R \in \Sigma} R)$, as in Line 10. We have $\Pi = (\bigcup_{R \in \Sigma} \Pi_R) \cup (\bigcup_{U \in \mathcal{U}} \Pi_U)$ where

$\mathit{\Pi}_R = \textsc{AppxPartition}(S, K \cap R, m/r)$ and $\mathit{\Pi}_U = \textsc{AppxPartition}(S, \mathsf{Closure}(U), \frac{3}{4}m)$. Recall that $r = \omega(1)$, and hence $m/r \leq \frac{3}{4}m$. By our induction hypothesis and Lemma 15,

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

which completes the proof. ◀

Algorithm 3 $\textsc{AppxPartition}(S, K, m)$

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

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 $\textsc{AppxPartition}(S, K, m)$ will return a regular partition $\mathit{\Pi}_{\text{opt}}$ of K such that $\sigma_S(\mathit{\Pi}_{\text{opt}}) \leq (1 + \varepsilon) \cdot \mathsf{opt}_{K,m}$. To solve our problem, we only need to call $\textsc{AppxPartition}(S, \mathbb{R}^2, 5n + 1)$, which will return a regular partition $\mathit{\Pi}_{\text{opt}}$ of $\mathbb{R}^2$ such that $\sigma_S(\mathit{\Pi}_{\text{opt}}) \leq (1 + \varepsilon) \cdot \mathsf{opt}_{\mathbb{R}^2, 5n+1}$. By the first statement of Lemma 9, we have $\mathsf{opt}_{\mathbb{R}^2, 5n+1} \leq \mathsf{opt}$. Therefore, it suffices to use the second statement of Lemma 9 to compute a function $f \in \Gamma_g^2$ such that $\sigma_S(f) = \sigma_S(\mathit{\Pi}_{\text{opt}}) \leq (1 + \varepsilon) \cdot \mathsf{opt}$.

Time complexity. If $m \leq r$, the procedure $\textsc{AppxPartition}(S, K, m)$ 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 $\textsc{AppxPartition}(S, K, m)$, we have the recurrence

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

which solves to $T(m) = n^{O(\sqrt{r}\log m + r)}$. Since our initial call is $\textsc{AppxPartition}(S, \mathbb{R}^2, 5n+1)$, 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.*

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, Sariel 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, Sariel 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. Sariel 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*. 1977.
17. Michael S Paterson and F Frances Yao. Optimal binary space partitions for orthogonal objects. *Journal of Algorithms*, 13(1):99–113, 1992.
18. 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.

Appendix

A Missing proofs

A.1 Proof of Lemma 6

Since $(y-y')^2 \geq 0$ for all $y, y' \in \mathbb{R}$, we have $\sum_{i=a'}^{b'} (y_i - f(x_i))^2 \geq \sum_{i=a}^{b} (y_i - f(x_i))^2$ for all $f \in \mathit{\Gamma}_g^1$. Thus, $\delta[a', b'] \geq \delta[a, b]$. To prove the second statement, notice that $\delta[a_{j-1}+1, a_j] \leq \sum_{i=a_{j-1}+1}^{a_j} (y_i - f[a,b](x_i))^2$ for all $j \in [r]$. Therefore,

$$\delta[a,b] = \sum_{i=a}^{b} (y_i - f[a,b](x_i))^2 \geq \sum_{j=1}^{r} \sum_{i=a_{j-1}+1}^{a_j} (y_i - f[a,b](x_i))^2 \geq \sum_{j=1}^{r} \delta[a_{j-1}+1, a_j],$$

which completes the proof.

A.2 Proof of Lemma 9

To see the first statement, let $f \in \mathit{\Gamma}_g^2$ and $R_1, \dots, R_k$ be the pieces of f, which are disjoint rectangles in $\mathbb{R}^2$. Without loss of generality, we may assume that each R_i is a regular rectangle; indeed, we can replace each R_i with the smallest regular rectangle R_i' containing all points $(x_i, x_i') \in R_i$ and one can easily verify that the new rectangles $R_1', \dots, R_k'$ are also disjoint. Furthermore, we may assume that each R_i is nonempty. Consider the *vertical decomposition* of $R_1, \dots, R_k$ defined as follows. For each top (top-left or top-right) vertex of each rectangle R_i, we shoot a upward ray from this vertex, which goes towards the infinity until hitting the boundary of some other rectangle R_j. Similarly, for each bottom (bottom-left or bottom-right) vertex of each rectangle R_i, we shoot a downward ray from this vertex, which goes towards the infinity until hitting the boundary of some other rectangle R_j. The boundaries of $R_1, \dots, R_k$ and the rays cut the plane into a set $\mathit{\Pi}$ of rectangles, which are regular since $R_1, \dots, R_k$ are regular rectangles. See Figure 4 for an illustration. Therefore, $\mathit{\Pi}$ is a regular OP of $\mathbb{R}^2$. Furthermore, $R_1, \dots, R_k \in \mathit{\Pi}$ by our construction. We claim that $|\mathit{\Pi}| \leq 5|f| + 1$ and $\sigma_S(\mathit{\Pi}) \leq \sigma_S(f)$. Since each rectangle R_i has at most four vertices, the total number of rays is at most $4k$. Suppose now we insert these rays one by one. Initially, the boundaries of $R_1, \dots, R_k$ cut the plane into $k+1$ regions. After we insert a ray, the total number of regions can increase at most 1. Therefore, at the end, the total number of regions (i.e., the number of rectangles in $\mathit{\Pi}$) is at most $5k+1$, i.e., $5|f|+1$. To see $\sigma_S(\mathit{\Pi}) \leq \sigma_S(f)$, we may assume $\sigma_S(f) < \infty$, i.e., $(x_i, x_i') \in \bigcup_{j=1}^{k} R_j$ for all $i \in [n]$. With this assumption, the only nonempty rectangles in $\mathit{\Pi}$ are $R_1, \dots, R_k$. Furthermore, by definition, we have $\delta_{R_j} \leq 1 + \sum_{(x_i, x_i') \in R_j} (y_i - f(x_i, x_i'))^2$ for all $j \in [k]$. It follows that

$$\begin{aligned} \sigma_S(\mathit{\Pi}) = \sum_{j=1}^{k} \delta_{R_j} &\leq \sum_{j=1}^{k} \left(1 + \sum_{(x_i, x_i') \in R_j} (y_i - f(x_i, x_i'))^2 \right) \\ &= |f| + \sum_{i=1}^{n} (y_i - f(x_i, x_i'))^2 \\ &= \sigma_S(f). \end{aligned}$$

Next, we prove the second statement of the lemma. Let $\mathit{\Pi}$ be a regular OP of $\mathbb{R}^2$. Suppose $R_1, \dots, R_k \in \mathit{\Pi}$ are the nonempty rectangles in $\mathit{\Pi}$. Note that $R_1, \dots, R_k$ are interior-disjoint. Furthermore, since $R_1, \dots, R_k$ are regular, the points $(x_1, x_1'), \dots, (x_n, x_n')$ are contained in their interiors. Therefore, we can pick $R_j' \subseteq R_j$ for $j \in [k]$ such that $R_1', \dots, R_k'$ are disjoint and R_j' contains the same subset of $\{(x_1, x_1'), \dots, (x_n, x_n')\}$ as R_j. For $j \in [k]$, let

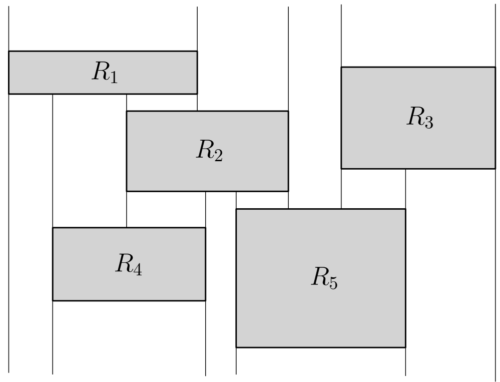

Figure 4 The vertical decomposition induced by the rectangles $R_1, \dots, R_5$

$f_j \in \mathbb{R}[x, x']_g$ be the polynomial that minimizes $\sum_{(x_i, x'_i) \in R'_j} (y_i - f_j(x_i, x'_i))^2$. We then define $f \in \Gamma_g^2$ as the function with pieces $R'_1, \dots, R'_k$ such that $f_{|R'_j} = f_j$ for $j \in [k]$. Clearly, f can be constructed in $n^{O(1)}$ time, because $|\Pi| \leq |\mathcal{R}_{\text{reg}}| = O(n^4)$. Also, one can easily verify from the construction that $\sigma_S(f) = \sigma_S(\Pi)$.

A.3 Proof of Lemma 11

Let Π be a regular OP of $\mathbb{R}^2$. For each $R \in \Pi_\bullet$, the boundary of R consists of (at most) four segments[1], which we call the *boundary segments* of R. Denote by $\mathcal{I}$ the set of the boundary segments of all rectangles in $R \in \Pi_\bullet$. We have $|\mathcal{I}| = O(|\Pi_\bullet|)$. Furthermore, since the rectangles in $\Pi_\bullet$ are interior disjoint, the segments in $\mathcal{I}$ do not cross each other. A classical result of [17] states that for a set of m non-crossing orthogonal segments in the plane, there exists a binary OP of $\mathbb{R}^2$ with $O(m)$ rectangles such that the interior of each rectangle is disjoint with the segments. In addition, according to the construction of [17], the binary OP is regular when the given segments are boundary segments of regular rectangles. Thus, there exists a regular binary OP Π' of $\mathbb{R}^2$ with $|\Pi'| = O(|\Pi_\bullet|)$ such that the interior of R' does not intersect any segment in $\mathcal{I}$ for all $R' \in \Pi'$. It follows that each $R' \in \Pi'$ is either contained in some $R \in \Pi_\bullet$ or interior-disjoint with all $R \in \Pi_\bullet$ and, for any $R' \in \Pi'_\bullet$, the latter case is impossible and we must have the former case, i.e., there exists $R \in \Pi_\bullet$ such that $R' \subseteq R$.

A.4 Proof of Lemma 12

Suppose that for any $R' \in \Pi'_\bullet$ there exists $R \in \Pi_\bullet$ such that $R' \subseteq R$. For a rectangle $R \in \Pi_\bullet$, we write $\Pi'_R = \{R' \in \Pi'_\bullet : R' \subseteq R\}$. Clearly, $\{\Pi'_R : R \in \Pi_\bullet\}$ is a partition of $\Pi'_\bullet$. We claim that $\sigma_S(\Pi'_R) - \delta_R \leq |\Pi'_R| - 1$ for any $R \in \Pi_\bullet$. Let $f \in \mathbb{R}[x, x']_g$ be the polynomial such that $\delta_R = 1 + \sum_{(x_i, x'_i) \in R} (y_i - f(x_i, x'_i))^2$. For any $R' \in \Pi'_R$, we have $\delta'_R \leq 1 + \sum_{(x_i, x'_i) \in R'} (y_i - f(x_i, x'_i))^2$. Note that for each $(x_i, x'_i) \in R$, there exists exactly

[1] Here we mean "generalized" segments including rays or lines.

one rectangle $R' \in \Pi'_R$ such that $(x_i, x'_i) \in R'$. Therefore, we have

$$\begin{aligned}\sigma_S(\Pi'_R) - \delta_R &\le \sum_{R' \in \Pi'_R} \left(1 + \sum_{(x_i,x'_i)\in R'} (y_i - f(x_i, x'_i))^2\right) - \delta_R \\ &= \sum_{R' \in \Pi'_R} \left(1 + \sum_{(x_i,x'_i)\in R'} (y_i - f(x_i, x'_i))^2\right) - \left(1 + \sum_{(x_i,x'_i)\in R} (y_i - f(x_i, x'_i))^2\right) \\ &= |\Pi'_R| - 1.\end{aligned}$$

Thus, $\sigma_S(\Pi') - \sigma_S(\Pi) = \sum_{R\in\Pi_\bullet} \sigma_S(\Pi'_R) - \sum_{R\in\Pi_\bullet} \delta_R \le \sum_{R\in\Pi_\bullet}(|\Pi'_R| - 1) = |\Pi'_\bullet| - |\Pi_\bullet|$.

A.5 Proof of Corollary 14

We shall used 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)** $|\Sigma| \le O(\sqrt{m})$, 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 $1/r(R)$ 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)** $|\Sigma| \le O(\sqrt{r})$, 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 each connected component of $K\backslash(\bigcup_{R\in\Sigma} R)$ intersects at most $\frac{3}{4}|\mathcal{R}|$ rectangles in $\mathcal{R}$. Let U be a connected component of $K\backslash(\bigcup_{R\in\Sigma} R)$. 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 $\frac{2}{3}|\mathcal{R}|$.

B Implementation details of our algorithm for univariate data

Recall that we want to compute $A(b)$ and all $f[a, b], \delta[a, b]$ where $a \in A(b)$ in $O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ time. To this end, we first do some preprocessing such that given a polynomial $f \in \mathbb{R}[x]_g$ and $a, b \in [n]$ with $a \le b$, we can compute $\sum_{i=a}^{b}(y_i - f(x_i))^2$ in $O(1)$ time. For all integers $p, q \ge 0$ such that $p, q \le 2g$, we compute the prefix sums of the sequence $(x_1^p y_1^q, \ldots, x_n^p y_n^q)$ of numbers. This can be done in $O(n)$ time since g is a constant. With these prefix sums, given integers $p, q \ge 0$ with $p, q \le 2g$ and indices $a, b \in [n]$ with $a \le b$, we can compute $\sum_{i=a}^{b} x_i^p y_i^q$ in $O(1)$ time, because $\sum_{i=a}^{b} x_i^p y_i^q = \sum_{i=1}^{b} x_i^p y_i^q - \sum_{i=1}^{a-1} x_i^p y_i^q$. Now observe that for a polynomial $f \in \mathbb{R}[x]_g$ the function $(y - f(x))^2$ is a polynomial of degree at most $2g$ with variables x and y. So we can write $(y - f(x))^2 = \sum_{p+q \leq 2g} e_{p,q} \cdot x^p y^q$ where the coefficients $e_{p,q}$ can be easily computed in $O(1)$ time given f. It follows that for $a, b \in [n]$ with $a \leq b$,

$$\sum_{i=a}^{b} (y_i - f(x_i))^2 = \sum_{p+q \leq 2g} \left(e_{p,q} \cdot \sum_{i=a}^{b} x^p y^q \right).$$

Therefore, with the computed prefix sums, we can compute $\sum_{i=a}^{b} (y_i - f(x_i))^2$ for any given $a, b \in [n]$ with $a \leq b$ in $O(1)$ time. It follows that knowing $f[a, b]$, one can computes $\delta[a, b]$ in $O(1)$ time, because $\delta[a, b] = \sum_{i=a}^{b} (y_i - f[a, b](x_i))^2$.

Now we are able to discuss how to compute all $A(b)$ and all $f[a, b], \delta[a, b]$ where $a \in A(b)$. Specifically, for a number $i \geq 0$ such that $(1 + \tilde{\varepsilon})^{i-1} - 1 \leq 2/\tilde{\varepsilon}$, we want to compute $a_i(b)$ and $f[a_i(b), b], \delta[a_i(b), b]$ for all right break points $b \in [n]$ in $O(n)$ time. We observe that the indices $a_i(b)$ satisfy the following monotonicity: for two right break points $b, b' \in [n]$ where $b \leq b'$, we have $a_i(b) \leq a_i(b')$. This allows us to solve the problem using a simple sliding-window approach shown in Algorithm 4, where Compute(S, i) computes $a_i(b)$ and $f[a_i(b), b], \delta[a_i(b), b]$ for all right break points $b \in [n]$. It is clear that Algorithm 4 runs in $O(n)$ time as long as in the while loop of Line 2-12, we can maintain $f[a, b]$ and $\delta[a, b]$ in $O(1)$ time whenever a or b changes. As discussed above, with our preprocessing, one can computes $\delta[a, b]$ in $O(1)$ time given $f[a, b]$. Therefore, our actual task here is to maintain $f[a, b]$ in $O(1)$ time. We observe that each change of a and b in the while loop of Line 2-12 is either $a \leftarrow a - 1$ or $b \leftarrow b - 1$. To maintain $f[a, b]$, we need the expression for $f[a, b]$ in terms of the points $(x_a, y_a), \ldots, (x_b, y_b)$. For a $(g + 1)$-dimensional vector $\beta = (\beta_0, \ldots, \beta_g)$, we define $\mathsf{poly}[\beta] \in \mathbb{R}[x]_g$ as the polynomial $\sum_{j=0}^{g} \beta_j \cdot x^j$. Also, we define

$$\mathbf{X}_{a,b} = \begin{pmatrix} 1 & x_a & \cdots & x_a^g \\ 1 & x_{a+1} & \cdots & x_{a+1}^g \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_b & \cdots & x_b^g \end{pmatrix} \text{ and } \mathbf{y}_{a,b} = (y_a, \ldots, y_b)^T.$$

It is well known that $f[a, b] = \mathsf{poly}[\beta_{a,b}]$ where $\beta_{a,b} = (\mathbf{X}_{a,b}^T \mathbf{X}_{a,b})^{-1} (\mathbf{X}_{a,b}^T \mathbf{y}_{a,b})$. Note that $\mathbf{X}_{a,b}^T \mathbf{X}_{a,b}$ is a $(g+1) \times (g+1)$ matrix and $\mathbf{X}_{a,b}^T \mathbf{y}_{a,b}$ is a $(g+1)$-dimensional vector. Furthermore, $\mathbf{X}_{a,b}^T \mathbf{X}_{a,b}$ and $\mathbf{X}_{a,b}^T \mathbf{y}_{a,b}$ can be easily maintained in $O(1)$ time for the operations $a \leftarrow a - 1$ and $b \leftarrow b - 1$ (simply by modifying each of their entries). With $\mathbf{X}_{a,b}^T \mathbf{X}_{a,b}$ and $\mathbf{X}_{a,b}^T \mathbf{y}_{a,b}$ in hand, $\beta_{a,b}$ and $f[a, b]$ can be directly computed in $O(1)$ time. This allows us to maintain $f[a, b]$ in $O(1)$ time in the while loop of Line 2-12. As a result, we obtain a linear-time approximation scheme for piecewise polynomial regression for univariate data, assuming the data points are pre-sorted.

C A sub-exponential time exact algorithm for bivariate data

We present a simple exact algorithm for piecewise polynomial regression for bivariate data, which runs in $n^{O(\sqrt{n})}$ time. Our algorithm first computes a regular OP Π of the plane such that $\sigma_S(\Pi) \leq \sigma_S(\Pi')$ for all regular OP Π' of the plane satisfying $|\Pi'| \leq 5n + 1$, and then uses the second statement of Lemma 9 to compute a function $f \in \Gamma_g^2$ such that $\sigma_S(f) = \sigma_S(\Pi)$ in $O(n \cdot |\Pi|) = O(n^2)$ time. We claim that $\sigma_S(f) = \mathsf{opt}$. It is clear that $\sigma_S(f) \geq \mathsf{opt}$. To see $\sigma_S(f) \leq \mathsf{opt}$, it suffices to show $\sigma_S(\Pi) \leq \mathsf{opt}$. Let $f^* \in \Gamma_g^2$ be the function such that $\sigma_S(f^*) = \mathsf{opt}$. Note that $|f^*| \leq n$, for otherwise f_{opt} has an "empty" piece which can be removed to make $\sigma_S(f^*)$ smaller. Therefore, by the first statement of Lemma 9,

Algorithm 4 COMPUTE(S, i) ▷ Computing $a_i(b)$, $f[a_i(b), b]$, $\delta[a_i(b), b]$

$b \leftarrow n$ and $a \leftarrow b$
while $b \geq 1$ **do**
    if b is a right break point **then**
        while $\delta[a, b] \leq (1 + \tilde{\varepsilon})^i - 1$ **do**
            if a is a left break point **then**
                $a_i(b) \leftarrow a$
                associate $f[a, b]$ and $\delta[a, b]$ with $a_i(b)$
            $a \leftarrow a - 1$
    $b \leftarrow b - 1$
    if $a > b$ **then** $a \leftarrow a - 1$

there exists a regular OP Π^* of $\mathbb{R}^2$ with $|\Pi^*| \leq 5n + 1$ such that $\sigma_S(\Pi^*) = \sigma_S(f^*) = \mathsf{opt}$. By the property of Π, we further have $\sigma_S(\Pi) \leq \sigma_S(\Pi^*) = \mathsf{opt}$. Hence, $\sigma_S(f) = \mathsf{opt}$.

Note that a set Π of interior-disjoint rectangles naturally induces a planar graph in which the vertices are the rectangles in Π and two vertices are connected by an edge if the two corresponding rectangles are neighboring to each other, i.e., their boundaries intersect at a segment (rather than a single point). The basic idea of our algorithm is to use the planar separator theorem, which states that one can partition the vertex set of a planar graph with m vertices into three parts V_1, V_2, Σ such that **(i)** there is no edge between V_1 and V_2, **(ii)** $|\Sigma| \leq 4\sqrt{m}$, and **(iii)** $|V_1| \leq \frac{2}{3}m$ and $|V_2| \leq \frac{2}{3}m$; the set Σ is called a *balanced separator*.

Algorithm 5 OPTPARTITION(S, K, m)

if $m \leq 10$ **then**
    solve the problem by brute-force
else
    $\mathcal{R}_K \leftarrow \{R \in \mathcal{R}_{\text{reg}} : R \subseteq K\}$, $\Pi_{\text{opt}} \leftarrow \emptyset$, opt $\leftarrow \infty$
    for all $\Sigma \subseteq \mathcal{R}_K$ with $|\Sigma| \leq 4\sqrt{m}$ **do**
        if the rectangles in Σ are interior-disjoint **then**
            $\mathcal{U} \leftarrow \mathsf{Components}(K \backslash (\bigcup_{R \in \Sigma} R))$
            $\Pi_U \leftarrow$ OPTPARTITION$(S, \mathsf{Closure}(U), \frac{2}{3}m)$ for all $U \in \mathcal{U}$
            $\Pi \leftarrow \Sigma \cup (\bigcup_{U \in \mathcal{U}} \Pi_U)$
            if $\sigma_S(\Pi) <$ opt **then** $\Pi_{\text{opt}} \leftarrow \Pi$ and opt $\leftarrow \sigma_S(\Pi)$
    return Π_{opt}

Let $K \subseteq \mathbb{R}^2$ be a regular region. Suppose we want to compute a regular OP Π of K such that $\sigma_S(\Pi) \leq \sigma_S(\Pi')$ for all regular OP Π' of K satisfying $|\Pi'| \leq m$. Note that we do *not* require $|\Pi| \leq m$. If $m = O(1)$, we solve the problem in $n^{O(m)} = n^{O(1)}$ time by brute-force: enumerating every set Π of at most m regular rectangles, checking if Π is a partition of K, and computing $\sigma_S(\Pi)$. Otherwise, we solve the problem as follows. Let Π^* be an (unknown) optimal regular OP of K with up to m rectangles, that is, $|\Pi^*| \leq m$ and $\sigma_S(\Pi^*) \leq \sigma_S(\Pi')$ for all regular OP Π' of K satisfying $|\Pi'| \leq m$. We guess a balanced separator Σ of the planar graph G_{Π^*} induced by Π^*, which corresponds to at most $4\sqrt{m}$ (interior-disjoint) regular rectangles in K (for convenience, we use the same notation Σ to denote the set of these rectangles). This separator separates the other vertices of G_{Π^*} into two subsets V_1 and V_2 of size at most $\frac{2}{3}m$ such that there is no edge between V_1 and V_2. Suppose our guess for Σ is correct, and consider the set $\mathcal{U}$ of connected components of $K \backslash (\bigcup_{R \in \Sigma} R)$. Each component $U \in \mathcal{U}$ contains some rectangles in $\Pi^* \backslash \Sigma$, whose corresponding vertices in G_{Π^*} induce a *connected* subgraph of G_{Π^*}. Therefore, these rectangles either all belong to V_1 or all belong to V_2. Because $|V_1| \leq \frac{2}{3}m$ and $|V_2| \leq \frac{2}{3}m$, the number of the rectangles in Π^* contained in U is at most $\frac{2}{3}m$. We recursively compute a regular OP Π_U for (the closure of) U such that $\sigma_S(\Pi_U) \leq \sigma_S(\Pi')$ for all regular OP Π' of (the closure of) U satisfying $|\Pi'| \leq \frac{2}{3}m$. Then we set $\Pi = \Sigma \cup (\bigcup_{U \in \mathcal{U}} \Pi_U)$, which is clearly a regular OP of K. We claim that, if our guess for Σ is correct, then $\sigma_S(\Pi) \leq \sigma_S(\Pi^*)$, and hence Π satisfies the desired property. Let $\Pi_U^* \subseteq \Pi^*$ be the subset of rectangles contained in U, for $U \in \mathcal{U}$. We know that $|\Pi_U^*| \leq \frac{2}{3}m$. Therefore, by the property of Π_U, we have $\sigma_S(\Pi_U) \leq \sigma_S(\Pi_U^*)$. It follows that

$$\sigma_S(\Pi) = \sigma_S(\Sigma) + \sum_{U \in \mathcal{U}} \sigma_S(\Pi_U) \leq \sigma_S(\Sigma) + \sum_{U \in \mathcal{U}} \sigma_S(\Pi_U^*) = \sigma_S(\Pi^*).$$

The entire algorithm is shown in Algorithm 5, where $\textsc{OptPartition}(S, K, m)$ computes a regular OP Π of the regular region K such that $\sigma_S(\Pi) \leq \sigma_S(\Pi')$ for all regular OP Π' of K satisfying $|\Pi'| \leq m$. The correctness of the algorithm follows directly from the discussion above. To solve our problem, we simply call $\textsc{OptPartition}(S, \mathbb{R}^2, 5n+1)$.

Time complexity. One easily verifies that in all recursive calls of $\textsc{OptPartition}(S, K, m)$, the region K is always a regular region (recall that a regular region is a subset of $\mathbb{R}^2$ that is the union of some regular rectangles) and hence the complexity of K is bounded by a polynomial in n. Therefore, the size of the set $\mathcal{U}$ computed in Line 7 of Algorithm 5 is also bounded by $n^{O(1)}$ in all recursive calls. Furthermore, since $|\mathcal{R}_{\text{reg}}| = O(n^4)$, the number of all subsets $\Sigma \subseteq \mathcal{R}_K$ with $|\Sigma| \leq 4\sqrt{m}$ considered in Line 5 is $n^{O(\sqrt{m})}$. It then follows that in a call $\textsc{OptPartition}(S, K, m)$, the total number of recursive calls made in Line 8 is bounded by $n^{O(\sqrt{m})}$ and all steps except the recursive calls can be done in $n^{O(1)}$ time. So if we use $T(m)$ to denote the time cost for the call $\textsc{OptPartition}(S, K, m)$, we have the recurrence $T(m) \leq n^{O(\sqrt{m})} \cdot (T(\frac{2}{3}m) + n^{O(1)})$. Solving this recurrence gives us $T(m) = n^{O(\sqrt{m})}$, which implies that the initial call $\textsc{OptPartition}(S, \mathbb{R}^2, 5n+1)$ takes $n^{O(\sqrt{n})}$ time.

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

D NP-hardness for bivariate data

In this section, we show that the piecewise-polynomial regression problem in $\mathbb{R}^d$ for $d \geq 2$ is NP-hard. This result is widely believed in the folklore, but we could not find a published record in the literature. So we give a proof for completeness.

Our reduction is from the planar rectilinear 3-SAT problem. A *planar rectilinear representation* of a 3-CNF boolean formula ϕ represents ϕ using horizontal and vertical segments in the plane in the following way. Each variable of ϕ is represented as a horizontal segment on the x-axis while each clause is represented a horizontal segment above the x-axis. Whenever a clause includes a variable, there is a vertical segment connecting two horizontal segments corresponding to the clause and the variable respectively. The vertical connections can be negative or positive according to whether the literal is negated or not. All segments are disjoint except that each vertical segment intersects with the two horizontal segments it connects. See Figure 5 for an illustration of planar rectilinear representation. In the planar rectilinear 3-SAT problem, the input of a 3-CNF boolean formula ϕ with its planar rectilinear representation, and the goal is to test if ϕ is satisfiable.

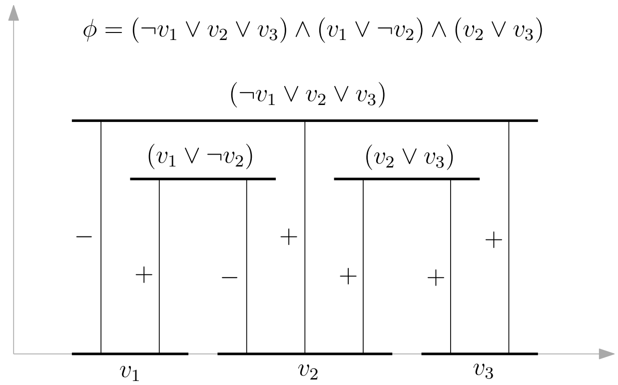

Figure 5 The planar rectilinear representation of $\phi = (\neg v_1 \vee v_2 \vee v_3) \wedge (v_1 \vee \neg v_2) \wedge (v_2 \vee v_3)$.

In order to describe our reduction, we introduce an intermediate problem called *piecewise polynomial perfect fitting* (PPPF), which is a variant of the piecewise-polynomial regression problem. Let $g \geq 0$ be a fixed integer and $\mathcal{R}$ be the family of orthogonal boxes in $\mathbb{R}^d$. In the PPPF problem, we are given a set $S = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}\}_{i=1}^n$ of data points, and our goal is to find a function $f \in \Gamma_{\mathcal{R}}^g$ with minimum number of pieces (i.e., minimum $|f|$) such that f *perfectly fits* S, i.e., $y_i = f(\mathbf{x}_i)$ for all $i \in [n]$.

▶ **Lemma 17.** *The PPPF problem in $\mathbb{R}^d$ with maximum degree g can be reduced in polynomial time to the piecewise polynomial regression problem in $\mathbb{R}^d$ with maximum degree g.*

Proof. Given a dataset $S = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}\}_{i=1}^n$, we reduce the PPPF problem on S (with maximum degree g) to an instance $\langle S, \lambda \rangle$ of piecewise polynomial regression (with maximum degree g). The only thing we have to determine is the parameter λ. Intuitively, we need to let λ be sufficiently small so that when evaluating the price of a function in Γ_g^d, the least square error is always more important than the number of pieces. For an axis-parallel box B in $\mathbb{R}^d$, we use err_B to denote the minimum $\sum_{\mathbf{x}_i \in B}(y_i - f(\mathbf{x}_i))^2$ for a d-variable polynomial function f with degree at most g. Let $\mathcal{B}$ be the set of *combinatorially different* boxes in $\mathbb{R}^d$, where two boxes B and B' are combinatorially different if $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \cap B \neq \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \cap B'$. Then we set λ to be a positive number smaller than err_B/n for all $B \in \mathcal{B}$ such that $\mathsf{err}_B > 0$. Since $|\mathcal{B}| = O(n^{2d})$, we can compute λ in polynomial time. We claim that the optimum of the PPPF instance $\langle S \rangle$ is k iff the optimum of the piecewise polynomial regression instance $\langle S, \lambda \rangle$ is λk. Suppose the optimum of the PPPF instance $\langle S \rangle$ is k. Then there exists a function $f \in \Gamma_{\mathcal{R}}^g$ with $|f| = k$ which perfectly fits S. Because of the existence of f, the optimum of the piecewise polynomial regression instance $\langle S, \lambda \rangle$ is at most λk. Furthermore, for any $k' < k$ disjoint boxes $B_1, \dots, B_{k'}$ such that $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subseteq \bigcup_{j=1}^{k'} B_j$, we have $\sum_{j=1}^{k'} \mathsf{err}_{B_j} > 0$; indeed, if $\sum_{j=1}^{k'} \mathsf{err}_{B_j} = 0$, then there exists a function in $\Gamma_{\mathcal{R}}^g$ with less than k pieces which perfectly fits S. It follows that for any $k' < k$ disjoint boxes $B_1, \dots, B_{k'}$ such that $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subseteq \bigcup_{j=1}^{k'} B_j$, we have $\sum_{j=1}^{k'} \mathsf{err}_{B_j} > \lambda n \geq \lambda k$. Therefore, $\sigma_S(f) \geq \lambda k$ for any $f \in \Gamma_{\mathcal{R}}^g$ with $|f| < k$. On the other hand, $\sigma_S(f) \geq \lambda k$ for any $f \in \Gamma_{\mathcal{R}}^g$ with $|f| > k$. So the optimum of the piecewise polynomial regression instance $\langle S, \lambda \rangle$ is λk. This completes the "only if" part of the claim. To see the "if" part, assume the optimum of the PPPF instance $\langle S \rangle$ is $k' \neq k$. Then the optimum of the piecewise polynomial regression instance $\langle S, \lambda \rangle$ is $\lambda k' \neq \lambda k$. This reduces the PPPF problem to piecewise polynomial regression. ◀

Next, we show how to reduce planar rectilinear 3-SAT to the PPPF problem in $\mathbb{R}^2$. For simplicity, we present the details of the reduction for the PPPF problem with maximum degree $g = 0$, and it can be easily generalized to a general g. When $g = 0$, the functions in $\Gamma^g_{\mathcal{R}}$ are piecewise constant functions.

Consider a given 3-CNF boolean formula ϕ and its planar rectilinear representation. Suppose ϕ has n variables and m clauses. We shall construct a set $S = \{((x_i, x'_i), y_i) \in \mathbb{R}^2 \times \mathbb{R}\}_{i=1}^N$ and determine a number k such that there exists a function $f \in \Gamma^2_0$ with $|f| \leq k$ such that $y_i = f(x_i, x'_i)$ for all $i \in [n]$ iff ϕ is satisfiable. Our set S consists of two types of points: *normal points* and *obstacle points*. We denote by S_1 the set of normal points and by S_2 the set of obstacle points. The y-coordinates of all points in S_1 are equal to 0, while all points in S_2 have nonzero distinct y-coordinates. Therefore, if a function $f \in \Gamma^2_0$ perfectly fits S, then each piece of f either covers only points in S_1 or covers a single point in S_2. It follows that the optimum (i.e., the minimum number of pieces of a function $f \in \Gamma^2_0$ that perfectly fits S) is exactly equal to $k_1 + |S_2|$, where k_1 is the minimum number of disjoint rectangles that cover all points in S_1 but do not contain (the xx'-projection images of) any points in S_2.

We first determine the x-coordinates and x'-coordinates of the normal points, i.e., the points in S_1. Let $v_1, \dots, v_n$ be the n variables of ϕ, $c_1, \dots, c_m$ be the clauses of ϕ, m_i be the number of clauses of ϕ that contains the variable v_i for $i \in [n]$. Define $L_+ = \{(i, j) :$ the clause c_j contains the literal $v_i\}$, $L_- = \{(i, j) :$ the clause c_j contains the literal $\neg v_i\}$, and $L = L_+ \cup L_-$. Without loss of generality, we can assume that $m_i \geq 2$ for all $i \in [n]$ (indeed, if a variable is only contained in one clause of ϕ, then we can choose the value of that variable to satisfy that clause and remove the clause and the variable from ϕ without changing the satisfiability of ϕ). Also, we may assume that each clause c_j has two or three literals (indeed, if a clause only has one literal, then we must choose the value of the variable corresponding to the literal to make this clause true and hence we can remove the clause and the variable from ϕ without changing the satisfiability of ϕ). Suppose the planar rectilinear representation of ϕ is given in the xx'-plane. In the representation, each variable v_i corresponds to a horizontal segment $\mathsf{seg}(v_i)$ on the x-axis, which each clause c_j corresponds to a horizontal segment $\mathsf{seg}(c_j)$ above the x-axis. We denote by $\mathsf{seg}(v_i, c_j)$ the vertical segment that connects the horizontal segments $\mathsf{seg}(v_i)$ and $\mathsf{seg}(v_j)$, for $(i, j) \in L$.

First, we replace each variable segment $\mathsf{seg}(v_i)$ with an *indented rectangle* D_i with m_i *peaks*. See the top two figures in Figure 6 for an illustration of the indented rectangles and peaks. On each vertex of D_i and the midpoint of each edge of D_i, we put a normal point. Therefore, we have in total $8m_i + 8$ normal points on D_i. See the bottom-left figure in Figure 6 for an illustration. For technical reasons, we rotate the indented rectangle D_i a little bit so that the normal points on D_i have distinct x- and x'-coordinates (see the bottom-right figure in Figure 6). We also use the notation D_i to denote the set of the $8m_i + 8$ normal points on the indented rectangle D_i for convenience. After we replace the variable segments with the indented rectangles, we let the vertical segments $\mathsf{seg}(v_i, c_j)$ for $(i, j) \in L$ connect to the peaks of the indented rectangles (each D_i has m_i peaks which one-to-one correspond to the m_i vertical segments incident to $\mathsf{seg}(v_i)$). We denote by $p_{i,j}$ the normal point on the peak of D_i that connects to $\mathsf{seg}(v_i, c_j)$, and denote by $p^-_{i,j}$ and $p^+_{i,j}$ the left and right adjacent points of $p_{i,j}$ in D_i.

Now we consider the clause segments $\mathsf{seg}(c_j)$ and the vertical segments $\mathsf{seg}(v_i, c_j)$. For a clause c_j with three literals, its *left, middle, right* variables refer to the variables corresponding to the left, middle, right vertical segments connecting to the clause segment $\mathsf{seg}(c_j)$, respectively. If a clause has only two literals, then it only has left and right variables. For

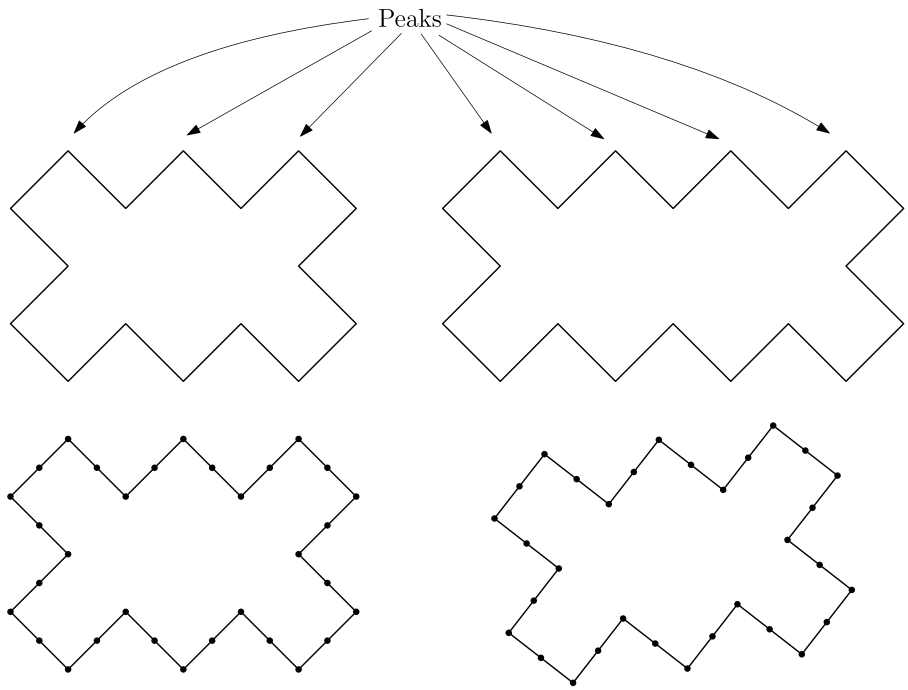

Figure 6 Indented rectangles with three (top-left) and four (top-right) peaks. We put on each vertex and the midpoint of each edge a normal point (bottom-left). We rotate the indented rectangle a little bit such that the normal points have distinct coordinates (bottom-right).

each vertical segment $\mathsf{seg}(v_i, c_j)$, we add two normal points $a_{i,j}$ and $b_{i,j}$ as follows. The point $a_{i,j}$ is very close to the midpoint of $\mathsf{seg}(v_i, c_j)$: if $(i, j) \in L_+$, then $a_{i,j}$ is slightly to the right of the midpoint; if $(i, j) \in L_-$, then $a_{i,j}$ is slightly to the left of the midpoint. The point $b_{i,j}$ is very close to the connecting point $e_{i,j}$ of $\mathsf{seg}(v_i, c_j)$ and $\mathsf{seg}(c_j)$: if $(i, j) \in L_+$, then $b_{i,j}$ is slightly to the southwest (or bottom-left) of $e_{i,j}$; if $(i, j) \in L_-$, then $b_{i,j}$ is slightly to the southeast (or bottom-right) of $e_{i,j}$. In addition, we slightly move the points $b_{i,j}$ vertically such that the following condition holds: for a clause c_j, we have $x'(b_{\mathsf{mid},j}) < \min\{x'(b_{\mathsf{left},j}), x'(b_{\mathsf{right},j})\}$ and $x'(b_{\mathsf{left},j}) \neq x'(b_{\mathsf{right},j})$, where $x'(\cdot)$ denotes the x'-coordinate and $v_{\mathsf{left}}, v_{\mathsf{mid}}, v_{\mathsf{right}}$ are the left, middle, right variables of c_j, respectively; if c_j only has two literals, then we only require $x'(b_{\mathsf{left},j}) \neq x'(b_{\mathsf{right},j})$. In other words, for each clause, we require that the b-points of its variables have distinct x'-coordinates and the b-point of its middle variable is always the lowest. Finally, for each clause c_j, we put a normal point s_j on the segment $\mathsf{seg}(c_j)$, whose x-coordinate is equal to the x-coordinate of $b_{\mathsf{mid},j}$, where v_{mid} is the mid variable of c_j; if c_j only has two literals, then we put s_j on the midpoint of $\mathsf{seg}(c_j)$. See Figure 7 for an illustration of the locations of the points $a_{i,j}, b_{i,j}, c_j$. Setting $S_1 = (\bigcup_{i=1}^{n} D_i) \cup (\bigcup_{(i,j)\in L} \{a_{i,j}, b_{i,j}\}) \cup (\bigcup_{j=1}^{m} \{s_j\})$, we finish the construction of the normal points.

Next, we describe the obstacle points, i.e., the points in S_2. As observed before, the minimum number of pieces of a function $f \in \mathit{\Gamma}_0^2$ that perfectly fits S is equal to $k_1 + |S_2|$, where k_1 is the minimum number of disjoint rectangles that cover all points in S_1 but do not contain (the xx'-projection images of) any points in S_2. Without any obstacle points, $k_1 = 1$ because we can cover all points in S_1 using a single rectangle. So we want to use the obstacle points to "force" a rectangle to only cover some certain subset of S_1 (in order to avoid the obstacle points). To this end, we first specify which subsets of S_1 we allow a

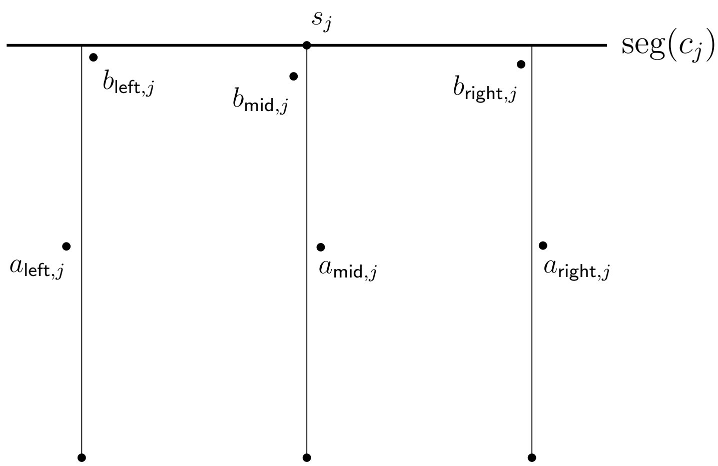

Figure 7 An illustration of the points $a_{i,j}, b_{i,j}$ and c_j. The clause c_j has a negated literal for its left variable and positive literals for its middle and right variables.

rectangle to cover. Recall that $p_{i,j}$ is the normal point on the peak of D_i that connects to $\mathsf{seg}(v_i, c_j)$, and $p^-_{i,j}$ and $p^+_{i,j}$ are the left and right adjacent points of $p_{i,j}$ in D_i. We define a collection of *legal subsets* of S_1 as follows.

(1) For $i \in [n]$, each pair of adjacent normal points in D_i form a legal subset.
(2) For $(i,j) \in L$, $\{s_j, b_{i,j}\}$, $\{a_{i,j}, b_{i,j}\}$, $\{p_{i,j}, a_{i,j}\}$ are legal subsets.
(3) For $(i,j) \in L_+$, $\{p_{i,j}, p^+_{i,j}, a_{i,j}\}$ and $\{p^+_{i,j}, a_{i,j}\}$ are a legal subset.
(4) For $(i,j) \in L_-$, $\{p_{i,j}, p^-_{i,j}, a_{i,j}\}$ and $\{p^-_{i,j}, a_{i,j}\}$ are a legal subset.
(5) Each single point in S_1 forms a legal subset.

▶ **Lemma 18.** *The boolean formula ϕ is satisfiable iff S_1 can be partitioned into at most $5|L| + 4n$ legal subsets.*

Proof. To show the "if" part, assume S_1 can be partitioned into at most $5|L| + 4n$ legal subsets. Let $\mathcal{P}$ be such a partition, i.e., $\mathcal{P}$ is a collection of at most $5|L| + 4n$ disjoint legal subsets that cover all points in S_1. We want to construct a satisfying assignment $\mathcal{A} : \{v_1, \dots, v_n\} \to \{\mathsf{true}, \mathsf{false}\}$ of ϕ. Define V as the set consisting of all *vertex* points of $D_1, \dots, D_n$ and all $b_{i,j}$ for $(i,j) \in L$. Similarly, define E as the set consisting of all *edge* points of $D_1, \dots, D_n$ and all $b_{i,j}$ for $(i,j) \in L$. We have $|V| = |E| = 5|L| + 4n$. Observe that any legal subset can cover at most one point in V (resp., E). This implies $|\mathcal{P}| \geq 5|L| + 4n$ and hence $|\mathcal{P}| = 5|L| + 4n$ Since $|\mathcal{P}| = |V|$ (resp., $|\mathcal{P}| = |E|$) and $\mathcal{P}$ covers all points in V (resp., E), every legal subset in $\mathcal{P}$ covers exactly one point in V (resp., E). We shall use this property to obtain the assignment $\mathcal{A}$ and prove it is a satisfying assignment. Consider a vertex point α of some D_i. Since $\alpha \in V \backslash E$, the legal subset in $\mathcal{P}$ that contains α must contain another point in $E \backslash V$, which can only be one of the two edge points adjacent to α in D_i. In other words, in the partition $\mathcal{P}$, every vertex point is *coupled* with an adjacent edge point (i.e., they belong to the same legal subset in $\mathcal{P}$). Furthermore, observe that if a vertex point of D_i is coupled with its clockwise (resp., counterclockwise) adjacent edge point, then *every* vertex point of D_i must be coupled with its clockwise (resp., counterclockwise) adjacent edge point. We now define our assignment $\mathcal{A}$ as follows. For all $i \in [n]$ such that every vertex point of D_i is coupled with its clockwise (resp., counterclockwise) adjacent edge point, we set $\mathcal{A}(v_i) = \mathsf{true}$ (resp., $\mathcal{A}(v_i) = \mathsf{false}$). We show $\mathcal{A}$ is a satisfying assignment by contradiction. Assume that $\mathcal{A}$ is not satisfying. Without loss of generality, we may assume that c_1 is an unsatisfied clause. Since $s_1 \notin V$, the legal subset in $\mathcal{P}$ that contains s_1 should contain another point in V, which must be $b_{i,1}$ for some $i \in [n]$ satisfying $(i, 1) \in L$. We consider the case where $(i, 1) \in L_+$, and the other case $(i, 1) \in L_-$ can be handled in the same way. Because c_1 is unsatisfied and $(i, 1) \in L_+$, we have $\mathcal{A}(v_i) = \mathsf{false}$. Therefore, each vertex point of D_i is coupled with its counterclockwise adjacent edge point; in particular, $p_{i,1}$ is coupled with $p_{i,1}^-$. This implies $\{p_{i,1}, p_{i,1}^+, a_{i,1}\} \notin \mathcal{P}$. Also, we have $\{p_{i,1}^+, a_{i,1}\} \notin \mathcal{P}$ (resp., $\{p_{i,1}, a_{i,1}\} \notin \mathcal{P}$), because every legal subset in $\mathcal{P}$ must contain one point in V (resp., E). Finally, we have $\{a_{i,1}, b_{i,1}\} \notin \mathcal{P}$, since $\{s_1, b_{i,1}\} \in \mathcal{P}$ and the legal subsets in $\mathcal{P}$ are disjoint. Now all legal subsets that contain the point $a_{i,1}$ are not in $\mathcal{P}$, contradicting the fact that $\mathcal{P}$ covers all points in S_1. As a result, $\mathcal{A}$ is a satisfying assignment.

To show the "only if" part, assume ϕ is satisfiable and let $\mathcal{A} : \{v_1, \dots, v_n\} \to \{\mathsf{true}, \mathsf{false}\}$ be a satisfying assignment of ϕ. We shall partition S_1 into $5|L| + 4n$ legal subsets. For each variable v_i such that $\mathcal{A}(v_i) = \mathsf{true}$, we construct $4m_i + 4$ (disjoint) legal subsets as follows. We first group each vertex point in D_i with its *clockwise* adjacent point in D_i (which is an edge point). In this way, we obtain $4m_i + 4$ legal subsets of size 2 which cover all normal points in D_i, where each peak $p_{i,j}$ is contained in the legal subset $\{p_{i,j}, p_{i,j}^+\}$. We then replace $\{p_{i,j}, p_{i,j}^+\}$ with the legal subset $\{p_{i,j}, p_{i,j}^+, a_{i,j}\}$ for all $j \in [m]$ such that $(i, j) \in L_+$. After this, we obtain $4m_i + 4$ legal subsets which are disjoint and cover all normal points in D_i and all $a_{i,j}$ for $j \in [m]$ satisfying $(i, j) \in L_+$. For each variable v_i such that $\mathcal{A}(v_i) = \mathsf{false}$, we construct $4m_i + 4$ (disjoint) legal subsets similarly. We first group each vertex point in D_i with its *counterclockwise* adjacent point in D_i, which gives us $4m_i + 4$ legal subsets covering all normal points in D_i where each peak $p_{i,j}$ is contained in the legal subset $\{p_{i,j}, p_{i,j}^-\}$. Then we replace $\{p_{i,j}, p_{i,j}^-\}$ with the legal subset $\{p_{i,j}, p_{i,j}^-, a_{i,j}\}$ for all $j \in [m]$ such that $(i, j) \in L_-$. After considering all variables $v_1, \dots, v_n$, we obtain in total $\sum_{i=1}^n (4m_i + 4) = 4|L| + 4n$ (disjoint) legal subsets. For convenience, we denote by $\mathcal{P}_1$ the collection of these legal subsets. Then $\mathcal{P}_1$ cover all normal points in $D_1, \dots, D_n$ and all $a_{i,j}$ for $(i, j) \in L_+$ such that $\mathcal{A}(v_i) = \mathsf{true}$ and for $(i, j) \in L_-$ such that $\mathcal{A}(v_i) = \mathsf{false}$. Next, we construct another collection $\mathcal{P}_2$ of $|L|$ (disjoint) legal subsets that cover all points in S_1 that are not covered by $\mathcal{P}_1$. First, for each clause c_j, pick an index $i_j \in [n]$ such that $(i_j, j) \in L$ and the literal of v_{i_j} in c_j makes c_j true under the assignment $\mathcal{A}$ (such an index i_j always exists since $\mathcal{A}$ is a satisfying assignment). Observe that the points $a_{i_1,1}, \dots, a_{i_m,m}$ are all covered by $\mathcal{P}_1$. We include in $\mathcal{P}_2$ the legal subsets $\{s_1, b_{i_1,1}\}, \dots, \{s_m, b_{i_m,m}\}$. Also, for each $(i, j) \in L \backslash \{(i_1, 1), \dots, (i_m, m)\}$, we include in $\mathcal{P}_2$ the legal subset $\{b_{i,j}\}$ if $a_{i,j}$ is covered by $\mathcal{P}_1$ or the legal subset $\{a_{i,j}, b_{i,j}\}$ if $a_{i,j}$ is not covered by $\mathcal{P}_1$. In this way, we obtain the collection $\mathcal{P}_2$ of $|L|$ legal subsets. Let $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$. It is easy to verify that **(1)** $|\mathcal{P}| = 5|L| + 4n$, **(2)** the legal subsets in $\mathcal{P}$ are disjoint, and **(3)** the legal subsets in $\mathcal{P}$ cover all points in S_1. This completes the "only if" part. ◀

With the above lemma in hand, the last step of our reduction is to use obstacle points to block all "illegal" subsets such that the pieces of a function $f \in \mathit{\Gamma}_0^2$ that perfectly fits S can only cover legal subsets (or a single obstacle point). Let U be the union of the minimum enclosing rectangles of the legal subsets. The locations of the normal points we pick guarantee the following property of legal subsets.

▶ **Fact 19.** *The minimum enclosing rectangle of a legal subset P only contains the normal points in P. Furthermore, the minimum enclosing rectangle of any illegal subset of S_1 is not contained in U.*

Proof. The first statement directly follows from how we locate the normal points. A remarkable case here is the legal subsets $\{s_j, b_{i,j}\}$ for $(i, j) \in L$. Recall that in our construction,

$x'(b_{\mathsf{mid},j}) < \min\{x'(b_{\mathsf{left},j}), x'(b_{\mathsf{right},j})\}$ and $x'(b_{\mathsf{left},j}) \neq x'(b_{\mathsf{right},j})$, where $x'(\cdot)$ denotes the x'-coordinate and $v_{\mathsf{left}}, v_{\mathsf{mid}}, v_{\mathsf{right}}$ are the left, middle, right variables of c_j, respectively. This property guarantees the minimum enclosing rectangles of $\{s_j, b_{\mathsf{left},j}\}$, $\{s_j, b_{\mathsf{mid},j}\}$, and $\{s_j, b_{\mathsf{right},j}\}$ only contains the normal points in $\{s_j, b_{\mathsf{left},j}\}$, $\{s_j, b_{\mathsf{mid},j}\}$, and $\{s_j, b_{\mathsf{right},j}\}$, respectively.

To see the second statement, it suffices to check for all *minimal* illegal subsets of S_1. Note that in our construction, every minimal illegal subset consists of two points in S_1. Thus, the statement follows from a simple but tedious case-by-case check for every pair of points in S_1 that do not form a legal subset. A remarkable case here is the illegal subsets formed by two normal points in D_i for some $i \in [n]$. Recall that when we replace each variable segment $\mathsf{seg}(v_i)$ with the indented rectangle D_i, we rotate D_i a little bit such that the normal points in D_i have distinct x- and x'-coordinates. The purpose of this rotation is just to guarantee that the minimum enclosing rectangle of any two non-adjacent normal points in D_i is not contained in U. (Without the rotation, the minimum enclosing rectangle of any two edge points in D_i with distance 2 is a segment and is contained in U. However, with the rotation, this is no longer the case.) We omit the tedious details here. ◀

Note that although the number of subsets of S_1 is exponential, the number of different minimum enclosing rectangles of these subsets is bounded by $|S_1|^4$ and these rectangles can be computed efficiently. For every minimum enclosing rectangle R that is not contained in U, we include in S_2 an obstacle point whose xx'-projection image is in $R\backslash U$. Then any rectangle in the xx'-plane that does not contain (the xx'-projection images of) any points in S_2 can only cover a legal subset of S_1. Therefore, by Lemma 18, $k_1 \leq 5|L| + 4n$ iff ϕ is satisfiable. Finally, let $S = S_1 \cup S_2$. We know that the optimum of the PPPF instance $\langle S \rangle$, which is equal to $k_1 + |S_2|$, is at most $5|L| + 4n + |S_2|$ iff ϕ is satisfiable. This completes our reduction from planar rectilinear 3-SAT to the PPPF problem with $g = 0$.

Extending the above reduction for a general constant g turns out to be easy. The normal points in S_1 are constructed in the same way. Let S_2 be the set of obstacles constructed above. We replace each obstacle point $a \in S_2$ with a set O_a of $g(|S_1| + |S_2|) + |S_2|$ new obstacle points whose xx'-projection images are very close to a. We choose the y-coordinates of the new obstacle points such that **(i)** the points in each O_a can be perfectly fit using a bivariate polynomial $f_a \in \mathbb{R}[x, x']_g$ and **(ii)** any $g + 2$ (normal and new obstacle) points that are not contained in O_a for any $a \in S_2$ cannot be perfectly fit using any bivariate polynomial in $\mathbb{R}[x, x']_g$. Let S_2' be the set of new obstacles. We claim that the optimum of the PPPF instance $\langle S = S_1 \cup S_2' \rangle$ is at most $5|L| + 4n + |S_2|$ iff ϕ is satisfiable. If ϕ is satisfiable, then we can cover the normal points using $k_1 = 5|L| + 4n$ disjoint pieces which avoid all (old) obstacle points and hence avoid all (new) obstacle points because of the locations of the new obstacles we choose. Then we cover the xx'-projection images of each set O_a using a single piece; this is possible because the points in each O_a can be perfectly fit using a bivariate polynomial $f_a \in \mathbb{R}[x, x']_g$. In this way, we constructed a function $f \in \Gamma_g^2$ with $|f| = 5|L| + 4n + |S_2|$ that perfectly fits S. Now suppose ϕ is unsatisfiable, and let $f \in \Gamma_g^2$ be a function that perfectly fits S. We show that $|f| > 5|L| + 4n + |S_2|$. We call the pieces of f containing at least one normal point *normal pieces*. The normal points contained in each normal piece of f must form a legal subset, for otherwise the piece will contain (the xx'-projection image) of an old obstacle point $a \in S_2$ and hence contain all points in O_a, which is impossible because $O_a \cup \{b\}$ cannot be perfectly fit using any bivariate polynomial in $\mathbb{R}[x, x']_g$ for any normal point $b \in S_1$. Then there are at least $5|L| + 4n + 1$ normal pieces, because ϕ is unsatisfiable. Furthermore, each legal piece can cover at most g points in S_2' because any subset of S consists of one normal point and $g + 1$ obstacle points cannot be perfectly fit using any bivariate polynomial in $\mathbb{R}[x,x']_g$. Now every set O_a has at least $(g+1)|S_2|$ points that are uncovered by the normal pieces. One easily verifies that these uncovered points require $|S_2|$ additional pieces to cover all of them, because any $g+2$ of them that are not contained in O_a for any $a \in S_2$ cannot be perfectly fit using any bivariate polynomial in $\mathbb{R}[x,x']_g$. Therefore, $|f| > 5|L| + 4n + |S_2|$.

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